z/OS

# DFSMS Managing Catalogs

*Version 2 Release 1*

SC23-6853-01

**Note**
Before using this information and the product it supports, read the information in "Notices" on page 257.

This edition applies to Version 2 Release 1 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

# Contents

# Figures

# Tables

# About this document

This document is intended to help you, a system programmer or storage administrator, build, maintain, and support catalogs.

For information about accessibility features of z/OS®, for users who have a physical disability, please see "Accessibility," on page 253.

## Required product knowledge

To use this document effectively, you should be familiar with these functions:

- Virtual storage access method (VSAM)
- Access method services
- Storage administration

## Referenced documents

The following publications are referenced in this document:

| Document Title | Order Number |
|---|---|
| *z/OS DFSMS Access Method Services Commands* | SC23-6846 |
| *z/OS DFSMS Using Data Sets* | SC23-6855 |

## z/OS information

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

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

To find the complete z/OS library, including the z/OS Information Center, go to the z/OS Internet library (http://www.ibm.com/systems/z/os/zos/bkserv/).

# How to send your comments to IBM

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

Use one of the following methods to send your comments:

1. Send an email to mhvrcfs@us.ibm.com.
2. Send an email from the "Contact us" web page for z/OS (http://www.ibm.com/systems/z/os/zos/webqs.html).
3. Mail the comments to the following address:
   IBM Corporation
   Attention: MHVRCFS Reader Comments
   Department H6MA, Building 707
   2455 South Road
   Poughkeepsie, NY 12601-5400
   US
4. Fax the comments to us, as follows:
   From the United States and Canada: 1+845+432-9405
   From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address.
- Your email address.
- Your telephone or fax number.
- The publication title and order number:
  z/OS V2R1.0 DFSMS Managing Catalogs
  SC23-6853-01
- The topic and page number that is related to your comment.
- The text of your comment.

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

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

# If you have a technical problem

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

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

# Summary of changes for z/OS V2R1, SC23-6853-01 (as updated March 2014)

The following changes are made to z/OS Version 2 Release 1 (V2R1) as updated March 2014..

### New

- A new section has been added with information on how to suspend a catalog. See "Suspending a Catalog" on page 97 for more information.
- New information has been added to simplify recovery and improve data availability for your installation. See "Recovering a BCS" on page 97 for more information.
- New parameters DELETE USERCATALOG NODISCONNECT and DEFINE USERCATALOG RECONNECT have been added to DELETE and DEFINE to keep alias information. See "Retaining Alias Information Across Catalog Recovery" on page 103 for more information.
- Reporting Capabilities of MODIFY CATALOG have been updated with operator commands MODIFY,CATALOG,REPORT,CATSTATS and MODIFY,CATALOG,REPORT,CATSTATX. See "Obtaining Information About Catalogs and CAS Activity" on page 122 and "Evaluating Catalog I/O Activity and Settings" on page 129 for more information.
- The MODIFY CATALOG Command has been updated with new parameters: [RECOVER,{LOCK | UNLOCK | SUSPEND | RESUME}(*ucat*)], [REPORT,CATSTATX({*catname* | *catprefix**})], [{RLSENABLE | RLSQUIESCE }{(*ucat*) | ,SYSTEM}] and [TASKMAX(*nnn*)]. See "MODIFY CATALOG Command Syntax" on page 147 for more information.
- The field name directory has been updated with new fields. See "Catalog Field Names" on page 235 for more information.
- A new chapter has been added for Record Level Sharing (RLS). See Chapter 13, "Accessing Catalogs for Record Level Sharing (RLS)," on page 249 for more information.

## z/OS Version 2 Release 1 summary of changes

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

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

# Chapter 1. Introduction to Catalogs

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

Cataloging data sets also simplifies backup and recovery procedures. Catalogs are the central information point for data sets; all VSAM data sets must be cataloged. In addition, all SMS-managed data sets must be cataloged.

DFSMS allows you to use catalogs for any type of data set or object. Many advanced functions require the use of catalogs, for example, the Storage Management Subsystem.

## Catalog Structure

A catalog consists of two separate kinds of data sets: a basic catalog structure (BCS); and a VSAM volume data set (VVDS). The BCS can be considered *the* catalog, whereas the VVDS can be considered an extension of the volume table of contents (VTOC).

The **basic catalog structure** is a VSAM key-sequenced data set. It uses the data set name of entries to store and retrieve data set information. For VSAM data sets, the BCS contains volume, security, ownership, and association information. For non-VSAM data sets, the BCS contains volume, ownership, and association information.

The **VSAM volume data set** is a VSAM entry-sequenced data set. A VVDS resides on every volume that contains a catalog or an SMS-managed data set that is cataloged. It contains the data set characteristics, extent information, and the volume-related information of the VSAM data sets cataloged in the BCS. If you are using the Storage Management Subsystem (SMS), the VVDS also contains data set characteristics and volume-related information for the non-VSAM, SMS-managed data sets on the volume.

The **Volume Table of Contents** and the VTOC index are system data sets that maintain extent and allocation information for a volume. The VTOC is used to find empty space for new allocations and to locate non-VSAM data sets. For all VSAM data sets, and for SMS-managed non-VSAM data sets, the VTOC is used to obtain information not kept in the VVDS.

VVDS records for VSAM data sets are called “VSAM volume records” (VVRs). Those for SMS-managed non-VSAM data sets are called “non-VSAM volume records” (NVRs). If a non-VSAM data set spans volumes, its NVR is in the VVDS of the data set's first volume. Because a BCS is a VSAM data set, it also has a VVR in the VVDS.

Every catalog consists of one BCS and one or more VVDSs. A BCS does not “own” a VVDS: more than one BCS can have entries for a single VVDS. Every VVDS that is connected to a BCS has an entry in the BCS.

For example, Figure 1 shows a possible relationship between two BCSs and three VVDSs on three disk volumes. “BCS.A” has entries for data sets residing on each of the three volumes. “BCS.C” has entries for data sets residing on volumes B and C. Because each volume has data sets cataloged, each volume contains a VVDS.

BCS.A resides on volume A with VVDS.A. Both the VVDS and the BCS have entries for each other. All three VVDSs are cataloged in BCS.A. BCS.C, residing on volume C, contains entries for VVDS.C and VVDS.B.

Notice that a VVDS has entries for *all* VSAM and SMS-managed data sets on its volume, whereas a BCS can have entries for data sets residing on any volume.

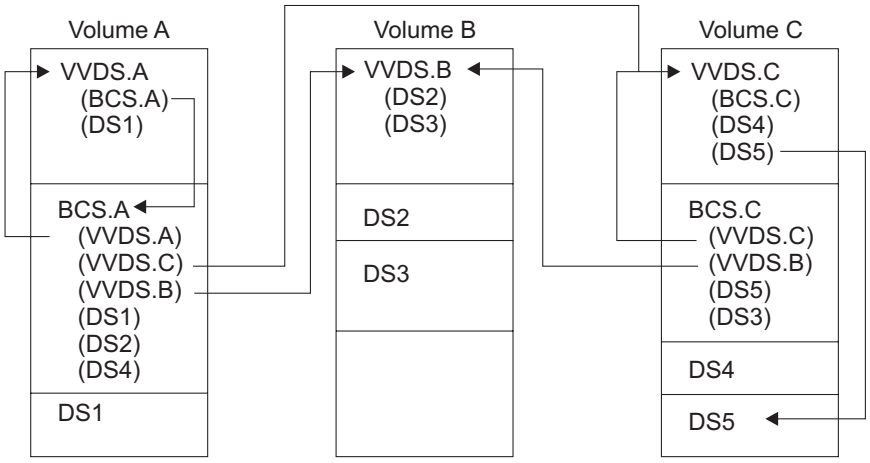



*Figure 1. Relationship of the BCS and the VVDS.* Data set names in parentheses are entries in the BCS or VVDS. Only selected arrows are drawn. See the text for a complete explanation of the relationships diagrammed in this figure.

Figure 1 illustrates how data set entries are contained in both the VVDS and the BCS. Information about a data set is also contained in the VTOC of the volume on which the data set resides, even if the data set is cataloged. To successfully perform all possible operations on a cataloged data set using the catalog, all three elements, the VVDS, BCS, and VTOC, must be synchronized. That is, any equivalent information contained in the BCS and VVDS entries for the data set, and the VTOC DSCB for the data set, must be the same. This is normally done automatically. However, if the catalog components, and the VTOC, become unsynchronized, see Chapter 7, “Analyzing Catalogs for Errors and Synchronization,” on page 109 for information on identifying and correcting the problems.

A configuration of catalogs depends on a master catalog. A master catalog has the same structure as any other catalog. What makes it a *master* catalog is that all BCSs are cataloged in it, as well as certain data sets called “system data sets” (for instance, SYS1.LINKLIB and other “SYS1” data sets). Master catalogs are discussed in “The Master Catalog” on page 24.

## Advantages of Using Catalogs

Catalogs offer advantages including improved performance, capability, usability, and maintainability.

## Performance

The catalog information that requires the most frequent updates is physically located in the VVDS on the same volume as the data sets, allowing faster access. A catalog request is expedited because fewer I/O operations are needed. Related entries, such as a cluster and its alternate index, are processed together.

## Capability

An catalog can have data sets cataloged on any number of volumes. The BCS can have as many as 123 extents on one volume. One volume can have multiple catalogs on it. All the necessary control information is recorded in the VVDS residing on that volume.

## Usability

With the commands provided, you can reorganize catalogs, move catalogs to devices of different types, merge two catalogs into one, split one catalog into two or more catalogs, share catalogs, and create portable copies.

Significant space savings for generation data groups are achieved in the catalog by reusing space when an old generation is deleted and by using an improved method of recording generation data groups.

## Maintainability

Maintainability is improved by simpler backup and recovery procedures and use of the DIAGNOSE and EXAMINE commands.

### Simpler Backup and Recovery Procedures

The BCS can be maintained independently from the data sets cataloged in it.

The dynamic information associated with a catalog data set (the data set characteristics) resides in the VVDS on the same volume as the catalog data set itself. The VVDS contains the data set characteristics that must be synchronized with the data set each time it is updated. Therefore, you can copy the data sets or the volume periodically for backup and recovery without causing the data set and VVDS portion of the catalog to be out of synchronization.

Information that can be out of synchronization in the catalog (for example, a frequently used relative byte address on volume and extents) is moved from the catalog to the VVDS. The VVDS resides on the same volume as the data set component. Therefore, you can back up data sets independently of the BCS.

The BCS maintains a record of the associated data sets with a sphere record. For example, the sphere record contains a record of a base cluster and its related alternate indexes. All the objects associated with the sphere record are processed before the sphere record is updated. You can therefore restart most processing at the point of interruption, without losing data or special processing.

You can also use the Integrated Catalog Forward Recovery Utility (ICFRU) to recover a damaged catalog to a correct and current status. This utility uses system management facilities (SMF) records that record changes to the catalog, updating the catalog with changes made since the BCS was backed up. ICFRU can also be used with shared catalogs or master catalogs, if the master catalog is not being used as a master at the time.

### Error Diagnosis

You can check the structure of a BCS with the EXAMINE command, and the content of a BCS or a VVDS with the DIAGNOSE command.

When analyzing a catalog, you must execute the VERIFY command before the EXAMINE command to verify that the catalog information for the catalog is current. If the catalog is open in the Catalog Address Space and you execute the EXAMINE command, the data in the catalog may not be current, and incorrect data may be displayed.

With EXAMINE, you can check the structural integrity of a BCS, just as you would check the structural integrity of any other catalog key-sequenced data set.

With DIAGNOSE, you can compare the BCS and the VVDS and thus verify catalog integrity. If an error is found, DIAGNOSE identifies the problem. Based on the result of the DIAGNOSE output, you can determine how to correct the error.

In addition to the DIAGNOSE command, you can use all the existing access method services and catalog diagnostic aids for the catalog.

Successful execution of a DIAGNOSE command depends on whether the input and compare data sets can be accessed. If you cannot read records from one of these data sets, the DIAGNOSE command terminates with an appropriate message and return code.

For more information on error diagnosis, refer to Chapter 10, "Catalog Diagnostic Information," on page 217.

## Using Access Method Services

You use access method services to define and maintain catalogs. Access method services commands can also be used to define and maintain catalog and non-catalog data sets. For a complete explanation of the usage of access method services, the required JCL, and examples, see *z/OS DFSMS Access Method Services Commands*.

Table 1 explains the access method services commands that are used with catalogs. For further information about the DEFINE command, see "Using the DEFINE Command" on page 6.

*Table 1. Access Method Services Commands for Catalogs*

| Command | Purpose |
|---|---|
| ALTER | Alters previously defined catalog entries or certain catalog attributes, including entries for tape volume catalogs. |
| CREATE | Creates tape library or tape volume entries. |
| DEFINE | Defines catalog entries. |
| DEFINE CLUSTER | Defines VVDSs and catalog entries. |
| DEFINE USERCATALOG | Defines user catalogs including tape volume catalogs (DEFINE USERCATALOG VOLCATALOG). |
| DELETE | Deletes catalog and VVDS entries, data sets, catalogs, VVDSs, and data set control blocks in the VTOC. |

*Table 1. Access Method Services Commands for Catalogs (continued)*

| Command | Purpose |
|---|---|
| DIAGNOSE | Determines whether the content of BCS or VVDS records is invalid or unsynchronized. Diagnose recognizes tape library and tape volume record types. It checks the cell structure of the volume catalog. |
| EXAMINE | Determines whether structural errors exist in the index or data component of a BCS. |
| EXPORT | Creates a backup copy of a catalog or copy that can be moved to another system or volume. |
| EXPORT DISCONNECT | Disconnects the catalog from the master catalog. |
| IMPORT | Restores an exported backup copy of a catalog, or makes a catalog that was previously exported from one system available for use in another system. |
| IMPORT CONNECT | Connects the catalog to the master catalog on the same system, or on any shared system. |
| LISTCAT | Lists catalog entries including entries for tape volume catalogs. |
| PRINT | Prints data set or catalog records. |
| REPRO | Copies catalogs; splits catalog entries between two catalogs; merges catalog entries into another user catalog or master catalog. REPRO supports tape volume catalogs. |
| VERIFY | Causes a catalog to reflect the end of a data set correctly after an error that prevented closing a catalog data set. The error might cause the catalog to be incorrect. |

**Restriction:** The access method services ALTER, CREATE, and DELETE commands should only be used to recover from tape volume catalog errors. Because access method services cannot change the library manager inventory in an automated tape library, ISMF should be used for normal tape library ALTER, CREATE, and DELETE functions.

## JCL Considerations

Under normal conditions, you do not have to specify a catalog on a DD statement when using access method services.

You might have to use DD statements to recatalog an incorrectly cataloged data set, or to define new system data sets.

The examples in this document occasionally use SMS functions. However, most examples can be used in either an SMS or a non-SMS environment, although minor changes might be necessary. Where allocation information can be included in a data class (for example, space attributes), they are explicitly coded. If management class or storage class is specified, it can be removed for non-SMS environments.

See *z/OS DFSMS Using Data Sets* for more details on allocating SMS-managed data sets. See *z/OS DFSMS Access Method Services Commands* for more details on JCL requirements for access method services.

## Using the DEFINE Command

Data sets of all types can be cataloged. This can be accomplished through JCL, dynamic allocation, access method services, IEHPROGM, or other utilities. If you are using the Storage Management Subsystem, data sets are automatically cataloged.

When you issue the access method services DEFINE command to create a catalog object, a catalog entry is built that describes the object. Before an object can be defined, there must be a catalog in which to define the object. The catalog is chosen according to the catalog search order.

When you define a catalog data set or object, you specify attributes to be associated with it. The attributes include, for example, the SMS Management Class name to be associated with the data set. After the object is defined, it can be processed with other access method services commands and with application programs.

Password protection is ignored for the protection of any data set cataloged in a catalog, including the catalogs themselves. This is a change from previous releases where passwords were ignored only for SMS-managed data sets.

When you define a catalog, cluster, or alternate index, you can specify attributes in several different ways. The parameter set for DEFINE USERCATALOG, MASTERCATALOG, CLUSTER, and ALTERNATEINDEX is directly related to the way the attributes are stored in the catalog. The catalog entries that describe a catalog, cluster, or alternate index are:

- The cluster entry, which describes the attributes of the cluster or catalog.
- The alternate index entry, which describes the attributes of the alternate index.
- The data entry, which describes the attributes of the data component of a catalog, cluster, or alternate index.
- The index entry, which describes the attributes of the index component of a catalog, key-sequenced cluster, or alternate index.

When you specify attributes as parameters of USERCATALOG, MASTERCATALOG, CLUSTER, or ALTERNATEINDEX, consider the following:

- Under SMS, you can specify a data class, management class, or storage class. You can specify data set size in kilobytes or megabytes as well as cylinders, tracks, or records.
- Attributes specified in the parameters are defined in the cluster or alternate index entry of a catalog if they pertain to that entry.
- Attributes specified in the parameters are defined for the data or index entries to which they pertain.
- If the same attribute is specified as a subparameter of DATA or INDEX, the value of the attribute specified at the DATA or INDEX level overrides the value of the attribute specified at the USERCATALOG, MASTERCATALOG, CLUSTER, or ALTERNATEINDEX level.

You can use the LISTCAT command with the ALL option to list catalog entries and to determine what various attributes are stored in the catalog. See *z/OS DFSMS Access Method Services Commands* for a description of the attributes for each type of entry.

## Notational Conventions

A uniform notation is used to describe the syntax of commands, or the format of control records. This notation is not part of the language. The following conventions can be used in this document:

**[ ]** Brackets enclose an optional entry. You can, but need not, include the entry. Examples are:
- **[***length***]**
- **[MF=E]**

**|** An OR sign (a vertical bar) separates alternative entries. You must specify one, and only one, of the entries unless you allow an indicated default. Examples are:
- **[REREAD | LEAVE]**
- **[***length* **| 'S']**

**{ }** Braces enclose alternative entries. You must use one, and only one, of the entries. Examples are:
- **BFTEK={S | A}**
- **{K | D}**
- **{***address* **| S | O}**

Sometimes alternative entries are shown in a vertical stack of braces. An example is:

```
MACRF={(R[C | P])}
{(W[C | P | L])}
{(R[C],W[C])}
```

In the example above, you must choose only one entry from the vertical stack.

**. . .** An ellipsis indicates that the entry immediately preceding the ellipsis can be repeated. For example:
- **(***dcbaddr***,[(***options***)],. . .)**

**' '** A ' ' indicates that a blank (an empty space) must be present before the next parameter.

**REQUIRED KEYWORDS AND SYMBOLS**

Entries shown **IN THE FORMAT SHOWN HERE** (notice the type of highlighting just used) must be coded exactly as shown. These entries consist of keywords and the following punctuation symbols: commas, parentheses, and equal signs. Examples are:
- **CLOSE , , , ,TYPE=T**
- **MACRF=(PL,PTC)**

The format (the type of highlighting) that is used to identify this type of entry depends on the display device used to view a softcopy document. The published hardcopy version of this document displays this type of entry in uppercase boldface type.

**DEFAULT VALUES**

Values shown **IN THE FORMAT SHOWN HERE** (notice the type of highlighting just used) indicate the default used if you do not specify any of the alternatives. Examples are:
- **[EROPT={ACC | SKP | ABE}]**
- **[BFALN={F | D}]**

The format (the type of highlighting) used to identify this type of entry depends on the display device used to view a softcopy document. The published hardcopy version of this document displays this type of value in underscored uppercase boldface type.

*User Specified Value*

Values shown *in the format shown here* (notice the type of highlighting just used) indicate a value to be supplied by you, the user, usually according to specifications and limits described for each parameter. Examples are:

- *number*
- *image-id*
- *count*

The format (the type of highlighting) that is used to identify this type of entry depends on the display device used to view a softcopy document. The published hardcopy version of this document displays this type of value in lowercase italic type.

# Chapter 2. Planning a Configuration of Catalogs

Before you define a catalog, there are some major issues that you should consider carefully. These issues involve not only the specific characteristics of the catalog, but also the characteristics of the configuration of catalogs that will reside on your system.

This chapter addresses issues related to the complete catalog configuration. Catalog protection, size, and performance attributes are discussed in other chapters.

## Using Catalogs with the Storage Management Subsystem

The Storage Management Subsystem (SMS) is a subsystem of DFSMS that automates data management, including data placement on volumes,cataloging, and backing up data.

### Catalog Rules for Data Managed by the Storage Management Subsystem

Under SMS, all permanent data sets must be cataloged. Only integrated catalog facility type catalogs can be used with SMS. Although a catalog contains entries for SMS-managed data, the catalog itself does not have to be SMS-managed, but it is recommended that catalogs containing entries for SMS-managed data also be SMS-managed. A catalog can contain entries for data sets that are SMS-managed, and entries for other data sets that are not managed by SMS.

All new SMS-managed data sets are cataloged when they are allocated, not when the job step terminates. SMS-managed generation data sets are cataloged in a "deferred roll-in" status at allocation, and then are "rolled-in" to the generation data group at step termination, if their disposition indicates they are to be cataloged. To set the disposition, use DISP=(NEW,CATLG).

You can change the GDG setting for reclaim processing as needed without having to IPL the system by specifying the keyword setting for GDS_RECLAIM(NO) in the IGDSMSxx member of SYS1.PARMLIB. If GDG reclaim processing is not desired the default is GDS_RECLAIM(YES). You can also issue the SETSMSGDG_RECLAIM(YES|NO) command to change the value specified in the IGDSMSxx member of SYS1.PARMLIB. This change is in effect only until you re-issue the command or IPL the system.

You cannot use password protection for catalogs or data sets cataloged in a catalog. Although you can define catalogs and data sets with passwords, SMS ignores the passwords. Use z/OS Security Server Resource Access Control Facility (RACF®) to protect data that is cataloged.

There are special RACF facility classes that are used with SMS. You need to define these classes to restrict the use of certain functions. For instance, you should not normally be able to specify a catalog to be searched unless you have authorization for the RACF directed catalog facility class. See "Storage Administration (STGADMIN) Profiles in the FACILITY Class" on page 88 for more information.

All SMS-managed data sets have entries in the VVDS. Non-VSAM data sets have non-VSAM volume records (NVRs). VSAM data sets have VSAM volume records (VVRs). This requires an SMS-managed volume to have a larger VVDS.

Temporary VSAM data sets that are SMS-managed also have VVDS entries, although they do not have BCS entries.

For more information on the Storage Management Subsystem, see *z/OS DFSMSdfp Storage Administration* . For more information on catalog considerations when moving to an SMS environment, see *z/OS DFSMS Implementing System-Managed Storage*.

## Defining Catalog and VVDS Data Classes

You can simplify the definition of new BCSs and explicitly defined VVDSs by creating data classes for each. The data classes should specify the characteristics most commonly used at your installation. You can override most attributes defined in the data class when you issue the DEFINE command. You can use a data class to define the catalog even if the catalog is not SMS-managed.

For BCSs, choose appropriate default size and performance attributes. See "Estimating Space Requirements for the BCS" on page 52, "Choosing Options to Adjust Catalog Performance" on page 55, and "Defining a Basic Catalog Structure" on page 56 for recommendations for BCS attributes. For RECORG, specify KS. Do not specify a key length or offset, or a logical record length. These values are set by the DEFINE USERCATALOG command.

For VVDSs, choose an appropriate size, usually 10 tracks primary and secondary space. There are no special performance attributes for VVDSs. For RECORG, specify ES. Allow DEFINE CLUSTER to set all other values.

## Defining a Catalog Management Class

You can define a special management class for BCSs. This management class can be shared with other critical system data sets that have the same requirements, to ensure that recent backup copies are available in case a catalog must be recovered.

In general, the management class you specify for BCSs should have the following attributes:

1. For expiration, NOLIMIT for all attributes.
2. For migration, NONE for command or auto migration.
3. For backup,
   a. For backup frequency, specify 0 to have DFSMShsm back up the catalog every occasion that it is changed.
   b. For administrator or user command backup, specify ADMIN to limit backup operations to storage administrators.
   c. For auto backup, specify YES.

Choose appropriate values for other backup attributes, depending on the needs of your installation. It is generally prudent to keep extra backup copies of catalogs, even after the catalog has been deleted.

Management class does not apply to VVDSs. VVDSs should only be backed up as part of a full volume dump.

## Defining a Catalog Storage Class

Storage class is used to define the performance objectives for a data set. Since catalogs have their own cache, specification of additional performance objectives through storage class is unnecessary.

If you want to control the placement of your catalogs, assign them a storage class with the Guaranteed Space attribute. The Availability objective for catalogs should be standard, although you can make it Continuous for the catalog of a critical application.

# Sharing Catalogs Among Systems

A shared catalog is a basic catalog structure that is eligible to be used by more than one system. It must be defined with SHAREOPTIONS(3 4), and reside on a shared volume. A DASD volume is initialized as shared using the MVS™ hardware configuration definition (HCD) facility. Note that the device must be defined as shared to all systems that access it. If some systems have the device defined as shared and some do not, catalog corruption will occur. Check with your system programmer to determine shared volumes. Note that it is not necessary that the catalog actually be shared between systems; the catalog address space assumes it is shared if it meets the criteria stated above. All VVDSs are defined as shared. Tape volume catalogs can be shared in the same way as other catalogs.

By default, catalogs are defined with SHAROPTIONS(3 4). You can specify that a catalog is not to be shared by defining the catalog with SHAREOPTIONS(3 3). Only define a catalog as unshared if you are certain it will not be shared. Place unshared catalogs on volumes that have been initialized as unshared. Catalogs that are defined as unshared and that reside on shared volumes will become damaged if referred to by another system.

If a catalog is shared, its catalog address control structures are refreshed when updates are made to the catalog from any system. This ensures that each system is using an up-to-date copy of the catalog at all times.

## Catalog Sharing Protocols

There are three protocols that are used to share catalogs between systems:

- VVDS mode
- Enhanced Catalog Sharing (ECS) mode
- Record Level Sharing (RLS) mode

### VVDS Mode Sharing

There is information necessary to communicate changes in a basic catalog structure (BCS) to other systems that are sharing the catalog. This information is stored in a special record in the VVDS on the volume the catalog is defined on. The information is used to ensure the consistency of the catalog records that are cached on any sharing subsystem. It is also used to update the BCS control block structure in those cases where a sharing system has extended beyond the current high used value or to a new extent. In addition, it is used to invalidate BCS data and index buffers when they have been updated from a sharing system.

The storing and retrieval of this information requires additional I/O to the volume containing the catalog. In some cases, this I/O overhead can become significant and have a noticeable impact on system or sysplex performance. VVDS Mode sharing is the default mode of sharing.

### Enhanced Catalog Sharing (ECS) Mode Sharing

This sharing protocol stores the information that describes changes to a shared catalog in the Coupling Facility. The I/O required for the VVDS mode protocol is eliminated, resulting in better sysplex-wide performance.

This protocol is used when three conditions are satisfied:

- The ECS cache structure is defined in a coupling facility that uses an installation-defined Coupling Facility Resource Manager (CFRM) policy
- A successful connection has been made to the ECS cache structure
- A catalog is referenced that has the ECSHARING attribute set

For additional information on enabling this protocol, see "Using Enhanced Catalog Sharing Mode" on page 42.

### Record Level Sharing (RLS) Mode Sharing

This sharing protocol provides not only more granular locking through the use of a CF lock structure but also greater buffering and caching through the use of the SMSVSAM address space and its associated buffer pools. A catalog in RLS mode completely eliminates the use of the SYSIGGV2 BCS reserve/enqueue, ISC, VLF, and ECS for that catalog, and the responsibility of locking, buffering, and caching falls onto the SMSVSAM address space.

This protocol is used when the following five conditions are satisfied:

- The LOG parameter is specified.
- The RLSENABLE attribute is specified.
- The STORCLAS and its cache set information are specified.
- The SMSVSAM address space is active on the system.
- The catalog is not currently open in either VVDS mode or ECS mode from any system in release 1.13 of z/OS or below.

For additional information on enabling this protocol, see Chapter 13, "Accessing Catalogs for Record Level Sharing (RLS)," on page 249.

## Sharing Catalogs Under the Storage Management Subsystem

A catalog can contain entries for SMS-managed and unmanaged data sets. A catalog containing entries for unmanaged data can itself be managed.

A catalog that contains entries for SMS-managed data does not have to reside on SMS-managed volumes. Likewise, a catalog that contains entries for data not managed by SMS can itself reside on an SMS-managed volume.

## Preventing Lockouts

CATALOG MANAGEMENT uses the SYSIGGV2 reserve while serializing access to catalogs. The SYSIGGV2 reserve is used to serialize the entire catalog BCS component across all I/O as well as to serialize access to specific catalog entries. The SYSZVVDS reserve is used to serialize access to associated VVDS records. The SYSZVVDS reserve along with the SYSIGGV2 reserve provide an essential mechanism to facilitate cross system sharing of catalogs.

**Preventing lockouts**: When your data sets re,side on the same volume as other shared catalogs, deadlocks can occur. An example is shown in Figure 2. In this example, SYS 1 and SYS 2 share DASD volumes, VOLSER1 and VOLSER2. The SYSIGGV2 reserve is held for CATALOG A by SYS 1 while trying to obtain a reserve for data set A. SYS 2 has a SYSIGGV2 reserve for CATALOG B while trying to obtain a reserve for data set B. Reserves for data set A or data set B could be for SYSVTOC or SYSZVVDS. You can prevent such deadlocks, by always converting the SYSIGGV2 reserves to SYSTEMS ENQUEUEs using Global Resource Serialization (GRS) or an equivalent product. The resources SYSZVVDS or SYSVTOC should be either both converted or both excluded in the GRS RNL lists. The important point about SYSZVVDS and SYSVTOC is that they both be treated the same way, either both converted or both excluded. You should review info apar II14297 which contains a number of items to consider when sharing catalogs and the GRS RNL definitions that cover sharing catalogs. For further information concerning catalog serialization, see *z/OS MVS Planning: Global Resource Serialization*.

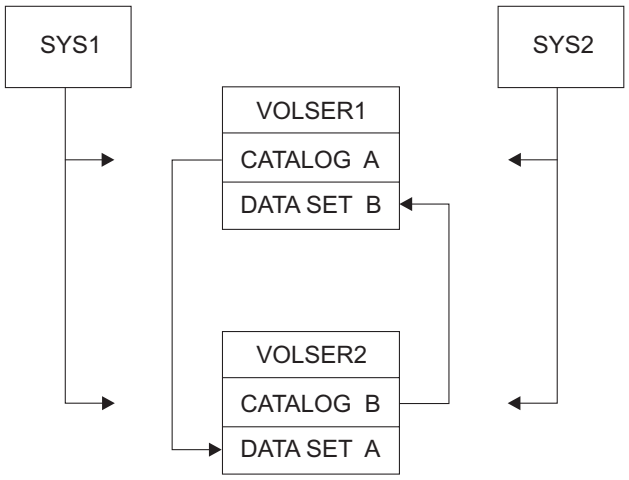



*Figure 2. Data Sets Residing on the Same Volume as Other Shared Catalogs*

Lockouts that occur because of a failure to convert the SYSIGGV2 reserve are not system failures for which the IBM® System Support personnel are able to provide assistance. You prevent these lockouts by ensuring that your support personnel has converted the SYSIGGV2 reserve.

In rare situations, a lockout may occur when a different address space (instead of the Catalog Address Space) obtains the resource SYSIGGV2 exclusive for a catalog as an ECS rebuild for the structure is in progress.

**Note:** It does not matter if the catalog is ECS-active.

Some applications that obtain SYSIGGV2 exclusive are DFDSS and IDCAMS IMPORT. These functions should not run simultaneously on catalogs when ECS is active since a rebuild can occur at any time. If contention occurred on a rebuild, issue D GRS,RES=(SYSIGGV2,*) command to determine the job (other than catalog) and the owner of the resource SYSIGGV2. Proceed to cancel the job and the rebuild should complete successfully. If the job is cancelled, it can rerun as soon as the rebuild completes. The other option is to deactivate ECS temporarily by issuing F CATALOG,DISCONNECT on one of the systems in the sysplex. This will cause all systems to quiesce ECS activity. When the applications are finished running, re-enable ECS by connecting the disconnected system back into ECS by issued

command (F CATALOG,ECSHR(CONNECT)) then turn on AutoAdd by issuing command (F CATALOG,ECSHR(AUTOADD)) to add catalogs back into ECS.

**Preventing a lockout due to resource SYSIGGV2 contention:** In rare situations, a lockout may occur when a different address space (instead of the Catalog Address Space) obtains the resource SYSIGGV2 exclusive for a catalog as an ECS rebuild for the structure is in progress. Note that it doesn't matter if the catalog is ECS-active or not. Some of the applications that obtain SYSIGGV2 exclusive are DFDSS and IDCAMS IMPORT. IBM recommends that these functions should not run simultaneously on catalogs when ECS is active since a rebuild could occur at any time. However, if contention does occur because of a rebuild, issue the `D GRS,RES=(SYSIGGV2,*)` command to determine the job and the owner of the resource SYSIGGV2. Proceed to cancel the job and the rebuild should then complete successfully. You can rerun the cancelled job as soon as the rebuild completes.

You can also resolve the problem by deactivating ECS temporarily with command `F CATALOG,DISCONNECT` on one of the systems in the sysplex. This will cause all systems to quiesce ECS activity. Once the applications are finished running, re-enable ECS by connecting the disconnected system back into ECS (`F CATALOG,ECSHR(CONNECT)`) and then turn on AutoAdd (`F CATALOG,ECSHR(AUTOADD)`) to add catalogs back into ECS.

SYSIGGV2 BCS resource is no longer raised for each catalog request to a catalog in RLS mode. Instead, the SMSVSAM address space will obtain data set and record locks from the CF lock structure on a catalog's behalf.

## Using the SYS% Conversion Facility

The purpose of the SYS% facility is to provide a method of managing system data sets with SMS in a multi-host environment. Each system requires duplicate system data sets (data sets with the high-level qualifier SYS1).

Because SMS requires uniquely named data sets, you normally would not be able to access the system data sets for one system from another system. However, you might need to update one system's SYS1 data sets from a different system.

The SYS% facility allows you to do this by permitting access to system data sets with alias names. The alias names have SYS for the first three characters, followed by another character that is *not* 1 (for example, SYSA). You use the alias to first orient the search to the appropriate master catalog, which is then searched for the requested system data set.

As an example, consider a complex with two systems, each using SMS, called SYSTEMA and SYSTEMB. To connect the systems, the master catalog of each system is defined as a user catalog in the master catalog of the other system. This can be seen in Figure 3 on page 15.

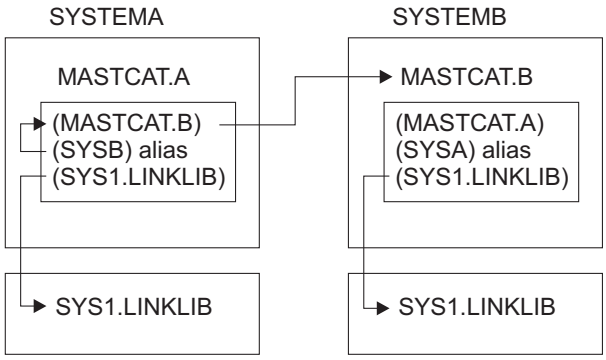



*Figure 3. Master Catalogs of Connected SMS Systems.* The pointers from MASTCAT.B to MASTCAT.A are not shown, to make the diagram clearer. Catalog entries are enclosed in parentheses.

In this example, MASTCAT.A is defined with the alias SYSA in MASTCAT.B; and MASTCAT.B is defined with the alias SYSB in MASTCAT.A. Each catalog contains a pointer to a different SYS1.LINKLIB system data set. If a job addresses the data set SYS1.LINKLIB, the SYS1.LINKLIB that is allocated depends on which system is running the job.

In order for a job running on SYSTEMA to access the SYS1.LINKLIB cataloged in MASTCAT.B on SYSTEMB, that job must orient itself to the other master catalog. This can be done by using the SYS% facility and the alias defined for the other master catalog.

If the SYS% facility is active, a job running on SYSTEMA can address SYSB.LINKLIB. Because SYSB is an alias of MASTCAT.B, MASTCAT.B is searched for a data set named SYSB.LINKLIB. Because one is not found, MASTCAT.B is again searched, only this time for SYS1.LINKLIB, which is found. The job then uses the SYS1.LINKLIB cataloged in MASTCAT.B, even though the job ran on SYSTEMA, and there is a SYS1.LINKLIB cataloged in SYSTEMA's master catalog.

If the alias is not defined (if you specify an alias that does not exist), the SYS% conversion routine will resolve to the current system.

**Example:** If you specify SYSX.LINKLIB on SYSTEMA, and SYSX is not defined as an alias to another system, then the request would go to SYS1.LINKLIB on SYSTEMA.

Make sure that you have located the correct system SYS1 data set. Also, if you specify SYSX.LINKLIB on SYSTEMA, and SYSX is defined as an alias to another system but the SYS1.LINKLIB does not exist on SYSTEMB, the request would go to SYS1.LINKLIB on SYSTEMA.

## Allocating New System Data Sets Under SMS

The SYS% facility is specifically used for allocating existing data sets. It is not meant to be used to allocate new data sets, for example, for a new system. However, the presence of SYS% aliases used with the SYS% facility allows for the allocation of new system data sets.

You can also use DEFINE PAGESPACE RECATALOG with the CATALOG parameter. With this method, you can recatalog the PAGE data set into the desired catalog by completing the following steps:

1. Create new page volumes using the DEFINE PAGESPACE command.

2. Move the new pagespace entries from the driver master catalog to the new master catalog using the REPRO MERGCAT command as shown in the following example:

```
//SYSPRINT DD  SYSOUT=*
//PLPA     DD  UNIT=3390,VOL=SER=S0HP22,
//             DISP=OLD
//SYSIN    DD *
   DEFINE  PAGESPACE  -
            ( FILE(PLPA) -
              NAME (SYS1.newsysname.PLPA) -
              CYLINDERS (1) -
              UNIQUE -
              VOLUME (S0HP22)  )
              CYLINDERS(1)
              UNIQUE -
              VOLUME(S0HP22)
//MOVEPAGE EXEC  PGM=IDCAMS
//UCATDVR  DD DSN=CATALOG.drivercat,DISP=SHR
//UCATNEW  DD DSN=CATALOG.newcat,DISP=SHR
//DD1      DD VOL=SER=S0HP22,UNIT=3390,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  REPRO ENTRIES(SYS1.newsysname.*) -
```

If the new system is SYSTEMB, and MASTCAT.B has been defined with the alias SYSB in MASTCAT.A, you can run allocation jobs on SYSTEMA to allocate data sets with the high-level qualifier SYSB. Once all the new data sets have been allocated (or at any time that you are ready), you can do a generic rename of the data sets, changing the SYSB qualifier to SYS1:

```
ALTER SYSB.*  NEWNAME(SYS1.*)  CATALOG(MASTCAT.B)
```

To successfully alter the names of the SYSB data sets from SYSTEMA, you must have appropriate RACF authority to the directed catalog facility class. See “Storage Administration (STGADMIN) Profiles in the FACILITY Class” on page 88.

This process of allocating new data sets with the high-level qualifier SYSB, and then renaming them by directing the catalog request, does not use the SYS% facility. After renaming the system data sets, MASTCAT.B does not contain any SYSB data sets. However, the SYS1 data sets in MASTCAT.B can be accessed by allocating them using the SYSB high-level qualifier, if the SYS% facility is activated.

## Enabling SYS% Conversion

The setting for the SYS% facility (“on” or “off”) is set in the SYSCAT*xx* member of SYS1.NUCLEUS. You can change this setting without an IPL by using the MODIFY CATALOG operator command:

```
MODIFY CATALOG,SYS%ON   or MODIFY CATALOG,SYS%OFF
```

Using the MODIFY command, you can use the SYS% facility selectively, only turning it on when you need it. See “Identifying the Master Catalog and Initial Configuration (SYSCATxx)” on page 37 for information on activating the SYS% facility at IPL.

## Restrictions on Using SYS% Conversion

The SYS% facility should only be used to allocate existing data sets. The SYS% facility can be used to build systems in an environment where the new and old data sets with the same names need to be cataloged in different catalogs at allocation time. For this reason, several functions are not supported in SYS% processing. Attempting them can cause undesired results or errors.

Do not attempt to use the SYS% facility to process SYS1 data sets with the DEFINE or ALTER commands. However, there is no restriction on using DEFINE or ALTER on data sets whose *real* data set name has a high level qualifier that can later be used by SYS%. For example, you can define a data set named SYSB.LINKLIB, and if SYSB is an alias to another system's master catalog, it is defined in that master catalog *under the name SYSB.LINKLIB*. Renaming the data set can then be accomplished if you have appropriate RACF authority to the directed catalog FACILITY class profile.

Errors result if you expect DEFINE, ALTER, or LISTCAT LEVEL to convert a SYS% alias to SYS1.

You can use the SYS% facility to delete data sets only if you use their fully-qualified data set names (a "discrete" deletion). You can do this during disposition processing or through access method services. For example, if the real data set name is "SYS1.LINKLIB", you can allocate and delete the data set using "SYSB.LINKLIB" if the alias "SYSB" is properly defined. Do not attempt to generically delete a group of SYS1 data sets using a SYS% alias by specifying "SYSB.*".

Do not use indirect volume serial numbers (for example, ******) when using SYS% conversion. Use specific volume serial numbers for all references to SYS1 data sets. Indirect volume serial numbers resolve to the IPL volume of the system running the job.

Be careful if you attempt to use SYS% conversion with the ISPF/PDF utilities, ISMF, and TSO commands. Unexpected results can occur, since these programs sometimes use generic functions.

## Integrity of Shared Catalogs and VVDSs

The catalog address space maintains information about when a catalog, either a BCS or a VVDS, is changed. Each system can identify when a catalog has changed, and a system refreshes its copy of the catalog in main memory only when necessary.

Sharing systems maintain control blocks for each BCS and VVDS accessed. Under normal conditions, these control blocks need no special maintenance.

However, if you cannot access a shared BCS or VVDS after recovery, refresh the control blocks for the BCS or VVDS. Use the MODIFY CATALOG command with CLOSE or UNALLOCATE to remove the control blocks for a BCS. The next request that accesses the BCS rebuilds the control blocks. The control blocks for a VVDS can be removed using the VCLOSE or VUNALLOCATE options of MODIFY CATALOG.

If you recover a shared BCS to a volume with a different volume serial number or device type than the original BCS had, use the access method services IMPORT CONNECT ALIAS command to update the catalog connector record for the BCS in the master catalog of each sharing system.

Sharing catalogs also affects catalog performance. See "The Effect of Sharing Catalogs on Cache Usage" on page 30 for more information on the performance of shared catalogs, and Chapter 8, "Working with the Catalog Address Space," on page 119 for more information on MODIFY CATALOG.

## Extended Alias Support

Alias entries are defined to allow a reference to a particular name to be translated to a different name for the actual data. This provides a means for users to access their data by a particular name without having to know the actual name of the data set that contains their data. This technique is particularly useful for migration from release to release of products.

For example, the names of libraries for different releases or versions of products can contain the release or version information. An alias can be created without this information, so users are not aware when the underlying library changes. This minimizes the impact on job control language and TSO changes as a result of migrations.

As an example, if SYS1.V1R6M0.PRODUCT is the name of a product library, an alias of SYS1.PRODUCT is created by this IDCAMS DEFINE command:

```
DEFINE ALIAS (NAME(SYS1.PRODUCT) RELATE(SYS1.V1R6M0.PRODUCT))
```

Users can now reference SYS1.PRODUCT, and if version 1 release 4 of the product is installed, the alias name can simply be changed to refer to the new library. This allows the new version of the library to be tested without disrupting current users.

A particular problem occurs in environments that share catalogs, particularly master catalogs. Different systems might be at different levels of software. Users would like to use the alias approach to minimize the effect of data set name changes. However, the actual data set name had to be specified when the ALIAS was defined; therefore all systems would see the same value.

A parameter for the DEFINE ALIAS command, SYMBOLICRELATE, allows the specification of the base data set name using system symbols. The above example could then be defined as:

```
DEFINE ALIAS (NAME(SYS1.PRODUCT) -
     SYMBOLICRELATE('SYS1.&PRODVR..PRODUCT'))
```

If the system symbol '&PRODVR' was set to 'V1R5M0' on System A, and 'V1R6M0' on System B, then a reference to the name SYS1.PRODUCT would access the proper data set, depending on what system the alias name was used from. To set system symbols, see the *z/OS MVS Initialization and Tuning Reference*. In this case, the alias name is resolved at the time of use, rather than at the time of definition. As sharing systems are ready to upgrade to the new data set, they only need change the definition of the appropriate system symbol or symbols to access the new data set by the original alias.

This support is available for defining aliases of user catalogs, or for non-VSAM (non-GDS) data sets. If the string containing the system symbols cannot be resolved (for example, the symbol might not be defined on the referencing system), the reference will probably fail. This is because there would be no data set name that matched the value specified in the SYMBOLICRELATE keyword.

The actual resolution of the symbolic string to a data set name is done at two possible times:

- If the resolved name is a catalog, it is done at catalog address space initialization or when the multi-level alias table is reinitialized
- If the resolved name is a non-VSAM data set, it is done at the time of reference to the data set by a catalog request

The symbolic string must not exceed 44 characters, including all name segments and periods.

**Note:** If the non VSAM data set has aliases that are defined using SYMBOLICRELATE parameter, the ALIAS entries are NOT deleted when you use the DELETE command. SYMBOLICRELATE aliases can resolve to different data sets depending on which system you are accessing and for this reason, DELETE will not remove the data set aliases defined using SYMBOLICRELATE as they may be valid on other systems.

## The Catalog Search Order

When an application or user refers to a cataloged data set or creates a data set that is to be cataloged, the configuration of catalogs must be searched to find the appropriate catalog. The catalog is chosen according to a search order determined by the aliases defined for the catalogs, the names of the catalogs, and the multilevel alias search level.

### The Standard Search Order

Most catalog searches should be based on catalog aliases. When appropriate aliases are defined for catalogs, the high-level qualifier of a data set name is identical to a catalog alias and identifies the appropriate catalog to be used to satisfy the request. However, some alternatives to catalog aliases are available for directing a catalog request, specifically the CATALOG parameter of access method services and the name of the catalog.

Table 2 summarizes the catalog search order for defining or locating a data set.

*Table 2. Standard Search Order for Catalog Requests*

| Defining a Data Set | Locating a Data Set |
|---|---|
| 1. Use the catalog named in the IDCAMS CATALOG parameter, if coded.<br>2. If the data set is a generation data set, the catalog containing the GDG base definition is used for the new GDS entry.<br>3. If the high-level qualifier is a catalog alias, use the catalog identified by the alias or the catalog whose name is the same as the high-level qualifier of the data set.<br>4. If no catalog has been identified yet, use the master catalog. | 1. Use the catalog named in IDCAMS CATALOG parameter, if coded. If the data set is not found, fail the job.<br>2. If the data set is a generation data set, the catalog containing the GDG base definition is used for the new GDS entry.<br>3. If not found, and the high-level qualifier is an alias for a catalog, search the catalog or if the high-level qualifier is the name of a catalog, search the catalog. If the data set is not found, fail the job.<br>4. Otherwise, search the master catalog. |

**Restriction:** For DEFINE USERCATALOG, the catalog will be added to the master catalog of the running system whether the CATALOG parameter is used or not.

To use an alias to identify the catalog to be searched, the data set or object name, or the generation data group base name, must be a qualified name.

The catalog search order is modified when the CATALOG parameter of access method services commands is used to direct the catalog request. When you specify a catalog in the CATALOG parameter, and you have appropriate RACF authority to the FACILITY class profile STGADMIN.IGG.DIRCAT, the catalog you specify is used.

For instance, DEFINE USERCATALOG CATALOG(SYS1.MASTER.ICFCAT) defines a catalog with a connector record in SYS1.MASTER.ICFCAT, even if SYS1.MASTER.ICFCAT is not the master catalog on the system where you issue the command. See "Storage Administration (STGADMIN) Profiles in the FACILITY Class" on page 88 for more information on the RACF directed catalog profile.

## The Multilevel Alias Facility

You can augment the standard catalog search order by defining multilevel catalog aliases. A multilevel catalog alias is an alias of two or more high-level qualifiers. You can define aliases of up to four high-level qualifiers. However, the multilevel alias facility should only be used when a better solution cannot be found. The need for the multilevel alias facility can indicate poor data set naming conventions. Before defining multilevel aliases, review your data set naming conventions.

### Overview of the Multilevel Alias Facility

Under the standard catalog search order, when you are using aliases, you might find that certain catalogs become over used. For instance, if the catalog becomes too large, recovery might become excessively disruptive, or the catalog might be required to process an excessive number of requests, resulting in frequent enqueues. If these catalogs contain large numbers of data sets with the same high-level qualifier, you can alleviate the problem with the multilevel alias facility. Using multilevel aliases, you can have data sets with the same high-level qualifier cataloged in *different* catalogs.

For example, your installation might have a project, PROJECT1, which requires many data sets and applications. Perhaps there are large collections of PROJECT1 data sets that have the same second qualifier: TEST or PROD. There are also some miscellaneous PROJECT1 data sets with neither TEST nor PROD as the second qualifier.

In this case, you might define catalog aliases as follows:
- Alias PROJECT1.TEST for catalog SYS1.ICFCAT.PRO1TEST,
- Alias PROJECT1.PROD for catalog SYS1.ICFCAT.PRO1PROD, and
- Alias PROJECT1 for catalog SYS1.ICFCAT.PROJECT1.

If the alias search level is 2, then data sets are cataloged as follows:

| Data Set | Catalog | Reason |
|---|---|---|
| PROJECT1.UTIL.CNTRL | SYS1.ICFCAT.PROJECT1 | The second qualifier is neither TEST nor PROD. |
| PROJECT1.PROD.DATA | SYS1.ICFCAT.PRO1PROD | There are two qualifiers, and the two qualifiers form the alias PROJECT1.PROD. |
| PROJECT1.PROD | SYS1.ICFCAT.PROJECT1 | There is only *one* qualifier in this data set name: PROJECT1. PROD is not a qualifier, so it is not used in the multilevel alias search. |
| PROJECT1.TEST.CNTRL | SYS1.ICFCAT.PRO1TEST | There are two qualifiers, and the two qualifiers form the alias PROJECT1.TEST. |

| Data Set | Catalog | Reason |
|---|---|---|
| PROJECT1.TEST.A.B | SYS1.ICFCAT.PRO1TEST | The first two qualifiers are used as the alias, since the search level is 2. The third qualifier is not used in the search. |

In this example, programs being tested (TEST) are isolated from production programs and data (PROD) and other miscellaneous files. This isolation is desirable for data protection and availability. Backup and recovery of one catalog would not affect projects using the other catalogs.

The alias search level is specified in the SYSCAT*xx* member of SYS1.NUCLEUS or LOAD*xx* member of SYS1.PARMLIB (see "Identifying the Master Catalog and Initial Configuration (SYSCATxx)" on page 37). It can also be changed without an IPL using the MODIFY CATALOG,ALIASLEVEL operator command (see "Changing the Multilevel Alias Search Level" on page 144).

When you use an alias to select catalogs, the use of a single level catalog name can also affect the selection. When a catalog has a single level name, that name behaves as an alias. For example, if a catalog has the name of ICFCAT, data sets beginning with ICFCAT, such as ICFCAT.DATA1, ICFCAT.DATA2, and ICFCAT.catalog.DATA are found in the ICFCAT catalog if the CATALOG parameter is not used to limit the search. See "Using LISTCAT in Examples" on page 63 for an example using the CATALOG parameter.

With the multilevel alias facility, this concept is extended to catalogs whose names have multiple levels. For instance, if MULTILEVELALIAS=2, and a catalog ICF.CAT exists, ICF.CAT will also behave as an alias. Data sets such as ICF.CAT.DATA1, ICF.CAT.DATA2, and ICF.CAT.catalog.DATA are found in the ICF.CAT catalog if no CATALOG parameter is used.

Using the multilevel alias facility and having short catalog names can influence generic searches for catalogs. The list of catalogs to be searched is more complex when a generic entry is not fully qualified. The generic portion of the name is searched first before all alias levels are satisfied. For example, if the name is SYS1.* and MULTILEVELALIAS=2, catalog management does not know which specific catalogs to select and selects all aliases and catalogs beginning with SYS1 and having two levels. For example, SYS1.CAT and SYS1.ALIAS could be selected. This list is then used to select the catalogs to be searched. Duplicates on the list are eliminated and then each catalog is searched for entries that match the search parameter. If SYS1.ALIAS points to ICF.CAT, then catalogs SYS1.CAT and ICF.CAT will be searched. SYS1.CAT is selected because its name matches the generic SYS1.*. ICF.CAT is selected because an alias that points to it, has a name that matches the generic SYS1.*.

Duplicate entries can appear if this alternate selection occurs. For example, if both SYS1.CAT and ICF.CAT contain an entry for SYS1.PROCLIB, SYS1.PROCLIB will be returned twice. SYS1.PROCLIB is returned once for SYS1.CAT where it was found, and once for ICF.CAT where it also occurs. You should determine which catalogs are involved because it might appear that you are receiving duplicate entries, but you can be receiving different entries with the same name from different catalogs.

In an environment with a multilevel alias level greater than 1, specifying a search such as ABC or ABC* may cause Catalog Management to orient a request to a different catalog. Currently, if catalog management fails to orient to a first catalog, it fails the request, even though it may have been a request for multiple catalogs.

For example, if a user specifies ABC, as in LISTCAT LEVEL(ABC) or ABC on ISPF option 3.4 with this search level, his request may be eligible to search 2 or more catalogs. If he doesn't have authority to access the first catalog, Catalog management fails the request because it can't orient to the first catalog.

## Choosing Aliases and an Alias Search Level

You should choose multilevel aliases carefully to avoid problems when trying to locate the data sets later. When you are using more than one alias level, the catalog is chosen that has the most matching qualifiers to the data set being searched for or cataloged.

**General Considerations for Multilevel Aliases:** Before selecting an alias level, or a specific multilevel alias, consider the following:

1. User catalog names perform like aliases. For example, if you have a catalog USER.ICFUCAT1 with the alias search level set at 2, then all data sets beginning with USER.ICFUCAT1 are cataloged in this catalog.
2. When defining an SMS-managed data set with multiple components, like an SMS-managed catalog key-sequenced data set, if a user specifies a component name that resolves to a *different* catalog than the data set itself, the definition fails.
3. When defining a catalog cluster or a generation data group, if the name of the cluster or generation data group matches an existing alias or user catalog name, the definition fails with a "duplicate name" error. This is to prevent the data or index component of the VSAM data set, or a generation data set, from becoming inaccessible.

For example, consider a situation where two user catalogs, ICFUCAT1 and ICFUCAT2, each have an alias. The alias search level is set at 3. There are two aliases defined:

1. Alias A.B for catalog ICFUCAT1
2. Alias A.B.C for catalog ICFUCAT2

If you define an entry-sequenced data set named A.B.C without specifying the data component name:

1. The cluster entry is in ICFUCAT1
2. The generated data component name is A.B.C.DATA

Since A.B.C is an alias for ICFUCAT2, any references to A.B.C.DATA are oriented to ICFUCAT2. Since A.B.C is cataloged in ICFUCAT1, any request for the data component results in a "data set not found" error.

Also, if you have a generation data group named A.B.C, it points to ICFUCAT1. However, the names of the generation data sets are in the form of A.B.C.G*xxxx*V*yy*. Even though the generation data sets are actually cataloged in ICFUCAT1 because its base is in ICFUCAT1, later references to any generation data set results in orientation to ICFUCAT2 and a "data set not found" error.

This problem can be avoided if no VSAM data set or generation data group is defined with the same name as an existing alias or catalog. It is your responsibility to ensure that when you add an alias to the catalog, you do not cause existing data sets to become inaccessible.

**Procedure for Choosing a Multilevel Alias:** You should take these steps before defining new aliases:

1. Do a LISTCAT LEVEL, using only the first qualifier of the new alias. For example, if the new alias is A.B.C, execute LISTCAT LEVEL(A). For aliases of only one level, the "first" qualifier is the alias name.
2. Check for any matches. If there are any matches and the listed data set does not currently reside in the catalog you are attempting to define the alias for, you might need to rename the data set or choose another alias if the data set becomes inaccessible.

For example, if you want to define the alias A.B for ICFUCAT2, use LISTCAT LEVEL(A) and analyze the output. In this example, A is an alias for ICFUCAT1 and A.B.C is an alias for ICFUCAT3:

```
CLUSTER ------ A.CLUSTER
     IN-CAT -- ICFUCAT1
DATA --------- A.DATA
     IN-CAT -- ICFUCAT1
CLUSTER ------ A.B.CLUSTER2
     IN-CAT -- ICFUCAT1
DATA --------- A.B.DATA2
     IN-CAT -- ICFUCAT1
CLUSTER ------ A.B.CLUSTER
     IN-CAT -- ICFUCAT1
DATA --------- A.B.DATA
     IN-CAT -- ICFUCAT1
CLUSTER ------ A.B.C.CLUSTER
     IN-CAT -- ICFUCAT3
DATA --------- A.B.C.DATA
     IN-CAT -- ICFUCAT3
INDEX -------- A.B.C.INDEX
     IN-CAT -- ICFUCAT3
ALIAS -------- A
     IN-CAT -- ICFMASTR
ALIAS -------- A.B.C
     IN-CAT -- ICFMASTR
```

Evaluation of the LISTCAT output:

The data sets A.CLUSTER and A.B.C.CLUSTER and their components will remain accessible. With an alias search level of 3, their aliases (A and A.B.C) continue to be oriented to the correct catalog.

However, A.B.CLUSTER2 and A.B.CLUSTER will become inaccessible if you define A.B as an alias for ICFUCAT2. These data sets are cataloged in ICFUCAT1, since they use the alias "A". After alias A.B is defined, searches for these data sets will be oriented to ICFUCAT2, and the data sets will not be found.

If you choose to rename a data set, the data and index components probably need to be renamed as well.

The following types of data sets might not be found by LISTCAT LEVEL:

1. Data sets defined with a catalog specified in the CATALOG parameter (bypassing the catalog search order) or with job or step catalogs.
2. Data sets whose data or index component names were defined by the user.

3. Data sets that became inaccessible because of the removal of an alias.

To identify data sets not cataloged according to catalog aliases, use the CATALOG parameter of LISTCAT, specifying the catalog containing the data set's entry. All user data sets should be cataloged according to catalog aliases. Otherwise, data can easily become lost, defeating the purpose of cataloging data.

# The Master Catalog

There is no structural difference between a master catalog and a user catalog. What makes a master catalog different is how it is used, and what data sets are cataloged in it. Each system has one active master catalog. The master catalog does not have to reside on the system residence volume.

## Contents of the Master Catalog

The master catalog for a system contains entries for all the user catalogs that are used on the system and the aliases pointing to them. Other data sets that you should catalog in the master catalog are shown in Table 3.

*Table 3. Master Catalog Entries*

| Entry types | Description/Example |
|---|---|
| User catalogs | Entries for all user catalogs used on the system. |
| Aliases | Entries for all aliases pointing to the user catalogs on the system. |
| Pagespaces | Entries for the pagespaces used on the system. |
| System software target libraries | The data sets used to run the system such as LINKLIB and ISPPENU. |
| Key operational data sets | Key operational data sets such as:<br>• PARMLIB data sets<br>• Parameter libraries for JES and TCP/IP<br>• SMS configuration data sets<br>• SMF data sets<br>• RACF databases<br>• Couple data sets |
| Subsystem data sets | Subsystem data sets, such as those for IMS™, DB2®, or CICS®, especially when they are replaced. |
| **Requirement:** Data sets required during the IPL process must be cataloged in the master catalog.<br>**Recommendation:** A master catalog should not contain entries for application or end user data sets. These should be contained in user catalogs that are pointed to by the appropriate aliases in the master catalog. | |

Figure 4 on page 25 shows the relationship between master and user catalogs.

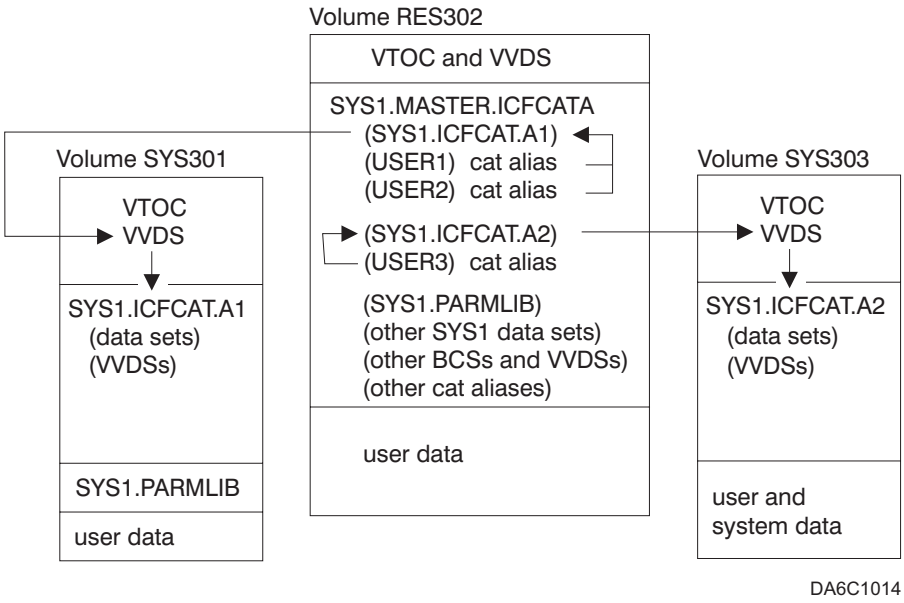



*Figure 4. Relationship Between Master and User Catalogs.* Parentheses indicate an entry in a catalog. Alias entries point to other entries in the same catalog. Arrows indicate the relationship between entries and the data sets they represent. The order of the entries in this example does not represent the order of entries in an actual catalog.

If your installation has multiple, interconnected systems, the master catalogs of each system can be connected to the master catalogs of each of the other systems. In other words, a master catalog on one system is a user catalog on the other systems.

You might want to combine SMS complexes into a single SMS complex and eliminate additional control data sets. To share the master catalog across an SMS complex see "Sharing a Master Catalog and IPL Volume" on page 28. If you are not running SMS, you can connect as many systems as is supported by the channel. RACF and appropriate alias naming conventions can prevent users on one system from cataloging data sets in the master catalog of another system. See Figure 3 on page 15 for an example of interconnected master catalogs.

For ease of backup and recovery of the master catalog, no user data sets should be cataloged in the master catalog. If you deny update access to the master catalog for most of your users, there is typically much less update activity for the master catalog.

Each system must identify its master catalog. You identify the master catalog in the LOAD*xx* member of SYS1.PARMLIB, or the SYSCAT*xx* member of SYS1.NUCLEUS. (See "Identifying the Master Catalog and Initial Configuration (SYSCATxx)" on page 37, and "Bypassing SYSCATxx with LOADxx" on page 39 for more information.)

## The Master Catalog During System Initialization

During a system initialization, the master catalog is read so that system data sets and catalogs can be located. Their catalog entries are placed in the cache that is selected by the user for the master catalog as they are read.

Catalog aliases are also read during system initialization, but they are placed in an alias table separate from the cache that is used for the master catalog.

Thus, if the master catalog only contains entries for system data sets, catalogs, and catalog aliases, the entire master catalog is in main storage by the completion of the system initialization.

## Creating and Using an Alternate Master Catalog

Because the master catalog is vital to the functioning of a z/OS system, you should create an alternate master catalog that can be used in a system initial program load (IPL) if the regular master catalog becomes damaged.

At minimum, an alternate master catalog contains entries for the system data sets necessary to IPL the system. After IPL, the original master catalog can be repaired or recovered, and the system can again be IPLed with the newly recovered master catalog.

The simplest procedure for creating an alternate master catalog is to use the access method services REPRO command to copy the master catalog into a defined new master catalog. After you have copied the master catalog into the newly defined master catalog, complete the following steps:

1. Use the newly defined catalogs as the master and the old master catalog as the alternate master catalog. This is necessary because the REPRO process changes the VVDS pointers to the output catalog. Although the system can still be IPLed with the old master, allowing you to recover the new master catalog if necessary, the new master should be used under normal circumstances.
2. Define the alternate master on a different volume than the volume of the original master. Otherwise, if the original master's volume is damaged, both the original master and alternate master are unavailable. If you desire volume IPL, initialize the new volume with IPL text, and copy the required system data sets to the new volume. Allocate the new system data sets with a different high level qualifier than SYS1 (e.g. SYS2), then rename them after they are cataloged in the alternate master. This is necessary because some system data sets are always in use. The alternate master then has entries for the same data sets defined in the original master.
3. To manually create a minimum alternate master catalog without using REPRO, create the catalog and define all the system data sets in the alternate master. Then define new storage index and page data sets.

The newly defined catalog should then be used as the master catalog, and the old master should be used as the alternate master. The alternate master should be defined on a different volume than the volume of the original master: otherwise, if the original master's volume is damaged, both the original master and the alternate master are unavailable.

After the alternate is created, changes to the entries for the system data sets are only reflected in the master catalog that is in use. If you IPL the system with the new master, changes are not reflected in the old master (now the alternate master). For example, if a system data set is moved to a different volume, update the alternate master by recreating it or recataloging the data set.

After defining the new master and copying the original master into it, create a SYSCAT*xx* member in SYS1.NUCLEUS to identify the alternate master. To use the alternate master catalog, specify at IPL time the two-character identifier of the SYSCAT*xx* member that contains the entry identifying the alternate master catalog. You can also define the alternate master in a LOAD*xx* member in SYS1.PARMLIB.

If you want to maintain the SYSCATLG member as the member identifying your current master catalog, and the SYSCATLG member points to the old master catalog, copy the member to a different SYSCAT*xx* member. Then update the SYSCATLG member to point to the catalog you just created.

The following example shows the procedure for creating an alternate master catalog. The new master catalog created is SYS1.ICFCAT.NEWMASTR. The old master catalog SYS1.ICFCAT.MASTER is used as the alternate master catalog, and is identified in the SYSCATAL member of SYS1.NUCLEUS. Both the new master and the old master use a multilevel alias search level of 1, no SYS% conversion, and the default number of CAS tasks.

## Example: Creating an alternate master catalog

```
//STEP01   EXEC  PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=A
//SYSIN     DD  *
   /*************************************************************/
   /* DEFINE THE NEW MASTER CATALOG                             */
   /* VOLUME ALTVOL DOES NOT CONTAIN THE ORIGINAL MASTER CATALOG*/
   /*************************************************************/
      DEFINE MASTERCATALOG -
            ( NAME(SYS1.ICFCAT.NEWMASTR) -
              CYLINDERS (5 1)  -
              VOLUME (ALTVOL) -
              ICFCATALOG)
/*
//STEP02   EXEC PGM=IDCAMS
//ALTERV   DD UNIT=3390,VOL=SER=ALTVOL,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
   /*************************************************************/
   /* COPY OLD MASTER CATALOG TO THE NEW MASTER CATALOG          */
   /*************************************************************/
        REPRO INDATASET(SYS1.ICFCAT.MASTER) -
              OUTDATASET(SYS1.ICFCAT.NEWMASTR)
/*
//STEP03   EXEC PGM=IEBGENER
//*************************************************************
//* COPY OLD SYS1.NUCLEUS(SYSCATLG) TO SYS1.NUCLEUS(SYSCATAL) *
//* (USE OLD MASTER AS THE ALTERNATE MASTER, NEW MASTER AS THE*
//*  DEFAULT MASTER)                                          *
//*************************************************************
//SYSPRINT DD SYSOUT=A
//SYSUT1   DD DSN=SYS1.NUCLEUS(SYSCATLG),DISP=OLD
//SYSUT2   DD DSN=SYS1.NUCLEUS(SYSCATAL),DISP=OLD
//SYSIN    DD DUMMY
//STEP04   EXEC PGM=IEBGENER
//*************************************************************
//* REPLACE OLD SYS1.NUCLEUS(SYSCATLG) MEMBER WITH POINTER TO  *
//* NEW MASTER CATALOG.                                       *
//*************************************************************
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  *
ALTVOL11  SYS1.ICFCAT.NEWMASTR
/*
//SYSUT2   DD  DSN=SYS1.NUCLEUS(SYSCATLG),DISP=OLD
//SYSIN    DD  DUMMY
```

*Figure 5. Creating an Alternate Master Catalog*

### Sharing a Master Catalog and IPL Volume

You can share a master catalog and share an IPL volume among multiple systems. The system data sets, SYS1.LOGREC and SYS1.STGINDEX are no longer fixed named and unable to be shared. They can now be shared and specified by the installation. In addition, a system symbolic, &SYSNAME, was introduced and can be used as part of data set name specifications for some parameters in PARMLIB. When you use &SYSNAME, the data set name specification becomes flexible and you do not need a separate parameter specification for each system in the sysplex.

For example, LOGREC=SYS1.LOGREC.&SYSNAME. The symbolic name, &SYSNAME, can also be used in other PARMLIB parameter specifications. You can use &SYSNAME for IEASYSxx parameters VIODSN=, PAGE=, SWAP=, DUPLEX=, and NONVIO=. You can use &SYSNAME for SMFPRMxx parameters DSNAME= and SID=.

## Using Symbolic References for Aliases

When sharing a master catalog, it might be desirable for an alias to reference a different data set or catalog, depending on the system it is accessed from. You can now define an alias for a catalog or a non-VSAM data set where the catalog name or non-VSAM data set name contains system symbols. This name is resolved at the time the alias entry is referenced, rather than when it is defined.

Aliases of data sets may not be used prior to the start of Master Scheduler Initialization. This is approximately the time that the system switches to full screen console mode. Thus, cataloged data set names contained in system initialization members that are read and processed prior to this time may NOT be specified as aliases. The actual data set names must be provided.

This capability simplifies staged introduction of products on members of a sysplex. Now a single entry in the shared master catalog can refer to different physical data sets on each member of the sysplex at the same time.

For more information, see the DEFINE ALIAS description of the SYMBOLICRELATE keyword in *z/OS DFSMS Access Method Services Commands*.

## Catalog Performance

This section is intended to help you to do tuning of DFSMS. This section documents information that is tuning information provided by DFSMS.

Performance should not be your main consideration when you define catalogs. It is more important to create a catalog configuration that allows easy recovery of damaged catalogs with the least amount of system disruption. However, there are several options you can choose to improve catalog performance without affecting the recoverability of a catalog.

### Factors Affecting Catalog Performance

The main factors affecting catalog performance are the amount of I/O required for the catalog and the subsequent amount of time it takes to perform the I/O. These factors can be reduced by caching catalogs in special caches used only by catalogs.

If the master catalog only contains entries for catalogs, catalog aliases, and system data sets, the entire master catalog is read into main storage during system

initialization. Because the master catalog, if properly used, is rarely updated, the performance of the master catalog is not appreciably affected by I/O requirements.

Performance can also be enhanced by specifying certain options when defining a catalog. These performance attributes are discussed in "Choosing Options to Adjust Catalog Performance" on page 55, and "Setting the Catalog Control Interval and Area Size" on page 50.

Sharing a catalog between systems also affects performance. See "The Effect of Sharing Catalogs on Cache Usage" on page 30.

In summary, catalog performance is mainly enhanced by caching the catalog. If a catalog is cached, the reduced I/O to the catalog reduces the effect of other factors on catalog performance. Otherwise, considerations affecting catalog recovery should take precedence over performance considerations.

## Caching Catalogs

The simplest method of improving catalog performance is to use cache to maintain catalog records within main storage or data space. Using cache reduces the I/O required to read records from catalogs on DASD.

Two kinds of cache are available exclusively for catalogs. The in-storage catalog (ISC) cache is contained within the catalog address space (CAS) in main storage. The catalog data space cache (CDSC) is separate from CAS and uses the MVS VLF component that stores the cached records in a dataspace. Both types of cache are optional, and each can be cancelled and restarted without an IPL.

Although you can use both types of catalog cache, you cannot cache a single catalog in both types of cache simultaneously. You must decide which catalogs benefit the most from each type of cache.

Catalog records are cached in the ISC or CDSC under the following conditions:

- For master catalogs, all records accessed sequentially or by key are cached except for alias records. Alias records are kept in a separate table in main storage.
- For user catalogs, only records accessed by key are cached.
- For each catalog, the records are cached in the CDSC if you have indicated the catalog is to use CDSC. Otherwise, the records are cached in the ISC, unless you have stopped the ISC for the catalog. If you stop both the CDSC and the ISC for a catalog, then records are not cached.

See "Starting and Stopping the Catalog Cache for a Catalog" on page 143 for information on using the MODIFY CATALOG command to manipulate the caches.

The ISC and VLF functions will not be used for any catalog in RLS mode; instead CAS will rely on the SMSVSAM address space to do the buffering and caching. Each buffer is independently registered with the CF cache structure connected by SMSVSAM from all sharing systems. Any record updates to the buffer from any system will trigger the CF to cross invalidate the copies in all other sharing systems. Once a buffer is invalidated, the most current copy can be read from the CF cache structure, if present, instead of performing an I/O to the DASD.

### In-Storage Catalog Cache

The in-storage catalog cache resides in main storage within the catalog address space. It is the default catalog cache. Each user catalog cached in ISC is given a

fixed amount of space for cached records. When a user catalog uses its allotted space in the ISC, the least recently used record is removed from the ISC to make room for the new entry.

Catalogs that are not frequently updated or shared with other systems use the ISC most effectively. The performance of the ISC is affected if the catalog is shared with another system. See "The Effect of Sharing Catalogs on Cache Usage" for more information.

Master catalogs, unlike user catalogs, are not limited to a set amount of storage. All eligible records in the master catalog are cached in the ISC as they are read. Thus, you should keep the number of entries in the master catalog to a minimum, so that the ISC for the master does not use an excessive amount of main storage.

Since ISC is the default catalog cache, catalogs are cached in the ISC unless you specify that the catalog is to use CDSC, or unless you use the MODIFY CATALOG operator command to remove the catalog from the ISC.

## Catalog Data Space Cache

The catalog data space cache (CDSC) resides in a data space that you define with the COFVLF*xx* member of SYS1.PARMLIB. The CDSC uses the virtual lookaside facility (VLF), which can be started using the START VLF operator command. See "Defining the Catalog Data Space Cache (COFVLFxx)" on page 40 for information on defining the CDSC.

You can add catalogs to the CDSC only by editing the COFVLF*xx* member to specify the catalogs, stopping VLF, then starting VLF. Because this releases the existing CDSC, catalog performance might be degraded for a while.

The CDSC can concurrently be used for any catalog that is not using the ISC. A single catalog cannot use both the CDSC and the ISC. Unlike the ISC, catalogs cached in the CDSC are not limited to a specific amount of storage. A catalog caches records until no space is left in the data space cache. Once the data space cache is full, the space occupied by the record least used is removed to make room for new records.

See "Monitoring the Catalog Address Space" on page 123 for information on monitoring the performance of the CDSC.

## The Effect of Sharing Catalogs on Cache Usage

If a catalog is defined with share options (3 4), and if it resides on a shared device, catalog management considers the catalog a shared catalog. A catalog is considered a shared catalog if it meets both of these conditions, even if it is not actually being shared among systems. Before each physical access to a shared catalog, special checking is performed to ensure that the ISC or CDSC contains current information. Checking also ensures that the access method control blocks for the catalog are updated in the event the catalog has been extended, or otherwise altered from another system. This checking maintains data integrity. It also affects performance because the VVR for a shared catalog must be read before using the ISC or CDSC version of the BCS record.

**Recommendation:** To avoid catalog corruption, define a catalog volume on a shared UCB and set catalog share options to (3 4) on all systems sharing a catalog.

A single catalog request can involve many physical references to the catalog, so the effect of this additional access to the VVDS can be significant. The volume

containing the VVDS will be reserved, and I/O will be performed. If the catalog has been defined to support Enhanced Catalog Sharing (ECS) by the ECSHARING attribute, most of this overhead will be eliminated.

Changes to shared catalogs are handled differently depending on whether the catalog uses the ISC or the CDSC.

If a catalog uses the ISC and a sharing system updates a record (any record, even if the record is not cached in this system's ISC), catalog management releases the *entire* ISC for the catalog and creates a new ISC for the catalog. Individual records changed by a sharing system are not identified and updated for ISC catalogs.

The CDSC, however, can identify individual records that a sharing system has updated. Thus, when a sharing system updates a record, the CDSC space used by the catalog is not necessarily released. CDSC space for a catalog is only "invalidated" (marked unusable and given back to the CDSC as "free space" which any catalog can use) if so many changes were made by a sharing system that catalog management could not maintain a record of all the changes. Otherwise, all changes made by the sharing system can be made to the CDSC record by record.

If a catalog is not really shared with another system, move the catalog to an unshared device or alter its share options to (3 3). To prevent potential catalog damage, never place a catalog with share options (3 3) on a shared device.

## Diagnosing a Catalog Performance Problem

This section documents the information you can use to diagnose and report a catalog performance problem, including:

- "Reporting a Catalog Performance Problem to the IBM Support Center"
- "Using GTF tracing for further diagnosis of catalog performance problems" on page 32
- "Possible causes and solutions for catalog performance problems" on page 33

### Reporting a Catalog Performance Problem to the IBM Support Center

In order to report a catalog performance problem when running DFSMS with z/OS V1R7 or higher, you must do the following:

1. Collect the following information:
   a. What information indicated an increase in CPU usage?
   b. Was this increase noticeable immediately after upgrading to a new release of z/OS?
   c. What release level was used as base for comparison?
   d. What are the PUT/RSU levels of the base used for comparison?
   e. Were any changes made in the catalog environment, such as:
      - Any new shared catalogs, or old catalogs that are now shared?
      - VLF or ISC changed catalogs?

        Note that SMF record type 41, subtype 3 gives details on VLF hit ratios and trimming for each VLF class. Based on this information, you may want to increase the MAXVIRT value in the COFVLFxx parmlib member for the IGGCAS VLF CLASS. Refer to VLF documentation in *z/OS MVS Programming: Authorized Assembler Services Guide* and *z/OS MVS Initialization and Tuning Reference*.
      - STRNO changes to catalogs?

- CATMAX or TASKMAX changes?
- Any other relevant changes to the system environment?

f. Have there been any increases in workload on the system, such as new applications, the addition of more TSO users, or increased batch workloads?

g. Have you noticed any increases in CPU consumption for other system address spaces, such as GRS, SMS, MASTER, or JES?

h. Are there specific time frames of increased Catalog CPU usage?

i. Are there specific jobs or users affected by the problem?

j. What type of serialization product are you using. For example, is GRS your serialization product?

k. Is the problem affecting overall system performance?

l. Do you have any DASD UCBs specified as LOCANY=YES in HCD? If so, when were the UCBs put above the line?

If the answers to these questions point to an overall increase in CAS CPU consumption (as opposed to specific users or jobs) then proceed to step 3. Otherwise, refer to "Possible causes and solutions for catalog performance problems" on page 33.

2. Collect comparison data of CAS CPU consumption between the base (previous) release and new DFSMS release. This information should represent total weekly or monthly CPU consumption for CAS broken down by TCB and SRB. If you are using RMF™, this is easiest if the Catalog address space is in a report performance group. You can use OEM applications that provide similar information so that a comparison can be made between the previous release and the current release.
3. Collect comparison data of CAS storage size between the base or previous release and new release of DFSMS.
4. Collect comparison data of CAS I/O rates between the base or previous release and new release of DFSMS.
5. If OMEGAMON® is available, then do an INSPECT of CAS drilling down on a few of the active TCBs to a CSECT level and the offsets within those CSECTs that are consuming CPU. Other OEM applications provide a similar capability. The important point is to get to the CSECT level of load modules. For example, you might find CSECT IGG0CLF5 in load module IGG0CLX0.

## Using GTF tracing for further diagnosis of catalog performance problems

You may also find the following information helpful for diagnosing catalog performance problems:

- Collect GTF trace data from the User and Catalog address space for DSP,SVC=(19,20,26,48,56,99) and set the following SLIP:

```
SLIP SET,IF,JOBNAME=jjjjj,ACTION=TRACE,LPAMOD=(IGC0002F,xxx),
        TRDATA=(STD,REGS,
        12R?+278?,+14,                    <===CAMLST
        12R?+278?+4?,+2C,                 <===ENTRY NAME
        12R?+270?,+1C,                    <===CTGPL
        11R?+CD4?,+1000,                  <===GFL workarea
        12R?+270?+4?,+2C,                 <===ENTRY NAME
        12R?+260,+263),END                <===CATMAN RC/RSN
        Where: jjjjj = the batch jobname or TSO userid
               xxx = the offset of procedure RESMCNTL in module
                     IGC0002F which is X'71A' for all current
                     releases of DFSMS (z/OS 1.7 thru 1.10).
```

Now, start GTF with the following options:

```
TRACE=SLIP,DSP,SVCP,JOBNAMEP
```

when prompted for a reply, enter:

```
R XX,SVC=(19,20,26,48,56,99),JOBNAME=(JJJJ,CATALOG),END.
```

Examine the time from the start of the SVC26 until the instruction fetch entry. The time for that should be in the range of 10-15 milliseconds.

- Collect GTF trace data with CCW data for the device where the Catalog resides. In the trace data you may see multiple reads to the VVDS, which is normal, or you may see continual re-reading of the same BCS record, which indicates a problem.

## Possible causes and solutions for catalog performance problems

The following suggestions can help you solve or avoid catalog performance problems:

1. If possible, convert from GRS ring to GRS star. The performance of GRS star is much better than GRS ring. See *z/OS MVS Planning: Global Resource Serialization*.
2. If you must use GRS ring check to see if the RESMIL parameter is set too high in the GRSCNFxx parmlib member. Refer to *z/OS MVS Planning: Global Resource Serialization* for details, but in general, this value should be set at 0 or OFF. Note that this suggestion only applies to a GRS Ring configuration
3. Make sure that the service class for CATALOG or GRS has not been changed from the default of SYSTEM.
4. Is the CATMAX setting too low? Issue an `F CATALOG,ALLOCATED` command to see how many catalogs are open and allocated. The value for CATMAX should be higher than the number of active catalogs.
5. Is the STRNO value for the catalog too low? A low STRNO value is indicated by queue wait times on SYSZRPLW/catname. Consult RMF ENQ delay reports to determine specific catalogs that might need to have STRNO bumped. For a highly active catalog, the default value of 2 should be increased to 5 or 7. Issue the following command to find the current STRNO value for the catalog:

   ```
   LISTCAT ENT(catname) ALL CAT(catname)
   ```

6. Are VLF or ISC active? Issue `F CATALOG,VLF` and `F CATALOG,ISC` to activate these options.
7. Catalog enqueue and enqueue resource problems: The resource SYSIGGV2 MUST be converted to a SYSTEMS enqueue when using catalogs in ECS mode. Failure to do so will result in damage to catalogs in ECS mode. This is applicable to all levels of DFSMS that support ECS (HDZ11F0, HDZ11G0, HDZ11H0, HDZ11J0, HDZ11K0, HDZ1180, HDZ1190, HDZ1A10).

   Note that you must treat resources SYSZVVDS and SYSVTOC the same way, either both converted or both excluded to prevent deadlocks
8. Do not place an entry for SYSIGGV1 in the SYSTEM inclusion RNL. Every SYSIGGV1 request that needs the SYSTEMS attribute already has it. Placing an entry for SYSIGGV1 in the SYSTEM inclusion RNL can degrade performance.
9. ISGAUDIT use: Due to the high number of enqueues and done by the catalog address space, if ISGAUDIT is being used and the SYSZVVDS and/or SYSIGGV2 names are not being filtered out, it is possible to see high CPU numbers attributed to the Catalog Address Space as a result of monitoring these. These can be removed from the monitor list to correct this.
10. Ensure there are no PER Slip Traps enabled, especially if more than one address space is involved.

11. If you are using a serialization product other than GRS, please review GRS APARs OW56028, OA01861 and OA01695.These APARs added new exit points in GRS for use by OEM products. Ensure that any GRS maintenance related to these exits and any maintenance provided by OEM applications that use these exits is installed.
12. WTO Buffer Shortage: This is indicated by the presence of system message IEA405E. When this occurs, the system issues a GQSCAN macro, which requires the CEDQ lock - this is a normal lock required by ENQ processing. Therefore due to the high number of ENQs required by normal SVC26 processing, catalog processing slows down and CPU usage appears high for the catalog address space.
13. In HDZ11G0 and higher versions of DFSMS, I/O statistics for catalogs and the Catalog Address Space will appear differently than earlier releases. Prior to z/OS V1R3, VSAM did the I/O to VSAM data sets, including catalogs. Starting with HDZ11G0 VSAM uses Media Manager to do all I/O. Prior to HDZ11G0 VSAM specifically omits the collection of Start-I/O or block counts when accessing a catalog. Media Manager does not differentiate between I/O to catalog or another type of data set. You may now see higher I/O counts for Catalog Address Space I/O requests. The actual I/O rates have not changed, simply the reporting of them.
14. To improve IDCAMS EXPORT processing of catalogs, specify the BUFND, BUFNI and BUFNO parameters. To specify BUFND and BUFNI you will need to use the INFILE parameter for EXPORT. Sample JCL is below:

```
//EXPRTCAT EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//INCAT    DD   DSN=MY.CATALOG,DISP=SHR,
//  AMP=('BUFND=XXX','BUFNI=YYY')
//OUTCAT   DD   DSN=MY.EXPORTED.CATALOG,DISP=(NEW,CATLG),
//  UNIT=SYSDA,SPACE=(CYL,(10,10)),BUFNO=ZZ
//SYSIN    DD   *
  EXPORT MY.CATALOG -
  INFILE(INCAT) -
  OUTFILE(OUTCAT) -
  TEMPORARY
/*
```

    To calculate the value for BUFND, use the number of CI's per CA for data component of the catalog. For BUFNI, compute the number of index records by dividing the High Used RBA of the index component by the index component CISIZE and add a value of 5 to 10 to that calculation. For BUFNO (ZZ) use a value in the range of 30 to 40.

## Defining the Catalog Configuration

The system data sets SYS1.NUCLEUS and SYS1.PARMLIB contain members used to define portions of your catalog configuration.

When you IPL your z/OS system, you must identify the master catalog to z/OS. This can be done either through the SYSCAT*xx* member of SYS1.NUCLEUS or the LOAD*xx* member of SYS1.PARMLIB.

If you use the SYSCAT*xx* member, z/OS issues the following message asking you to identify which member you are using. This is done during the nucleus initialization program (NIP) time. You respond with the last two characters in the SYSCAT*xx* member name (the *xx* value):

```
IEA347A SPECIFY MASTER CATALOG PARAMETER
```

If you enter a blank line in response to this message, z/OS uses "LG" as the parameter, identifying SYSCATLG.

If you use the LOAD*xx* member to identify the master catalog, the system uses the master catalog specified in the member, and the master catalog message is not issued.

MVS also uses information you define in the COFVLF*xx* member of SYS1.PARMLIB to identify the catalogs that are candidates for the catalog data space cache (or VLF cache). Catalogs not identified in the COFVLF*xx* member are eligible for the standard in-storage catalog cache.

Another member of SYS1.PARMLIB, SMFPRM*xx*, is used to identify the record types that the system management facilities records. SMF can be used to record changes to catalogs.

## The intersection of SYSCATxx, LOADxx, IGGCATxx, and MODIFY CATALOG command

There are several places to define catalog configuration values: SYSCATxx, LOADxx, IGGCATxx, and the MODIFY CATALOG command. Table Table 4 shows which attributes you can specify in each. However, for complete information, see the following:

- SYSCATxx member of SYS1.NUCLEUS, see "Identifying the Master Catalog and Initial Configuration (SYSCATxx)" on page 37.
- LOADxx member of SYS1.PARMLIB, see *z/OS MVS Initialization and Tuning Reference*.
- IGGCATxx member of SYS1.PARMLIB, see *z/OS MVS Initialization and Tuning Reference*.

The system applies the parameters in IGGCATxx, LOADxx, and SYSCATxx in the following order:

1. Parmlib member IGGCATxx, if specified, takes the highest priority, followed by...
2. Parmlib member LOADxx followed by...
3. SYSCATxx member of SYS1.NUCLEUS followed by...
4. System defined defaults

IGGCATxx overlaps extensively with the MODIFY CATALOG command, as you can see in Table 4. The system applies IGGCATxx values at IPL time as well as at the restart of the Catalog Address Space, while command configuration changes are temporary, lasting only until the next IPL.

**Note:**

Some of these attributes are specified by subparameters of the CONTENTION, ECSHR, ENABLE, or DISABLE parameters of the MODIFY CATALOG command.

*Table 4. Intersection of SYSCATxx, LOADxx, IGGCATxx, and MODIFY CATALOG command*

| Keyword | Attribute | IGGCATxx | LOADxx | SYSCATxx | MODIFY CATALOG |
|---|---|---|---|---|---|
| | Master catalog name | No | Yes | Yes | No |
| | Master catalog type | No | Yes | Yes | No |

*Table 4. Intersection of SYSCATxx, LOADxx, IGGCATxx, and MODIFY CATALOG command (continued)*

| Keyword | Attribute | IGGCATxx | LOADxx | SYSCATxx | MODIFY CATALOG |
|---|---|---|---|---|---|
| | Master catalog volume | No | Yes | Yes | No |
| **ALIASLEVEL** | Number of data set name qualifiers to be used in the multilevel alias facility catalog search | Yes | Yes | Yes | Yes |
| **ALLOCLCK** | Contention detection settings for ALLOCLCK resource | Yes | No | No | CONTENTION |
| **AUTOADD** | AUTOADD function | Yes | Yes | Yes | ECSHR |
| **CATMAX** | Maximum number of catalogs that can be opened concurrently in CAS | Yes | No | No | Yes |
| **DELFORCEWNG** | Warning message IDC1997I or IDC1998I | Yes | No | No | ENABLE \| DISABLE |
| **DELRECOVWNG** | Warning message IDC1999I | Yes | No | No | ENABLE \| DISABLE |
| **DSNCHECK** | Syntax checking on names being added to a catalog | Yes | No | No | ENABLE \| DISABLE |
| **DUMPOFF**<br>**DUMPON** | CAS dynamic dumping | Yes | No | No | Yes |
| **EXTENDEDALIAS** | Ability to create extension records for user catalog aliases on the current system | Yes | No | No | ENABLE \| DISABLE |
| **GDGFIFOENABLE** | FIFO ordering of GDG | Yes | No | No | No |
| **NOTIFYEXTENT** | Monitor percentage of the maximum extents possible for a currently allocated catalog | Yes | No | No | Yes |
| **SYMREC** | Creation of SYMREC records | Yes | No | No | ENABLE \| DISABLE |
| **SYS%OFF**<br>**SYS%ON** | SYS% conversion | Yes | No | Yes | Yes |
| **SYSIGGV2** | Contention detection settings for SYSIGGV2 resource | Yes | No | No | CONTENTION |
| **SYSZTIOT** | Contention detection settings for SYSZTIOT resource | Yes | No | No | CONTENTION |
| **SYSZVVDS** | Contention detection settings for SYSZVVDS resource | Yes | No | No | CONTENTION |
| **TAPEHLQ** | Tape volume catalog high level qualifier | Yes | Yes | Yes | No |
| **TASKMAX** | Maximum number of concurrent user service tasks | Yes | No | No | Yes |
| **TASKMIN** | CAS service task lower limit (minimum number of CAS service tasks or requests) | Yes | Yes | No | Yes |
| **TASKTABLESIZE** | CAS task table size | Yes | No | Yes | No |
| **UPDTFAIL** | Warning message IEC390I | Yes | No | No | ENABLE \| DISABLE |
| **VVDSSPACE** | Catalog volume data set space allocation | Yes | No | No | Yes |

*Table 4. Intersection of SYSCATxx, LOADxx, IGGCATxx, and MODIFY CATALOG command (continued)*

| Keyword | Attribute | IGGCATxx | LOADxx | SYSCATxx | MODIFY CATALOG |
|---|---|---|---|---|---|
| **VVRCHECK** | Enhanced VVR checking on VVDS I/O | Yes | No | No | ENABLE \| DISABLE |

If you wish to suppress the IEA347A message that prompts for the master catalog name at IPL time, you must use the LOADxx member. Note that this means that the system uses the default TASKTABLESIZE value. If you want a larger TASKTABLESIZE , you must specify it in the SYSCATxx parmlib member and you cannot supress the IPL prompt.

## Identifying the Master Catalog and Initial Configuration (SYSCATxx)

| Attributes Defined in SYS1.NUCLEUS(SYSCATxx) |
|---|
| Master catalog's volume |
| Master catalog type |
| SYS% facility on or off |
| Multilevel alias search level |
| Catalog address space service task lower limit |
| Master catalog name |
| Tape volume catalog high level qualifier |
| AUTOADD indicator |
| CAS Service Task Table Size (TASKTABLESIZE) |

Once you have defined your master catalog, and have cataloged the necessary system data sets in it, you must identify the master catalog to the system to use it as a master catalog. The SYSCAT*xx* member of SYS1.NUCLEUS contains the information for identifying the master catalog to MVS.

If you use the LOAD*xx* member to identify the master catalog, the system uses the master catalog specified in the member, and the master catalog message is not issued.

You can have multiple SYSCAT*xx* members. The default name is SYSCATLG. Each copy of the SYSCAT*xx* member can identify different master catalogs.

The following is the format of the SYSCAT*xx* member of SYS1.NUCLEUS (the member can contain only one record):

**Byte** **Contents**

**1** volser

is the volume serial number of the master catalog's volume. This must be in the first 6 bytes of the record.

**7** catalog type

is the catalog type and SYS% facility default, where:

**blank or 0**
Indicates a catalog master catalog. In this case, leave bytes 8, 9, and 10 blank. Only the volume serial number and the catalog name can be specified for a master catalog.

**1** indicates a catalog, with the SYS% facility turned *off*,

**2** indicates a catalog with the SYS% facility turned *on*.

This value must be defined in the seventh column of the record. The setting for the SYS% conversion facility can be changed after IPL for a current session with the MODIFY CATALOG,SYS% operator command.

**8** alias level

specifies the multilevel alias search level. The default is 1 and the maximum is 4.

This value must be defined in the eighth column of the record. If you want the default value of 1, either specify 1 or leave the eighth column blank.

This value can later be changed for a current session with the MODIFY CATALOG,ALIASLEVEL operator command.

**9** CAS task low limit

specifies the catalog address space service task lower limit (maximum number of CAS service tasks or requests), in hexadecimal. The value can range from X'18' to X'FF', but the default is X'3C'.

This value must be defined in the ninth and tenth columns. Specify the default value (either explicitly, or by leaving these two columns blank). If the catalog address space needs more services tasks, it creates them.

**11** catalog name

specifies the name of the master catalog. The name can be up to 44 characters.

To change to a different master catalog, you must IPL the system and specify a different SYSCAT*xx* member.

**55** tape volume catalog high level qualifier (optional, otherwise blank)

specifies the first name qualifier of all volume catalogs (volcats) in the system. The value is specified as 1 to 8 characters. See “Defining Names for a Tape Volume Catalog” on page 59. Leave this blank if you have no tape volume catalog member.

**63** AUTOADD feature (optional, otherwise blank)

specifies that the autoadd function is enabled when the first connection is made to the coupling facility by the catalog address space.

The value **Y** must be specified in column 63 of the record.

**65** TASKTABLESIZE (optional, otherwise blank)

is a three-digit decimal number that specifies the size of the service task table allocated by CAS in common storage. The maximum value is 400. If no value is supplied, or the value supplied is less than 200, a value of 200 is used.

**Note:**

1. 10% of the service tasks specified are reserved for system use, such as recursive catalog calls resulting from the need to allocate catalogs for a request. The remaining service tasks are available for user catalog requests.
2. This parameter cannot be specified in the SYSCAT entry of a LOAD*xx* member.

TASKMAX, which you can specify in parmlib member IGCATxx or the Modify Catalog command, differs from this value in that it is a soft limit on the maximum number of concurrent non-CAS service requests that can be running at any given time and can be changed at any time. TASKTABLESIZE, on the other hand, can only be changed with an IPL. TASKMAX can be no larger than 90% of TASKTABLESIZE.

The following is a sample step to create the SYS1.NUCLEUS member SYSCATLG. This job describes a master catalog named SYS1.MASTERA.ICFCAT on volume SYSRES. The SYS% to SYS1 conversion facility is turned off. The multilevel alias search level is set at one. The catalog address space service task lower limit default of X'3C' is used by leaving columns 9 and 10 blank.

```
//SYSCAT  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT2   DD DSN=SYS1.NUCLEUS(SYSCATLG),DISP=SHR,DCB=(RECFM=U)
//SYSUT1   DD *
SYSRES11  SYS1.MASTERA.ICFCAT                                    y
//*
```

## Bypassing SYSCATxx with LOADxx

| Attributes Defined in SYS1.PARMLIB(LOADxx) |
| --- |
| Same attributes as SYS1.NUCLEUS(SYSCAT*xx*) |

The LOAD*xx* member of SYS1.PARMLIB can be used to specify the master catalog and other information needed to initialize a system. If the master catalog is specified in this member, the operator is not asked to specify a SYSCAT*xx* member during system initialization.

To identify the master catalog in the LOAD*xx* member, use the SYSCAT statement. The following is the format of the SYSCAT statement for the LOAD*xx* member:

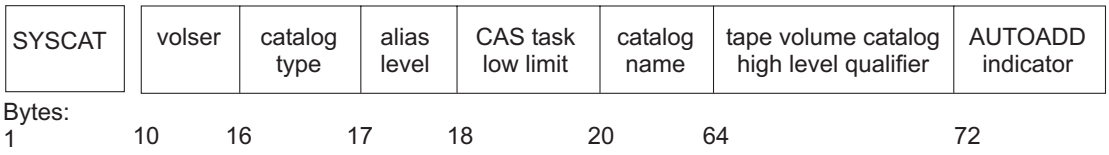



*Figure 6. The SYSCAT statement for the LOADxx member.*

The information following the SYSCAT keyword is the same as that in the SYSCAT*xx* member of SYS1.NUCLEUS. (See "Identifying the Master Catalog and Initial Configuration (SYSCATxx)" on page 37 for an explanation of the values.) The high level qualifier of the tape volume catalog is specified with 1 to 8 characters or the default is SYS1. When the system is initialized with SYSP=*xx*, the tape volume catalog high level qualifier specified in the LOAD*xx* member is activated.

The following SYSCAT statement in LOAD*xx* replaces the SYSCATLG member created in "Identifying the Master Catalog and Initial Configuration (SYSCATxx)" on page 37:

```
SYSCAT   SYSRES11  SYS1.MASTERA.ICFCAT                                  y
```

For complete information on the format and use of the LOAD*xx* member of SYS1.PARMLIB, see *z/OS MVS Initialization and Tuning Reference*.

## Defining the Catalog Data Space Cache (COFVLFxx)

| Attributes Defined in SYS1.PARMLIB(COFVLFxx) |
| --- |
| Catalogs eligible for the catalog data space cache<br>Amount of storage used for the catalog data space cache |

The COFVLF*xx* member of SYS1.PARMLIB determines which catalogs are eligible for the catalog data space (VLF) cache. Tape volume catalogs can be in the VLF cache. The cache is not actually started until you enter the START VLF operator command, which is explained in *z/OS MVS System Commands* . For an explanation of the catalog data space cache, and how to determine which catalogs should be eligible for data space caching, see "Caching Catalogs" on page 29.

Although the COFVLF*xx* member can also be used by other components, the discussion here focuses strictly on its use with catalogs. For more information on this SYS1.PARMLIB member, see *z/OS MVS Initialization and Tuning Reference*.

Besides specifying the catalogs eligible for the catalog data space cache, you can also specify the amount of storage that you will allow VLF to use to cache catalog records.

The syntax of the CLASS statement for defining the catalog data space cache is:

| Statement | Keywords |
| --- | --- |
| **CLASS** | **NAME(IGGCAS)**<br>**EMAJ***(catname)*<br>**[EMAJ***(catname)***]**<br>**[...]**<br>**MAXVIRT({ 4096**\|*size***})** |

where:

**NAME(IGGCAS)**
specifies the class name for the catalog data space cache.

**EMAJ(***catname***)**
specifies the name of a catalog eligible for catalog data space caching.

You can include any number of EMAJ parameters on the CLASS statement, but each EMAJ parameter must specify one, and only one, catalog.

Catalogs cached in the catalog data space cache cannot be simultaneously cached in the in-storage catalog. For a complete discussion of caching catalogs, see "Caching Catalogs" on page 29.

**MAXVIRT({4096**|*size***})**
specifies the maximum virtual storage that VLF can use to cache catalog

records. The decimal value is multiplied by 4096 (4K blocks). The minimum value for size is 256 decimal (256 equates to one megabyte). The maximum value for size is 524288 decimal. If you specify a decimal value that is not in the valid range, VLF defaults to 4096 decimal. To determine the optimum size for MAXVIRT, see"Evaluating Catalog Data Space Cache Performance" on page 127. You can also refer to the *z/OS MVS Initialization and Tuning Reference*.

For example, the following CLASS statement identifies four catalogs as eligible for catalog data space caching:

```
CLASS   NAME(IGGCAS)
        EMAJ(USER.ICFCAT.PROJECT1)
        EMAJ(USER.ICFCAT.PROJECT2)
        EMAJ(USER.ICFCAT.PRODUCT1)
        EMAJ(USER.ICFCAT.PRODTEST)
        MAXVIRT(256)
```

You can monitor and evaluate the performance of the catalog data space cache with the information supplied by the MODIFY REPORT,CACHE operator command. See "Monitoring the Catalog Address Space" on page 123 for more information.

## Recording SMF Records for Catalog Events (SMFPRMxx)

You can use the system management facilities to record certain catalog events. These records can be used to keep track of catalog activity, and can be used during catalog recovery. These records are also valuable in diagnosing catalog problems, and should always be recorded.

With complete SMF information, you can determine which backup copy of a catalog is the most recent, and you can use the Integrated Catalog Facility Forward Recovery Utility to update a newly recovered catalog. You can also develop reporting applications to analyze or track catalog usage.

Table 5 lists the SMF record types used with the catalog. Define the SMFPRM*xx* member of SYS1.PARMLIB so that these record types are recorded. For complete information about defining this member, see *z/OS MVS Initialization and Tuning Reference*.

You can use the IFASMFI6 macro to map the fields of these records. For more information about using SMF data and the IFASMFI6 macro, and about the contents of these record types, see *z/OS MVS System Management Facilities (SMF)*.

*Table 5. SMF Record Types used with Catalogs*

| Type | Description |
|---|---|
| 36 | Records successful exports of catalogs. Contains the date and time of the export, and information needed for importing the backup copy. Written on successful completion of EXPORT. |
| 60 | Records any changes to VVDS records, both VVRs and NVRs. Written when a VVDS record is inserted, updated, or deleted. |
| 61 | Records changes to the BCS during DEFINE processing. Written when a record is inserted or updated. |
| 65 | Records changes to the BCS during DELETE processing. Written when a record is deleted or updated. |
| 66 | Records changes to the BCS during ALTER processing. Written when a record is updated, inserted, or deleted. |

## Using Enhanced Catalog Sharing Mode

ECS mode provides a substantial performance benefit for catalogs that are shared between one or more systems in a sysplex.

To use ECS mode, you must include the ECS structure in the Coupling Facility Resource Manager (CFRM) policy. You must also define (or alter) one or more catalogs to set the ECSHARING attribute.

### Defining the CFRM Policy for Enhanced Catalog Sharing

The following ECS structure information should be included in the IXCMIAPU job that formats the couple data set containing the Coupling Facility Resource Manager (CFRM) policy:

**Name:** SYSIGGCAS_ECS

**Size:** Use the following parameters to determine the maximum structure size. See the *ES/9000 Processor Resource/Systems Manager™ Planning Guide* for the algorithms that use these parameters to estimate the size of the cache structure. You can also use the CFSIZER to determine how large the structure should be and to generate the JCL. CFSIZER is available at the following website: *www.ibm.com/s390/pso/*

- Total directory entry count (TDEC): The expected total number of ECS-active catalogs in the sysplex at any one time plus 25%. The 25% is a recommended buffer so that the structure never becomes full. If it does, then the CF default algorithm of reclaiming the least-recently-used entries will take effect. If the structure ever does become full, either catalogs will not be activated or a performance degradation will be experienced for some catalogs because of the overhead involved in rebuilding entries that were reclaimed. The maximum number of ECS-active catalogs that are supported on any one system is 1024.
- Total data area element count (TDAEC): Same as TDEC
- Maximum storage class (MSC): 1
- Maximum castout class (MCC): 1
- Maximum data area size (MDAS): 1
- Data area element characteristic (DAEX): 4
- Adjunct Assignment Indicator (AAI): 0 (zero)
- Directory portion of directory-to-data ratio (R_de): 1
- Data portion of directory-to-data ratio (R_data): 1

The size of the structure must be supplied by the installation in the CFRM policy. (The ECS facility does not specify a size.)

**Initial Size:**
Since ECS supports dynamic structure size alteration, an initial size (with a buffer smaller than the recommended 25% for the maximum size described above) can be specified in the CFRM policy. See "Managing Coupling Facility Resources" in *z/OS MVS Setting Up a Sysplex* for information on specifying an initial size.

**Preference List:**
Determined by the installation. The ECS cache structure does not require a nonvolatile CF.

**Exclusion List:**
Determined by the installation.

The ECS cache structure supports the rebuild and alter functions.

## Enabling a Catalog for ECS Mode

A new attribute, ECSHARING, is available on the IDCAMS DEFINE and ALTER commands. Setting this attribute will make a catalog eligible for sharing using the ECS protocol, as opposed to the VVDS protocol. However, the system will not actually use the ECS protocol unless:

- There is an active connection to the ECS cache structure AND
- The ECS mode has been activated by the MODIFY CATALOG, ECSHR(AUTOADD) command.

You can alter the ECSHARING attribute of a catalog at any time. If you remove the attribute and the catalog is currently using ECS mode, it will be converted back to VVDS mode on all systems that are sharing it. This is convenient if you find that the catalog must be accessed by a system that does not support the ECS protocol. A catalog cannot be shared using both the ECS and VVDS protocols at the same time. Therefore, it is necessary to stop ECS mode for a catalog if access is needed by a system that does not support the ECS protocol. This can be temporarily accomplished by using the MODIFY CATALOG ECSHR(REMOVE,catname) command. This does not remove the ECSHARING attribute, but does remove the catalog from ECS mode on all systems. It can only be returned to ECS mode by issuing a MODIFY CATALOG ECSHR(ENABLE,catname) command.

## Restrictions on ECS Mode Usage

There are several restrictions on using ECS mode:

- ECS and VVDS protocols cannot be used simultaneously for a catalog. This is enforced by the catalog address space. If you attempt to use a catalog that is currently ECS-active from a non-ECS system in the sysplex, the associated catalog request fails with return code RC228 and reason code RSN26.

  **Warning:** Access to ECS catalogs is restricted to within a parallel sysplex. Attempts to alter ECS-active catalogs from outside the sysplex result in data integrity errors. If you must access a catalog that is or has been ECS-active in a sysplex from outside the sysplex, you must first explicitly deactivate the catalog by issuing the following operator command from one of the systems within the sysplex:

  ```
  MODIFY CATALOG,ECSHR(REMOVE,catname)
  ```

  If you must access multiple or an unknown number of ECS-eligible catalogs outside the sysplex, follow the procedure described in "Disconnecting from the structure" on page 45 to disconnect all systems in the sysplex from the structure. Taking a system down or out of the sysplex does NOT properly deactivate ECS-active catalogs.
- No more than 1024 catalogs can currently be shared using ECS from a single system.
- All systems sharing the catalog in ECS mode must have connectivity to the same Coupling Facility, and must be in the same global resource serialization (GRS) complex.
- When you use catalogs in ECS mode, convert the resource SYSIGGV2 to a SYSTEMS enqueue. Otherwise, the catalogs in ECS mode will be damaged.

## Activating ECS

To activate ECS, perform the following steps:

1. Define the ECS structure in the CFRM policy and activate the policy. This action should connect all ECS-eligible systems to the ECS structure.
2. Specify the ECSHARING attribute for the desired catalogs so that they are ECS-eligible.
3. Issue the MODIFY CATALOG,ECSHR(AUTOADD) command on one system to enable AutoAdd throughout the sysplex. (AUTOADD can also be enabled at IPL time, see "Operational Considerations" for details on how to do this.) This will cause all catalogs that are eligible to be automatically activated on their next reference.

   **Restriction:** Autoadd should never be enabled if all participating systems are not connected because once autoadd is enabled, it will automatically add catalogs to the ECS structure on the next reference on each system. For example, if System A is connected and System B is not connected and autoadd is enabled, catalogs will automatically be added on System A, but not on System B. Therefore, two systems would try to share the same catalog using different sharing protocols (System A uses ECS; System B uses VVDS sharing). Because data integrity would be compromised in this situation, ECS detects and prevents this situation by rejecting System B's catalog requests with RC228 RSN26. If this situation occurs, issue the MODIFY CATALOG, ECSHR(DISCONNECT) command on all connected systems.
4. If a catalog is eligible (has the ECSHARING attribute) but still does not activate automatically and the MODIFY CATALOG, ECSHR(STATUS) command shows a status of Inact(NonECSAcc) because the catalog is currently available for only non-ECS users, then use the MODIFY CATALOG,ECSHR(ENABLE,catname) command to activate the catalog.).

## Operational Considerations

This section includes information about the following operational considerations:

- Connecting to the structure
- The autoadd function
- Quiecsing ECS activity
- Disconnecting from the structure
- Rebuilding the ECS structure

**Connecting to the Structure:** The Catalog Address Space (CAS) automatically connects to the ECS structure in the coupling facility as soon as possible during its initialization. Operator message IEC377I is issued to indicate the success or failure of this action. (You will normally see this message two times during the IPL process.) If your CFRM policy does not have the ECS structure defined, you will receive the "Not Connected" variation of the IEC377I message; this is an informative message and can be ignored. If at any time the connection is lost, you can connect (or disconnect) to (or from) the structure using the MODIFY CATALOG command.

**The Autoadd Function:** Autoadd adds ECS-eligible catalogs to the ECS structure on the next reference to the catalog. When CAS connects to the ECS structure, it checks to see if autoadd is already enabled for the sysplex. If so, CAS checks to see if autoadd is enabled for this system as well. If autoadd is not enabled, ECS will not automatically enable it. You can activate autoadd automatically during IPL by turning on the autoadd indicator in the SYSCATxx member of SYS1.NUCLEUS by specifying the value y in column 63, as described in "Identifying the Master Catalog and Initial Configuration (SYSCATxx)" on page 37. Or, an operator can manually activate autoadd if there are no ECS-ineligible systems that will be

blocked from catalog access by the activation of the ECS autoadd function. Autoadd remains active until one of the following events occur:

- ECS activity is quiesced in the sysplex. (ECS activity is quiesced during a rebuild but reactivates autoadd at the completion of the rebuild unless a rebuild error occurs.) See "Quiecsing ECS Activity."
- All systems in the sysplex are down at the same time, such as during a power outage.
- Only one system is ECS-active and a CAS restart or an IPL is performed.

**Quiecsing ECS Activity:** ECS quiesces activity by turning off the autoadd function and removing all ECS-active catalogs from the ECS structure. All catalogs then revert to VVDS sharing mode. ECS activity is quiesced in a sysplex when one of the following events occur:

- An ECS-active system normally disconnects from the ECS structure. A normal disconnect means that the system issuing the disconnect continues to function and make catalog requests. In this situation, ECS activity is quiesced to ensure that catalog requests on the disconnected system are not rejected because catalogs are ECS-active on other systems. Both CAS restarts and IPLs are considered abnormal disconnects and do not result in quiescing ECS activity. A normal disconnect is initiated by the MODIFY CATALOG,ECSHR(DISCONNECT) command.
- A rebuild of the ECS structure is initiated. This can be done by an operator, another component, or by ECS when it is informed of an ECS structure failure or loss of CF connection. ECS activity is quiesced for the life of the rebuild. Once the rebuild has completed without error, ECS restores the ECS state at the time the rebuild was initiated; that is, if autoadd was on when the rebuild was initiated, autoadd is reactivated and all eligible catalogs are re-enabled automatically by ECS. If autoadd was not on when the rebuild was initiated, autoadd will not be activated at the completion of the rebuild. If a CF error occurs during an ECS rebuild, the customer must manually reactivate ECS after the CF has stabilized.

**Recommendation:** In order to minimize catalog availability problems and operator intervention, do not set the ECSHARING attribute for a catalog unless all sharing systems will be ECS-active.

**Disconnecting from the structure:** If it becomes necessary to stop ECS activity, use the `MODIFY CATALOG,ECSHR(DISCONNECT)` command to quiesce ECS activity (see "Quiecsing ECS Activity") and disconnect from the ECS structure in the coupling facility. This command initiates a normal disconnect which will cause all other systems in the sysplex to quiesce their ECS activity, but the other systems will not disconnect from the structure.

To disconnect all the systems in a sysplex, issue the disconnect command on one system, wait for all systems to quiesce, then enter (route) a disconnect command on each of the other systems. Performing the disconnect in this manner initiates only one quiesce operation and limits the communications between systems necessary for the quiesce to be processed. You can use the `D XCF,STR,STRNAME=SYSIGGCAS_ECS` command to display the active systems connected to the cache structure.

**Rebuilding the ECS Structure:** If you find that the ECS structure is damaged, you can initiate a rebuild of the structure by issuing the following command:

```
SETXCF START,REBUILD,STRNAME=SYSIGGCAS_ECS
```

See *z/OS MVS System Commands* for details on the SETXCF command.

During the rebuild, ECS activity is automatically quiecsed and reactivated as described in “Quiecsing ECS Activity” on page 45. The contents of the original ECS structure are discarded and a new structure is created. The new structure is populated with the most current sharing information for each catalog as they are referenced.

## RACF Considerations

The security administrator can define RACF profiles to control the use of coupling facility structures. If an IXLSTR.structure_name facility class is defined, then a PERMIT specifying the RACF user ID assigned to Catalog must be specified in this facility class definition. See "Connection Services" in the *z/OS MVS Programming: Sysplex Services Guide* for more detailed information about authorizing coupling facility requests.

# Chapter 3. Defining Catalogs

You create a catalog by defining its structure with the access method services DEFINE USERCATALOG command. The ICFCATALOG parameter is the default. DEFINE USERCATALOG VOLCATALOG defines a tape volume catalog that only contains tape library and tape volume entries. Access method services can also be used to define catalog aliases (with DEFINE ALIAS), and VVDSs (with DEFINE CLUSTER). Alias is not supported for tape volume catalogs.

Before you define a catalog, determine how large you want the catalog to be, and on which volume you want to place it. Also, determine the appropriate performance attributes for the catalog. All of these subjects are covered in this chapter.

See "Setting the Catalog Control Interval and Area Size" on page 50 for information on choosing an appropriate control interval size for the index.

## Using Indirect Volume Serials with Cloned zFS Data Sets

Starting with z/OS V1R12, you can make a clone, or copy, of a zFS and use indirect volume serials to access it. This lets you define a catalog entry for a zFS to specify different volume serials for different systems, which might allow the use of existing processes for cloning systems when zFS is used for the version root file system.

**Note:**

1. This support is limited to single volume zFS data sets.
2. Before attempting the physical copy, make sure the zFS data set is unmounted. If it is not unmounted, it could lead to serialization issues that cause the copy to fail. For more information on the DFSMSdss physical copy operation, refer to DFSMSdss DOC APAR OA39443 or refer to *z/OS DFSMSdss Storage Administration*.

Do the following steps to clone and catalog a zFS data set with an indirect volume serial:

1. Clone a zFS by making copies of the existing zFS data sets using COPY with the PHYSINDYNAM (PIDY) parameter for the following reasons:
   - Using PHSYINDYNAM creates an uncataloged copy of the data set. Other method create a catalog entry. You need the uncataloged version so that it can be accessed by the catalog entry with the indirect volser.
   - Using PHYSINDYNAM (PIDY) lets you use the same name for your original and copied zFS.

   The following JCL shows a recommended COPY command to clone a zFS:

```
//STEPT07 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY DS(INC(ZFS.LDS)) -
PIDY ( -
(1P0301) -
) -
OUTDY ( -
(1P0302) -
```

```
) -
REPLACEU -
ALLDATA(*) -
ALLEXCP
```

See COPY command for DFSMSdss in *z/OS DFSMSdss Storage Administration* for information about the PHYSINDYAM parameter.

2. Create a system symbolic in IEASYMxx for &VOL01 for each system that will have a copy of the zFS data set.

   See IEASYMxx (symbol definitions) in *z/OS MVS Initialization and Tuning Reference*.

3. Enter a DEFINE CLUSTER command with the RECATALOG parameter using the system symbolic (indirect volser) for ZFS.LDS:

```
DEFINE CLUSTER -
   (NAME(ZFS.LDS) -
    LINEAR -
    VOLUMES(&VOL01) -
    RECATALOG)
```

   See DEFINE CLUSTER in *z/OS DFSMS Access Method Services Commands*.

4. A subsequent IDCAMS LISTCAT command for the LDS will show the following:

```
CLUSTER ------- ZFS.LDS
...
DATA ------- ZFS.LDS.DATA
...
VOLUME
  VOLSER------------&VOL01
DEVTYPE------X'00000000'
```

   Note that the LDS now has a device type of X'00000000'.

   See the LISTCAT command in *z/OS DFSMS Access Method Services Commands*.

Figure 7 on page 49 provides a visual representation of the zFS Cloning Support Using Indirect VOLSERs.

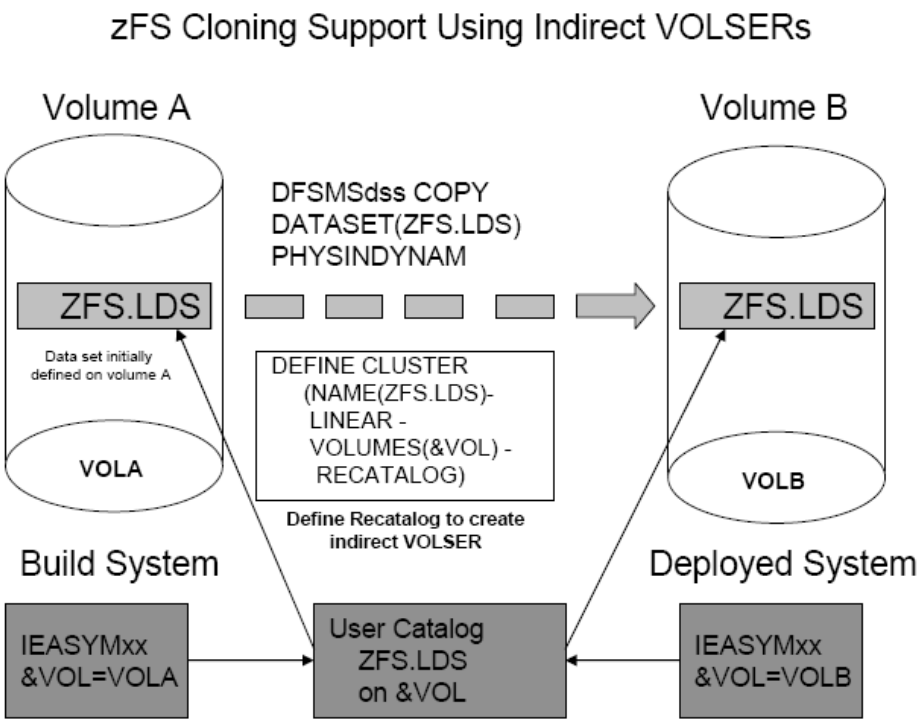



*Figure 7. zFS Cloning Support Using Indirect VOLSERs*

## Determining Catalog Size

Ideally, a catalog should be defined with enough space so that it does not grow into secondary extents. Excessive secondary extents can decrease catalog performance. You only need to consider reorganizing the catalog to eliminate secondary extents if their number becomes excessive.

A catalog can have up to 123 extents but can only occupy space on a single volume.

Before defining a catalog, carefully consider how that catalog will be used. Estimate the number of data sets and OAM object collections that will be defined in the catalog. The size of the catalog is directly affected by the number of entries in the catalog, and by whether you are using the Storage Management Subsystem.

### Assigning Space to a Catalog

You can specify the required space for a catalog in kilobytes, megabytes, tracks, cylinders, or records. This value can be specified directly on the DEFINE USERCATALOG command, or indirectly, through an SMS data class defined for catalogs that contains a space attribute. You can specify the appropriate data class on the DATACLASS parameter, or you can allow the data class ACS routine to assign the appropriate data class.

If you specify space in records, the amount of space is determined by the average record size. This is specified on the RECORDSIZE parameter, and the default is 4086. In general, you should not change the default values for average and maximum record size.

You must always specify a space parameter at the USERCATALOG (cluster) level. Space is allocated to the BCS according to the following rules:

1. If space is only defined as a subparameter of USERCATALOG, the space specified is assigned to the data component of the catalog. Additional space is allocated to the index according to the size of the space request.
2. If space is specified as subparameters of USERCATALOG and DATA, then the data component is assigned the requested space and the index is allocated space according to the space request for DATA. The USERCATALOG space request is ignored.
3. If space is specified as subparameters of USERCATALOG, DATA, and INDEX, then the data and index components are assigned the requested space. The USERCATALOG space request is ignored. If INDEX space is specified, space must also be specified for DATA.

If you are defining the BCS on a volume that *does not* contain a VVDS, a VVDS is also defined and allocated with the BCS. This implicitly defined VVDS is allocated with a default primary and secondary space amount. The default space allocation will be the values that the operator specified on the most recent `F CATALOG,VVDSSPACE` command. If that command has not been issued since IPL, the default space allocation will be TRACKS(10 10).

If you want to define a VVDS with a space allocation different from the default, you must define a VVDS on the volume (using DEFINE CLUSTER) before you define the BCS.

## Setting the Catalog Control Interval and Area Size

You can specify the control interval sizes, or the catalog can be allowed to select the CISIZE. The BCS is a catalog key-sequenced data set; therefore, the standard control interval and control area calculations are used.

To set a specific control interval size (CISIZE), use one of the following guidelines:

- From 512 bytes to 8KB, CISIZE must be specified in multiples of 512 bytes
- From 8 kilobytes to 32 kilobytes, CISIZE must be specified in multiples of 2 kilobytes.

Selecting a 4096 or the 8192 byte control interval size provides an acceptable compromise between minimizing data transfer time and reduces the occurrence of a record that spans a control interval. The resulting values for the catalog should be the same as for a key-sequenced data set with spanned records.

Additional considerations include:

- For data records less than 1 kilobyte, specify a smaller control interval size (less than 32 768) for the data component. Most of the processing is random and little benefit is gained from larger control interval sizes.
- Large or multivolume data records (catalog records larger than the CISIZE) require a second read to obtain the record and this has a direct impact on performance. Specifying a larger control interval size can provide improved

performance for catalog-key sequenced data sets with many volumes (or alternate data indices), or Generation Data Groups containing large, multivolume data sets.

For more information on CISIZE considerations, see *z/OS DFSMS Using Data Sets*

The control area size for the data component is the smaller of the primary allocation quantity, secondary allocation quantity, or one cylinder.

The control area size is never smaller than a track or greater than one cylinder. It should be large enough to contain a maximum-length record; the default maximum record length for the BCS (a spanned record data set) is 32 400 bytes.

To optimize catalog performance, choose a control area of one cylinder. This can be accomplished if you allocate space in cylinders, or if you specify a number of kilobytes, megabytes, records, or tracks that are one cylinder or larger.

**Restriction:** If you add entries to a catalog in ascending sequence, it may result in control interval splits. For more information, see *z/OS DFSMS Using Data Sets*.

## Estimating Catalog Size

Because a catalog uses variable-length, spanned records, it is not possible to precisely calculate the amount of space that a catalog requires. However, since secondary extents do not cause problems for catalogs, it is not necessary to be precise in making a size estimate. The following information serves only as an approximation for your catalog space requirements.

### Estimating Space Requirements for a Tape Volume Catalog

A volume catalog (VOLCAT) is a catalog that contains only tape library and tape volume entries. A *general* VOLCAT contains all tape library entries and any tape volume entries that do not point to a specific VOLCAT. A *specific* VOLCAT cannot contain tape library entries. It contains a specific group of tape volume entries based on the tape volume serial numbers (tape volsers).

Table 6 lists the tape library and tape volume entries and approximates the number of bytes needed to maintain information in the tape volume catalog. The values do not necessarily represent the length of any one record.

*Table 6. Estimated Space Needed by the Tape Volume Catalog*

| Record | Number of Bytes |
|---|---|
| tape library entry | 320 |
| tape volume entry | 275 |

**Restriction:** A specific VOLCAT cannot contain tape library entries. It contains a specific group of tape volume entries based on the tape volume serial numbers (tape volsers).

Use the following steps to estimate the space allocation for a tape volume catalog:

1. Estimate the number of tape library entries and tape volume entries to be cataloged in the VOLCAT. Use only tape volume entries for a specific VOLCAT because it cannot contain tape library entries.
2. Using these figures, determine the total space requirement, in bytes. This figure is the minimum amount of space that the VOLCAT requires.

3. Divide the total number of bytes by 1024 to determine the number of kilobytes, or by 1048576 to determine the number of megabytes. Round the result up to the nearest whole integer, and specify KILOBYTES or MEGABYTES as appropriate.

## Estimating Space Requirements for the BCS

The amount of space a BCS requires depends on the type and number of data sets and objects cataloged in it. The amount of space in the BCS used by each type of data set or object can vary, depending on the:

- Length of the data set or component names
- Number of volumes per data set
- Number of relationships between components
- Number of alternate indexes for a data set
- Number of paths
- Presence of security information
- Presence of Storage Management Subsystem information.

Table 7 lists the various types of data sets and objects, and approximates the number of bytes needed in the BCS to maintain information about a data set or object of that type. The values do not necessarily represent the length of any one record.

Data sets used by NFS or DFM will have 455 bytes added to their published size to accommodate new cell types.

*Table 7. Estimated Space Needed for Each Type of Data Set or Object*

| Data Set or Object Type | Number of Bytes |
|---|---|
| generation data group | 350 |
| generation data set | 200 |
| alias entry (see note) | 150 |
| non-VSAM data set or OAM object collection | 200 |
| user catalog connector (see note) | 200 |
| VSAM entry-sequenced, linear, or relative record data sets | 400 |
| VSAM key-sequenced data set | 650 |
| alternate index | 400 |
| path | 190 |

**Note:** To simplify the calculation for a user catalog connector record and to account for all of the catalog's aliases, count each catalog connector as requiring 32400 bytes.

Use the following steps to estimate the space allocation for a BCS:

1. Estimate the number of each type of data set or object that will be cataloged in the BCS. Using these figures, determine the total space requirement, in bytes, for the BCS. This figure is the minimum amount of space that the BCS requires.
2. Increase this quantity by the amount of free space required, and add additional space to allow for growth and any inaccuracies in the calculation. For example, if you define free space as 20% of each control interval and area, multiply by 1.7.

3. Divide the total number of bytes by 1024 to determine the number of kilobytes, or by 1048576 to determine the number of megabytes. Round the result up to the nearest whole integer, and specify KILOBYTES or MEGABYTES as appropriate.
4. Choose an appropriate secondary allocation. It is best if the secondary allocation is larger than the equivalent of one cylinder, so that the control area is defined as one cylinder.

The number of kilobytes or megabytes in a control area of one cylinder for different IBM DASD devices can be determined by multiplying the number of tracks per cylinder by the track capacity, and dividing by 1024 (kilobytes) or 1048576 (megabytes).

If you want to allocate space in tracks or cylinders, you must perform some additional calculations:

1. Divide the total number of bytes (including free space) by the data control interval size. Round up to the nearest whole integer. This is the number of required data control intervals.
2. Determine the number of control areas required by dividing the number of required control intervals by the number of control intervals that fit into your selected control area. Round up to the nearest integer.
3. Specify TRACKS or CYLINDERS as appropriate. If you specify TRACKS, space can be allocated in cylinders if the number of tracks is more than one cylinder.

**Estimating Space for an Extended Format BCS:** A BCS is limited to 4 GB unless you define it as an extended format BCS, which means it can use extended addressability. Using extended addressability, the size limit for a BCS is determined by the control interval size multiplied by 4 GB. For example, a control interval size of 4 KB yields a maximum data set size of 16 TB, while a control interval size of 32 KB yields a maximum data set size of 128 TB. No increase in processing time is expected for extended format data sets that grow beyond 4 GB. To use extended addressability, the BCS must be SMS managed and defined as extended format.

You can specify extended format for a BCS using SMS data class DSNTYPE=EXT parameter and subparameters R (meaning required) or P (meaning preferred) on the ISMF DATA CLASS DEFINE/ALTER panel. Use R to ensure the BCS is extended. The Extended Addressability value must be set to Y (Yes).

The only extended format option available for a BCS is extended addressable. This means that BCSs cannot be compressed or striped.

## Estimating Space Requirements for the VVDS

The VVDS contains VSAM volume records (VVRs) that hold information about VSAM data sets residing on the volume. The VVDS also contains non-VSAM volume records (NVRs) for SMS-managed non-VSAM data sets on the volume. If an SMS-managed non-VSAM data set spans volumes, only the first volume contains an NVR for that data set.

The system automatically defines a VVDS with 10 tracks primary and 10 tracks secondary space, unless you explicitly define it. If you want to explicitly define a VVDS, for example, as part of volume initialization, IBM recommends that you use the default TRACKS(10 10) for the space allocation. If you need a VVDS with more space, there are two approaches to obtaining it:

- The recommended approach is to keep the defaults TRACKS(10,10s) and, if required, obtain more space through secondary allocations as needed. For

instance, if you determine that you need a 60 track VVDS, keep the default allocation so that you do not over allocate your VVDS and obtain additional space as required through secondary allocation. This is recommended because a large VVDS allocation can effect performance in some products, ADRDSSU for example, which process the entire contents of the VVDS data set in some functions.

- You can estimate the amount of VVDS space you need to allocate by estimating the number and type of data sets that will reside on the volume. (Note that data sets used by NFS or DFM will have 500 bytes added to their published size to accommodate new cell types) Then, use the information in Table 8 to estimate the total amount of space needed for the data sets that will reside in the VVDS.

  To estimate the total amount of space needed:
  1. Add the space required by each data set.
  2. Add 8 kilobytes for use by the VVDS.
  3. Multiply the result by 1.2 to leave room for errors in the calculation.
  4. Divide the result by 1024 to determine the number of kilobytes, and round up to the nearest integer.

  Specify the result in the KILOBYTES parameter of the DEFINE CLUSTER command. Choose an appropriate secondary space allocation, usually at least half of the primary allocation.

*Table 8. Estimated Space Needed by the VVDS*

| Data Set Type | SMS-managed Volume | Non-SMS-managed Volume |
|---|---|---|
| VSAM key-sequenced data set or alternate index | 530 | 480 |
| VSAM entry-sequenced or relative record data set | 370 | 320 |
| VSAM linear data set | 340 | 290 |
| non-VSAM data set | 100 | 0 |

**Note:** These numbers assume 3 qualifier data set and catalog names of 8 characters per qualifier. Key-sequenced data sets and alternate indexes are assumed to have 64 character keys. SMS class names are assumed to be eight characters.

**More VVDS sizing considerations:** Consider the following when estimating the space needed for a VVDS:

- If you want to allocate space in tracks, you must divide the total number of bytes by 4096 (the control interval for a VVDS). Then, divide the result by the number of 4096 blocks per track for the device.
- Track allocations may be rounded up to the next cylinder value. For example, a define request for 5455 tracks will likely allocate 5460 tracks, which is evenly divisible by 15 and is the next cylinder boundary.
- The VVDS can hold a maximum of 1048575 control intervals (CIs). This limits defining of primary and secondary allocation sizes to 87375 tracks or 5825 cylinders or fewer. This limit is also in effect when extending the VVDS. Thus, if extending the VVDS by the secondary allocation amount causes it to exceed the CI limit, the extend will fail. If you know the average number of VVRs per CI in the VVDS, you can estimate the maximum number of data sets the VVDS can support on the volume (IDCAMS PRINT of the VVDS can assist in coming to an estimate of average VVR's per CI).
- You can explicitly create a VVDS that can optionally be placed in Extended Addressable Storage (EAS), if available. Allocation amounts which are intended for EAS space may be rounded up to a Multi-Cylinder Unit (MCU). The

potential rounding up of EATTR(OPT) allocations has the effect of lowering the VVDS maximum size to CI values which are consistent with the MCU boundary.

- Explicit definitions of a VVDS with EATTR(OPT) above the last usable MCU while under the VVDS CI maximum will succeed, but the allocation will be rounded down to the last usable MCU boundary. Additionally, extending of the VVDS which was both explicitly defined and EATTR(OPT) will fail, if extending the VVDS will exceed the last usable MCU boundary prior to the VVDS CI maximum.

## Choosing Options to Adjust Catalog Performance

You can enhance catalog performance by carefully selecting VSAM attributes when you define the catalog. Some catalog characteristics cannot be altered once the catalog is defined, so you should pay careful attention when selecting catalog characteristics.

Because the BCS of a catalog is a VSAM key-sequenced data set, you can enhance BCS performance by using the same considerations that you would use for similar key-sequenced data sets. For more information on performance considerations for VSAM data sets, see *z/OS DFSMS Using Data Sets*.

The following sections address VSAM attributes for performance and recommended selections as they apply to catalogs in non-RLS mode. These VSAM attributes are ignored for catalogs in RLS mode. All parameters discussed are used on the DEFINE USERCATALOG command.

### Specifying the Number of Concurrent Requests

It is possible to specify the number of concurrent read requests for a BCS with the STRNO attribute. You can have from 2 to 255 concurrent read requests. Only one write request is allowed at a given time.

Initially define the catalog with **STRNO(3)**. You can monitor the adequacy of this value by using the Resource Measurement Facility™ (RMF) to watch for enqueues on the resource **SYSZRPLW.***catname***, and I/O contention on catalog volumes. SYSZRPLW** is the major name of the resource, and *catname* (the name of the catalog) is the minor name. An indication of enqueue contention is given by the I/O service times, not necessarily the number and frequency of enqueues on this resource.

You can change the STRNO attribute with the access method services ALTER command. This will take effect following a close and subsequent reopen of the catalog. You can close the catalog using the MODIFY CATALOG,CLOSE console command. The next request to the catalog will cause the catalog to be reopened.

If a catalog request is for update, an ENQ with exclusive control is issued on the BCS itself. However, most requests to the BCS are read requests, and those result in shared enqueues on the BCS.

Because requests for the VVDS are provided for dynamically, the number of concurrent requests is variable. There can be multiple updates to the VVDS concurrently, if the updates do not change the length of the VVDS records, and if the updates occur in different control intervals. An update to a record gets an exclusive enqueue on a control interval, and if the update changes the length of the VVDS record, the update request gets an exclusive enqueue on the VVDS itself.

## Other Catalog Performance Options

All the options discussed in this section are alterable, except for RECORDSIZE. If you decide your initial selection is inappropriate, you can use the access method services ALTER command to change it.

**Buffers**
You can specify buffer size on three different parameters:

**BUFFERSPACE**
The required buffer space for your catalog is determined by catalog. You only need to specify this if you want extra buffer space for your catalog. In general, allow catalog to determine buffer space. The number of buffers specified applies to each processor.

**BUFND**
Specifies the number of buffers for transmitting data between virtual and auxiliary storage. The default, STRNO+1, is usually adequate. When a catalog contains large GDGs or other spanned records, the number should be increased accordingly to a minimum of MAXLRECL/CISIZE+STRNO.

**BUFNI**
Specifies the number of buffers for transmitting index entries between virtual and auxiliary storage. The default, STRNO+2, is adequate for 3 levels of index in the catalog. If the catalog index exceeds 3 levels of index, the minimum BUFNI equals the number of levels of index - 1 + STRNO.

**FREESPACE**
The free space allows catalog updates without an excessive number of control interval and control area splits.

The following parameter affects catalog performance but can only be specified when the catalog is initially defined.

**RECORDSIZE**
The record size information is ignored when a catalog is defined. It will always set the record size information for a catalog as though RECORDSIZE(4086 32400) was specified. The maximum value should always be the default if large records could be generated. Large catalog records are generated for GDGs that have many GDSs. Master catalogs with user catalog connector records that contain a large number of aliases also have large catalog records.

# Defining a Basic Catalog Structure

Before you define a catalog, determine which attributes to assign to the catalog.

- "Sharing Catalogs Among Systems" on page 11 discusses catalog sharing and share options.
- "Choosing Options to Adjust Catalog Performance" on page 55 discusses performance attributes.
- "Estimating Space Requirements for the BCS" on page 52 discusses how to estimate the size for a BCS and how to define an extended addressable BCS.

Carefully consider these sections before you define a new catalog; some of the options you can specify when defining a catalog cannot be altered.

**Note:** BCSs cannot be defined in cylinder-managed space.

For a discussion of attributes that can be altered for an existing catalog, see "Altering Catalog Attributes" on page 74.

Use the access method services DEFINE USERCATALOG ICFCATALOG command to define the basic catalog structure of a catalog. Use the access method services DEFINE USERCATALOG VOLCATALOG to define a catalog that only contains tape library and tape volume entries. See the z/OS DFSMS Access Method Services Commands for more detail.

If the BCS is being defined on a volume that contains a VVDS, then the existing VVDS is cataloged in the newly created BCS. If there is no VVDS on the volume, the DEFINE USERCATALOG ICFCATALOG job allocates one for you, using the default attributes for the VVDS (see "Defining a VVDS (catalog Volume Data Set)" on page 60 for information on defining a VVDS). If the job tries to create a VVDS and cannot (for example, if there is not enough space on the volume), the job terminates before the BCS is defined.

You must specify ICFCATALOG on the DEFINE USERCATALOG command. Otherwise, you will create a catalog catalog, the default.

## Example: Defining a catalog

The following example defines a catalog with the attributes recommended in this document. The catalog defined, SYS1.ICFCAT.TEST, is placed on volume SYS305, and is allocated with 15 megabytes primary and secondary space.

```
//DEFCAT   JOB  ...
//DEFCAT  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  DEFINE USERCATALOG             -
        ( NAME(SYS1.ICFCAT.TEST) -
          MEGABYTES(15 15)       -
          VOLUME(SYS305)         -
          ICFCATALOG             -
          FREESPACE(10 10)       -
          STRNO(3)               -
       DATA( CONTROLINTERVALSIZE(4096) -
             BUFND(4)  )         -
       INDEX( BUFNI(4)  )
/*
```

Catalog definition can be simplified by creating a data class for catalogs. Then, you can define new catalogs using the data class, and the appropriate attributes are assigned according to the data class definition.

## Example: Defining a tape volume catalog - General

A general VOLCAT is the default tape volume catalog. A general VOLCAT contains all tape library entries and any tape volume entries that do not point to a specific VOLCAT. Each system can have access to only one general VOLCAT. The general VOLCAT must be defined prior to bringing the tape libraries online. All General Tape Volume Catalogs must be defined in the Master Catalog of the processing system. All systems connected to the IBM 3495 Tape Library Dataserver must use the same high level qualifier for their tape volume catalogs. If different versions exist, different systems can be accessing different sets of tape volume catalogs. See "Defining Names for a Tape Volume Catalog" on page 59 for more detail. The procedure for specifying the tape volume catalog high level qualifier is shown in "Bypassing SYSCATxx with LOADxx" on page 39.

This example defines an SMS-managed tape volume catalog named SYS1.VOLCAT.VGENERAL.

```
//DEFVCAT     JOB     ...
//STEP1       EXEC   PGM=IDCAMS
//SYSPRINT    DD     SYSOUT=A
//SYSIN       DD     *
     DEFINE USERCATALOG -
           (NAME(SYS1.VOLCAT.VGENERAL) -
           VOLCATALOG -
           VOLUME(338001) -
           CYLINDERS(1 1))
/*
```

### Example: Defining a tape volume catalog - Specific

A specific VOLCAT is a tape volume catalog that contains a specific group of tape volume entries based on the tape volume serial numbers (tape volsers). A specific VOLCAT cannot contain tape library entries. See “Defining Names for a Tape Volume Catalog” on page 59 for more detail concerning restrictions when different systems are connected to the Tape Library Dataserver.

This example defines an SMS-managed tape volume catalog named SYS1.VOLCAT.VT. This tape volume catalog would contain all tape volume entries that have a label beginning with the character "T". For example, one entry could be TAPE01.

```
//DEFVCAT     JOB     ...
//STEP1       EXEC   PGM=IDCAMS
//SYSPRINT    DD     SYSOUT=A
//SYSIN       DD     *
     DEFINE USERCATALOG -
           (NAME(SYS1.VOLCAT.VT) -
           VOLCATALOG -
           VOLUME(338001) -
           CYLINDERS(1 1))
/*
```

## Defining Aliases for a User Catalog

To use a catalog, the system must be able to determine which data sets should be defined in that catalog. The simplest way to accomplish this is to define aliases for the catalog. “Choosing Aliases and an Alias Search Level” on page 22 discusses general considerations for choosing aliases. Before defining an alias, carefully consider the effect the new alias will have on old data sets. A poorly chosen alias could make some data sets inaccessible.

You can define aliases for the catalog in the same job in which you define the catalog by including DEFINE ALIAS commands after the DEFINE USERCATALOG command. You can use conditional operators to ensure the aliases are only defined if the catalog is successfully defined. After the catalog is defined, you can add new aliases or delete old aliases.

Catalog aliases are defined in the master catalog, which contains an entry for the user catalog.

- By default, the number of aliases a catalog can have is limited by the maximum record size for the master catalog. If the master catalog is defined with the default record sizes, there is a practical maximum of 3000 aliases per catalog, assuming the aliases are only for high-level qualifiers. If you use multilevel aliases, fewer aliases per catalog can be defined.
- On a system at a z/OS V1R13 level or higher, you can increase the number of aliases possible from 3000 to a theoretical limit of over 500,000 aliases depending

on the alias name length. You can exploit this increased number of aliases by specifying the EXTENDEDALIAS enable feature on the MODIFY CATALOG command as follows:

```
F CATALOG,ENABLE(EXTENDEDALIAS)
```

Once a user catalog connector extension record is created, it will exist for the rest of the life of the user catalog connector. For example, when all the association entries in the user catalog connector extension record are deleted, the user catalog connector extension record will still exist with an empty association cell. Also, there will be no associated order for the association entries of the user catalog connector to be returned.

By default, EXTENDEDALIAS is disabled. You should only enable EXTENDEDALIAS to increase the number of possible catalog aliases when all systems in the sysplex are z/OS V1R13 or greater.

Note that you can disable this function using the MODIFY CATALOG command as follows:

```
F CATALOG,DISABLE(EXTENDEDALIAS)
```

You cannot define an alias if a data set cataloged in the master catalog has the same high-level qualifier as the alias. The DEFINE ALIAS command fails with a "duplicate data set name" error. For example, if a catalog is named PAYROLL.TESTSYS.ICFCAT, you cannot define the alias PAYROLL for any catalog.

## Example: Defining aliases for a user catalog

The following job defines two aliases for SYS1.ICFCAT.TEST, USER01 and PROJECTA:

```
//DEFALIAS JOB ...
//ALIAS    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
       DEFINE ALIAS -
              (NAME(USER01)     -
              RELATE(SYS1.ICFCAT.TEST))

       DEFINE ALIAS -
              (NAME(PROJECTA)     -
              RELATE(SYS1.ICFCAT.TEST))
/*
```

The NAME parameter identifies the alias, and the RELATE parameter identifies the catalog for which the alias is being defined.

# Defining Names for a Tape Volume Catalog

Use the access method services DEFINE USERCATALOG VOLCATALOG to define a catalog that only contains tape library and tape volume entries. See the z/OS DFSMS Access Method Services Commands for more detail. Table 9 describes the naming conventions for a tape volume catalog.

*Table 9. Naming Conventions for a Tape Volume Catalog*

| Name | Type |
|---|---|
| xxxxxxxx.VOLCAT.VGENERAL | general tape volume catalog |
| xxxxxxxx.VOLCAT.Vx | specific tape volume catalog |

The high level qualifier of the tape volume catalog, xxxxxxxx, is 1 to 8 characters. It is specified by the LOAD*xx* member in SYS1.PARMLIB. See "Bypassing

SYSCATxx with LOADxx" on page 39 for a description of the format. When the system is initialized with SYSP=*xx*, the tape volume catalog high level qualifier specified in the LOAD*xx* member is activated. After initialization, you should use the MODIFY CATALOG,REPORT command to verify that the proper tape volume catalog high level qualifier is selected. For an example of the MODIFY CATALOG,REPORT output showing the high level qualifier, see "Monitoring the Catalog Address Space" on page 123.

All systems connected to the Tape Library Dataserver must use the *same* high level qualifier for their tape volume catalogs. All systems must be initialized with a LOAD*xx* member containing the same tape volume catalog high level qualifier. If different versions exist, different systems can be accessing different sets of tape volume catalogs.

Another problem can occur when one system does not specify the high level qualifier. In this case, the tape volume catalog high level qualifier defaults to SYS1. A second system is initialized with a LOAD*xx* member specifying something other than SYS1 as the tape volume catalog high level qualifier. Table 10 describes the types of errors that can occur when two different systems, using the Tape Library Dataserver, use different high level qualifiers for tape volume catalogs.

*Table 10. Errors When Using Different Tape Volume Qualifiers*

| Error | Description |
|---|---|
| not found | the target tape volume catalog does not exist |
| synchronization errors | systems are making updates to different versions of tape volume catalogs and are out of synchronization with the real inventory |

To guarantee accuracy, all systems connected to the Tape Library Dataserver should use the *same* high level qualifier for their tape volume catalogs.

## Defining a VVDS (catalog Volume Data Set)

A VVDS can be defined either:

- Explicitly, using DEFINE CLUSTER; or
- Implicitly, when the first catalog or SMS-managed data set is defined on the volume.

**Note:**

1. VVDSs cannot be defined in cylinder-managed space.
2. Do not use an SMS storage class when defining a VVDS because unexpected results can occur when a VVDS is defined as SMS managed.

A VVDS is defined with the name SYS1.VVDS.V*volser*, where *volser* is the volume serial number of the volume containing the VVDS. SYS1.VVDS.V*volser* does not have to be cataloged in the master catalog.

An explicitly defined VVDS is not related to any BCS until a data set or catalog object is defined on the volume. As data sets are allocated on the VVDS volume, each BCS with catalog or SMS-managed data sets residing on that volume is related to the VVDS.

An explicit definition of a VVDS does not update any BCS and, therefore, can be performed before the first BCS in the installation is defined.

Explicitly defining a VVDS is usually appropriate when you are initializing a new volume. If you are not running SMS, and a volume already contains some non-VSAM data sets, it is appropriate to allow the VVDS to be defined implicitly. The default space allocation will be the values that the operator specified on the most recent `F CATALOG,VVDSSPACE` command. If that command has not been issued since IPL, the default space allocation will be TRACKS(10 10).

If you are explicitly defining a VVDS, see "Estimating Space Requirements for the VVDS" on page 53 for an explanation of how to estimate the size of the VVDS.

### Example: Defining a VVDS

The following job defines and allocates a VVDS on volume SER003 with 10 tracks of space:

```
//DEFVVDS JOB ...
//DEFINE  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    DEFINE CLUSTER -
           (NAME(SYS1.VVDS.VSER003) -
           TRACKS(10 10) -
           VOLUMES(SER003) -
           NONINDEXED)
/*
```

## Using One Catalog As a Model for Another Catalog

When you define a BCS or VVDS, you can use an existing BCS or VVDS as a model for the new one. The attributes of the existing data set are copied to the newly defined data set, unless you explicitly specify a different value for an attribute. You can override any of a model's attributes.

If you do not want to change or add any attributes, you need only supply the entry name of the object being defined and the MODEL parameter. When you define a BCS, you must also specify the volume and space information for the BCS.

When you use the MODEL parameter, ensure that the job is not terminated due to allocation problems when you explicitly do any of the following:

- Specify a different type of device with the VOLUMES parameter.
- Change the size of records, buffer space, or control intervals with the RECORDSIZE, BUFFERSPACE, or CONTROLINTERVALSIZE parameters.
- Change the unit of allocation with the CYLINDERS, TRACKS, KILOBYTES, MEGABYTES, or RECORDS parameters.

When you explicitly specify any of the above parameters for the BCS or VVDS to be defined, you might need to make corresponding changes to other related parameters.

If MODEL is specified as a parameter of USERCATALOG, the following steps occur:

1. The attributes of the model are copied for the BCS being defined.
2. Any attributes explicitly specified as parameters in the DEFINE command override those of the model.

If MODEL is specified as a parameter of CLUSTER (at the cluster level) but is not specified as a subparameter of the DATA or INDEX parameter, the following steps occur:

1. The attributes of the model are copied for the VVDS being defined.
2. Any attributes explicitly specified as parameters of CLUSTER override those of the model for the VVDS.
3. The attributes of the model's data and index components are copied for the VVDS's data and index components.
4. Attributes explicitly specified as parameters of CLUSTER are reproduced to the data and index components, overriding those of the model.
5. Attributes explicitly specified with subparameters of the DATA or INDEX parameters override the previous two steps.

If MODEL is specified both as a subparameter of DATA or INDEX and as a parameter of CLUSTER, the following steps occur:

1. The attributes of the CLUSTER model are copied for the cluster entry of the defined VVDS.
2. Any attributes explicitly specified as parameters of CLUSTER override those of the model for the VVDS.
3. Attributes explicitly specified as parameters of CLUSTER are reproduced to the VVDS's data and index components.
4. Attributes of the model specified with the MODEL subparameter of the DATA or INDEX parameters are copied, overriding the previous step.
5. Attributes explicitly specified with the subparameters of the DATA or INDEX parameters are copied, overriding the previous two steps.

## Example: Using a model to define a BCS

In this example, the catalog SYS1.ICFCAT.NEWCAT is defined using SYS1.ICFCAT.MODEL as a model. The MODEL parameter specifies the name of the model, and the catalog that contains the entries for the model. Because catalogs contain their own entries, you must specify the name of the catalog twice in the MODEL parameter.

```
//DEFCAT2  JOB  ...
//MODEL   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    DEFINE USERCATALOG -
           (NAME(SYS1.ICFCAT.NEWCAT) -
            ICFCATALOG -
            MODEL(SYS1.ICFCAT.MODEL SYS1.ICFCAT.MODEL) -
            VOLUME(VSER03) -
            CYLINDERS(15 5))
/*
```

# Chapter 4. Maintaining Catalogs

A major task of managing catalogs is maintaining an existing configuration of catalogs. You might need to change the size or location of a catalog to improve performance or maintain data security, delete obsolete catalogs, or remove volumes from a system. This chapter discusses some of the major tasks of catalog maintenance.

## Retrieving Information From a Catalog or VTOC

Information about a catalog or data set can be obtained using:

- Access method services (IDCAMS)
- ISMF
- IEHLIST
- The DFSMS attribute call service IGWASMS
- The TSO service routine IKJEHCIR
- The macros CAMLST and SHOWCAT
- The Catalog Search Interface (CSI) option.

### Listing the Contents of a Catalog

You can list catalog records using the access method services LISTCAT command, or the ISMF line operator CATLIST. CATLIST produces the same output as LISTCAT, but places the output in a data set that can be browsed.

You can use the LISTCAT output to monitor catalog data sets. The statistics and attributes listed can be used to help determine if you should reorganize, recreate, or otherwise alter a catalog data set to improve performance or avoid problems.

For example, you can use the values for High Used RBA and High Allocated RBA to help avoid out-of-space conditions for a data set or catalog. If the High Used RBA is less than the High Allocated RBA, then at least one control area split can occur without adding another secondary extent to the data set. Of course, you do not need to be concerned about secondary extents unless the volume is full or the data set already has a large number of them. If a data set has a large number of secondary extents, you might want to recreate the data set in a single extent.

Most information concerning non-catalog data sets is maintained in the VTOC.

The statistical information contained in the self-describing entries for a BCS is not correct. Catalog management does not maintain information about the statistical attributes of a BCS.

The cluster entry name for a BCS is 44 bytes of zeros. The name of the data component is the name you gave the catalog. The name of the index component is generated according to the regular catalog rules. The rules for catalog generated names are described in *z/OS DFSMS Using Data Sets*.

#### Using LISTCAT in Examples

A catalog's self-describing entries are contained in the catalog itself. A catalog's connector record, which associates the catalog to its aliases is contained in the master catalog.

When you are listing the self-describing entries for a catalog, specify the name of the catalog in the CATALOG parameter. For example, the following step can be used to list the self-describing entry for SYS1.ICFCAT.VSYS303:

```
//LSTSDENT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    LISTCAT ALL ENTRIES(SYS1.ICFCAT.VSYS303) -
       CATALOG(SYS1.ICFCAT.VSYS303)
/*
```

You can use LISTCAT to determine which VVDSs are connected to a BCS. You can use this information to determine which VVDSs to compare to a BCS when you use the DIAGNOSE command. For example, the following step lists the VVDSs connected to SYS1.ICFCAT.VSYS303:

```
//LSTVVDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
   LISTCAT LEVEL(SYS1.VVDS) CATALOG(SYS1.ICFCAT.VSYS303)
/*
```

You can use LISTCAT to list the aliases associated with a catalog. Specify ALL with the catalog name in the ENTRIES parameter. The aliases are listed in the Associations group for the user catalog. If you specify a catalog in the CATALOG parameter, specify the master catalog. The following example lists the aliases associated with SYS1.ICFCAT.VSYS303:

```
//LSTALIAS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    LISTCAT ALL ENTRIES(SYS1.ICFCAT.VSYS303)
/*
```

You can use LISTCAT to display all fields associated with tape library and tape volume entries. Specify LIBRARYENTRIES to list tape library entries. Specify VOLUMEENTRIES to list tape volume entries. This example lists the tape library entry named ATLLIB1. Specify ALL to list all information associated with the tape library entry ATLLIB1.

```
//LISTCLIB JOB  ...
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
     LISTCAT ENTRIES(ATLLIB1) -
        LIBRARYENTRIES -
        ALL
```

This example lists all the tape volume entries whose names begin with the letters 'VA' in the tape library named ATLLIB1.

```
//LISTCLIB    JOB     ...
//STEP1       EXEC   PGM=IDCAMS
//SYSPRINT    DD     SYSOUT=A
//SYSIN       DD     *
      LISTCAT -
           VOLUMEENTRIES(VA*) -
           LIBRARY(ATLLIB1) -
           ALL
```

For more details on using LISTCAT, see *z/OS DFSMS Access Method Services Commands*. For details on using CATLIST, see *z/OS DFSMS Using the Interactive Storage Management Facility*.

## Printing a Catalog or VVDS

You can print the contents of a BCS or VVDS with the PRINT command, but the only circumstance where it might be useful is when you need to determine which catalogs are connected to a VVDS. This might be necessary to determine which BCSs to specify in a DIAGNOSE command, or when you are recovering a volume.

The names of the first 36 BCSs connected to a VVDS are in the first record of the VVDS. If you print this record using the DUMP format, you can read the names of the BCSs in the character format portion of the dump.

The following step can be used to print the first record of a VVDS. Because VVDSs are not normally found by catalog searches, use the INFILE parameter to specify a DD statement defining the VVDS.

```
//PRNTVVDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VVDS     DD DSN=SYS1.VVDS.VSPOOL1,DISP=SHR,
//            UNIT=SYSDA,VOL=SER=SPOOL1,AMP=AMORG
//SYSIN    DD *
       PRINT INFILE(VVDS) COUNT(1)
/*
```

A catalog or VVDS can also be printed using DFSMSdss.

## Listing a Volume Table of Contents (VTOC)

The most convenient and flexible way to generate a data set list from a VTOC is to use the data set application of ISMF. With ISMF, you can generate a list of data sets based on many different filtering criteria, and execute commands against the data sets listed. You can also save the lists.

The VTOCLIST line operator can also be used to generate a VTOC listing for a data set in the IEHLIST format. For more information about using ISMF, see *z/OS DFSMS Using the Interactive Storage Management Facility*.

If you want to use batch processing to get a VTOC listing, you can use the IEHLIST utility. The listing can be formatted or dumped in hexadecimal. For more information, see *z/OS DFSMSdfp Utilities*.

## Obtaining Information from an Application Program

An application program can access a catalog and retrieve information or perform other tasks. This can be done by using macros, or by using calls to system services or programs like access method services (IDCAMS).

The following is a list of macros or other callable services that can be used with catalogs, and a brief description of their use:

*Table 11. Macros and System Services for Accessing Catalogs*

| Service | Description |
|---|---|
| CATALOG and CAMLST CAT | Macros used to catalog a non-catalog data set. |
| CATALOG and CAMLST RECAT | Macros used to recatalog a non-catalog data set. |
| CATALOG and CAMLST UNCAT | Macros used to uncatalog a non-catalog data set, if it is not managed by the Storage Management Subsystem. |

*Table 11. Macros and System Services for Accessing Catalogs (continued)*

| Service | Description |
|---|---|
| CATALOG SEARCH INTERFACE | Provides an alternative to LISTCAT that allows customizing of the output. You can display as little or as much information as you want. Also includes some information that LISTCAT does not provide. Three Assembler H sample programs and one REXX program are provided as part of CSI. See Chapter 11, "Catalog Search Interface User's Guide," on page 223. For more information, see *HLASM Programmer's Guide* |
| IGWASMS | A DFSMS attribute call service used to identify the SMS classes for a data set, and determine whether the data set is a PDSE. For more information, see *z/OS DFSMSdfp Advanced Services*. |
| IKJEHCIR | A TSO programming service that can be used to retrieve a list of data sets with a matching high-level qualifier, and their associated data set types. It can also be used to identify the volume serial numbers and device types associated with a name, and find the next level qualifiers for a name. For more information, see *z/OS TSO/E Programming Services*. |
| LOCATE and CAMLST NAME | Macros used to retrieve information about a data set. For more information, see *z/OS DFSMSdfp Advanced Services*. |
| SHOWCAT | A macro used to retrieve information about a catalog data set or generation data group. For more information, see *z/OS DFSMS Macro Instructions for Data Sets*. |

# Changing the Size or Contents of a Catalog

The main issues concerning the size or content of a catalog are availability and recoverability, not performance. Catalog performance is normally unaffected by the size of the catalog, or by the aliases defined for it.

When deciding whether a catalog is too big or small, or whether it has the wrong combination of entries (based on catalog aliases), consider how loosing that catalog would affect your installation with respect to how long it would take to recover the catalog.

Splitting, merging, or reorganizing catalogs can be time-consuming and disruptive to your users. Therefore, only perform these actions after carefully weighing the possible benefits against the lost availability of the catalog.

## Splitting Catalogs or Moving Catalog Entries

You can split a catalog into two catalogs or to move a group of catalog entries, if you determine that a catalog is either unacceptably large or that it contains too many entries for critical data sets.

If the catalog is unacceptably large (a catalog failure would leave too many entries inaccessible), then you can split the catalog into two catalogs. If the catalog is of an acceptable size but contains entries for too many critical data sets, then you can simply move entries from one catalog to another.

**Attention:** Performing REPRO on a catalog while data sets are open in the source catalog might result in a loss of information if any of those data sets extend, or other catalog updates are made. The changes might not be copied to the target catalog, resulting in a mismatch between the information contained in the VVDS and the new target BCS. This might cause the data sets to be inaccessible or receive errors when they are used.

To split a catalog or move a group of entries, use the access method services REPRO MERGECAT command. The following steps should be followed to split a catalog or to move a group of entries:

1. Use ALTER LOCK to lock the catalog. If you are moving entries to an existing catalog, lock it as well.
2. If you are splitting a catalog, define a new catalog with DEFINE USERCATALOG LOCK.
3. Use LISTCAT to obtain a listing of the catalog aliases that you are moving to the new catalog. Use the OUTFILE parameter to define a data set to contain the output listing.
4. Use EXAMINE and DIAGNOSE to ensure that the catalogs are error-free. Fix any errors indicated.
5. Use REPRO MERGECAT to split the catalog or move the group of entries. When splitting a catalog, the OUTDATASET parameter specifies the catalog created in 2. When moving a group of entries, the OUTDATASET parameter specifies the catalog that is to receive the entries. This step can take a long time to complete.

   Use the ENTRIES or LEVEL parameter to specify which catalog entries are to be removed from the source catalog and placed in the catalog specified in OUTDATASET.

   **Note:**

   a. The use of the LEVEL or ENTRIES parameter will not move extended aliases to the new catalog. For more information on extended aliases, see "Extended Alias Support" on page 18.
   b. In some cases, use of the LEVEL or ENTRIES parameter may cause data sets to no longer be found. For example, consider a data set named AAA.LOADLIB that has an alias of BBB.LOAD:
      - If a REPRO MERGECAT is run with LEVEL parameter specified as LEVEL(BBB), the alias will not move to the output catalog. And after the alias BBB is deleted and redefined to point to the output catalog, the data set cannot be located using the alias.
      - If a REPRO MERGECAT is run with the LEVEL parameter specified as LEVEL(AAA) the data set and the alias will move to the output catalog. But unless the alias BBB is deleted and redefined to point to the output catalog, the data set cannot be located through the alias.

      For this example, running two REPRO MERGECAT job/steps with a specification of LEVEL(AAA) and LEVEL(BBB) and then deleting and redefining of the aliases AAA and BBB to point to the output catalog will allow the data set to be accessed through both the alias and the data set name.

      The ENTRIES parameter can cause similar problems, which can be resolved using a similar method.

   If this step fails for any reason, see "Recovering from a REPRO MERGECAT Failure" on page 68.
6. Use the listing created in 3 to create a sequence of DELETE ALIAS and DEFINE ALIAS commands for each alias. These commands delete the alias to the original catalog, and redefine them as aliases for the catalog that now contains entries belonging to that alias name.

   The DELETE ALIAS/DEFINE ALIAS sequence must be run on each system that shares the changed catalogs.

Any data sets with extended aliases that may have been affected by the REPRO MERGECAT must have their individual extended aliases deleted and redefined. For more information on extended aliases, see "Extended Alias Support" on page 18. Data sets whose aliases are cataloged at identical LEVELs can be moved without a need for further delete/define activity for the individual data sets.

7. Unlock both catalogs using ALTER UNLOCK.

## Merging Catalogs

You might find it beneficial to merge catalogs if you have many small or seldom-used catalogs. An excessive number of catalogs can complicate recovery procedures and waste resources such as CAS storage, tape mounts for backups, and system time performing backups.

Merging catalogs is accomplished in much the same way as splitting catalogs. The only difference between splitting catalogs and merging them is that in merging, you want all the entries in a catalog to be moved to a different catalog, so that you can delete the obsolete catalog.

The following steps should be followed to merge two catalogs:

1. Use ALTER LOCK to lock both catalogs.
2. Use LISTCAT to list the aliases for the catalog you intend to delete after the merger. Use the OUTFILE parameter to define a data set to contain the output listing.
3. Use EXAMINE and DIAGNOSE to ensure that the catalogs are error-free. Fix any errors indicated.
4. Use REPRO MERGECAT *without* specifying the ENTRIES or LEVEL parameter. The OUTDATASET parameter specifies the catalog that you are keeping after the two catalogs are merged. This step can take a long time to complete.
5. Use the listing created in step 2 to create a sequence of DELETE ALIAS and DEFINE ALIAS commands to delete the aliases of the obsolete catalog, and to redefine the aliases as aliases of the catalog you are keeping.

   If this step fails for any reason, see "Recovering from a REPRO MERGECAT Failure."
6. Use DELETE USERCATALOG to delete the obsolete catalog. Specify RECOVERY on the DELETE command.

   If your catalog is shared, run the EXPORT DISCONNECT command on each shared system to remove unwanted user catalog connector entries.
7. Use ALTER UNLOCK to unlock the remaining catalog.

You can also merge entries from one tape volume catalog to another using REPRO MERGECAT. REPRO retrieves tape library or tape volume entries and redefines them in a target tape volume catalog. In this case, VOLUMEENTRIES needs to be used to correctly filter the appropriate entries. The LEVEL parameter is not allowed when merging tape volume catalogs.

## Recovering from a REPRO MERGECAT Failure

Depending on the number of entries being processed, REPRO MERGECAT can require a lot of time to complete execution. If the REPRO MERGECAT job fails, for example, if the system goes down during execution, first determine what caused the job to fail.

As REPRO MERGECAT processes a catalog, it transfers entries from one catalog to another, deleting the entries from the source catalog. Thus, if REPRO MERGECAT has not completed, the target catalog contains the only valid entries for some of your data sets. For that reason, *do not* delete the target catalog simply because REPRO MERGECAT failed.

If the failure was caused by errors external to catalog management and access method services, simply rerun the REPRO MERGECAT job.

If the REPRO MERGECAT job cannot complete, use REPRO MERGECAT to *return* the entries from the target catalog that were moved there from the source catalog. This should return you to your starting point. Use the access method services DIAGNOSE command to help determine the catalog errors that must be corrected. Also, use EXAMINE to check the internal structure of the catalog.

Entries moved by REPRO MERGECAT are now cataloged in the new catalog, and they cannot be referenced from the old catalog. You should not use REPRO MERGECAT to make a backup copy of a catalog, or to create new catalog for another system. Using REPRO MERGECAT will make it impossible to refer to the data sets from the original catalog.

To help avoid REPRO MERGECAT failures, you should always use EXAMINE and DIAGNOSE before using REPRO MERGECAT, and fix any indicated errors. If REPRO MERGECAT cannot process a record because of an error in the record, it normally bypasses the record and issues a message. After using REPRO MERGECAT, carefully inspect all messages to determine if any entries were skipped. Then use DIAGNOSE on any skipped entries to determine the errors and the appropriate recovery procedures.

## Changing the Size of a BCS

You can change the size of a catalog if the catalog is much larger than necessary, or if it has grown into excessive secondary extents. Before changing the size of a catalog, consider merging small catalogs or splitting large catalogs.

When you change the size of a catalog, the catalog is also reorganized.

If you are decreasing the size of the catalog, you must be certain to define the new catalog with enough space to contain all the entries in the existing catalog. The catalog must also account for the free space defined for the catalog.

Before changing any catalog always:

1. Ensure you have a good backup of a valid catalog. Using IDCAMS DIAGNOSE and EXAMINE on the catalog and correcting any errors before taking the backup will ensure you can fall back to a valid backup if there are problems. Having more than one copy is always a good idea in case one copy becomes damaged or lost. This backup can be a full volume dump of the volume the catalog is on, a DFDSS backup or IDCAMS EXPORT. Having all three gives you more options if you need to back out of the catalog changes.
2. Use IDCAMS DIAGNOSE and EXAMINE to ensure that the catalog does not have errors.
3. Consider using IDCAMS LISTCAT ALL on the catalog to be changed, in case there are problems. Because this may produce a large volume of output you may wish to redirect the output to a data set, either by pointing the SYSPRINT DD statement to a data set or by using the OFILE keyword on the LISTCAT command.

4. If you are changing a catalog that is shared across systems, a LISTCAT of the master catalog with the ALL keyword is needed to enable a redefine of any aliases that point to the user catalog that is being changed, unless you already have a job set up to define the aliases. If you already have a job, verify that it is current by comparing the LISTCAT output to the job.
5. Quiesce all activity against the catalog that is to be changed.

Run the following commands as the **first** part of the change process and, if there are errors, make the required corrections before proceeding with changing the catalog.

```
//DIAGNOSE EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//DIAGDD   DD   DISP=SHR,DSN=your.user.catalog
//SYSIN    DD   *
  DIAGNOSE ICFCATALOG INFILE(DIAGDD)
//EXAMINE1 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
    INDEXTEST NODATATEST ERRORLIMIT(1000)
//EXAMINE2 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
    NOINDEXTEST DATATEST ERRORLIMIT(1000)
//LISTCAT  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  LISTCAT CAT(your.user.catalog) ALL
```

The following steps show how to change the size of a catalog. The catalog used in the example is called ICFCAT.USER.VSYS303.

1. Lock the catalog.

   ```
   //LOCKCAT EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=A
   //SYSIN    DD *
      ALTER ICFCAT.USER.VSYS303 LOCK
   /*
   ```

2. Issue the following command on each system that shares the catalog:

   ```
   F CATALOG,NOVLF(ICFCAT.USER.VSYS303)
   ```

3. If the catalog is shared with other systems and those other systems do not all share the same master catalog, then EXPORT DISCONNECT the catalog on all other systems other than the system you are running the jobs to resize the catalog on.

   ```
   //EXPDISC  EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=*
   //SYSIN    DD *
              EXPORT ICFCAT.USER.VSYS303 DISCONNECT
   ```

4. Export the BCS with the EXPORT command. The aliases of the catalog are saved with the exported copy, and can be used in later steps to redefine the aliases. You must create a sequential data set for the catalog export, if one does not already exist.

   ```
   //EXPORT  EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=A
   //EXPDD    DD DSN=CATBACK.ICFCAT.USER.VSYS303,DISP=OLD
   //SYSIN    DD *
   ```

```
   EXPORT ICFCAT.USER.VSYS303 -
       OUTFILE(EXPDD) -
       TEMPORARY
/*
```

5. Delete the BCS with the RECOVERY option.

```
//DELCAT  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  DELETE ICFCAT.USER.VSYS303 -
       RECOVERY -
       USERCATALOG
/*
```

6. Redefine the BCS with the desired space and performance attributes.

```
//DFNEWCAT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    DEFINE USERCATALOG -
          (NAME(ICFCAT.USER.VSYS303) -
           VOLUME(SYS303) -
           MEGABYTES(15 5) -
           ICFCATALOG -
           LOCK -
           FREESPACE(20 20) -
           STRNO(3)  -
           )  -
           DATA( CONTROLINTERVALSIZE(4096) -
                 BUFND(4)  )  -
           INDEX( CONTROLINTERVALSIZE(4096) -

                    BUFNI(4)  )
/*
```

7. Import the BCS using the IMPORT command. Specify INTOEMPTY on the IMPORT command.

   Also, specify ALIAS, so that the aliases exported with the catalog are redefined.

```
//IMPORT  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    IMPORT INDATASET(CATBACK.ICFCAT.USER.VSYS303) -
           OUTDATASET(ICFCAT.USER.VSYS303)        -
           ALIAS   -
           LOCK  -
           INTOEMPTY
/*
```

8. If the catalog is shared with other systems that do not share the same master catalog then you will need to IMPORT CONNECT on all other systems that share the catalog.

```
//IMPRTCON EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
        IMPORT CONNECT -
            OBJECTS((ICFCAT.USER.VSYS303) -
            DEVICETYPE(3390) -
            VOLUMES(SYS303)))
```

9. If the catalog is shared with other systems that do not share the same master catalog then you will need to define the aliases for that catalog on the other systems that do not share the same master catalog.
10. Run the following commands and, if there are any errors, make the required corrections before unlocking the catalog.

```
//DIAGNOSE EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//DIAGDD   DD   DISP=SHR,DSN=your.user.catalog
//SYSIN    DD   *
  DIAGNOSE ICFCATALOG INFILE(DIAGDD)
//EXAMINE1 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
    INDEXTEST NODATATEST ERRORLIMIT(1000)
//EXAMINE2 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
  NOINDEXTEST DATATEST ERRORLIMIT(1000)
//LISTCAT  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  LISTCAT CAT(your.user.catalog) ALL
.
```

11. If there were no errors in the previous step , you can now unlock the catalog:

```
//LOCKCAT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    ALTER ICFCAT.USER.VSYS303 UNLOCK
/*
```

## Recovering from a REPRO NOMERGECAT Failure

If the job fails, determine what caused the REPRO NOMERGECAT to fail. Because the source catalog is copied into an empty target catalog, the REPRO operation cannot be restarted if an error occurs.

Before you restart the copy operation, you must delete and redefine the target catalog. Make sure volumes, that contain objects to be copied, are restored.

After a successful REPRO of one catalog into an empty target catalog, the VVRs are changed to point to the target catalog. Subsequent processing occurs in the target catalog; entries can no longer be deleted from the source catalog.

## Changing the Size of a VVDS

A VVDS automatically reclaims space used by deleted records. It grows into secondary extents only when there is no space for a new record. Even if the VVDS grows into a few secondary extents, the performance and reliability of the VVDS are not affected.

If you still need to change the size of a VVDS, you must backup all data sets that have an entry in the VVDS, remove these data sets from the volume, delete the VVDS, redefine the VVDS, and finally restore the data sets from their backups to the volume.

**Restriction:** A VVDS is not subject to control interval splits.

The following steps are suggested for this process:

1. Obtain exclusive use of the volume by: stopping work, varying the volume offline to sharing systems, or quiescing sharing systems.

2. Remove all data sets from the volume that have an entry in the VVDS. All catalog, SMS-managed data sets, and their catalogs should be removed using DFSMSdss (logical dump) or access method services (EXPORT, or REPRO followed by DELETE).

   With volumes that are not SMS-managed, you can use DFSMSdss to dump all catalog data sets from the volume with one command. With SMS-managed volumes, use DFSMSdss to dump the entire volume.
3. Delete the VVDS specifying RECOVERY.
4. Define a new VVDS specifying the desired size. The default recommended size is trk(10,10).
5. Restore the data originally on the volume, using DFSMSdss or access method services (use the same utility used to dump the data sets).

You can use DIAGNOSE to determine any differences between the rebuilt VVDS and each BCS that had data sets on the volume. Because all existing data should have been recovered to the volume, there should be no differences.

## Renaming a Catalog

You cannot directly rename a catalog with the ALTER command. To rename a catalog, you must define a new catalog with the desired name and copy the old catalog into it. You can do this with the REPRO NOMERGECAT command.

REPRO NOMERGECAT is used to copy a catalog into an empty target catalog. Refer to *z/OS DFSMS Access Method Services Commands* to determine if the REPRO NOMERGECAT option is appropriate for you.

All aliases to the old catalog must be redefined as aliases of the new catalog. Aliases are not automatically oriented to the new catalog.

The following steps show how to rename ICFCAT.USER.VSYS303 to ICFCAT.REPRO.VSYS303:

1. Before you rename a catalog, make sure it does not have structural and logical errors; to do this, run IDCAMS DIAGNOSE and EXAMINE jobs.
2. When you use NOMERGECAT, recovery is required in the event of a failure. Make sure you have a current back up of the catalog.
3. Lock the old catalog using ALTER LOCK.
4. List the aliases of the old catalog using LISTCAT. Specify the name of the old catalog in the ENTRIES parameter, and specify the ALL parameter. This lists all the aliases of the catalog in the ASSOCIATIONS group of the LISTCAT listing.

   Specify a data set using the OUTFILE parameter to contain the LISTCAT listing. This data set can be used in step 8 on page 74.
5. Define the new catalog with the desired name, size, and other attributes. You can use the old catalog as a model.
6. Copy the original catalog into the new catalog using the REPRO NOMERGECAT command. Specify the old catalog in the INDATASET parameter, and the new catalog in the OUTDATASET parameter. Do not specify any other parameters.

   Refer to "Recovering from a REPRO NOMERGECAT Failure" on page 72.

   You can use REPRO with the MERGECAT option for this step. Refer to *z/OS DFSMS Access Method Services Commands* to determine if the MERGECAT option is appropriate for you.

7. Delete the original catalog with the RECOVERY option. This also deletes the old aliases.
8. Use the listing of aliases created with LISTCAT to create a sequence of DEFINE ALIAS commands for each alias. Specify the name of the new catalog in the RELATE parameter.

## Altering Catalog Attributes

When you initially define a catalog, you choose its attributes according to your expectations of what the catalog requires. These requirements can change over time. However, only a portion of a catalog's attributes can be altered.

The following are the important attributes that can be altered:
- Buffer sizes (BUFFERSPACE, BUFND, BUFNI)
- FREESPACE
- MANAGEMENTCLASS
- SHAREOPTIONS
- STORAGECLASS
- STRNO
- WRITECHECK.

To alter these alterable attributes:

1. Use the access method services ALTER command to change the desired attribute. For complete information on the ALTER command, and for examples of altering catalog attributes, see *z/OS DFSMS Access Method Services Commands*.
2. Close the catalog with MODIFY CATALOG,CLOSE so that the CAS control blocks for the catalog are refreshed.

The following attributes are unalterable:
- CONTROLINTERVALSIZE
- DATACLASS
- RECORDSIZE
- REPLICATE or NOREPLICATE for index records.

If you want to alter attributes that are unalterable:

1. Lock the catalog using ALTER LOCK.
2. Export the catalog with EXPORT.
3. Delete the catalog with DELETE RECOVERY.
4. Define a catalog with the same name on the same device with the desired attributes. Specify LOCK so that the new catalog cannot be used.
5. Import the catalog into the newly defined catalog with IMPORT INTOEMPTY. Allow the command to default to UNLOCK, so that the catalog is unlocked.

## Moving, Connecting, and Disconnecting Catalogs

You can move a catalog to another volume, even to a volume of a different device type, by using the EXPORT and IMPORT commands or by using DFSMSdss.

When you use IMPORT CONNECT for a tape volume catalog, you need to specify VOLCATALOG.

You can also move the catalog to another system by connecting it to the system and disconnecting it from the original system

If you connect a catalog to a sharing system, ensure that the share options for the catalog are (3 4); this is the default for the DEFINE USERCATALOG command. If the share options are (3 3), first use the ALTER command to change the share options to (3 4). The device on which the catalog resides must be defined as shared for each system.

## Moving a Catalog to a Different Volume

The procedure for moving a catalog to a different volume is essentially the same as that described in "Changing the Size of a BCS" on page 69.

Before moving any catalog always:

1. Ensure you have a good backup of a valid catalog. Using IDCAMS DIAGNOSE and EXAMINE on the catalog and correcting any errors before taking the backup will ensure you can fall back to a valid backup if there are problems. Having more than one copy is always a good idea in case one copy becomes damaged or lost. This backup can be a full volume dump of the volume the catalog is on, a DFDSS backup or IDCAMS EXPORT. Having all three gives you more options if you need to back out of the catalog move.
2. Use IDCAMS DIAGNOSE and EXAMINE to ensure that the catalog does not have errors.
3. Consider using IDCAMS LISTCAT ALL on the catalog to be moved, in case there are problems. Because this may produce a large volume of output you may wish to redirect the output to a data set, either by pointing the SYSPRINT DD statement to a data set or by using the OFILE keyword on the LISTCAT command.
4. If you are moving a catalog that is shared across systems, a LISTCAT of the master catalog with the ALL keyword is needed to enable a redefine of any aliases that point to the user catalog that is being moved, unless you already have a job set up to define the aliases. If you already have a job, verify that it is current by comparing the LISTCAT output to the job.
5. Quiesce all activity against the catalog that is to be moved.

Run the following commands and, if there are errors, make the required corrections before proceeding with moving the catalog.

```
//DIAGNOSE EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//DIAGDD   DD   DISP=SHR,DSN=your.user.catalog
//SYSIN    DD   *
  DIAGNOSE ICFCATALOG INFILE(DIAGDD)
//EXAMINE1 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
    INDEXTEST NODATATEST ERRORLIMIT(1000)
//EXAMINE2 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
    NOINDEXTEST DATATEST ERRORLIMIT(1000)
//LISTCAT  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  LISTCAT CAT(your.user.catalog) ALL
```

1. Lock the catalog.

```
//LOCKCAT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  ALTER ICFCAT.USER.VSYS303 LOCK
/*
```

2. Issue the following command on each system that shares the catalog:

   ```
   F CATALOG,NOVLF(ICFCAT.USER.VSYS303)
   ```

3. If the catalog is shared with other systems and those other systems do not all share the same master catalog, then EXPORT DISCONNECT the catalog on all other systems other than the system you are running the jobs to resize the catalog on.

   ```
   //EXPDISC  EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=*
   //SYSIN    DD *
     EXPORT ICFCAT.USER.VSYS303 DISCONNECT
   ```

4. Export the BCS with the EXPORT command. The aliases of the catalog are saved with the exported copy, and can be used in later steps to redefine the aliases. You must create a sequential data set for the catalog export, if one does not already exist.

   ```
   //EXPORT  EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=A
   //EXPDD    DD DSN=CATBACK.ICFCAT.USER.VSYS303,DISP=OLD
   //SYSIN    DD *
     EXPORT ICFCAT.USER.VSYS303 -
     OUTFILE(EXPDD) -
     TEMPORARY
   /*
   ```

5. Delete the BCS with the RECOVERY option.

   ```
   //DELCAT  EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=A
   //SYSIN    DD *
     DELETE ICFCAT.USER.VSYS303 -
     RECOVERY -
     USERCATALOG
   /*
   ```

6. Redefine the catalog specifying the new volume. In the example, the new volume is SYSNEW. You may also change the space and performance attributes at this time.

   ```
   //DFNEWCAT EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=A
   //SYSIN    DD *
     DEFINE USERCATALOG -
     (NAME(ICFCAT.USER.VSYS303) -
     VOLUME(SYSNEW) -
     MEGABYTES(15 5) -
     ICFCATALOG -
     LOCK-FREESPACE(20 20) -
     STRNO(3)  -
     REPLICATE )  -
     DATA( CONTROLINTERVALSIZE(4096) -
     BUFND(4)  )  -
     INDEX( CONTROLINTERVALSIZE (4096) -
     BUFNI(4)  )
   /*
   ```

7. Import the BCS using the IMPORT command. Specify INTOEMPTY on the IMPORT command.

   Also, specify ALIAS, so that the aliases exported with the catalog are redefined.

```
//IMPORT  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  IMPORT INDATASET(CATBACK.ICFCAT.USER.VSYS303) -
  OUTDATASET(ICFCAT.USER.VSYS303)        -
  OBJECTS((ICFCAT.USER.VSYS303   -
  VOLUMES(SYSNEW))       -
  ALIAS   -
  LOCK  -
  INTOEMPTY
/*
```

8. If the catalog is shared with other systems that do not share the same master catalog then you will need to IMPORT CONNECT on all other systems that share the catalog.

```
//IMPRTCON EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  IMPORT CONNECT -
  OBJECTS((ICFCAT.USER.VSYS303) -
  DEVICETYPE(3390) -
  VOLUMES(SYSNEW)))
```

9. If the catalog is shared with other systems that do not share the same master catalog then you will need to define the aliases for that catalog on the other systems that do not share the same master catalog.
10. Run the following commands and, if there are errors, correct them before unlocking the catalog.

```
//DIAGNOSE EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//DIAGDD   DD   DISP=SHR,DSN=your.user.catalog
//SYSIN    DD   *
  DIAGNOSE ICFCATALOG INFILE(DIAGDD)
//EXAMINE1 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
    INDEXTEST NODATATEST ERRORLIMIT(1000)
//EXAMINE2 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  EXAMINE NAME(your.user.catalog) -
  NOINDEXTEST DATATEST ERRORLIMIT(1000)
//LISTCAT  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  LISTCAT CAT(your.user.catalog) ALL
.
```

11. If you did not unlock the catalog when you imported it, unlock it with the ALTER UNLOCK command.

    Alternatively, if you do not want to do the DELETE, DEFINE, and IMPORT described in Steps 5 through 7, you can import the catalog to the new volume using the IMPORT command with the OBJECTS parameter. Note that IMPORT deletes the old catalog on the previous device and moves the catalog to the specified different volume.

```
//NEWVOL EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  IMPORT INDATASET(CATBACK.ICFCAT.USER.VSYS303) -
  OUTDATASET(ICFCAT.USER.VSYS303) -
  OBJECTS((ICFCAT.USER.VSYS303 -
```

```
  VOLUMES(SYSNEW)))   -
  ALIAS -
  LOCK
/*
```

In this example, the new volume is SYSNEW, and is specified in the OBJECTS parameter.

If the target device type has different device characteristics, the new catalog can have different data and index component control interval sizes. If you do not want the attributes of the catalog to change use the method described in Steps 4 through 6; that is, delete the catalog with DELETE RECOVERY, define a catalog with the desired attributes on the new volume, and import the original catalog into the newly defined catalog.

Catalog clusters with RACF discrete profiles use the catalog volume for RACF discrete profile checking. If a catalog is moved to a new volume, all catalog clusters with a discrete profile that reside in that catalog must have their RACF discrete profiles updated with the new catalog volume.

### Updating Catalog Connector Records

A connector record is a record in the master catalog for a user catalog. For shared catalogs, this record requires updating if another system has moved the catalog.

If a connector record must be updated due to a change in device type or volume serial number for the catalog, use the IMPORT CONNECT ALIAS command. Specifying ALIAS preserves any aliases already defined for the catalog.

Use the OBJECTS parameter to indicate the changed device type and volume serial number.

A moving user catalogs example follows:

```
//IMPORTCN EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  IMPORT CONNECT ALIAS -
     OBJECTS((ICFCAT.USER.VSYS303 VOLUMES(SYSNEW) -
        DEVT(3390))) -
     CAT(SYS1.MASTERB.ICFCAT)
/*
```

## Moving a Catalog to a Different System

You might need to move a catalog from one system to another. In this case, you are not necessarily changing the physical location of the catalog, but you are changing how the catalog can be accessed and used.

To move a catalog from one system to another system, the catalog must reside on a volume used by the receiving system. If the volume containing the catalog is already shared by the receiving system, you do not have to change the physical location of the catalog. If the volume is not shared, move the catalog to a volume used by the receiving system.

Once the catalog is on a volume that the receiving system can access, use the IMPORT CONNECT command to connect the catalog to the system's master catalog. If you do not want the original system to use the catalog, use EXPORT DISCONNECT to break the connection between the original master catalog and the user catalog you are moving. If the catalog's share options are (3 4) and the device is defined as shared, the catalog can be shared by both systems.

It is not necessary to lock a catalog when using IMPORT CONNECT or EXPORT DISCONNECT. Jobs oriented to a catalog should end normally if the catalog is disconnected. However, if you are permanently moving a catalog to a different system, ensure that users remove references to the catalog from their jobs.

When you use IMPORT CONNECT, the aliases defined for the catalog on one system are not transferred or redefined on the receiving system, even if you specify ALIAS. If you are permanently moving a catalog to a different system, simply import an exported copy of the catalog (without specifying CONNECT), naming the receiving system's master catalog in the CATALOG parameter. Specify ALIAS, so that the aliases are redefined.

Another way to preserve aliases is to use LISTCAT to obtain a list of the aliases. Then, after you have connected the catalog to the receiving system, use DEFINE ALIAS on the receiving system to define the aliases.

The following example shows how to connect a user catalog on SYSTEMA to SYSTEMB using IMPORT CONNECT, and then how to remove the catalog from SYSTEMA. Before removing the catalog from SYSTEMA, the aliases to the catalog are listed, so they can later be defined for the catalog on SYSTEMB. The LISTCAT output is directed to a data set that can later be edited to produce a series of DEFINE ALIAS commands to define the aliases on the receiving system. The alias associations are listed in the Associations group in the LISTCAT output.

```
//SYSMVCT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//LISTDATA DD DSN=LISTCAT.ALIAS.LISTING,DISP=OLD
//SYSIN    DD *
    LISTCAT ALL -
            ENTRIES(ICFCAT.USER.VSYS303) -
            OUTFILE(LISTDATA)

    IMPORT CONNECT -
           OBJECTS((ICFCAT.USER.VSYS303 -
              DEVICETYPE(3390) -
              VOLUMES(SYS303))) -
           CATALOG(SYSTEMB.MASTER.CATALOG)

    EXPORT ICFCAT.USER.VSYS303 -
           DISCONNECT -
           CATALOG(SYSTEMA.MASTER.CATALOG)
/*
```

## Establishing and Breaking Connections between BCSs and VVDSs

Occasionally, you might need to establish or break a connection between the master catalog and a user catalog, or between BCSs and VVDSs. For example, the DIAGNOSE command issues a message if a BCS and VVDS are connected, but there are no data sets on the VVDS's volume that are cataloged in the connected BCS. If this is the expected condition, breaking the connection between the VVDS and the BCS eliminates unwanted messages from DIAGNOSE.

See Figure 4 on page 25 for an illustration of how master and user catalogs, and VVDSs, are connected.

A BCS is connected to the master catalog of each system that shares the catalog. However, when the BCS is defined, it is only connected to the master catalog on

the same system. To connect the catalog to other sharing systems, use the IMPORT CONNECT command. A catalog does not have to exist in order to connect it to a master catalog.

If you specify ALIAS on the IMPORT CONNECT command, any aliases that are already defined in the target master catalog for the catalog are maintained. Aliases that are not already defined in the target master catalog are not added, however. ALIAS can be specified even if no aliases are defined for the catalog.

Alternatively, you might need to disconnect a catalog from a system. In this case, use EXPORT DISCONNECT. This command removes the catalog's entry from the master catalog, and deletes all associated aliases. You do not have to lock a catalog before disconnecting it. Any jobs oriented to the catalog should end normally.

The catalog does not have to exist in order to disconnect it. For instance, if another system has deleted the catalog, any sharing system can disconnect that catalog.

You cannot use DELETE NOSCRATCH to delete a catalog's entry in the master catalog.

The system connects VVDSs with BCSs as the need arises. However, if you want to explicitly establish a connection between a VVDS and a BCS, you can use DEFINE RECATALOG, specifying the VVDS as the entry name, and specifying the BCS in the CATALOG parameter.

To break the connection between a BCS and a VVDS, use DELETE NOSCRATCH, specifying the VVDS as the entry name, and the BCS in the CATALOG parameter. If the VVDS is available, it is checked to determine if the BCS in the CATALOG parameter has any data sets on the volume. If data sets are found on the volume, the DELETE command will fail. If the VVDS is not available, no check is performed.

## Deleting Catalogs and Catalog Entries

Use the access method services DELETE command to delete catalogs, cataloged data sets, objects, tape library entries, and tape volume entries. With this command, you can delete anything you can define or create with access method services.

The access method services ALTER, CREATE, and DELETE commands should only be used to *recover* from tape volume catalog (VOLCAT) errors. Because access method services cannot change the library manager inventory in an automated tape library, ISMF should be used for normal tape library ALTER, CREATE, and DELETE functions. DELETE can also be used for deleting erroneous records from BCSs, VVDSs, and VTOCs.

This section focuses on deleting catalogs and other system or sensitive data. For information on using DELETE to delete the entries for a data set (not the data set itself) from a BCS, VVDS, or VTOC, see “Updating the Catalog After Recovery” on page 103.

Before using the DELETE command, review the command description in *z/OS DFSMS Access Method Services Commands*.

## Deleting Catalogs for Recovery

When you are recovering a catalog, you can temporarily delete the catalog without deleting any data sets cataloged in it. Normally, it is not necessary to delete a catalog before recovering it. If you want to change the size or unalterable attributes of a catalog, however, delete the catalog for recovery, define the catalog with the desired size and attributes, and then recover the catalog.

To delete a catalog for recovery, use the DELETE RECOVERY command. When used on a BCS, DELETE RECOVERY:

- Deletes the connector record for the BCS from the master catalog
- Deletes all aliases for the catalog from the master catalog
- Deletes the VVDS records that describe the BCS
- Deletes the VTOC DSCBs for the BCS
- Does not delete or alter any data set cataloged in the BCS.

You can also delete a VVDS for recovery using DELETE RECOVERY. This is normally only done when you are rebuilding a VVDS. When used on a VVDS, DELETE RECOVERY:

- Deletes the entry for the VVDS from the specified BCS
- Deletes the VTOC DSCB for the VVDS
- Does not delete the BCS and VTOC entries for the data sets and catalog objects that were reflected in the VVDS
- Does not delete the entries for the VVDS in any other BCSs.

Before deleting a VVDS, ensure that sharing systems have been quiesced or have varied the shared DASD offline.

You cannot delete a VVDS that contains entries for data sets cataloged in the master catalog. Because any catalog data set or SMS-managed data set is inaccessible after the VVDS is deleted, delete any BCS on the volume before rebuilding the VVDS. The BCS can later be recovered from a backup copy. If you can, export the BCS before deleting the VVDS.

If a VVDS is not cataloged in any BCS, you must establish a connection with a BCS before deleting the VVDS. This can be done with DEFINE RECATALOG, specifying the VVDS and a catalog in the CATALOG parameter. It can also be done by defining a catalog data set on the volume with the VVDS. You can then delete the catalog data set before deleting the VVDS.

## Deleting a Catalog Permanently

In general, a catalog should be empty before you permanently delete it. If you choose, however, you can delete the catalog and all catalog data sets, SMS-managed non-catalog data sets, or objects cataloged in it.

If the BCS is empty (it contains only records describing itself and the VVDS on its volume), simply delete it using DELETE USERCATALOG. This removes the entries for the BCS from the VTOC and VVDS, and removes the entry in the master catalog. If the BCS is shared with other systems, use EXPORT DISCONNECT on the sharing systems to finish deleting the catalog.

If the BCS no longer exists, but has a connector record in the master catalog, use EXPORT DISCONNECT to delete the connector record.

If the BCS is not empty, and you want to delete the catalog and SMS-managed data sets cataloged in it as well, specify FORCE on the DELETE USERCATALOG command. When you use FORCE:

- The BCS and its aliases are deleted.
- All catalog data sets and objects cataloged in the BCS are deleted.
- All SMS-managed non-catalog data sets are deleted.
- All non-catalog data sets not managed by SMS are uncataloged, but they are not deleted.
- No VVDSs are deleted.

When you delete a non-empty catalog using FORCE, catalog data sets and objects cannot be erased, even if you specify ERASE on the DELETE command. You should delete any sensitive data individually before deleting the non-empty catalog.

## Deleting a VVDS Permanently

A VVDS must be connected to a BCS before it can be deleted. A VVDS must be empty (contain only self-describing entries) before it can be permanently deleted. You cannot permanently delete a VVDS on an SMS-managed volume unless the volume is empty of data sets, because all managed data sets must have VVDS entries.

If the VVDS is not connected to any BCS, use DEFINE RECATALOG to catalog the VVDS. Alternatively, define a catalog cluster on the volume with the VVDS. This catalogs the VVDS in the BCS.

## Deleting Catalog Aliases

To simply delete an alias, use the DELETE ALIAS command, specifying the alias you are deleting.

To delete all the aliases for a catalog, use EXPORT DISCONNECT, to disconnect the catalog. The aliases are deleted when the catalog is disconnected. When you again connect the catalog (using IMPORT CONNECT), the aliases remain deleted.

## Removing All catalog Data from a Volume

You can delete all catalog data sets on a volume with the ALTER REMOVEVOLUMES command if, and only if, the volume is not managed by the Storage Management Subsystem. If the master catalog is specified as the entry name on the ALTER REMOVEVOLUMES command and the target volume is SMS-managed, the volume becomes unusable and you must recover the volume.

When the master catalog is specified, the ALTER REMOVEVOLUMES command deletes all catalog data sets from the specified volume, including any BCS or VVDS on the volume. If a BCS resides on a volume, first export the BCS before using ALTER REMOVEVOLUMES. You can then recover the catalog from the exported copy.

There are three reasons you might want to use ALTER REMOVEVOLUMES to perform catalog volume cleanup:

To recover all and only catalog data sets on a volume,
To delete a BCS on a volume with no VVDS, or
To delete a VVDS that you cannot connect to a BCS.

If you are trying to recover only selected catalog data sets, use the DELETE command instead of ALTER REMOVEVOLUMES.

If you are removing the catalog objects from a volume, use the DIAGNOSE command before and after ALTER REMOVEVOLUMES, to check for lost data or other problems. Use both the ICFCATALOG and VVDS parameters of DIAGNOSE.

Because ALTER REMOVEVOLUMES deletes VVDSs, any connections between a VVDS and BCSs on other volumes are broken. You can reconnect the VVDS to the BCSs using DEFINE RECATALOG. However, when a data set on the volume is accessed, the VVDS is recataloged in the appropriate BCS automatically. After the VVDS is removed, you can delete the VVDS's record from BCSs on other volumes using DELETE NOSCRATCH.

## Catalog Record-Level Sharing

If you rename, alter, move, recover or change the size of a basic catalog structure when using catalog record-level sharing (RLS), you must use the AMS SHCDS CFREPAIR command to correct RLS information in the catalog. The catalog must be import connected on all systems to the master catalog before the CFREPAIR command can be used. The AMS SHCDS CFREPAIRDS command may be used to correct the RLS information for individual RLS data sets. However, caution must be taken to identify all data sets used as RLS data sets. Otherwise, data may be lost.

If you are using catalog RLS support and decide to no longer use it, the AMS SHCDS CFRESET command is used to reset applicable RLS indicators in the catalog. See *z/OS DFSMS Access Method Services Commands* for more information about the SHCDS command.

## Deleting Sensitive Data

Typically, when you delete a data set, only the catalog, VVDS, and VTOC information is removed. The information on the disk or tape that the data set occupied is unchanged; only the means of locating and accessing the information is actually deleted. Until that space is used again, the information could be read by a program that can find the data.

To protect sensitive information, you can erase the information when you delete it. Information is erased by overwriting it with binary zeros before the space is made available for other allocations.

You can control the erasure of data with the Resource Access Control Facility (RACF), the DEFINE command, or the DELETE command. Data is erased according to the following rules:

1. If the RACF generic or discrete profile specifies ERASE, the data is erased.
2. If ERASE is specified on the DELETE command, the data is erased, even if the RACF profile specifies NOERASE.
3. If NOERASE is specified on the DELETE command, the data is erased only if the RACF profile specifies ERASE. If NOERASE is desired, use a RACF command to change the attribute before using the DELETE command.
4. If a catalog cluster or alternate index is deleted, and it was defined with ERASE specified, the data is erased unless NOERASE is specified on the DELETE command.

For more information about the RACF ERASE attribute, see "RACF-Controlled ERASE Options" on page 91.

# Chapter 5. Protecting Catalogs

The protection of data includes:
- Data security—the safety of data from theft or intentional destruction
- Data integrity—the safety of data from accidental loss or destruction.

Data can be protected either indirectly, by preventing access to programs that can be used to modify data, or directly, by preventing access to the data itself. Catalogs and cataloged data sets can be protected in both ways.

To protect your catalogs and cataloged data, use the authorized program facility (APF) and the Resource Access Control Facility (RACF). Catalog passwords and USVRs are no longer used.

Because of its special use, you cannot export, or import a VVDS, nor can you alter its attributes by using the ALTER command.

## Authorized Program Facility Protection for Access Method Services

The authorized program facility (APF) limits the use of sensitive system services and resources to authorized system and user programs.

For information about using APF for program authorization, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

All access method services load modules are contained in SYS1.LINKLIB, and the root segment load module (IDCAMS) is link-edited with the SETCODE AC(1) attribute. These two characteristics ensure that access method services executes with APF authorization.

APF authorization is established at the job step task level. If a load request is satisfied from an unauthorized library during the execution of an APF authorized job step, the task is abnormally terminated. It is the installation's responsibility to ensure that a load request cannot be satisfied from an unauthorized library during access method services processing.

The following situations could cause the invalidation of APF authorization for access method services:
- An access method services module is loaded from an unauthorized library, or invoked by an unauthorized program.
- A user-security verification routine (USVR) is loaded from an unauthorized library during access method services processing.
- An exception installation or user exit routine is loaded from an unauthorized library during access method services processing.
- A user-supplied special graphics table is loaded from an unauthorized library during access method services processing.

Because APF authorization is established at the job step task level, access method services is not authorized if invoked by an unauthorized application or terminal monitor program.

Under the time sharing option (TSO), if the system does not have the TSO Command Package Program Product, you can authorize your terminal monitor program by link-editing it with the SETCODE AC(1) attribute. You must enter the names of those access method services commands requiring APF authorization to execute under TSO in the authorized command list (AUTHCMD) in the SYS1.PARMLIB member IKJTSO*xx* or added to the CSECT IKJEGSCU. See *z/OS TSO/E Customization* for more information.

The restricted functions performed by access method services that cannot be requested in an unauthorized state are:

- DEFINE—when the RECATALOG parameter is specified
- DEFINE—when the define is for an alias of a UCAT
- DELETE—when the RECOVERY parameter is specified
- EXPORT—when the object to be exported is a BCS
- IMPORT—when the object to be imported is a BCS
- PRINT—when the object to be printed is a catalog
- REPRO—when a BCS is copied or merged
- VERIFY—when a BCS is to be verified.
- SHCDS—all functions

# Resource Access Control Facility (RACF) Protection

You should use RACF to protect your data sets and catalogs. Only RACF and APF are used with SMS.

## RACF Authorization Checking

To open a catalog as a data set, you must have ALTER authority and APF authorization. When defining an SMS-managed data set, the system only checks to make sure the user has authority to the data set name and SMS classes and groups. The system selects the appropriate catalog, without checking the user's authority to the catalog. You can define a data set if you have ALTER or OPERATIONS authority to the applicable data set profile.

Deleting any type of RACF-protected entry from a RACF-protected catalog requires ALTER authorization to the catalog or to the data set profile protecting the entry being deleted. If a non-catalog data set is SMS-managed, RACF does not check for DASDVOL authority. If a non-catalog, non-SMS-managed data set is being scratched, DASDVOL authority is also checked.

Altering the passwords in a RACF-protected catalog entry requires ALTER authority to the entry being altered, or the OPERATIONS attribute. ALTER authority to the catalog itself is not sufficient for this operation.

For ALTER RENAME, the user is required to have the following two types of authority:

1. ALTER authority to either the data set or the catalog
2. ALTER authority to the new name (generic profile) or CREATE authority to the group.

Be sure that RACF profiles are correct after you use REPRO MERGECAT on a catalog that uses RACF profiles. If the target and source catalogs are on the same volume, the RACF profiles remain unchanged.

REPRO MERGECAT will preserve RACF discrete profiles when the target and source catalog are on different volumes. Profiles will be updated with the target

volume, except when the protected data set is DFSMShsm migrated. Profiles for DFSMShsm migrated data sets must be manually changed using RACF commands. Be sure to verify the integrity of discrete profiles after MERGECAT. You should use generic profiles to avoid this situation.

Non-catalog tape data sets defined in a catalog can be protected by:
- Controlling access to the tape volumes; or
- Controlling access to the individual data sets on the tape volumes.

Note that if you run RACF in "warn" mode, you may receive indications of access violations. Catalog processing uses two-step verification for many types of functions. The first test checks to see if the user has authority to the specific data set. If this request fails the security check, the system will attempt to verify if the user has the appropriate authority to the containing catalog. If this request succeeds, the access is granted. However, in warn mode a message will be produced for the first security check that failed, even though the user passes the stated security checks for the access. These messages can be ignored, as they will disappear when RACF is no longer running in 'warn' mode.

## Generic Profile-Checking Facility

RACF provides a generic profile-checking facility. With the always-call capability of catalogs, you can consolidate the access authorization requirements of several similarly named and similarly used data sets under a single generic profile definition. A generic profile is used to protect one or more data sets that have identical access requirements. For example, you can build a generic profile named ‘userid.*’ to protect all data sets cataloged under the same high-level qualifier. For more information, see *z/OS Security Server RACF Command Language Reference*.

catalog data sets that are protected by generic profiles are not RACF-indicated in the catalog. Therefore, RACF is always called for any access to data sets that are cataloged. If the data set is not protected by either a discrete profile or a generic profile, no protection is in effect. The catalog does not have to be RACF-protected in order for its data sets to be RACF-protected.

For catalog clusters that are cataloged, all the components of the catalog cluster are protected by one profile (the profile that protects the cluster name). This profile can be discrete or generic. You do not need to create profiles that protect the index and data components of a cluster.

Data sets protected with discrete profiles are flagged as “RACF-indicated.” If a data set protected by a discrete profile is moved to a system where RACF is not installed, no user is given authority to access the data set. However, if the data set is protected with a generic profile, it is not flagged as “RACF-indicated”; therefore, access authority is determined by normal catalog password protection.

## Controlling Catalog Functions with RACF Profiles in the FACILITY Class

By defining and controlling access to profiles in the FACILITY class, you can control who can use certain catalog functions. Besides defining these profiles, you must activate the FACILITY class for these functions to be protected.

These profiles can be assigned to an owner. For example, the person responsible for managing catalogs can be assigned ownership of these profiles. A profile owner can then list, modify, or delete the profiles as needed.

The following RACF commands show how to define a FACILITY profile, authorize a user to perform the functions restricted by the profile, and activate the FACILITY class. The profile defined is IGG.CATLOCK, which is assigned to user CATADMIN, and user USER01 is authorized to use the profile.

```
RDEFINE  FACILITY  IGG.CATLOCK  UACC(NONE) OWNER(CATADMIN)

PERMIT  CLASS(FACILITY)  IGG.CATLOCK  ID(USER01) ACCESS(READ)

SETROPTS CLASSACT(FACILITY)
```

The RDEFINE command creates the profile and gives it a universal access authority (UACC) of NONE. Because READ authority to the profile allows a user to perform the protected function, you must use a UACC of NONE to limit the use of the protected function. The PERMIT command is used to authorize the appropriate users or groups to perform the protected function. If the FACILITY class is already active, the SETROPTS command is not necessary.

## Controlling Who Can Lock a Catalog (IGG.CATLOCK Profile)

The IGG.CATLOCK profile, in conjunction with normal security checking, controls who can lock a catalog and who can access a locked catalog.

If you have READ access to the IGG.CATLOCK profile and ALTER authority to the catalog, you can lock or unlock a catalog. If you have READ access to the IGG.CATLOCK profile, you can access and repair a locked catalog. If the IGG.CATLOCK profile is not defined, or the FACILITY class is not active, you cannot lock or unlock a catalog.

In previous releases, a user defined as privileged or trusted to RACF would automatically have read access to this facility class. However, you must now explicitly authorize those privileged or trusted users to this facility class if they need access to locked catalogs. There are products that are privileged (such as DFSMShsm) because they perform a variety of actions against data sets and catalogs. Without requiring explicit authorization to the facility class, these products could inadvertently access or change locked catalogs, causing damaged catalogs or unexpected results. If these products or components are to be used while accessing a locked catalog, they must be explicitly authorized to the facility class. It is recommended they be authorized only for the duration of the specific need.

To ensure the integrity of catalogs, restrict authority to the IGG.CATLOCK profile to only the necessary people or system components. To define entries in a catalog, users only need UPDATE authority to the data set profile protecting the catalog. Therefore, you should consider specifying UACC(UPDATE) for the data set profiles protecting user catalogs. To delete entries in a catalog, users need either ALTER authority to the data set or ALTER authority to the catalog. We recommend that you only give users ALTER authority to their own data sets.

See “Locking a Catalog” on page 96 for an explanation of catalog locking and “Recovering a BCS” on page 97 for an example of locking a catalog during recovery.

## Storage Administration (STGADMIN) Profiles in the FACILITY Class

To control the ability to perform functions associated with storage management, define profiles in the FACILITY class whose profile names begin with STGADMIN (storage administration). The STGADMIN.IGG facility classes are only intended for SMS data sets.

If defined, these profiles are checked before a user is allowed to perform the protected function. Users must have read access to the specific profile in order to use the protected functions. If these profiles are not defined, other RACF checking is still made to verify authority.

Note that the following STGADMIN.IGG statement permissions are checked with LOG=NONE:

- STGADMIN.IGG.DELNVR.NOBCSCHK
- STGADMIN.IGG.DEFDEL.UALIAS
- STGADMIN.IGG.DEFINE.RECAT
- STGADMIN.IGG.DELETE.RENAME

Some FACILITY profiles are not checked if the caller is using the system key or is running in supervisor state. These profiles are:

STGADMIN.IGG.DEFDEL.UALIAS
STGADMIN.IGG.DEFNVSAM.NOBCS
STGADMIN.IGG.DEFNVSAM.NONVR
STGADMIN.IGG.DELETE.NOSCRTCH
STGADMIN.IGG.DELGDG.FORCE
STGADMIN.IGG.DELGDG.RECOVERY
STGADMIN.IGG.DELNVR.NOBCSCHK
STGADMIN.IGG.DIRCAT
STGADMIN.IGG.LIBRARY
STGADMIN.IGG.DEFINE.RECAT

Define the following classes to protect catalog functions. For a complete list of STGADMIN profiles, see *z/OS DFSMSdfp Storage Administration*.

**STGADMIN.IDC.DIAGNOSE.CATALOG**
protects the ability to use the access method services DIAGNOSE command against catalogs.

**STGADMIN.IDC.DIAGNOSE.VVDS**
protects the ability to use the access method services DIAGNOSE command against a VVDS when a comparison is made to a BCS.

**STGADMIN.IDC.EXAMINE.DATASET**
protects the ability to use the access method services EXAMINE command on catalogs.

**STGADMIN.IGG.ALTER.SMS**
controls the ability to alter the storage class and management class of an SMS-managed data set. If the profile is not built, the user must have RACF authority to the storage class and the management class to alter it. To use this profile, the administrator must have ALTER access to the data set whose storage or management class is to be changed.

**STGADMIN.IGG.ALTER.UNCONVRT**
protects the ability to alter an SMS-managed catalog data set to an unmanaged catalog data set.

**STGADMIN.IGG.DEFDEL.UALIAS**
allows you to define or delete an alias related to a usercatalog without further authorization checking.

**STGADMIN.IGG.DEFNVSAM.NOBCS**
controls the ability to define or alter a NVR for a data set without affecting

the BCS entry if one exists. This profile is only checked by authorized services using the LOCATE macro, not by utilities like IDCAMS.

**STGADMIN.IGG.DEFNVSAM.NONVR**
controls the ability to define or alter a BCS for a data set without affecting the VVDS entry if one exists. This profile is only checked by authorized services using the LOCATE macro, not by utilities like IDCAMS.

**STGADMIN.IGG.DELETE.NOSCRTCH**
protects the ability to delete the BCS entry for an SMS-managed data set without deleting the data set itself (for example, using DELETE NOSCRATCH). This protects against functions that uncatalog data sets.

**STGADMIN.IGG.DELGDG.FORCE**
protects the ability to use DELETE FORCE on a generation data group that contains an SMS-managed generation data set. The DELETE GDG FORCE command deletes SMS generation data sets referenced by the generation data group. It also removes the generation data group entry in the catalog.

**STGADMIN.IGG.DELGDG.RECOVERY**
this command deletes the generation data group and uncatalogs the SMS generation data sets. When you use this command, the generation data group entry is deleted from the catalog and generation data sets remain unaffected in the VTOC, and if SMS managed, in the VVDS.

**STGADMIN.IGG.DELNVR.NOBCSCHK**
protects the ability to delete the VVDS entry (the NVR) for an SMS-managed non-catalog data set and to bypass the catalog name and BCS entry checking. If there is a BCS entry or if the catalog name contained in the NVR does not match the catalog provided in the request, the function is denied unless the user has authority to this profile.

**STGADMIN.IGG.DIRCAT**
protects the ability to direct a catalog request to a specific catalog, bypassing the normal catalog search. A directed catalog request is one in which the catalog name is explicitly passed to catalog management in the CATALOG parameter of access method services commands.

**Note on catalog requests in SMS:** In an SMS environment, all the catalog requests against SMS-managed data sets should be satisfied by the normal catalog search order. You must be authorized to this facility class in order to direct the catalog request to a specific catalog, unless you are using one of the following commands:

- LISTCAT
- DEFINE ALIAS of a usercatalog
- IMPORT CONNECT
- EXPORT CONNECT
- LISTCAT LEVEL, and other catalog commands that list the catalog in a generic manner.

**STGADMIN.IGG.DLVVRNVR.NOCAT**
protects the ability to delete a VVR or NVR without an associated catalog. Users having RACF READ authority to the facility class will need no other RACF authority to the master catalog to perform the DELETE NVR or DELETE VVR functions.

**Note:** Access to this facility class should be restricted to users who understand the risk involved in deleting a VVR or NVR entry from a VVDS.

**STGADMIN.IGG.DELETE.RENAME**
controls the ability to delete data set entries flagged as "rename in process". Attempts without the facility class for data sets flagged in this manner receive message IDC3009I with a return code of 90 and a reason code of 54. The "rename in progress" flag is ignored for users having RACF READ authority to the facility class and issuing a DELETE, and the entry is deleted. This facility class is intended for maintenance purposes.

**STGADMIN.IGG.LIBRARY**
protects the ability to DEFINE, DELETE or ALTER tape library and tape volume entries.

**STGADMIN.IGG.DEFINE.RECAT**
controls the ability to DEFINE RECATALOG a data set without having any authorization to the data set. The only data set authorization is:

- Users must have ALTER authority to the target and source catalog while performing a REPRO MERGECAT
- Users must have UPDATE authority to the target catalog while performing a DEFINE RECATALOG

The primary purpose of this RACF facility class is for REPRO MERGECAT command processing. Historically, there was a security restriction in REPRO MERGECAT processing where in Catalog Management requires the catalog administrator who executes the REPRO MERGECAT command to have ALTER authority to the data set(s). With this RACF facility class, the REPRO MERGECAT function does not require ALTER authority to the data set(s) being moved.

In order to use the REPRO MERGECAT command, you must do the following RACF set-up:

- ALTER authority to both source and target catalogs
- READ authority to the following RACF facility classes:
  - STGADMIN.IGG.DELETE.NOSCRTCH
  - STGADMIN.IGG.DEFINE.RECAT

## RACF-Controlled ERASE Options

DELETE processing removes catalog and VTOC information; it also makes the associated DASD space available for a new allocation. By default, this process does not erase the data from the disk. Data is erased by overwriting it with binary zeros. Sensitive data should be erased before its space is made available.

You can use RACF commands to specify an ERASE or NOERASE attribute in generic or discrete profiles. When so specified, these attributes become default attributes for:

- DELETE processing of an alternate index or cluster cataloged in a catalog.
- Scratch and partial release processing of non-catalog data sets (see *z/OS DFSMSdfp Advanced Services* for more information).

See “Deleting Sensitive Data” on page 83 for more information on deleting sensitive data.

# Chapter 6. Backing Up and Recovering Catalogs

Because catalogs are essential system data sets, it is important that you maintain backup copies. The more recent and accurate a backup copy, the less impact a catalog outage will have on your installation.

Besides the backup and recovery of BCSs, you should also develop a strategy for backing up VVDSs, VTOCs, and VTOC indexes.

This chapter considers the backup and recovery of BCSs and VVDSs, and indirectly, the VTOC. For more information about data set back up and recovery, see the following:

- *z/OS DFSMSdfp Storage Administration*
- *z/OS DFSMS Using Data Sets*
- *z/OS DFSMSdss Storage Administration*
- *z/OS DFSMShsm Storage Administration*

## Developing a Backup and Recovery Strategy

A primary consideration for backing up catalogs is backup frequency. In general, a catalog recovery takes less time the more recently the backup copy was taken. However, continuously creating backup copies can be a drain on your system. Daily catalog backup should be sufficient for most catalogs.

Backup procedures can be simplified if you are using the Storage Management Subsystem. You can define a management class for your catalogs and other system data sets, defining an appropriate backup frequency. The system then creates backups of the catalog according to your management policy.

Your backup strategy should include running IDCAMS DIAGNOSE and EXAMINE functions against each catalog before you perform a back up. This ensures that you back up only valid catalogs.

The BCS can be backed up as a data set using the EXPORT command, DFSMSdss, or DFSMShsm. The aliases defined for the catalog are saved with the backup copy when EXPORT or DFSMShsm is used, or when logical dump is used with DFSMSdss. When you recover the catalog by importing it, you can have the saved aliases redefined and merged with existing aliases. DFSMSdss and DFSMShsm redefine the aliases automatically. However, when you use the IMPORT command, you must specify ALIAS to have aliases redefined or retained.

Before recovering a BCS, you should lock the catalog to prevent access during the recovery. Before you can lock the catalog, no job can have the catalog allocated with a disposition of OLD. This disposition can be defined on a DD statement. It can also result if the catalog is used in the OUTDATASET parameter of access method services commands. The RACF FACILITY class profile IGG.CATLOCK must be defined to allow the use of the LOCK parameter with ALTER, DEFINE USERCATALOG, and IMPORT to those users having READ authority to this class. All other access is restricted from LOCK use.

The VVDS and VTOC should not be backed up as data sets, but are backed up as part of a full volume dump using DFSMSdss or DFSMShsm. The entries in the

VVDS and the VTOC are backed up with the data sets they describe when the data sets are backed up with the IDCAMS EXPORT command, DFSMShsm, or DFSMSdss logical dump.

There are two ways that a VVDS or VTOC can be recovered:
1. Restore the volume containing the VVDS or VTOC, or
2. Rebuild the VVDS and VTOC by recovering the data sets on the volume.

Restoring the volume is the easiest way to recover a VVDS or VTOC. However, this is seldom practical because the data sets restored will not be current. To rebuild the VVDS, you must delete it and then recover all VSAM and SMS-managed data sets which were on the volume.

The VTOC can be backed up with volume dumping, and can be restored by restoring the volume. To rebuild a VTOC, you have to use ICKDSF to initialize the volume. Then, all data (not only VSAM data sets), must be recovered to the volume. Do not try to repair a VTOC by manually rebuilding damaged records.

BCS backups and volume dumps can be stored on DASD or tape. More than one backup should be kept, since the most recent backup might also be damaged.

See *z/OS DFSMS Implementing System-Managed Storage* for a more extensive discussion of backup and recovery considerations.

## Backing Up a Catalog

The two parts of a catalog, the BCS and the VVDS, require different backup techniques: the BCS can be backed up like other data sets, whereas the VVDS can only be backed up as part of a volume dump. The entries in the VVDS and VTOC are backed up when the data sets they describe are exported with access method services, logically dumped with DFSMSdss, or backed up with DFSMShsm.

### Backing Up a BCS

You can use the access method services EXPORT command, the DFSMSdss logical DUMP command, or the DFSMShsm BACKDS command to back up a BCS. You can later recover the backup copies using the same utility used to create the backup: the access method services IMPORT command for exported copies; the DFSMSdss RESTORE command for logical dump copies; and the DFSMShsm RECOVER command for DFSMShsm backups.

The copy created by these utilities is a “portable” sequential data set that can be stored on a tape or direct access device, which can be of a different device type than the one containing the source catalog.

When these commands are used to back up a BCS, the aliases of the catalog are saved in the backup copy. The source catalog is not deleted, and remains as a fully functional catalog. The relationships between the BCS and VVDSs are unchanged.

You cannot permanently export a catalog by using the PERMANENT parameter of EXPORT. The TEMPORARY option is used even if you specify PERMANENT or allow it to default.

To ensure the integrity of the copy, access to the BCS is serialized by these commands. This serialization prevents update access but allows read access from the system performing the command. For a catalog in non-RLS mode, if you are

using multi-system global resource serialization (GRS) or equivalent product to convert the catalog RESERVE to GLOBAL ENQUEUE, other sharing systems can also have read access to the catalog. Otherwise, the RESERVE (with SYSIGGV2) makes the catalog inaccessible from other systems for the duration of the command. For a catalog in RLS mode, a different serialization technique is used since neither RESERVE nor GLOBAL ENQUEUE with SYSIGGV2 is raised for both read and update accesses. The access method services EXPORT command, the DFSMSdss logical DUMP command, and the DFSMShsm BACKDS command all adapt this serialization technique to perform a backup. See 'Advanced Customization Guide' for more information.

These commands do not back up corresponding entries in any related VVDS except those describing the BCS itself.

For more information on using DFSMSdss, see *z/OS DFSMSdfp Storage Administration*. For more information on using DFSMShsm, see *z/OS DFSMShsm Storage Administration*.

**Restriction:** You cannot use IDCAMS REPRO or other copying commands to create and recover BCS backups.

### Backing Up a Master Catalog

A master catalog can be backed up as any other BCS. You should use EXPORT, DFSMSdss, or DFSMShsm for the backup. Another way to provide a backup for the master catalog is to create an alternate master catalog. For information on defining and using an alternate master catalog, see "Creating and Using an Alternate Master Catalog" on page 26.

You should also make periodic volume dumps of the master catalog's volume. This dump can later be used by the stand-alone version of DFSMSdss to restore the master catalog, if you cannot access the volume from another system.

### Backing up a VVDS

The VVDS should not be backed up as a data set to provide for recovery. To back up the VVDS, back up the volume containing the VVDS, or back up all data sets described in the VVDS (all VSAM and SMS-managed data sets). If the VVDS ever needs to be recovered, recover the entire volume, or all the data sets described in the VVDS.

You can use either DFSMSdss or DFSMShsm to back up and recover a volume or individual data sets on the volume. For further information, see *z/OS DFSMSdfp Storage Administration* and *z/OS DFSMShsm Storage Administration*.

## Recovering a Catalog

Normally, a BCS is recovered separately from a VVDS. A VVDS usually does not need to be recovered, even if an associated BCS is recovered. However, if you need to recover a VVDS, and a BCS resides on the VVDS's volume, you must recover the BCS as well. If possible, you should export the BCS before recovering the volume, and then recover the BCS from the exported copy. This ensures a current BCS.

Before recovering a BCS or VVDS, try to recover damaged records. If damaged records can be rebuilt, you can avoid a full recovery. BCS records can be recovered using access method services DELETE and DEFINE commands with appropriate

parameters. VVDS and VTOC records can be recovered using the DELETE command and by recovering the data sets on the volume. See "Updating the Catalog After Recovery" on page 103 for more information on recovering individual entries.

## Locking a Catalog

You should restrict access to a catalog when you are recovering it or when you are performing other maintenance procedures which involve redefining the catalog. If you do not restrict access to the catalog (by locking it, by terminating user sessions, or by another method), users might be able to update the catalog during recovery or maintenance and create a data integrity exposure. Locking the catalog eliminates the need to terminate user sessions during catalog recovery or maintenance.

You can only lock user catalogs. You cannot lock a master catalog. While the catalog is locked, unauthorized requests to access the catalog fail with a return code indicating that the catalog is temporarily unavailable. Jobs entered with JCL fail with a JCL error. You cannot make JCL jobs wait until the catalog is unlocked. The catalog is also unavailable to any system that shares the catalog.

After you have completed the catalog recovery or maintenance, unlock the catalog so that normal operations can resume.

| To lock or unlock a catalog with a serialized close across the sysplex, you use the LOCK and UNLOCK parameters of the access method services ALTER, DEFINE, or IMPORT commands. You use the ALTER command to lock or unlock an existing catalog; the DEFINE command to lock a newly defined catalog (the default is to define an unlocked catalog); and the IMPORT command to lock or unlock a catalog that you are importing. In order to lock or unlock a catalog, you must have READ access to the IGG.CATLOCK profile in RACF, and ALTER authority to the catalog. If the catalog is shared between systems, you may want to ensure you provide access to those users who may need to access the catalog while it is locked (such as VTAM®, described below).

| Another way to lock or unlock a catalog with a serialized close across the sysplex is to use the following modify commands:

```
F CATALOG,RECOVER,LOCK(ucatname*)
F CATALOG,RECOVER,UNLOCK(ucatname*)
```

**Exception:** Catalogs are not unlocked during a system IPL. If you lock a catalog, and there is a system failure, the catalog is still locked after you IPL the system. This can cause problems if a locked catalog contains entries for data sets needed during IPL.

For example, if the catalog containing entries needed for VTAM is locked, VTAM cannot be started. Because VTAM is needed to start TSO, and TSO must be active to issue the ALTER command or submit a batch IDCAMS job, you cannot use ALTER UNLOCK to unlock the catalog.

As long as TSO is available, you can simply use ALTER UNLOCK to unlock the catalog and allow the IPL to complete. However, you can also authorize VTAM to the IGG.CATLOCK profile. This allows VTAM to access a locked catalog. If you authorize VTAM to IGG.CATLOCK, you should also authorize any other components which are needed to start VTAM.

Jobs such as VTAM and any other critical system resource should be given CATLOCK authority prior to locking any user catalog. If TSO is not available, and VTAM cannot be started because it does not have access to the IGG.CATLOCK profile, you must use a card reader to enter an IDCAMS ALTER UNLOCK job into the system to unlock the catalog.

If the catalog is shared between systems, it may be unlocked from any of the shared systems. Thus the catalog does not need to be unlocked from the system that locked it. This provides an alternative way to recover when the system that locked the catalog cannot be used to unlock it.

## Suspending a Catalog

Another way to restrict access to the catalog is to suspend a catalog. Similar to locking a catalog, you can only suspend user catalogs. You cannot suspend a master catalog. While the catalog is suspended, unauthorized requests to access the catalog will be suspended automatically in the user's address space waiting on SYSZIGG4 BCS ENQ. Because the requests are suspended, they will not return back to the user with any return code. The catalog is also unavailable to any system that shares the catalog. After you have completed the catalog recovery or maintenance, resume the catalog so that normal operations can continue.

To suspend or resume a catalog with a serialized close across the sysplex, use the SUSPEND and RESUME parameters of the access method services ALTER, DEFINE, or IMPORT commands. The ALTER command is used to suspend or resume an existing catalog; the DEFINE command is used to suspend or resume a newly defined catalog; and the IMPORT command to suspend or resume a catalog during a restore. In order to suspend or resume a catalog, you must have READ access to the IGG.CATLOCK profile in RACF, and ALTER authority to the catalog. If the catalog is shared between systems, you may want to ensure you provide access to those users who may need to access the catalog while it is suspended.

Another way to suspend or resume a catalog with a serialized close across the sysplex is to use the following modify commands:

```
F CATALOG,RECOVER,SUSPEND(ucatname*)
F CATALOG,RECOVER,RESUME(ucatname*)
```

If the catalog is shared between systems, it may be resumed from any of the shared systems. Thus the catalog does not need to be resumed from the system that suspended it. This provides an alternative way to recover when the system that suspended the catalog cannot be used to resume it.

## Recovering a BCS

You can recover a BCS that was backed up with the access method services EXPORT command, the DFSMSdss logical DUMP command, or the DFSMShsm BACKDS command or automatic backup. To recover the BCS, use the IDCAMS IMPORT command, the DFSMSdss RESTORE command, or the DFSMShsm RECOVER command.

When you recover a BCS using these commands, you do not need to delete and redefine the target catalog unless you want to change the catalog's size or other characteristics, or unless the BCS is damaged in such a way as to prevent the usual recovery. The recovered catalog is reorganized when you use IMPORT or RECOVER, but not when you use RESTORE.

Aliases to the catalog can be defined if you use DFSMSdss, DFSMShsm, or if you specify ALIAS on the IMPORT command. If you have not deleted and redefined the catalog, all existing aliases are maintained, and any aliases defined in the backup copy are redefined if they are not already defined.

If you do not first delete the catalog you are recovering (specifying RECOVERY), the catalog is deleted and redefined according to the attributes of the backup copy. If you delete and redefine the catalog before you recover it (and the newly defined catalog is empty), the backup copy is copied into the new catalog. Specify INTOEMPTY on the IMPORT command if you define the catalog before importing it.

When using IMPORT with the OUTFILE parameter, the DD statement defining the catalog must have a disposition of OLD.

Before you recover a BCS, lock the BCS. After you recover the catalog, update the BCS with any changes which have occurred since the last backup. You can use the access method services DIAGNOSE command to identify certain unsynchronized entries.

To simplify catalog recovery and improve data availability for your installation, use the Integrated Catalog Forward Recovery Utility (ICFRU) to update the BCS to a current status. For more information, see Chapter 9, "Integrated Catalog Forward Recovery Utility (ICFRU)," on page 163.

The following are the steps needed to recover a BCS using the IMPORT command. For further discussion of using DFSMSdss and DFSMShsm in BCS recovery, see *z/OS DFSMSdss Storage Administration* and *z/OS DFSMShsm Storage Administration*. The catalog being recovered is SYS1.ICFCAT.PROJECT1.

1. If the catalog is used by the job scheduler for any batch jobs, hold the job queue for all job classes except the one you use for the recovery. Interactive users are returned appropriate messages if a catalog is locked and they try to use it. However, batch jobs fail with a JCL error if they try to use a locked catalog.
2. Lock the catalog so that access is restricted. Note that in order to lock or unlock a catalog, you must have READ access to the IGG.CATLOCK profile in RACF, and ALTER authority to the catalog. See "Locking a Catalog" on page 96 for more information.

   ```
   //LOCKCAT EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=A
   //SYSIN    DD *
         ALTER SYS1.ICFCAT.PROJECT1 LOCK
   /*
   ```

3. Use the Integrated Catalog Forward Recovery Utility to create an updated EXPORT backup copy of the BCS, using the last backup copy and the appropriate SMF records.
4. Import the most current backup copy of the BCS (which contains the BCS's aliases as they existed when the backup was made) using the access method services command IMPORT ALIAS LOCK. The most current backup is the one created in step 3. If the aliases are not needed, do not specify the ALIAS parameter. If you do not specify LOCK, the catalog is unlocked.

   ```
   //RECOVER  EXEC PGM=IDCAMS
   //BACKCOPY DD DSN=BACKUP.SYS1.ICFCAT.PROJECT1,DISP=OLD
   //SYSPRINT DD SYSOUT=A
   //SYSIN    DD *
       IMPORT INFILE(BACKCOPY) -
   ```

```
        OUTDATASET(SYS1.ICFCAT.PROJECT1) -
        ALIAS -
        LOCK

 /*
```

5. If you did not use the Integrated Catalog Forward Recovery Utility to create an updated backup copy of the BCS, you need to manually check for recent changes to the BCS which are not reflected in the recovered copy.

   Use the SMF records which record changes to the catalog, and any tape management records you keep. An DFSMShsm audit might also be helpful.

   The backup or portable copy of a catalog reflects the contents of the catalog at the time it was backed up with EXPORT, DFSMSdss, or DFSMShsm. Any subsequent ALTER, DEFINE or DELETE operations are not reflected in the catalog when it is imported.

   The access method services command DIAGNOSE can be used after importing the catalog, to help assess activity not reflected in the imported catalog. However, DIAGNOSE is only useful for VSAM or SMS-managed data sets on DASD. Tape data sets cannot be processed by DIAGNOSE. List the VVDSs connected to the catalog to determine which VVDSs to compare with the BCS.

   If you were able to list the non-VSAM data sets before recovering the catalog, compare the list taken before the recovery with one taken after the recover. Make any needed changes by following the procedures in “Updating the Catalog After Recovery” on page 103.

   For complete information on using DIAGNOSE, see “Analyzing a Catalog for Synchronization Errors” on page 109.

6. Use the EXAMINE and DIAGNOSE commands to check the integrity of the recovered catalog's structure and content. If EXAMINE indicates structural errors, you need to recover the next most recent backup copy, repeating these steps. If DIAGNOSE indicates errors, address the errors as appropriate.

```
//EXAMINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
   EXAMINE NAME(SYS1.ICFCAT.PROJECT1) -
           INDEXTEST -
           DATATEST
/*
```

7. If you recovered the catalog onto a volume with a different volume serial number or device type (for example, if the catalog was damaged due to volume damage), and the catalog is shared with other systems, use IMPORT CONNECT ALIAS to update the catalog connector records in the master catalogs of the sharing systems. See “Recovering Shared Catalogs” for more information on recovering shared catalogs.

8. Unlock the catalog with the command ALTER UNLOCK to allow general access.

```
//UNLCKCAT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    ALTER SYS1.ICFCAT.PROJECT1 UNLOCK
/*
```

9. Free the job queue if you put it on hold.

## Recovering Shared Catalogs

When two or more systems share access to a user catalog and the user catalog is recovered on one of those systems to a different volume or device type, you will need to update or replace the catalog connector record in the master catalogs on

the other systems. Otherwise, the recovery considerations for shared catalogs are the same as those described in “Recovering a BCS” on page 97.

The best way to update the catalog connector record on the sharing systems is to use the access method services IMPORT CONNECT ALIAS command. This command maintains any aliases already defined for the catalog. You can use this command even if the catalog has no aliases defined for it.

If you want to delete the aliases for a catalog, you can use EXPORT DISCONNECT followed by IMPORT CONNECT.

For example, the user catalog SYS1.ICFCAT.SHARED, which has many aliases, resides on volume 339001 (a 3390), and is shared by SYSTEMA and SYSTEMB. If SYS1.ICFCAT.SHARED is successfully recovered by SYSTEMA to volume 339002 (another 3390), SYS1.ICFCAT.SHARED is inaccessible to SYSTEMB because its connector record in SYSTEMB's master catalog has an incorrect volume serial number. Executing the following step on SYSTEMB updates the volume serial number and preserves the aliases already defined for the catalog:

```
//CONNECT  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
    IMPORT CONNECT ALIAS -
         OBJECTS((SYS1.ICFCAT.SHARED  -
                  DEVICETYPE(3390)    -
                   VOLUMES(339002)))
/*
```

If you use EXPORT DISCONNECT, and IMPORT CONNECT, all aliases for the user catalog are deleted from SYSTEMB and must be redefined if needed.

### RLS Considerations When Recovering Shared Catalogs

If you use VSAM record-level sharing (RLS), the AMS SHCDS CFREPAIR or CFREPAIRDS command should be used. When a catalog is recovered, but before making the catalog available, either use the CFREPAIR command to reconstruct critical RLS information in the catalog or use the CFREPAIRDS command to correct the RLS information for individual RLS data sets. The catalog must be import connected on all systems to the master catalog before the CFREPAIR or CFREPAIRDS command can be used (see “Recovering Shared Catalogs” on page 99). See *z/OS DFSMS Access Method Services Commands* for more information about the SHCDS command.

## Recovering a Master Catalog

A master catalog cannot be recovered while it is being used as a master catalog. To recover the catalog, you must first make it a user catalog.

This can be accomplished in either of two ways:

1. IPL the system using an alternate master catalog. The old master catalog should be identified as a user catalog in the alternate master catalog. After you recover the old master, you can IPL the system using the recovered catalog.
2. Recover the catalog from a shared system. The catalog is defined as a user catalog in the other system's master catalog, and could be addressed on the sharing system as a user catalog.

You can also recover the master catalog using DFSMSdss if you have a physical dump of the volume containing the master catalog.

## Recovering an Unavailable Catalog

If a catalog becomes unavailable, that is, it cannot be opened and accessed correctly, you should determine the cause of the problem before attempting a recovery.

The following sequence of system messages indicates that the system was unable to open the catalog:

| message id | return code | reason code |
|---|---|---|
| IEC331I | return code 004 | reason code 40 |
| IEC161I | return code 004 | reason code 80 |
| IEC331I | return code 004 | reason code 86 |

These messages are usually accompanied by an IDC3009I message.

Before recovering the catalog, determine if the problem is caused by a broken connection between the catalog and the master catalog. Use LISTCAT ALL to look at the entry for the catalog, comparing the volume serial number and device type listed to the actual ones. If the catalog connector record is inaccurate or missing, use IMPORT CONNECT ALIAS to connect the BCS to the appropriate master catalog.

These messages might also indicate that the VVDS is unavailable. If so, recover the VVDS.

If the catalog is properly connected to the master catalog and the VVDS is available, then recover the catalog. You might need to delete the catalog's record in the VVDS, and its DSCB in the VTOC, before importing a backup copy. The access method services DELETE USERCATALOG RECOVERY command removes the VVDS and VTOC entries for the catalog.

## Recovering a VVDS

Before recovering a VVDS, decide if the VVDS is systematically damaged, or if only certain entries in the VVDS are damaged. If you cannot open the VVDS, for example, when you try to print it or access data sets which have entries in it, then the VVDS is probably systematically damaged, and should be recovered in its entirety.

If you can open the VVDS, run DIAGNOSE to determine which entries are damaged. You can then use the access method services DELETE command followed by data set recovery to recreate the VVDS entries for the affected data sets, and avoid a total VVDS recovery.

If you decide you must recover the VVDS in its entirety, then all data sets represented in the VVDS must be recovered, either individually or by a full-volume restore. Use DFSMSdss or DFSMShsm for volume recovery.

If you are not using SMS, then only VSAM data sets are affected. Otherwise, all data sets on the volume are affected, and must be recovered.

Before recovering a volume, it is necessary to get the volume offline, so that users cannot allocate resources on the volume as you try to restore it. Use the following procedure to get the volume offline if it is not managed by the Storage Management Subsystem:

1. Use the VARY command to get the volume offline.

2. Use the DISPLAY command to determine if the volume has been successfully varied offline, or if resources are still allocated on the volume.
3. Use MODIFY CATALOG to unallocate the VVDS or any catalogs on the volume which are allocated. Use the VUNALLOCATE parameter to unallocate the VVDS. Use the UNALLOCATE command to unallocate the catalog.

If the volume is SMS-managed, set the SMS VOLUME STATUS to DISALL before using the VARY command. Then, check for allocations with the DISPLAY command, and use MODIFY CATALOG if necessary.

After you have gotten control of the volume, use DFSMShsm or DFSMSdss to recover it. For information on using the DFSMShsm RECOVER command to recover a volume with incremental backup copies, see *z/OS DFSMShsm Storage Administration*. For information on volume recovery with DFSMSdss, see *z/OS DFSMSdfp Storage Administration*.

## Recovering Tape Volume or Tape Library Entries

Access method services cannot change the library manager inventory in an automated tape library. ISMF should be used for normal tape library create functions. The access method services CREATE LIBRARYENTRY or CREATE VOLUMEENTRY commands should be used only to recover from tape volume catalog errors.

### Creating a Tape Library Entry:

The CREATE LIBRARYENTRY command creates a tape library entry.

This example creates an entry for a tape library named ATLLIB1.

```
//CREATLIB JOB  ...
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
     CREATE LIBRARYENTRY -
           (NAME(ATLLIB1) -
           LIBRARYID(12345) -
           LIBDEVTYPE(3495) -
           LOGICALTYPE(AUTOMATED) -
           NUMBERSLOTS(15000) -
           NUMBEREMPTYSLOTS(1000) -
           NUMBERSCRATCHVOLUMES(MEDIA1(500) MEDIA2(400)) -
           SCRATCHTHRESHOLD(MEDIA1(200) MEDIA2(100)) -
           DESCRIPTION('TEST LIBRARY ATLLIB1') -
           CONSOLENAME(TESTCON)
```

### Creating a Tape Volume Entry:

The CREATE VOLUMEENTRY command creates tape volume entries.

This example creates a tape library entry for a volume with volser AL0001.

```
//CREATVOL JOB  ...
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
     CREATE VOLUMEENTRY -
           (NAME(VAL0001) -
           LIBRARYNAME(ATLLIB1) -
           STORAGEGROUP(*SCRTCH*) -
           USEATTRIBUTE(SCRATCH) -
           NOWRITEPROTECT -
           LOCATION(LIBRARY) -
           SHELFLOCATION(10098SHELF) -
```

```
            OWNERINFORMATION('JOHN SMITH,RMKD222') -
            ENTEREJECTDATE(1990-03-18) -
            EXPIRATIONDATE(2000-12-31) -
            WRITEDATE(1991-01-02) -
            MOUNTDATE(1991-01-02))
/*
```

## Retaining Alias Information Across Catalog Recovery

When deleting and defining user catalogs, new parameters on DELETE and DEFINE are provided to keep alias information, and subsequently restore the alias information on a re-define of the same catalog name. Use DELETE USERCATLOG NODISCONNECT to retain the catalog's alias information, and DEFINE USERCATLOG RECONNECT to restore the catalog's alias information. The new instance of the catalog can be defined on a different volume and device type then the prior instance. The purpose of retaining and restoring aliases when deleting and re-defining a catalog is to ensure new data set defines will not be oriented to other catalogs during this window.

## Restoring a Full Volume With Catalogs in ECS mode

Full volume restores on volume pools can occasionally result in catalog integrity compromise when:

- Existing catalogs use ECS on the volume.
- Catalogs were previously accessed on the UCB of the target volume of the restore.

In these instances, before performing the full volume restore you must remove the catalogs fom the ECS structure using one of the following methods:

- Remove individual catalogs by name from the ECS structure using the MODIFY CATALOG,ECSHR(REMOVE,*catname*) command.

  Make sure that you do not issue any MODIFY CATALOG,ECSHR(AUTOADD) or MODIFY CATALOG,ECSHR(ENABLEALL) commands until after the restore is completed, because these commands reactivate the catalogs.
- Disconnect all systems from the coupling facility using the MODIFY CATALOG,ECSHR(DISCONNECT) command. Because this method might adversely affect system performance, do this only if you do not know the names of the affected catalogs.

If you do perform a full volume restore before removing the catalogs from the ECS structure, and systems accessing the catalogs are shut down, you must IPL those systems with AUTOADD disabled and then issue the MODIFY CATALOG ECSHR(REMOVE,*catname*) commands to remove the catalogs from ECS. Once this is complete, you can add catalogs to ECS.

# Updating the Catalog After Recovery

After you recover a catalog, use the access method services DIAGNOSE command to help determine how the recovered catalog differs from the current status of data sets, VVDSs, and BCSs on your system. Some activities that alter data sets will change the information in the BCS or VVDS records, and you will need to update your catalog to reflect any changes which have occurred since the catalog was last backed up.

The DIAGNOSE command identifies differences between BCS and VVDS records for VSAM and SMS-managed non-VSAM data sets. To identify inaccurate BCS records for other non-VSAM data sets, use LISTCAT NONVSAM, if possible, before and after the recovery.

The access method services DEFINE and DELETE commands can be used to update the catalog. With these commands, you can: recatalog a data set, thus updating the information in the catalog records; delete a data set; delete a data set's record in the BCS or VVDS; and delete a data set's DSCB in the VTOC.

Table 12 shows the activities that might have occurred since the last catalog backup, and the tasks required to update the catalog.

*Table 12. Activities That Downgrade a Basic Catalog Structure (BCS)*

| Activity Causing Downgrading | Data Set Type | Location of Information | Action Needed To Update the Catalog |
|---|---|---|---|
| **Add** | | | |
| Records | N<br>V | VTOC/tape labels<br>VVDS | None |
| Extents | N<br>V | VTOC/tape labels<br>VTOC/VVDS | None |
| Volumes | N<br>V | BCS/VTOC/tape labels [1]<br>BCS/VTOC/VVDS | Recatalog the data set [2] |
| Candidate Volumes | V | BCS | Add the volume with ALTER ADDVOLUMES |
| Data Sets | N<br>V | BCS/VTOC/tape labels [1]<br>BCS/VTOC/VVDS | Recatalog the data set [2] |
| **Remove** | | | |
| Records | N<br>V | VTOC<br>VVDS | None |
| Extents | N<br>V | VTOC<br>VTOC/VVDS | None<br>Not applicable to VSAM |
| Volumes | N<br>V | BCS/VTOC [1]<br>BCS/VTOC/VVDS | Recatalog the data set [2]<br>Not applicable to VSAM |
| Candidate Volumes | V | BCS | Remove volume with ALTER REMOVEVOLUMES |
| Data Sets | N<br>V | BCS/VTOC/tape labels [1]<br>BCS/VTOC/VVDS | DELETE NOSCRATCH to remove BCS record |
| **Modify** | | | |

*Table 12. Activities That Downgrade a Basic Catalog Structure (BCS) (continued)*

| Activity Causing Downgrading | Data Set Type | Location of Information | Action Needed To Update the Catalog |
|---|---|---|---|
| Records | N<br>V | (data set only)<br>VVDS | None |
| Extents | N<br>V | VTOC<br>VTOC/VVDS | None<br>Not applicable to VSAM |
| Volumes | N<br>V | BCS/VTOC/tape labels [1]<br>BCS/VTOC/VVDS | Recatalog the data set [2]<br>Not applicable to VSAM |
| Data Sets | N<br>V | BCS/VTOC [1]<br>BCS/VTOC/VVDS | Recatalog the data set [2] |

**Notes:**
**N** Non-VSAM data set
**V** VSAM data set

1. SMS-managed non-VSAM data sets also have VVDS entries.
2. To recatalog a data set, first use DELETE NOSCRATCH to remove BCS record for the data set. Then, use DEFINE to recreate the entry. If the data set is VSAM or SMS-managed, specify RECATALOG on the DEFINE command, so that the BCS record is rebuilt using information from the VVDS.

## Recataloging Data Sets and VSAM Objects

After a BCS recovery, some entries in the recovered BCS might not accurately reflect the current characteristics of a data set or VSAM object. The VVDS, VTOC, and tape labels should contain the accurate information for existing data sets. Data sets can only be recataloged into the catalog specified in the VVR/NVR unless they are pagespace, swapspace or SYS1 data sets.

For SMS-managed data sets and all VSAM data sets and associations, BCS entries can be recataloged using the access method services DEFINE command with the RECATALOG option. If there is a BCS entry for the data set, first remove it using DELETE NOSCRATCH.

For non-VSAM, non-SMS-managed data sets, you can delete the BCS entry using the DELETE NOSCRATCH command. Then, you can catalog the data set with DEFINE NONVSAM. If the BCS entry for the data set is missing, you can use DEFINE NONVSAM without first using DELETE. For information on DEFINE NONVSAM, see *z/OS DFSMS Access Method Services Commands*.

To recatalog a data set or VSAM object using the DEFINE RECATALOG command, the corresponding entries in the VVDS and VTOC must exist. This information is used to rebuild the BCS entries, along with information supplied in the command. Data sets can only be recataloged into the catalog specified in the VVR/NVR unless they are pagespace, swapspace or SYS1 data sets.

You might also need to recatalog a data set or VSAM object if you used DELETE with the NOSCRATCH option to delete a BCS entry, or if you restored a volume.

To successfully recatalog a data set or VSAM object, you generally must supply the entry's name, volume, and any ownership, protection, or expiration attributes defined for the entry. Occasionally, you might need to specify the space attribute, and the data set organization. When recataloging alternate indexes, you must also specify the object to which the index is related.

When recataloging VSAM data sets, do not specify component names. They are obtained from the VVDS.

If the data set or VSAM object is protected by a discrete RACF profile, it must be recataloged with that profile. This information is kept in the VTOC or VVDS.

## Recataloging a VVDS

After recovery, a BCS might not contain entries for all the VVDSs on volumes where the BCS has data sets. In this case, you might want to recatalog the VVDS so that the BCS contains entries for all connected VVDSs.

If you want to recreate the BCS entry for a VVDS, use the access method services DEFINE CLUSTER command with the RECATALOG option. Specify the name, volume of the VVDS, and NONINDEXED. The BCS entry is rebuilt using information in the VVDS and the command. A VTOC entry for the VVDS must also exist.

## Deleting BCS Records

When you delete a data set, the BCS, VVDS, and VTOC entries for the data set are removed. If you later recover a BCS, there might be BCS entries for data sets which have been deleted. In this case, the data sets do not exist, and there are no entries for them in the VVDS or VTOC. To clean up the BCS, delete the BCS entries.

For non-VSAM data sets and VSAM clusters, alternate indexes, and page spaces, the DELETE command with the NOSCRATCH option removes the BCS entries. If you define the appropriate RACF FACILITY class, only authorized users are allowed to use NOSCRATCH on SMS-managed data sets.

When you use the NOSCRATCH option, the VVDS and VTOC are not changed. Using this option is the same as uncataloging a data set. After deleting the BCS entry, you can recatalog the data set.

The BCS record for a VVDS might only be deleted if there are no records in the VVDS for data sets cataloged in the BCS. When the BCS record for a VVDS is deleted, the back-pointer to the BCS in the VVDS is also deleted.

You might also need to delete VSAM truename records. VSAM data sets and objects which have more than one component have more than one BCS entry. For example, a key-sequenced data set has entries for the data component and the index component (“truename” records), as well as an entry for the data set itself (“sphere” record).

In order to use a VSAM data set or object, there must be a sphere record for it. If that record is missing, delete the truename records for the data set's components using DELETE TRUENAME. Then recatalog the data set with the DEFINE command. If the sphere record is accessible, then the sphere record and associated truename records can be deleted using the NOSCRATCH option of DELETE.

You can only delete a truename record if the associated sphere record is missing or inaccessible. This problem can be identified using the DIAGNOSE command. You cannot define a data set with the same name as a data set with a missing sphere record: the data set's name cannot be reused until the truename records are deleted.

### Deleting VVDS Records and VTOC DSCBs

Occasionally, a VVDS might have a record for a data set which no longer exists, or for which there is no corresponding BCS entry. The VVDS contains VVRs for all VSAM data sets and objects, and on SMS-managed volumes, NVRs for non-VSAM data sets.

You can use the access method services DELETE command to clean up a VVDS and to remove records for data sets which no longer exist or which have become uncataloged. Specify either VVR or NVR to delete the appropriate type of record.

The DELETE command only deletes the VVR or NVR if there is no corresponding entry in a BCS for the data set. When the VVDS record is deleted, the corresponding DSCB in the VTOC is also scratched. If there is no VVDS record for the data set, the DSCB is still scratched. In order to delete a VVR, specify the component name (the "truename") to be deleted, rather than the cluster name.

STGADMIN.IGG.DLVVRNVR.NOCAT protects the ability to delete a VVR or NVR without an associated catalog. For more information, see the note on catalog requests under **STGADMIN.IGG.DIRCAT** in "Storage Administration (STGADMIN) Profiles in the FACILITY Class" on page 88.

## Recovering Data Sets

After recovering a volume or data set, you might have to recover some individual data sets. The sequence of commands you use should depend on the existence of entries for the data set in the BCS, VVDS, and VTOC. Normally, you only have to restore the latest backup copy of the data set. In some circumstances, you should first delete the data set's catalog or VTOC entries before recovery.

The following table shows the possible relationships between entries in the BCS, VVDS, and VTOC for a single data set. The recovery procedures noted are further explained after the table.

| BCS | VVDS | VTOC | Recovery Action |
|---|---|---|---|
| entry present | entry present | entry present | normal |
| entry present | entry present | entry missing | normal |
| entry present | entry missing | entry present | normal |
| entry present | entry missing | entry missing | normal |
| entry missing | entry present | entry present | clean up, then recover |
| entry missing | entry present | entry missing | clean up, then recover |
| entry missing | entry missing | entry present | clean up, then recover |
| entry missing | entry missing | entry missing | normal |

Following is an explanation of the appropriate recovery procedures:

**normal**
In these cases, you do not need to perform special actions before recovering a

data set. For VSAM data sets and objects, import an exported copy. For non-VSAM data sets, replace the data set with a backup copy.

**clean up, then recover**

In these cases, where there are VVDS or VTOC entries for a data set, but no BCS entry, first delete the VVDS or VTOC entries. Then recover the data set using the normal procedure.

If you do not first clean up the VVDS, importing a VSAM data set or restoring an SMS-managed data set may create duplicate VVDS records. This duplication may cause problems when you try to use the data set.

If there is only a DSCB for a VSAM data set, you cannot import the data set unless you scratch the DSCB. For an explanation of the procedure for deleting VVDS records and VTOC DSCBs, see "Deleting VVDS Records and VTOC DSCBs" on page 107.

# Chapter 7. Analyzing Catalogs for Errors and Synchronization

This chapter describes how you can analyze your catalogs of DASD data sets. Proper catalog functioning requires that the structure of the BCS remain sound. It also requires that data set information contained in the BCS, VVDS, and VTOC be synchronized. That is, the characteristics of a data set (extent information, SMS class names, etc.) must be the same in the data set's entries in the BCS, VVDS, and VTOC.

Structurally damaged or unsynchronized catalogs can cause jobs that use the catalog to fail or produce incorrect results if the wrong data set is processed. If the number of corrupt entries becomes large, you might have to recover the catalog.

It is strongly recommended that you perform the DIAGNOSE and EXAMINE functions regularly for all catalogs in your installation. They can be combined with regular backup functions, for example. Performing these functions regularly ensures that backups taken of catalogs are structurally valid and also helps identify, as early as possible, when a catalog has possibly been damaged.

## Analyzing a BCS for Structural Errors

If a catalog is causing jobs to fail and you cannot trace the failure to unsynchronized entries in the catalog, the catalog might be structurally unsound. Structural problems affect BCS characteristics as a VSAM key-sequenced data set, not as a catalog.

You must have READ authority for the RACF STGADMIN.IDC.*xxx* profile, if it is defined. For more information about STGADMIN RACF profiles, see “Storage Administration (STGADMIN) Profiles in the FACILITY Class” on page 88.

To test the structure of a BCS, perform the following steps:

1. Execute the access method services VERIFY command to verify that the VSAM information for the catalog is current. If the catalog is currently open in the catalog address space and the EXAMINE command is executed, the data that is contained in the catalog may not be current.
2. Execute the access method services ALTER command to lock the catalog and prevent updates to the catalog while you are inspecting its structure.
3. Execute the access method services EXAMINE command. With this command, you can test both the index and data components of a BCS.
4. If the BCS has structural errors, recover the catalog with the most recent structurally sound backup copy.

For more information on using EXAMINE, see *z/OS DFSMS Access Method Services Commands* and *z/OS DFSMS Using Data Sets*.

## Analyzing a Catalog for Synchronization Errors

Catalog entries might become unsynchronized, so that information about the attributes and characteristics of a data set are different in the BCS, VVDS, and VTOC. These differences may make a data set inaccessible or otherwise unusable.

To analyze a catalog for synchronization errors, you can use the access method services DIAGNOSE command. With this command, you can analyze the content of catalog records in the BCS and VVDS, and compare VVDS information with DSCB information in the VTOC.

Besides checking for synchronization errors, DIAGNOSE also checks for invalid data, or invalid relationships between entries.

To use DIAGNOSE ICFCATALOG, you must have READ authority to the RACF STGADMIN.IDC.DIAGNOSE.CATALOG profile if it is defined. DIAGNOSE VVDS requires READ authority to STGADMIN.IDC.DIAGNOSE.VVDS. See "Storage Administration (STGADMIN) Profiles in the FACILITY Class" on page 88 for more information on STGADMIN profiles.

If you are using DFSMShsm, catalog synchronization is checked during the daily data management cycle, and synchronization errors are reported. This eliminates the need to use DIAGNOSE separately, unless you are recovering a catalog. Whenever you recover a catalog, use DIAGNOSE to help identify catalog records that must be updated or that are otherwise in need of individual recovery. Updating catalog records after recovery is discussed in "Updating the Catalog After Recovery" on page 103.

## Using the DIAGNOSE Command

You can use the DIAGNOSE command to accomplish two main tasks:

1. Check the dependent content of catalog records
2. Check the validity of the content of catalog records.

There are no special parameters needed for simply checking the content of a record. As with all DIAGNOSE jobs, specify the ICFCATALOG parameter when checking a BCS, and the VVDS parameter when checking a VVDS.

DIAGNOSE recognizes tape library and tape volume record types. DIAGNOSE checks the cell structure of the volume catalog.

To compare catalog information with VTOC information, you must use DIAGNOSE VVDS, to analyze a VVDS. When you use DIAGNOSE ICFCATALOG, only the BCS and related VVDSs are analyzed.

All records of the input BCS or VVDS are processed unless they are explicitly included or excluded.

### Checking the Dependent Content of a Record

To check the dependent content of catalog records, use the COMPAREDD or COMPAREDS parameters. If you are analyzing a BCS, specify related VVDSs. If you are analyzing a VVDS, specify related BCSs. You can limit dependency checking to selected BCSs or VVDSs, rather than specifying all related BCSs or VVDSs.

For example, the VVDS is checked for an entry that is consistent with the BCS entry. This is not a complete check of the external data set or its entry. It is only a consistency check between the two.

If you are comparing:

- A BCS, the VVDS record is checked for dependency.
- A VVDS, the BCS record, and the VTOC DSCB are checked for dependency.

The input BCS or VVDS determines which entries in the BCSs or VVDSs are used for comparison. Only entries referenced in the input BCS or VVDS are checked.

If dependency checking is not specified, the external pointers to the BCS are not used or checked. The DIAGNOSE command does not identify VVDSs that should be specified for dependency checking.

To determine which VVDSs are connected to a BCS, use LISTCAT LEVEL(SYS1.VVDS), and specify the BCS in the CATALOG parameter. To determine which BCSs are connected to a VVDS, use PRINT COUNT(1) to print the first record of the VVDS, which contains the names of the related BCSs.

## Limiting the Scope of DIAGNOSE

You can specify which records are to be analyzed, or you can exclude certain records from analysis. In this way, you can limit the analysis to records you suspect are in error. If you are analyzing a VVDS, you can also focus on particular catalogs, including or excluding VVDS records belonging to certain catalogs.

This is done using the INCLUDE or EXCLUDE parameters:
- INCLUDE—check only those entries meeting the specification
- EXCLUDE—check the complement of the entries meeting the specification.

Although INCLUDE and EXCLUDE define the scope of the catalog analysis, some entries that are not to be checked may need to be scanned during the processing of the command. Errors in scanned records may result in messages, even though the entries were not to be checked.

Table 13 outlines what happens in processing when INCLUDE and EXCLUDE are specified with DIAGNOSE.

*Table 13. DIAGNOSE Processing When INCLUDE or EXCLUDE are Specified*

| Data Set Type | Normal Processing | INCLUDE Processing | EXCLUDE Processing |
|---|---|---|---|
| VSAM cluster | Cluster and all components and paths | VSAM cluster and components but no paths | VSAM cluster and components but no paths |
| VSAM component | Cluster and all components and paths | Only the component | Only the component |
| VSAM path | Path and related alternate index or base cluster | Only the path entry | Only the path entry |
| non-VSAM | Entry and any aliases | Entry only | Entry only |
| non-VSAM alias | Alias and the base entry | Alias entry only | Alias entry only |
| generation data group base | generation data group base, generation data sets, and aliases | generation data group base and generation data sets | generation data group base and generation data sets |
| generation data set | generation data group base, generation data sets, and aliases | generation data set only | generation data set only |

You can change the GDG setting for reclaim processing as needed without having to IPL the system by specifying the keyword setting for GDS_RECLAIM(NO) in the IGDSMSxx member of SYS1.PARMLIB. If GDG reclaim processing is not desired the default is GDS_RECLAIM(YES). You can also issue the SETSMSGDG_RECLAIM(YES|NO) command to change the value specified in the IGDSMSxx member of SYS1.PARMLIB. This change is in effect only until you re-issue the command or IPL the system.

### Processing Considerations for DIAGNOSE

DIAGNOSE checks the content of BCS and VVDS records in the following order:

1. entry or record format
2. any associations (in the BCS only)
3. miscellaneous length and context
4. BCS and VVDS dependencies (if the COMPARE DD option is specified)

DIAGNOSE issues message IDC21364I if errors are discovered in any of these steps. However, once an error is discovered, DIAGNOSE stops processing that entry, and proceeds to the next entry. Thus, additional errors might be hidden.

Because the DIAGNOSE command checks the content of catalog records, if the records contain corrupted information, there is the possibility that the DIAGNOSE job will abend. For example, corrupted length field values could lead DIAGNOSE to attempt to access invalid storage. If DIAGNOSE abends because of corrupted data in the catalog, try to determine the incorrect record, delete it, and recover the data set from a backup copy.

Another point to keep in mind is that DIAGNOSE can indicate that an entry is in error if another job is processing the entry at the same time as DIAGNOSE. For example, if DFSMShsm is migrating a data set while DIAGNOSE is checking the entry for the data set, you might receive an error for the entry. If you suspect that a DIAGNOSE indicated error is erroneous, use DIAGNOSE again and specify the entry in the INCLUDE parameter.

## Analyzing DIAGNOSE Output

If DIAGNOSE finds errors, it issues messages explaining the errors. The record in error is also dumped, unless you specify NODUMP.

The messages provided by DIAGNOSE can result in the following summaries:

- A list of all entries that had no errors
- A list of entries that had errors
- A list of volume serial numbers that were found to be associated with the BCS that were not encountered during a BCS entry scan
- A list of BCS names found to be associated with a given VVDS that were not encountered during a VVDS entry scan
- A list of comparison members that were not encountered during processing
- A list of included or excluded members that were not encountered during processing.

The output of DIAGNOSE generally consists of three error messages. The first, message IDC21364I, provides the following information:

- The name (and type) of data entry being checked
- The key of the record for a BCS or the RBA of the record for a VVDS
- The offset to the start of the cell causing the error
- A reason that describes the error.

The second message is usually IDC21365I, which provides a display of the records in error. If this record was previously displayed, the record is not displayed again. If the record is a DSCB, the first 44 bytes of the record are not displayed.

The third message produced by DIAGNOSE is IDC21363I, which is a summary of all the errors found by DIAGNOSE.

## Overview of DIAGNOSE Messages

DIAGNOSE issues three general types of messages:

- Invocation error messages, which identify errors in command syntax.
- Execution error messages, which identify errors found while analyzing BCS or VVDS records, or VTOC DSCBs. These errors indicate that the input BCS or VVDS is damaged, and the indicated conditions should be corrected to maintain a usable catalog.
- Summary messages, which identify conditions that are possibly, but not necessarily, errors, and summarize the results of the DIAGNOSE job.

Table 14 summarizes the messages that DIAGNOSE issues, and indicates the message type and associated condition code. Complete information concerning most of these messages can be found in *z/OS MVS System Messages, Vol 1 (ABA-AOM)* through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*.

*Table 14. DIAGNOSE Messages*

| Message Number | Condition Code | Message Type |
|---|---|---|
| IDC01360I | 0 | Summary |
| IDC01371I | 0 | Execution |
| IDC01379I | 0 | Execution |
| IDC11361I | 4 | Summary; may be syntax |
| IDC11362I | 4 | Summary |
| IDC11367I | 4 | Summary |
| IDC11373I | 4 | Summary |
| IDC11374I | 4 | Summary |
| IDC11375I | 4 | Summary |
| IDC21363I | 8 | Summary |
| IDC21364I | 8 | Execution |
| IDC21365I | 8 | Execution |
| IDC21372I | 8 | Execution; may be syntax |
| IDC31366I | 12 | Syntax |
| IDC31368I | 12 | Syntax |

*Table 14. DIAGNOSE Messages (continued)*

| Message Number | Condition Code | Message Type |
|---|---|---|
| IDC31369I | 12 | Execution |
| IDC31370I | 12 | Execution; may be syntax |
| IDC31376I | 12 | Execution |
| IDC31377I | 12 | Execution |

**Note:**

Condition Code:
**0** Not an error condition; informational only
**4** Possible error condition, processing continues
**8** Error condition, processing continues
**12** Severe error, processing terminates

## Example: DIAGNOSE Output

Figure 8 on page 115 shows an example of the output created when comparing a BCS to a VVDS. The records in error are dumped. An error limit of 1 is used, so that only one error is listed.

```
   /*********************************/
   /* DIAGNOSE THE BCS OF DIAGCAT1:   */
   /*  -PASS THE BCS NAME ON CATDD    */
   /*  -COMPARE THE BCS AND THE VVDS  */
   /*   OF SYS1.VVDS.V339001          */
   /*  -DUMP ENTRIES IN ERROR         */
   /*  -AFTER ONE ERROR, STOP THE JOB */
   /*********************************/
   DIAGNOSE ICFCATALOG -
            INFILE(CATDD) -
            COMPAREDD(VVDSDD) -
            ERRORLIMIT(1)

IDC21364I ERROR DETECTED BY DIAGNOSE:

  ICFCAT ENTRY:  SAMPLE.KSDS1.DATA (D)

  RECORD:  SAMPLE.KSDS1 /00

  OFFSET:  X'0073' <------------------------- Offset to start of cell
                                                causing error
  REASON:  12 - CATLG AND VVDS NAMES UNEQUAL

IDC21365I ICFCAT RECORD DISPLAY:

  RECORD:  SAMPLE.KSDS1 /00

000000  00DB0034  C3004600  2DE2C1D4  D7D3C54B  D2E2C4E2  F1404040  40404040  40404040  *....C....SAMPLE.KSDS1           *
000020  40404040  40404040  40404040  40404040  40404040  40000012  01E5E2C1  D4E3C5E2  *                    ....VSAMTES*
000040  E3408033  7F00000F  0019C400  490012E2  C1D4D7D3  C54BD2E2  C4E2F14B  C4C1E1C1  *T ..".....D....SAMPLE.KSDS1.DATA*
000060  00001201  FFFFFFFF  FFFFFFFF  0080337F  00000F00  IE0400F3  F3F9F0F0  F1305020  *..............".........33900.&.*
000080  09820000  00200000  00000000  00000000  00001AC9  004A0013  E2C1D4D7  D3C54BD2  *...................I....SAMPLE.K*
0000A0  E2C4E2F1  4BC9D5C4  C5E70000  1201FFFF  FFFFFFFF  FFFF0080  337F0000  0F001E04  *SDS1.INDEX..............".......*
0000C0  00F3F3F9  F0F0F130  50200982  00000020  00000000  00000000  000000              *.339001.&....................   *

IDC21365I VVDS RECORD DISPLAY:

  RECORD:  X'00002000'

000000  01170040  E9000000  000012E2  C1D4D7D3  C54BD2E2  C4E2F14B  C4C1E3C1  000DE2C1  *... Z......SAMPLE.KSDS1.DATA..SA*
000020  C1C1C1C1  4BD2E2C4  E2F10008  C4C9C1C7  C3C1E3F1  0DE2C1D4  D7D3C54B  D2E2C4E2  *AAAA.KSDS1..DIAGCAT1.SAMPLE.KSDS*
000040  F1000035  21402000  00000600  00000100  00018000  00000000  00280000  00006400  *1....  ..........................*
000060  00FFFFFF  FFFFFFFF  FF000000  00000000  00000000  00000000  62608000  60000000  *............................-.-.-...*
000080  00004000  00001400  00000000  00000002  00000000  64000000  00000000  00000000  *................................*
0000A0  00000000  00000000  00000000  00000000  00000000  01000000  00000000  00000000  *................................*
0000C0  00000000  00000000  00000028  00000000  00000000  00000000  00003E23  80010000  *................................*
000100  00001400  01000000  06000000  06000100  00000000  0027FF                      *......................          *

IDC31369I MAXIMUM ERROR LIMIT REACHED, PROCESSING TRUNCATED

IDC21363I THE FOLLOWING ENTRIES HAD ERRORS:

  SAMPLE.KSDS1.DATA (D) - REASON CODE:   12

IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

IDC0002I IDCAMS PROCESSING COMPLETE.  MAXIMUM CONDITION CODE WAS 8
<?Pub *0000003537>
```

DA6C1010

*Figure 8. Sample DIAGNOSE Output*

Message IDC21364I indicates that the entry record for SAMPLE.KSDS1.DATA, the data component of a VSAM cluster, is in error. This record is dumped in message IDC21365I, which also identifies the cluster name, SAMPLE.KSDS1. The corresponding VVDS record is also dumped.

The error detected is "CATLG AND VVDS NAMES UNEQUAL" and the associated reason code is 12. The offset provided points to the beginning of a volume cell. The cell contains the volume serial number 339001, which indicates the VVDS is SYS1.VVDS.V339001. Thus, the entry for SAMPLE.KSDS1.DATA in the BCS does not agree with the entry in SYS1.VVDS.V339001.

An inspection of the VVDS entry shows that the name SAMPLE.KSDS1 appears as SAAAAA.KSDS1. To recover, correct the VVDS record by deleting it and then importing SAMPLE.KSDS1.

Message IDC31369I indicates that the DIAGNOSE job was ended prematurely, because of the error limit.

Finally, message IDC21363I lists all the errors detected in this job before the error limit was reached. This message summarizes information from all IDC21364I messages issued during the job.

## Recovering from Errors Identified by DIAGNOSE

The recovery procedures for DIAGNOSE errors depend on whether the error exists in a BCS entry or in a VVDS entry.

### Recovering Damaged BCS Entries

Use the following steps to recover a damaged BCS entry:

1. Remove the sphere or base record, if it exists.

   The damage detected might not be in a sphere or base record. If it is not, the entry name of the sphere or base record is indicated in messages IDC21364I and IDC21365I.
2. Remove any remaining association records.

   You can re-execute the DIAGNOSE command after you remove the sphere or base record to identify any unwanted truename or association entries in the BCS. You can remove these entries by using the DELETE command with the TRUENAME parameter.
3. Reintroduce the removed entries into the catalog.

   After the damaged entries have been removed, you can redefine the data sets. For VSAM and SMS-managed non-VSAM data sets, you should specify the RECATALOG option of the DEFINE command.

If you are recovering generation data group entries, use the same procedure. However, you must reintroduce the current generation data sets into the catalog in the proper order after the generation data group has been redefined. You can use the LISTCAT command to determine the current generation data sets.

See “Deleting BCS Records” on page 106 and “Recataloging Data Sets and VSAM Objects” on page 105 for more information about correcting BCS entries.

### Recovering Damaged VVDS Entries

Use the following steps to recover a damaged VVDS entry:

1. Remove the entries in the BCS for the data set, if they exist.

   Before the damaged VVDS records can be removed, you must remove the entries in the BCS. See “Deleting BCS Records” on page 106 for more details on removing BCS entries.
2. Remove the damaged VVDS records.

   After you have removed the BCS entries, you can remove the VVDS records by using the DELETE command and specifying VVR or NVR. DELETE VVR or NVR also removes the Format 1 DSCB from the VTOC.
3. Recover the data set from a backup copy.

If a backup copy of the data set does not exist and the data set can be opened, you can attempt to recover some of the data. Depending on the extent and type of damage in the VVDS record, you might be unable to recover any data. The data that you do recover might be damaged or out of sequence.

See “Deleting VVDS Records and VTOC DSCBs” on page 107 for more information about removing VVDS records.

# Chapter 8. Working with the Catalog Address Space

You can use the MODIFY CATALOG operator command to work with the catalog address space (CAS) and catalogs cached in the in-storage catalog or the catalog data space cache. Using MODIFY CATALOG, you can lessen the impact of catalog recovery, diagnose problems with CAS, and adjust CAS performance.

By using MODIFY CATALOG, you can remove damaged catalog control blocks from CAS without performing a system IPL. You can also remove inaccurate VVDS control blocks after a volume recovery.

You can also use this command to obtain reports on CAS and the catalog data space cache.

## The Catalog Address Space

Catalog functions are performed in the catalog address space (CAS). Most catalog modules and control blocks are located in the catalog address space above 16MB. This reduces the required virtual storage in a user's private area needed to perform catalog functions.

During the initialization of a z/OS system, all user catalog names identified in the master catalog, their aliases, and their associated volume serial numbers are placed in tables in CAS. The system creates the number of CAS service tasks you specify in the CAS service task lower limit parameter. A table called the CRT keeps track of these service tasks.

Changes to the master catalog are automatically reflected in the CAS tables. The information in the master catalog is normally the same as the information in CAS. For shared catalogs, the catalog address spaces on all the sharing systems are updated, maintaining data integrity for your systems.

When a user requests a catalog function, a service task is assigned for that request. This task is assigned a CAS ID, which can later be used to end or abnormally end the request if, for any reason, the request is not satisfied.

CAS also maintains a number of special tasks for its own use: the mother task, allocate task, analysis task, modify task, and the asynchronous events task. The asynchronous events task processes asynchronous events as they are signalled.

The CAS mother task keeps track of all CAS service tasks and other functions of the catalog address space. If the mother task is ended or abnormally ended, then all service tasks are ended and CAS is restarted.

The CAS allocate task performs VVDS and catalog allocation as needed. This marks a resource as being used by the task requesting it. If the allocate task allocates a catalog to CAS, the catalog remains allocated until the next system IPL, or until MODIFY CATALOG,UNALLOCATE is used on the catalog.

The CAS analysis task is dedicated to checking CAS for errors. This error checking is performed periodically.

The CAS modify task is used by the MODIFY CATALOG operator command. Only one MODIFY command can be processed at one time. If the modify task is active and another MODIFY CATALOG command is entered, the second command is rejected.

While they are processing, some of the MODIFY CATALOG options require control of specific system resources. If the resource required by the modify task is not available, the task waits a limited time for the resource. If the request is not completed in the allotted time, the CAS modify task abnormally ends with abend code A1A. A new modify task is then attached, so that the MODIFY CATALOG command is still available.

The MODIFY command can be entered at any console that can submit operator commands.

## Using MODIFY CATALOG with System Maintenance Procedures

Operator Commands Discussed in This Section:
- MODIFY CATALOG,CLOSE
- MODIFY CATALOG,SYS%
- MODIFY CATALOG,UNALLOCATE
- MODIFY CATALOG,VCLOSE
- MODIFY CATALOG,VUNALLOCATE

The MODIFY CATALOG command can simplify certain system maintenance procedures. However, MODIFY CATALOG options do not prevent users from using catalog resources. If a maintenance procedure is adversely affected by not having exclusive control of a catalog or VVDS, either do not use MODIFY CATALOG or use it in conjunction with other commands.

### Recovering a Volume Containing a BCS or VVDS

When you recover a volume, vary the volume offline to sharing systems. This is done with the VARY command. However, VARY might not work because a VVDS or BCS on the volume is allocated to the catalog address space. Use MODIFY CATALOG,ALLOCATED to determine if there are any open BCSs on the volume.

You can unallocate a BCS from CAS with the MODIFY CATALOG,UNALLOCATE command. The VVDS can be unallocated with the MODIFY CATALOG,VUNALLOCATE command. Neither of these MODIFY CATALOG commands locks the BCS or VVDS, so issue the VARY command before unallocating them. The DISPLAY command can be used to help determine if the VVDS or BCS on the volume is allocated.

When you recover a volume, the physical location of the volume's VVDS might change. Because the catalog address space maintains control blocks for each VVDS, this change of location might make VSAM and SMS-managed data sets on the volume inaccessible.

If you unallocate the VVDS, the control blocks are rebuilt after a request for the volume is processed by catalog management. If data sets remain inaccessible after the recovery, close the VVDS with MODIFY CATALOG,VCLOSE. If data sets remain inaccessible, the problem is not with the control blocks for the VVDS.

The same considerations hold true if there is a BCS on the volume: the control blocks for the BCS are rebuilt by the first request that uses the catalog. If you have problems accessing the catalog, unallocate it or close it. Doing either solves problems of inaccurate control blocks.

Do not use MODIFY CATALOG in place of varying a volume offline. Closing or unallocating a VVDS or a BCS does not prevent users from accessing a volume. MODIFY CATALOG is best used to unallocate a BCS or VVDS from the catalog address space, so that the VARY command can be performed.

## Applying PTFs to the Catalog Component

To activate program temporary fixes (PTFs) and other service applications to the modules that make up the catalog component, do either of the following:

- IPL the system. You must perform an IPL if maintenance changes any load modules other than IGG0CLX0, since those load modules are in SYS1.LPALIB. You must also specify CLPA in response to message IEA101A, or use the MLPA function to temporarily bring in the changed modules.
- Refresh the LINKLIST LOOKASIDE by issuing the MODIFY LLA,REFRESH command. Then perform a restart of the Catalog Address Space. A restart can be done by issuing the MODIFY CATALOG,RESTART operator command; see "Restarting the Catalog Address Space" on page 140 for more information. This method can be used if the only load module changed by maintenance is IGG0CLX0.

If you change load modules other than IGG0CLX0 and do not follow the procedures that are outlined above, maintenance will not be properly applied to the running system. You might receive unpredictable results, or problems that maintenance should have fixed might still occur.

## Applying PTFs to Systems Using the Storage Management Subsystem

When you are using the Storage Management Subsystem, you can use the SYS% conversion facility to allow you to apply PTFs (program temporary fixes) or other maintenance to all sharing systems from one system. For a complete description of the SYS% facility, see "Using the SYS% Conversion Facility" on page 14.

If the SYS% facility has not been activated, you can use MODIFY CATALOG,SYS%ON to activate it. When you are finished applying the PTFs, you can deactivate the facility with MODIFY CATALOG,SYS%OFF. The current setting of SYS% (on or off) can be determined using MODIFY CATALOG,REPORT.

Before you can use the SYS% facility to orient jobs to system data sets on shared systems, you must define appropriate alias names for each master catalog.

For example, assume you need to apply a fix to SYS1.LINKLIB on three systems: SYSTEMA, SYSTEMB, and SYSTEMC. The SMP/E job runs on SYSTEMA.

Each master catalog must be connected to the other master catalogs as a user catalog, and assigned a four character alias whose first three letters are SYS. The fourth character can be any valid character. In this case, SYSTEMB has the alias SYSB, and SYSTEMC has the alias SYSC. These aliases are defined in the master catalog for SYSTEMA.

The SMP/E job can then be directed to apply the fix to SYS1.LINKLIB (on SYSTEMA), SYSB.LINKLIB (on SYSTEMB), and SYSC.LINKLIB (on SYSTEMC).

When allocating SYSB.LINKLIB, the system uses the SYSB alias to orient the catalog request to SYSTEMB's master catalog. SYSTEMB's master catalog is then searched for SYSB.LINKLIB. Finding no data set by that name, SYSTEMB's master catalog is again searched for SYS1.LINKLIB, which it finds. The job proceeds using SYSTEMB's SYS1.LINKLIB.

The same procedure is then used to allocate SYSC.LINKLIB.

See "Restrictions on Using SYS% Conversion" on page 16 for restrictions on using the SYS% facility.

## Obtaining Information About Catalogs and CAS Activity

Operator Commands Discussed in This Section:
- MODIFY CATALOG,ALLOCATED
- MODIFY CATALOG,CONTENTION
- MODIFY CATALOG,ENTRY
- MODIFY CATALOG,LIST
- MODIFY CATALOG,LISTJ
- MODIFY CATALOG,ECSHR
- MODIFY CATALOG,REPORT
- MODIFY CATALOG,REPORT,PERFORMANCE
- MODIFY,CATALOG,REPORT,CATSTATS
- MODIFY,CATALOG,REPORT,CATSTATX
- MODIFY CATALOG,TAKEDUMP

You can use the MODIFY CATALOG command to list information about catalogs currently allocated to the catalog address space. Sometimes you need this information so that you can use another MODIFY command to close or otherwise manipulate a catalog in cache.

Table 15 provides an overview of the reporting capabilities of the MODIFY CATALOG command.

*Table 15. Reporting Capabilities of MODIFY CATALOG*

| Option | Message | Purpose |
|---|---|---|
| ALLOCATED | IEC348I | Provides information on all catalogs referred to since the last IPL. |
| CONTENTION | IEC359I | Provides information on the reason classes or Catalog resources for which contention detection is available. It also indicates the current wait-time and any associated actions. |
| DUMP | | Provides dynamic dumping of the catalog address space during diagnostic testing. |
| ENTRY | IEC349I | Provides the storage address and PTF level of a catalog management load module, so that a serviceability level indication processing (SLIP) trap can be set.<br><br>For more information on SLIP traps, refer to *z/OS MVS System Command*sh. |
| LIST | IEC347I | Provides the task identification and address of catalog address space tasks, so that the ID can be used in other MODIFY CATALOG commands. The listing also supplies information about job names, elapsed time of the job, and other selected information. |

*Table 15. Reporting Capabilities of MODIFY CATALOG (continued)*

| Option | Message | Purpose |
|---|---|---|
| LISTJ | IEC3471 | Provides the information about a catalog address space task that is operating on behalf of a user job. |
| ECSHR | IEC3801 | Provides information about the current status of the enhanced catalog sharing (ECS) function, and catalogs that might be using the facility. |
| REPORT | IEC359I | Provides general information on the status of the catalog address space. Current settings for CAS parameters and features are listed as well as CAS specialty tasks addresses and any CAS entry points that have been intercepted.<br><br>There are other forms of the REPORT command that provide information on other aspects of the catalog address space. |
| REPORT,CATSTATS | IEC359I | Displays information about I/O activity and catalog settings for all active catalogs. Information displayed here can be used to assess the usage of a particular catalog over time, as well as certain settings that can impact on catalog performance. See "Evaluating Catalog I/O Activity and Settings" on page 129 for further discussion.<br><br>Other forms of the REPORT command provide information on different aspects of the catalog address space. |
| REPORT,CATSTATX | IEC359I | Extended catalog statistics displays information about I/O activitity and catalog settings for a specific catalog or a particular set of catalogs. Additional data not included in REPORT,CATSTATS is included here in a format intended to provide this extended information. See "Evaluating Catalog I/O Activity and Settings" on page 129 for further discussion.<br><br>Other forms of the REPORT command provide information on different aspects of the catalog address space. |
| REPORT, PERFORMANCE | IEC359I | Displays information about the performance of specific events that catalog processing invokes. Each line shows the number of times (*nnn*) that event has occurred since IPL or the last reset of the statistics by MODIFYCATALOG,REPORT,PERFORMANCE(RESET), and the average time for each occurrence (*nnn.nnn*). The unit of measure of the average time (*unit*) is either milliseconds (MSEC), seconds (SEC), or the average shown as hours, minutes, and seconds (*hh:mm:ss.th*).<br><br>Other forms of the REPORT command provide information on different aspects of the catalog address space. |
| TAKEDUMP | IEC359I | Provides an SVCDUMP that contains the correct dump options to obtain all of the data needed to diagnose catalog problems. Use this command instead of the MVS DUMP command because some of the information needed for catalog problems might not be provided. |

**Note:** This table does not reflect the detailed syntax of these commands. See "MODIFY CATALOG Command Syntax" on page 147 for syntax information.

## Monitoring the Catalog Address Space

Using the MODIFY CATALOG,REPORT command, you can obtain general information about the catalog address space. This information can be used to evaluate your current catalog environment setup. If you determine your current setup is inadequate, you can change it with another MODIFY CATALOG command, or by changing the SYSCAT*xx* member of SYS1.NUCLEUS.

The following example shows the output of an unqualified MODIFY CATALOG,REPORT command:

```
IEC359I CATALOG REPORT OUTPUT
*CAS*************************************************************
*  CATALOG COMPONENT LEVEL   = HDZ2210                           *
*  CATALOG ADDRESS SPACE ASN = 0014                              *
*  SERVICE TASK UPPER LIMIT  =  180                              *
*  SERVICE TASK LOWER LIMIT  =   60                              *
*  HIGHEST # SERVICE TASKS   =    7                              *
*  # ATTACHED SERVICE TASKS  =    7                              *
*  MAXIMUM # OPEN CATALOGS   = 1,024                             *
*  ALIAS TABLE AVAILABLE     = YES                               *
*  ALIAS LEVELS SPECIFIED    = 1                                 *
*  SYS% TO SYS1 CONVERSION   = OFF                               *
*  CAS MOTHER TASK           = 009AC680                          *
*  CAS MODIFY TASK           = 009FC7E0                          *
*  CAS ANALYSIS TASK         = 009FC380                          *
*  CAS ALLOCATION TASK       = 009FC5B0                          *
*  CAS ASYNC TASK            = 009FC150                          *
*  CAS SYSPLEX COMMAND TASK  = 00994C58                          *
*  CAS SYSPLEX QUIESCE TASK  = 00994E88                          *
*  VOLCAT HI-LEVEL QUALIFIER = SYS1                              *
*  NOTIFY EXTENT             =   80%                             *
*  DEFAULT VVDS SPACE        = ( 10, 10) TRKS                    *
*  ENABLED FEATURES          = DSNCHECK SYMREC UPDTFAIL          *
*  DISABLED FEATURES         = DELFORCEWNG VVRCHECK              *
*  DISABLED FEATURES         = AUTOTUNING BCSCHECK               *
*  DISABLED FEATURES         = DELRECOVWNG EXTENDEDALIAS         *
*  DISABLED FEATURES         = ECS AUTOADD DUMPON                *
*  INTERCEPTS                = (NONE)                            *
*CAS*************************************************************
```

The first two entries supply information on the component level of catalog management, and the ASID for the catalog address space.

The next four entries supply information about the service tasks available to process catalog requests. In this example, you can see that the **service task lower limit** is adequate for the **# attached service tasks**. The **service task upper limit** defaults to 180; it is set either at system initialization by the SYSCAT*xx* member of SYS1.NUCLEUS, the IGGCAT*xx* parmlib member or by the the MODIFY CATALOG,TASKMAX command. The **highest number of service tasks** is equal to the highest value the current number of tasks field has reached.

Catalog management creates tasks as necessary, eventually causing the **# attached service tasks** to exceed the **service task lower limit**. As catalog requests subside, the number of tasks attached and available for processing requests is reduced until the lower limit is reached.

**Maximum number of open catalogs** is the value set by the MODIFY CATALOG,CATMAX command. This is the maximum number of catalogs that might be open to CAS simultaneously. If the maximum is reached, catalog management closes the least recently used catalog before opening another catalog. Normally, a catalog remains open once it has been opened.

**Alias table available** entry indicates whether there is a problem with the catalog alias table. This field should always say YES. If it says NO, try restarting the catalog address space. Performance is affected if catalog management does not have catalog aliases in the catalog alias table.

**Hint:** The alias table might not have been created if there was not enough virtual storage available during system initialization. Other messages are issued to indicate this problem. If restarting CAS fails to create the alias table, determine why there is not enough virtual storage, and make the necessary changes to your system.

**Alias levels specified** is the search level for the multilevel alias facility. It is set either at system initialization by the SYSCAT*xx* member of SYS1.NUCLEUS, or by the MODIFY CATALOG,ALIASLEVEL command.

**SYS% to SYS1 conversion** entry indicates whether the SYS% facility is on or off.

The next four entries indicate the hexadecimal addresses of the indicated tasks.

**VOLCAT hi-level qualifier** entry indicates the high level qualifier used for the tape volume catalogs of this system.

**ENABLED FEATURES** entry lists the features that have been enabled with the IGGCAT*xx* parmlib member or the MODIFY CATALOG,ENABLE command.

**DISABLED FEATURES** entry lists the features that have been disabled with the IGGCAT*xx* parmlib member or the MODIFY CATALOG,DISABLE command.

**INTERCEPTS** entry lists up to three entry points that have been intercepted by vendor products (that is, the vendor code gets control at one of those entry points first, before IBM catalog code does). If no entry points have been intercepted, **(NONE)** appears.

## Monitoring the Catalog Address Space Performance

The MODIFY CATALOG,REPORT,PERFORMANCE command can be used to examine certain events that occur in the catalog address space. These events represent points at which catalog code calls some function outside of the catalog component, such as enqueues, I/O, or allocation. All such events are tracked, except for lock manager requests and GETMAIN/FREEMAIN activity. An example of the output from this command is:

```
IEC359I CATALOG PERFORMANCE REPORT
*CAS************************************************
*  Statistics since 12:14:49.61 on  09/29/2004       *
*  -----CATALOG EVENT----   --COUNT--  ---AVERAGE---  *
*  Entries to Catalog            313     23.418  SEC  *
*  BCS ENQ Shr                   291      0.084 MSEC  *
*  BCS ENQ Excl                   20      0.082 MSEC  *
*  BCS DEQ                       352      0.082 MSEC  *
*  VVDS RESERVE CI                14      0.077 MSEC  *
*  VVDS DEQ CI                    14      0.176 MSEC  *
*  VVDS RESERVE Shr               61      0.153 MSEC  *
*  VVDS RESERVE Excl               2      0.099 MSEC  *
*  VVDS DEQ                       63      0.147 MSEC  *
*  SPHERE ENQ Excl                 2      0.054 MSEC  *
*  SPHERE DEQ                      2      0.033 MSEC  *
*  RPL ENQ                         3      0.210 MSEC  *
*  RPL DEQ                         3      0.244 MSEC  *
*  BCS Get                     2,181      1.102 MSEC  *
*  BCS Put                         2     12.836 MSEC  *
*  BCS Erase                       3     34.630 MSEC  *
*  VVDS I/O                       81      9.310 MSEC  *
*  VLF Define Major                1      3.879 MSEC  *
*  VLF Identify                4,596      0.000 MSEC  *
*  RMM Exit                        8      0.005 MSEC  *
```

```
*  Tape Exit                      8      0.001 MSEC  *
*  BCS Allocate                   3     13.102 MSEC  *
*  SMF Write                      9      0.197 MSEC  *
*  VVDS Format                    5     23.878 MSEC  *
*  IXLCONN                        2      1.511 MSEC  *
*  MVS Allocate                   2    267.105 MSEC  *
*  Lookup/Pin UCB                 8      0.329 MSEC  *
*  Unpin UCB                      8      0.340 MSEC  *
*  Capture UCB                    3      0.062 MSEC  *
*  Uncapture UCB                  6      0.061 MSEC  *
*  RACROUTE Auth                 47      5.511 MSEC  *
*  RACROUTE Define                3    537.213 MSEC  *
*  DADSM Allocate SMS             5     15.163  SEC  *
*CAS************************************************
```

The following example shows the output of an unqualified MODIFY CATALOG,REPORT command:

```
IEC359I CATALOG REPORT OUTPUT
*CAS************************************************************
*  CATALOG COMPONENT LEVEL   = HDZ2210                          *
*  CATALOG ADDRESS SPACE ASN = 0014                             *
*  SERVICE TASK UPPER LIMIT  =  180                             *
*  SERVICE TASK LOWER LIMIT  =   60                             *
*  HIGHEST # SERVICE TASKS   =    7                             *
*  # ATTACHED SERVICE TASKS  =    7                             *
*  MAXIMUM # OPEN CATALOGS   = 1,024                            *
*  ALIAS TABLE AVAILABLE     = YES                              *
*  ALIAS LEVELS SPECIFIED    = 1                                *
*  SYS% TO SYS1 CONVERSION   = OFF                              *
*  CAS MOTHER TASK           = 009AC680                         *
*  CAS MODIFY TASK           = 009FC7E0                         *
*  CAS ANALYSIS TASK         = 009FC380                         *
*  CAS ALLOCATION TASK       = 009FC5B0                         *
*  CAS ASYNC TASK            = 009FC150                         *
*  CAS SYSPLEX COMMAND TASK  = 00994C58                         *
*  CAS SYSPLEX QUIESCE TASK  = 00994E88                         *
*  VOLCAT HI-LEVEL QUALIFIER = SYS1                             *
*  NOTIFY EXTENT             =   80%                            *
*  DEFAULT VVDS SPACE        = ( 10, 10) TRKS                   *
*  ENABLED FEATURES          = DSNCHECK SYMREC UPDTFAIL         *
*  DISABLED FEATURES         = DELFORCEWNG VVRCHECK             *
*  DISABLED FEATURES         = AUTOTUNING BCSCHECK              *
*  DISABLED FEATURES         = DELRECOVWNG EXTENDEDALIAS        *
*  DISABLED FEATURES         = ECS AUTOADD DUMPON               *
*  INTERCEPTS                = (NONE)                           *
*CAS************************************************************
```

This command can be useful in identifying performance problems that you suspect are related to catalog processing. For example, if the average time for ENQS that is shown in the report seems excessive, it might indicate some problems in the GRS configuration, or parameter specifications. High I/O times might indicate problems:

- With channel or device load
- With volumes that are suffering a high number of I/O errors
- With volumes that have excessively high RESERVE rates or long RESERVE durations.

In the previous example, the last entry identifies an average of over 15 seconds per DADSM allocate request. This might indicate some problem with reserved volumes, or I/O errors.

To reset the information in this report, you can issue the MODIFY CATALOG,REPORT,PERFORMANCE(RESET) command.

## Monitoring Catalog Contention

The MODIFY CATALOG,CONTENTION command can be used display the reason classes or Catalog resources for which contention detection is available. It will also indicate the current wait-time and any associated actions.

```
IEC359I CATALOG CONTENTION REPORT
*CAS*****************************************************
* RESOURCE        THRESHOLD(MIN)    ACTION(S)           *
*CAS*****************************************************
* SYSZTIOT           10             N                   *
* SYSZVVDS            7             NR                  *
* SYSIGGV2        9,999             N                   *
* ALLOCLCK        INACTIVE          N                   *
*CAS*****************************************************
* ACTION KEY                                            *
* N = NOTIFY OPERATOR  R = REDRIVE REQUEST              *
*CAS*****************************************************
```

Threshold wait times are in minutes and valid values are zero (inactive), and 5 minutes (minimum) to 9,999 minutes (maximum). Actions are displayed in the action column. N, for notify, is the default action and is always on and cannot be disabled. R, for re-drive, can be set by the MODIFY CATALOG,CONTENTION(*reasonclass*,*wait_time*,*action-flags*) command.

## Evaluating Catalog Data Space Cache Performance

In order to evaluate the catalog data space cache, use the MODIFY CATALOG,REPORT,CACHE command. You can use the command to evaluate the cache performance for a specified catalog, or for all catalogs that are currently being cached. The numbers shown in this report are in decimal. If a 'K' appears after the number, it means that the number has been divided by 1024. Following is an example of the output for a catalog cache statistic report:

```
IEC359I CATALOG CACHE REPORT
*CAS*****************************************************************
*   HIT% -RECORDS- -SEARCHES --FOUND-- -DELETES- -SHR UPD- --PURGE-- *
*                                                                     *
* SYS1.MVSRES.MASTCAT                          (ISC)                  *
*    42%    1,578       679       291        13         0         3  *
* SYS1.PROD.UCAT                               (VLF)                  *
*    37%    2,304     1,014       372        27        42         0  *
*                                                                     *
*CAS*****************************************************************
```

The report on SYS1.MVSRES.MASTCAT, currently using ISC caching, shows that 42% of the records requested from this catalog were already in storage. Therefore, no I/O operation was necessary. SYS1.PROD.UCAT is currently being cached through the VLF function, and shows similar statistics. Twenty percent is the break-even point for catalog caching. If the hit percentage is lower than 20, then the processor storage and cycles needed to maintain the data space are not worth the I/O operations saved. This figure is a rough estimate. If central processing unit (CPU) utilization and processor storage are not limiting factors, a lower hit ratio is tolerated.

When you evaluate the cache performance for a catalog, you need to consider how long the catalog has been using the cache. If the cache has only been available for a catalog for an hour, the hit ratio will likely be low. However, if the catalog has been using the cache for a week, expect a good hit ratio. The hit ratio is an

indication of cache usage while the cache is available. The values do not accumulate between performing IPLs, stopping and starting VLF, or restarting the catalog address space.

To obtain a good comparative analysis of how different catalogs are using the cache, produce periodic reports during an active session (between IPLs). This shows you which catalogs are getting good hit ratios quickly, and those which are either reaching the break-even point slowly or not at all. If you want to reset the statistics for a catalog, you can use the MODIFY CATALOG,CLOSE command to close that catalog. When it is next referenced, a new cache structure will be built.

**Note:** The MODIFY CATALOG,CLOSE command does **not** clear the PURGE statistic.

To calculate the hit ratio, divide the number of hits by the number of searches of the data space. The Catalog Cache Report example also lists the number of catalog records in the cache and the number of deleted or updated records in the cache. Note that for catalogs using VLF, the count of records in the cache might not be accurate due to VLF trimming. The number that is shown represents the number of records that the catalog has added to VLF, not the number that might actually be in the VLF cache.

Hit ratio results are relative to a particular installation, workload, or reference pattern. It is possible (but not guaranteed) to increase hit ratios by dedicating more storage space to a VLF-managed catalog. For more information on how to dedicate more storage space to a VLF-managed catalog, refer to *z/OS MVS Initialization and Tuning Reference*.

The master catalog will always be searched for data set names that do not have an alias to a user catalog. For example, non-existent data set names and SMS-managed temporary data sets are not cataloged so all searches for them would show up as a "miss" of the master catalog cache. This may cause the hit ratio for the master catalog cache to appear lower than expected.

The PURGE count displays the number of times a particular catalog cache has been purged and is reset at an IPL (Initial Program Load). Catalog purges commonly occur for two reasons:

- ISC-cached catalog purges occur when a change is made from an external system and is recognized by the current system.
- The number of changes within a system exceeds the maximum that can be recorded between two catalog access events. The specific number of changes that can be recorded is an internal value and varies based on several conditions, all of which are outside of the user's control.

Catalog management allows for record level granularity updates to the VLF cache in a cross system sharing environment. Record updates for a catalog using ISC caching cause the entire ISC cache to be purged when a change is made from another system. Nonzero values under the PURGE column for a catalog using ISC caching indicate this is happening for that catalog. Deletes or updates to a shared catalog on one system cause a corresponding deletion of records in the VLF cache on the other systems. Catalog management accomplishes this by using a list of entries in the catalog that describes VVR.

Records might not be found in the cache of the corresponding system because they were never accessed on that system. The cache is a caching system where records

are not added to the cache until they are accessed. The catalog caching report shows a value for sharing (SHR UPD). This sharing number reflects how many records that were found and deleted from the VLF cache because they were updated or deleted on a shared system. The deleted number given in the catalog data space report reflects:

- The *total* count of records updated or deleted on the current system AND
- Those records updated or deleted on the shared system that are found in the current system's cache.

If more updates or deletions occur in a given time than can be processed by the sharing system, the entire VLF cache on the sharing system is invalidated. This is shown in the MODIFY CATALOG,REPORT,CACHE output under the heading PURGE. This number should always be small, otherwise this indicates that more updates are occurring than can be processed by the sharing system. If the purged number grows dramatically over a short period of time, the catalog is probably not a good candidate for caching.

The numbers shown for record counts, records, searches, hits, deletes, and sharing might not have a one-to-one correlation to the catalog functions being performed. This is because catalog management maintains other records, such as truename records, that can be cached and will add to the statistics.

If a catalog is not using the catalog data space cache to your satisfaction, you can temporarily make it ineligible for CDSC by using MODIFY CATALOG,NOVLF. Alternatively, after closing or deallocating the catalog, you can use the ALLOCATE,NOVLF subparameter to allocate the catalog to CAS without the use of VLF.

To make the catalog permanently ineligible for future data space cache use, modify the COFVLF*xx* member of SYS1.PARMLIB, and stop and restart VLF.

## Evaluating Catalog I/O Activity and Settings

In order to evaluate catalog I/O activity, use the MODIFY CATALOG,REPORT,CATSTATS command. You can use the command to evaluate the activity and settings for a specified catalog, or for all catalogs that are currently open and active. The numbers shown in this report are in decimal. If a 'K' appears after the number, it means that the number has been divided by 1024. Following is an example of the output for a Catalog I/O Statistics report:

```
IEC359I CATALOG I/O STATS REPORT
*CAS*****************************************************************
*     ADDS  UPDATES     GETS   GETUPD  DELETES  BUFNI  BUFND STRNO *
*                                                                   *
* SYS1.PROD.UCAT1                                                   *
*     356      321    1,472      947        5       4      4     2 *
* SYS1.MVSRES.MASTCAT                                               *
*      36       33    2,394       10        0       4      4     2 *
*CAS*****************************************************************
```

The cumulative statistics reported here (Adds, Updates, Gets, Getupd, and Deletes) are from the time the catalog was opened following IPL, or following the issuance of MODIFY CATALOG,RESET command. The following describes the columns for this report:

**ADDS**
The number of catalog entries added to this catalog over the reported interval.

**UPDATES**
The number of catalog entries modified in this catalog over the reported interval.

**GETS**
The number of catalog entries read from this catalog over the reported interval.

**GETUPD**
The number of catalog entries gotten for update from this catalog over the reported interval.

**DELETES**
The number of catalog entries deleted from this catalog over the reported interval.

**BUFNI**
The number of I/O buffers VSAM uses for transmitting catalog index entries for this catalog.

**BUFND**
The number of I/O buffers VSAM uses for transmitting catalog entries for this catalog.

**STRNO**
The number of concurrent catalog positioning requests that VSAM should manage for accessing the catalog.

If you are interested in more detailed information for a particular catalog or set of catalogs, you can utilize the MODIFY CATALOG,REPORT,CATSTATX(catname) or MODIFY CATALOG,REPORT,CATSTATX(catprefix*) command. Following is an example of the output for extended catalog statistics:

```
IEC359I EXTENDED CATALOG STATS
*CAS*************************************************************
* CATALOG NAME     = SYS1.UCAT.TEST                              *
* INSERTS (ADDS)   = 12,486                                      *
* UPDATES          = 3,758                                       *
* RETRIEVES        = 21,816                                      *
* RETRIEVES FOR UP = 9,760                                       *
* ERASES (DELETES) = 3,456                                       *
* CA-RECLAIMS      = 16                                          *
* CA-REUSES        = 4                                           *
* BUFNI SETTING    = 4                                           *
* BUFND SETTING    = 4                                           *
* STRNO SETTING    = 2                                           *
* AVG ELAPSED TIME =  32.359 MSEC                                *
* AVG CPU TIME     = 462.718 USEC                                *
*CAS*************************************************************
```

As with the CATSTATS report output, the numbers shown are in decimal. If a 'K' appears after the number, it means that the number has been divided by 1024.

In addition to the items found in the CATSTATS format of the report, two additional fields appear:

**CA-RECLAIMS**
The number of CA reclaims performed by VSAM on this catalog over the reported interval. Note that for RLS Catalogs this information will show up as N/A as the values are not currently available.

**CA-REUSES**
The number of CA reuses performed by VSAM on this catalog over the

reported interval. Note that for RLS Catalogs this information will show up as N/A as the values are not currently available.

**AVG ELAPSED TIME**
The average elapsed time for all requests to this catalog over the reported interval.

**AVG CPU TIME**
The average cpu time for all requests to this catalog over the reported interval.

## Obtaining Task Identifiers Needed by Other MODIFY Commands

Some MODIFY CATALOG commands require that you supply a task address or ID, or a volume serial number or catalog name, in order to perform the desired task. The ALLOCATED and LIST parameters can be used to obtain these task identifiers.

For MODIFY CATALOG commands that require a task ID or address (ABEND, END), use the LIST parameter. For commands that require a catalog name, you can use either LIST or ALLOCATED.

The main difference between LIST and ALLOCATED is that LIST provides information about a specific task using the task's job name, ID, or storage address. ALLOCATED lists information about all catalogs allocated, even those not assigned to the catalog address space. ALLOCATED also tells you the type of cache to which a catalog is assigned.

### Interpreting MODIFY CATALOG,LIST Output

The following is an example of the output for MODIFY CATALOG,LIST:

```
IEC347I LIST CATALOG TASK(S)
*CAS*****************************************************************
* FLAGS -  TASK ADDRESS - JOBNAME  / STEPNAME - ELAPSED TIME - ID  *
* -W---L     005AB2A8     ACCTING  / SORTSTEP     00.08.26     04  *
*****************************************************************
* O-OLDEST, W-WAIT, A-ABEND, E-ENQ, R-RECALL, L-RLS               *
*CAS*****************************************************************
```

The job ACCNTING has been waiting for more than 8 minutes. This is excessive time for a catalog request, and indicates a problem. The "L" indicator shows that this request is currently waiting on a response from the RLS address space for some requested function.

You can also use the MODIFY CATALOG,LISTJ(ACCTING),DETAIL command to obtain a more detailed set of information about this particular catalog request. This information is useful to determine if it is waiting on a particular catalog that is unavailable, or an ENQ resource on which there is contention. The information from the LISTJ DETAIL command is useful when reporting a suspected problem to IBM service. Using either the END or ABEND parameters and the task ID or address, you can stop the request. See “Ending a Catalog Request Task” on page 139 for more information about ending CAS tasks.

The flags have the following meaning:

**O** The task is the oldest active task. This is only indicated if all active CAS tasks are listed.

**W** The task is waiting for the completion of some event, for example, an ENQ or a tape mount.

**A** The task is abnormally ending.

**E** The task is waiting for an ENQ on a catalog resource.

**R** The task is suspended while the requester's address space is recalled to perform a needed function or obtain information. For example, mounting a volume, verifying security, renaming or erasing data, or extending a data set can cause a recall. If the recall lasts for a very long time, examine the system log for messages indicating the delay. This recall has no relationship to any DFSMShsm function.

**L** The task is suspended waiting for a response from the RLS address space for a request for an RLS function. When this flag displays, you should follow the RLS-specific diagnostic procedures for gathering information to report this problem.

The TASK ADDRESS field gives the hexadecimal address of the CAS service task.

The JOBNAME / STEPNAME field gives the name of the job and step which initiated the catalog request.

The ELAPSED TIME field gives the “hours.minutes.seconds” that the task has been active in CAS.

The ID field gives the CAS identifier for the task.

## Interpreting MODIFY CATALOG,LISTJ(jobname),DETAIL Output

The following is an example of the output for MODIFY CATALOG,LISTJ(jobname),DETAIL command:

```
IEC347I LIST CATALOG TASK(S)
*CAS*****************************************************************
* JOB/STEP Name: IBMUSER /IEFPROC    ASN: 0034  TCB: 008C57D0        *
* CAS TCB: 00897D08 Task Number: 01 TCB Comp Code: 00000000          *
* CCX: 068E9000  CCA: 7F73A000  CCAPROB: 00000000                    *
* CTGPL: 0622A103 7F48FD2A 7F48FCCE 7F48FDE0                         *
*        0400FF04 7F48FCFC 00000000                                  *
* Request Type: GFL                                                 *
*   CTGENT:                                                         *
*   CTGCAT:                                                         *
* CCASRCH: SYS1                                                     *
* Oriented to: SYS1.MVSRES.MASTCAT                                  *
* Waiting for completion of: BCS Read                               *
*        at 00B8734C for 00.00.01                                   *
*CAS*****************************************************************
```

This command is primarily designed to provide detailed information about a particular catalog request. Use this command to obtain more information about a request that showed excessive processing time after the MODIFY CATALOG,LIST command was issued. Much of this information is designed for IBM Service personnel, but there are some fields that can help you do real-time problem diagnosis:

- CCASRCH - the last name of an entry that is attempting to be accessed (or was accessed) in the catalog the request is oriented to.
- Oriented to - shows the catalog that is being accessed for the indicated request
- Waiting for completion of - indicates a specific event outside of the catalog code that the request is waiting for. This includes the address where catalog code will continue executing when the request is complete, and the length of time the request has been waiting

This information might allow you to perform other real-time diagnosis depending on what the request is waiting on, and what catalog or data set is indicated. For example, if it shows it is waiting on the completion of a BCS ENQ, you might try issuing a D GRS,C command to see if there is contention on that catalog name, and what job in the system might be causing that contention.

There is other information that might or might not be displayed in response to this command, depending on the type and state of a request. The above example does not show all of this information.

## Interpreting MODIFY CATALOG,ALLOCATED Output

The following is an example of the output for MODIFY CATALOG,ALLOCATED:

```
*CAS*****************************************************************
*
* FLAGS -VOLSER-USER-CATALOG NAME                                    %
*
* Y-I-R- ZSUSR1 0001 CATALOG.SUE.TEST                                1
*
* Y-I-R- ZSSYS1 0001 CATALOG.ZSSMPE                                  1
*
* Y-I-R- CATA12 0001 CATALOG.IMSUCAT                                 1
*
* YSI-R- VTFM01 0001 CATALOG.VTFM                                    1
*
* Y-I-R- CATA12 0001 CATALOG.A12UCAT                                 1
*
* Y-I-R- ZSUSR1 0001 CATALOG.ITSCUSR                                 5
*
* Y-I-R- STIODF 0001 CATALOG.IODF                                    1
*
* Y-I-R- Z19CAT 0001 CATALOG.Z19MCAT                                 1
*
*********************************************************************
```

In this example, all catalogs except the master catalog reside on SMS-managed volumes. All but two use the catalog data space (VLF) cache.

Explanations for the output are as follows:

**FLAGS**

The flags have the following meanings:

**Y/N** The catalog is (Y) or is not (N) allocated to the catalog address space.

**S** The catalog is managed by the Storage Management Subsystem.

**V** The catalog is using the catalog data space cache (VLF).

**I** The catalog is using the in-storage catalog.

**C** The catalog is closed.

**D** The catalog has been deleted.

**R** The catalog is using cross systems sharing.

**A** The catalog is a tape volume catalog.

**E** The catalog is shared and is using the Enhanced Catalog Sharing facility.

**K** The catalog has been locked by an IDCAMS ALTER LOCK or IMPORT LOCK command.

**VOLSER**

The VOLSER field shows the volume serial number of the volume containing the catalog.

**USER** The USER field contains a count of the number of address spaces that have allocated this catalog. This Catalog Address Space number is normally one.

**%** The % column shows the percentage of allocated extents for each catalog in the list. This percentage indicates the extent usage by either the data or index component, whichever is higher, for that particular catalog. For catalogs that are marked as closed or deleted, this value is N/A.

## Interpreting MODIFY CATALOG,REPORT,DUMP Output

The following is an example of the output for MODIFY CATALOG,REPORT,DUMP:

```
*CAS*************************************************************
*    STATUS  RETURN CODE  REASON CODE  MODULE ID  COUNT          *
*     aaa       bbb          ccc          dd       eee          *
*CAS*************************************************************
```

The variables have the following values:

**aaa** The dump status, which is either ON or OFF.

**bbb** The catalog return code in decimal. The range is 0 to 255, or '***'.

**ccc** The catalog reason code in decimal. The range is 0 to 255, or '***'.

**dd** The catalog module identifier. or '**'.

**eee** The match count in decimal. The range is 0 to 999.

This number decrements each time an error that matches the return code, reason code, and module identifier is detected within the catalog address space. When the count becomes zero, a dump will be taken for that occurrence of the error.

The MODIFY CATALOG,DUMPON command allows the specification of asterisks for any two of the fields return code, reason code, or module identifier. The asterisks indicate that field should not participate in a match for a detected error. For example, **DUMPON(132,*,FO)** will match on any return code 132 issued from IGG0CLFO.

## Interpreting MODIFY CATALOG,ECSHR(STATUS) Output

The following is an example of the output for MODIFY CATALOG,ECSHR(STATUS):

```
*CAS*************************************************************
*  CF Connection: AutoAdd                                       *
* ------------------CATALOG------------------ -----STATUS------  *
* SYS1.VOLCAT.VGENERAL                       Inact(NotShrable)  *
* SYS1.MVSRES.MASTCAT                        Active             *
* UCAT1.USERCAT                              Active             *
* UCAT2.USERCAT                              Inact(Disconnect)  *
* UCAT3.USERCAT                              Inact(NonECSAcc)   *
*CAS*************************************************************
```

The above example shows that the catalog connection to the coupling facility structure that is used for Enhanced Catalog Sharing is active. If the connection did not exist, return and reason codes that are associated with the failure to connect would be displayed.

The information listed for the status of the ECS facility includes error information if the connection has failed. The possible values for the connection status are:

**Autoadd**
The system is connected to the ECS structure, and the automatic add function is enabled.

**Connect Failure**
While attempting to connect to the ECS structure, an error was returned from IXLCONN. The return and reason codes from IXLCONN are also displayed.

**Connected**
The system is connected to the ECS structure in the coupling facility; AUTOADD is disabled.
**Inact(CFFail)**
The system has disconnected because a coupling facility failure was detected.
**Inact(Disconnect)**
ECS is disconnected from the ECS structure in the coupling facility due to a MODIFY CATALOG command with the ECSHR(DISCONNECT) parameter.
**Inact(Restart)**
The system disconnected while processing a CAS restart.
**Quiescing**
ECS is currently active, but is in the process of disconnecting.
**Rebuild Connected**
ECS has connected to the new structure during rebuild processing.
**Rebuild Cleanup**
ECS has received and processed the rebuild cleanup event.
**Rebuild Quiesced**
ECS activity has been quiesced due to a rebuild request.
**Rebuild Stopped**
A rebuild of the ECS structure that was in progress has been stopped.
**Unavailable**
The catalog is not currently allocated and open in the catalog address space (CAS).
**Unknown**
The system status of ECS facility is unknown.

If the connection has been terminated because of a service error, a return code, a reason code, and the module ID will be shown under the system connection status line. The return code and reason code of a coupling facility service call will also be displayed if available.

After the system status is displayed, the ECS status of all catalogs that have been referenced since the last system IPL is displayed. The values of status that might appear are:
**Active** ECS is active for the catalog.
**Inact(CFFail)**
The catalog is inactive because the coupling facility failed.
**Inact(CFFull)**
An attempt was made to activate the catalog, but the ECS structure in the coupling facility is full.
**Inact(Disconnect)**
The catalog is inactive because the system disconnected from the ECS structure.
**Inact(MaxCats)**
An attempt was made to activate the catalog, but the maximum number of catalogs that are allowed in the ECS structure has been reached.
**Inact(NeverConn)**
The catalog is inactive because the system is not connected to the ECS structure in the coupling facility.
**Inact(NonECSAcc)**
The catalog is inactive because the last system to access the catalog was non-ECS 1.5. system.

**Inact(NotElig)**
The catalog is inactive because it is not ECS-eligible; it does not have the ECSHARING attribute.

**Inact(NotShrable)**
The catalog is inactive because it either is not on a shared volume or does not have SHAREOPTIONS(3,4).

**Inact(Removed)**
The catalog is inactive due to a MODIFY CATALOG command with the ECSHR(REMOVE,...) parameter.

**Inact(Unknown)**
The ECS status of this catalog has never been set or is otherwise unknown.

The status of a catalog is the last status known to the ECS facility. If the status changes, it will not be reflected in this display until the next catalog request that uses that catalog.

## Detecting Catalog Resource Contention

Unless resource contention monitoring has been disabled for a resource, the system monitors the following resources for contention:

- ALLOCLCK, which is an internal CAS lock that serializes access to CAS allocation task responsible for most catalog allocation events.
- SYSIGGV2, which is used to serialize access to associated Catalog records.
- SYSZTIOT, which is used to control access to task input/output table resources.
- SYSZVVDS, which is used to serialize access to associated VVDS records. The SYSZVVDS reserve, along with the SYSIGGV2 reserve, provide a mechanism to facilitate cross system sharing of catalogs.

The system checks the catalog address space (CAS) for tasks waiting for the resource beyond the specified wait time (by default, 10 minutes). Once a task is identified as waiting beyond the specified wait time, the system writes a SYMREC record to the logrec data set and issues message IEC393I displaying information about the waiting task or tasks. If the task is still waiting after 5 more minutes, this message is repeated. This message will be repeated every 15 minutes thereafter, if nothing changes. If a new hung task is identified or an old hang is resolved, the notification is reset to the new state at the time of the next system check (within 30 seconds).

In addition to notification, the system can also re-drive. When re-drive is active, the first time a service task with an active resource passes the contention threshold, the service task is abended and the request is resubmitted to catalog for processing.

The abend is a 91A-13, which will produce a 246-rsn dump, unless suppressed, and a LOGREC with error code 246 is generated out of module IGG0CLA9 for the service task. When the service task is abended, all resources it currently holds are freed, and the contention caused due to this request is removed.

The original request submitted to catalog is then again assigned to a catalog service task for processing (which could be, but need not be, the same task). The service task will process the request as if it were newly submitted, which leads to three possible results:

1. The task could again run to contention, which would lead to an new notification and LOGREC after the wait time threshold is past. The only difference in processing is that the re-drive action will not occur on the re-driven request. The second request behaves as if the only action were notification.

2. The task while re-processing the request could fail before contention occurs. Not all tasks can be re-driven with out error, and additionally may require additional cleanup. For example, the task was in the middle of defining a KSDS and was able to create the DATA portion of the KSDS on the volume prior to being re-driven. The presence of the DATA portion on the volume will cause an error on the re-driven request, and will need to be cleaned up in order to proceed. The re-drive action abends the service task without back-out processing.
3. The task could run to completion without contention occurring again. This possibility exists if the original cause of the contention was a deadly embrace of multiple resources among catalog service tasks. For example, another service task got resource A and then went after resource B, while a second service task got resource B and then went after resource A. This situation would cause sustainable contention and could be resolved by either service task being re-driven.

## Example of Detecting Catalog Resource Contention

In this example, the SYSZTIOT resource is used with the default wait time (10 minutes) and action (notification only). The information in this example also applies to the other resources to which resource contention monitoring applies (ALLOCLCK, SYSIGGV2, and SYSZVVDS).

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

```
F CATALOG,TAKEDUMP
```

To get more information about the waiting jobs, use the information in message IEC393I as follows:

- To get job name information, issue one of the following commands:
  - To gather additional information about the waiting task, issue the following command:

    ```
    F CATALOG,LISTJ(jobname)
    ```
  - To gather information about all currently executing service tasks, issue the following command:

    ```
    F CATALOG,LIST
    ```
- After getting task information, you can redrive the holder of the SYSZTIOT so that the waiting task may complete:

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

  If this command doesn't resolve the contention, you can use the following command to terminate the waiting task:

  ```
  F CATALOG,ABEND(taskid)
  ```
- To list all the tasks currently holding SYSZTIOT, issue the following command:

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

  This is useful if the task with an exclusive hold on SYSZTIOT is not a CAS task.

Unless specified otherwise in the IGGCATxx parmlib member, the system, by default, lets a task wait for the SYSZTIOT task for 10 minutes before writing a

SYMREC record to the logrec data set and issuing message IEC393I. You can specify a different wait time using the following command:

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

See "MODIFY CATALOG Command Syntax" on page 147 for *wait_time* limits.

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

Unless specified otherwise in the IGGCATxx parmlib member, the only action the system takes by default is notify (writing a SYMREC record to the logrec data set and issuing message IEC393I). You can specify that the system also re-drive the task using the following command:

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

See "MODIFY CATALOG Command Syntax" on page 147 for *action-flags* , which can be used to specify re-drive.

If you specify a re-drive action for SYSZTIOT on the MODIFY command, you must respecify that value after each system IPL. A non-default *action-flags* value does, however, persist through restarts of the CAS task.

Additional information:

- You can display the current SYSZTIOT wait time and action by using the MODIFY CATALOG,CONTENTION command without parameters (see "MODIFY CATALOG Command Syntax" on page 147).
- For information about using SYMREC records, see Logrec Information in *z/OS DFSMSdfp Diagnosis*.
- For information about system message IEC393I, see *z/OS MVS System Messages, Vol 7 (IEB-IEE)*
- For information about the MODIFY CATALOG command, see "MODIFY CATALOG Command Syntax" on page 147.

## Fixing Temporary Catalog Problems

Operator Commands Discussed in This Section:

MODIFY CATALOG,ABEND
MODIFY CATALOG,CLOSE
MODIFY CATALOG,END
MODIFY CATALOG,RESTART
MODIFY CATALOG,UNALLOCATE
MODIFY CATALOG,VCLOSE
MODIFY CATALOG,VUNALLOCATE

Occasionally, the control blocks for a catalog kept in the catalog address space might be damaged. You might think the catalog is damaged and in need of recovery, when only the control blocks need to be rebuilt. If the catalog appears damaged, try rebuilding the control blocks first. If the problem persists, recover the catalog.

There might also be situations where a job gets an enqueue lockout from a catalog, or a catalog request is not being satisfied for some reason. In these cases, the job needs to be ended and redriven, if possible.

It is also possible that a catalog request that is currently being processed, cannot be properly retried after being interrupted by these commands. Use these commands when all other means of correcting an ongoing catalog error have failed.

Table 16 gives an overview of the error recovery capabilities of the MODIFY CATALOG command.

*Table 16. Error Recovery Capabilities of MODIFY CATALOGS*

| Option | Purpose |
|---|---|
| ABEND | To end a CAS task abnormally. This should only be used after you have unsuccessfully tried END, or when you are ending the CAS allocate, analysis, or modify tasks. |
| CLOSE | To release all CAS storage for the specified catalog. The catalog is not locked, and is re-opened to CAS by the next catalog request that accesses the catalog. The catalog is not unallocated from CAS, and the common services area storage used by the catalog is not freed. |
| END | To end a CAS task. You can choose to redrive the request, or simply end it. This is the preferred method of ending a CAS task, and should be used before attempting ABEND. |
| RESTART | To abnormally end the CAS mother task and restart it in a new address space. This option should only be used when your only other option is to IPL the system. If RESTART fails, you must IPL. |
| UNALLOCATE | To unallocate and close a catalog allocated to CAS from CAS without releasing the CAS storage. As with CLOSE, the control blocks for the catalog are rebuilt with the next request for the catalog, and the catalog is not locked.<br><br>This command is useful for getting a volume offline. |
| VCLOSE | To close the VVDS that resides on the specified volume. The VVDS is opened by the next request that tries to access it. |
| VUNALLOCATE | To unallocate all VVDSs from the catalog address space. This can help in getting a volume offline for recovery. The VVDS is not locked, and the next request that tries to access it opens the VVDS. However, VVDSs are not allocated to the catalog address space until MODIFY CATALOG,NOVUNALLOCATE is issued. |

**Note:** This table does not reflect the detailed syntax of these commands. See "MODIFY CATALOG Command Syntax" on page 147 for syntax information.

## Ending a Catalog Request Task

Using the LIST parameter, you can determine that a catalog request task is taking too much time to execute. If your resource monitor program indicates you have an enqueue lockout on a resource held by CAS, the listing gives you the ID of the task with the enqueue. You might want to use the MODIFY command with the LISTJ and DETAIL keywords to get more information about the associated catalog request.

In these situations, you will want to end the task. Sometimes redriving the task allows the request to be successfully satisfied, especially if there is an enqueue lockout. The timing of the resource requests on the redrive are probably different than when the enqueue occurred, so that the requests are satisfied.

Use the following procedure to end a catalog request task:

1. Use MODIFY CATALOG, LIST to obtain the task ID or address.

2. If you are trying to end the CAS allocate, analysis, or modify task, skip this step. If you are trying to end a user task, use one of the following commands:
   a. Use MODIFY CATALOG,END(*xx*){REDRIVE} if the task is in enqueue lockout, or you want the request to be redriven.
   b. Use MODIFY CATALOG,END(*xx*){NOREDRIVE} if you want to permanently end the task.
   c. Use MODIFY CATALOG,END(*xx*){REDRIVE|NOREDRIVE}[FORCE] if you want to end a task abnormally, even if it is in recall.

      **Restriction:** Do not use FORCE unless the address space or task that the service task is operating on behalf of has ended abnormally. You can use the MODIFY CATALOG,LIST command to find the name of the user job and task the service task is processing.
3. If the preceding step failed, or you skipped it, use MODIFY CATALOG,ABEND(*id*)[,FORCE] to end the task and redrive the request once.

   **Restriction:** Do not use FORCE unless the address space or task that the service task is operating on behalf of has ended abnormally.

## Refreshing a Catalog's Control Blocks

The catalog address space is designed with internal checks that allow it to identify, and subsequently rebuild, damaged catalog control blocks. However, not all problems can be identified.

If you find that attempts to access a particular catalog are resulting in recurrent abnormal endings, rebuild the control blocks in the catalog address space. This should be done using the CLOSE or UNALLOCATE parameters. If the damaged control blocks are for a VVDS, use VCLOSE or VUNALLOCATE.

Each of these parameters causes the control blocks to be released and rebuilt on a subsequent request that tries to access the catalog. Which parameter you use depends on whether you want the catalog unallocated and closed (UNALLOCATE), or closed but left allocated (CLOSE).

When rebuilding the control blocks for a VVDS, use VCLOSE whenever possible. With VCLOSE, you specify the particular VVDS whose control blocks you suspect are in error. VUNALLOCATE can also be used, but this parameter unallocates all VVDSs. You cannot unallocate a specific VVDS.

The UNALLOCATE parameter can be used to unallocate all catalogs at once. It can also be used to unallocate a catalog on a volume that you need to vary offline. Use VUNALLOCATE if you need to unallocate a VVDS to get a volume offline.

## Restarting the Catalog Address Space

The catalog address space is designed to restart with a minimum of interruption to your system. Requests that are being processed are generally able to be restarted from the beginning. Requests that are made while the address space is in the process of restarting are temporarily suspended.

The catalog address space is critical to the functioning of your system, and there is always the possibility that a restart might fail. However, the use of the restart facility might also prevent an IPL or clear other error conditions that are a result of problems associated with the use of catalogs. For example, the following problems are typically corrected by a restart:

- Inability to vary a volume containing a catalog offline, when a MODIFY CATALOG,CLOSE command does not release the volume
- ABENDs in the catalog address space that relates to lack of storage (such as 878 ABENDs)
- ABENDs in the catalog address space that might indicate damage to control blocks (such as repeated 0C4 ABENDs at the same location)
- ENQ lockouts, particularly on the SYSIGGV2 resource, when the MODIFY CATALOG,ABEND command will not remove the task in error
- Installing catalog maintenance to correct a problem when an IPL is not necessary or feasible

Do not use RESTART to refresh catalog or VVDS control blocks or to change catalog characteristics. The use of other MODIFY command formats are designed to accomplish this on a catalog-by-catalog basis. There is a risk that the catalog address space restart might fail for some unanticipated reason. If this occurs, it will be necessary to IPL the system to recover the address space. However, a restart failure is a very unlikely occurrence.

It is also possible that a catalog request that is currently being processed cannot be properly retried after being interrupted by these commands. For example, this could occur with a DEFINE command of a VSAM data set that is partially completed at the time of the restart. It is recommended that you try to quiesce system activity as much as feasible before doing a restart, or at least attempt to minimize the use of catalogs.

When you enter MODIFY CATALOG,RESTART, message IEC363D is displayed as follows:

```
IS THIS RESTART RELATED TO AN EXISTING CATALOG PROBLEM (Y OR N)?
```

- If you answer 'Y', the message IEC364D is displayed as follows:

  ```
  HAS AN SVC DUMP OF THE CATALOG ADDRESS SPACE ALREADY BEEN TAKEN (Y OR N)?
  ```

  – If you answer 'Y', the catalog address space ends abnormally with an 81A ABEND, which causes a restart.
  – If you answer 'N', an SVC dump is automatically taken before the catalog address space ends abnormally with an 81A ABEND, which causes a restart.
- If you answer 'N', the catalog address space ends abnormally with an 81A ABEND, which causes a restart.

All catalog tasks that were in process at the time of the 81A ABEND are restarted from the beginning.

The restart of CAS in a new address space should be transparent to all users. However, even when all requests are redriven successfully and receive a return code 0, the system might produce indicative dumps on the console, the system log, and on user job logs. There is no way to suppress these indicative dumps.

As noted above, a request might not successfully be restarted. If this is the case, the appropriate return and reason code information, associated messages, and possibly system dumps will be produced.

The catalog address space is designed to recover from cross-memory failures that can occur during CAS restart. CAS recognizes and recovers from the following abend codes, that might occur during a restart: 052, 058, 066, 070, 073, and 0D*x*. You can ignore any indicative dumps produced by the system for these abend codes. Only the final catalog return code, which should be 0, is significant.

## Making Temporary Modifications to the Catalog Environment

Operator Commands Discussed in This Section:
MODIFY CATALOG,ALIASLEVEL
MODIFY CATALOG,ALLOCATE
MODIFY CATALOG,CATMAX
MODIFY CATALOG,CLOSE
MODIFY CATALOG,{ISC|NOISC}
MODIFY CATALOG,{SYS%ON|SYS%OFF}
MODIFY CATALOG,TASKMAX
MODIFY CATALOG,UNALLOCATE
MODIFY CATALOG,{VLF|NOVLF}
MODIFY CATALOG,ENABLE(DELFORCEWNG)
MODIFY CATALOG,ENABLE(DSNCHECK)
MODIFY CATALOG,DISABLE(DELFORCEWNG)
MODIFY CATALOG,DISABLE(DSNCHECK)

The MODIFY CATALOG command allows you to alter a number of attributes that are initialized at IPL time. Many of these attributes are initialized by the SYSCAT*xx* member of SYS1.NUCLEUS. The maximum catalog value (CATMAX) is also set at IPL time to 1024.

The parameters discussed in this section allow you to tailor your system according to temporary needs, without requiring you to bring down your system and IPL using a different SYSCAT*xx* member. Other parameters allow you to gain closer control over storage, so that you can free up storage used by catalog management for your own jobs.

Table 17 gives an overview of the temporary system tailoring capabilities of the MODIFY CATALOG command.

*Table 17. Temporary System Tailoring Capabilities of MODIFY CATALOG*

| Option | Purpose |
|---|---|
| ALIASLEVEL | To change the number of qualifiers used to determine the alias of a user catalog. This is the multilevel alias search level. |
| ALLOCATE | To allocate a catalog to the catalog address space. Using the NOISC or NOVLF subparameters, you can prevent the allocated catalog from using the specified cache. |
| CATMAX | To close all open catalogs and set a maximum to the number of catalogs that might be open in CAS. When the maximum is reached, the least recently used catalog is closed. This conserves storage. |
| CLOSE | To close a catalog and remove it from CAS, and to free up the CDSC or ISC storage used by the catalog. |
| DISABLE | To disable specific optional features. See the discussion of the DISABLE keyword for features that can be specified. |
| ENABLE | To enable specific optional features. See the discussion of the ENABLE keywords for features that can be specified. |
| ISC and NOISC | To either place a catalog in the in-storage catalog cache, or remove it from that cache. |
| SYS%ON and SYS%OFF | To either activate or deactivate the SYS% facility, to convert searches for SYS% data sets to searches for SYS1 data sets. |
| TASKMAX | To specify an upper limit to the number of service tasks to process catalog requests. Once the limit is reached, new requests must wait. Setting an upper limit reduces the storage used by CAS. |

*Table 17. Temporary System Tailoring Capabilities of MODIFY CATALOG (continued)*

| Option | Purpose |
|---|---|
| UNALLOCATE | To close and unallocate a catalog from CAS, and to free up the CDSC or ISC storage used by the catalog. |
| VLF and NOVLF | To either assign a catalog to the catalog data space cache, or to prevent placement of additional records from a catalog into that cache. Records are not removed from the CDSC with these parameters. |

**Note:** This table does not reflect the detailed syntax of these commands. See "MODIFY CATALOG Command Syntax" on page 147 for syntax information.

## Starting and Stopping the Catalog Cache for a Catalog

When your system is initialized during IPL, catalogs are assigned as eligible for either the in-storage catalog cache or the catalog data space cache. These two caches have different performance benefits. Although a catalog might perform well in one type of cache under most circumstances, you might find occasion to remove a catalog from cache, or to move it to a different type of cache. If a catalog using the in-storage cache (ISC) receives a lot of update activity, you might want to remove it from cache until the update activity is finished.

You can move any catalog in the catalog data space to the in-storage catalog cache. Once removed from the catalog data space, a catalog automatically begins using the in-storage catalog cache, unless you use the NOISC parameter to prevent it. If you use the NOISC parameter, the catalog is not cached.

The objective of in-storage cache is to cache only those records that are read directly. Records will not be cached for other types of requests, and therefore in-storage cache is not then in effect. The ISC flag will be turned on the first time a direct read is done. This flag will remain on until that record is no longer in cache. At that point it will remain off until the next direct read to the catalog.

Since the COFVLF*xx* member of SYS1.PARMLIB controls which catalogs use the catalog data space cache, under normal conditions you cannot move a catalog from the in-storage catalog cache to the catalog data space. If a catalog is entered in the COFVLF*xx* member, and you have used MODIFY CATALOG,NOVLF to prevent it from using the catalog data space, then issuing the MODIFY CATALOG,VLF command returns the catalog to using the catalog data space.

To change a catalog from ISC to VLF cache management:
1. Update the COFVLFxx parmlib member to add the catalog EMAJ name to the IGGCAS class and recycle VLF
2. Issue F CATALOG,CLOSE(cat)
3. Issue F CATALOG,NOISC(cat)
4. Issue F CATALOG,VLF(cat).

**Note:** Catalogs will not appear in MODIFY CATALOG,REPORT,CACHE as using VLF until the next reference to the catalog that requires caching be done. There are conditions that may remove a catalog from VLF and cause it to be added dynamically as catalog operations continue. These operations are transparent to catalog operation and are not identified externally when they happen

To change a catalog from VLF to ISC cache management:
1. Issue F CATALOG,CLOSE(cat)
2. Issue F CATALOG,NOVLF(cat)

3. Issue F CATALOG,ISC(cat)
4. To make this change permanent, remove the catalog EMAJ name from the IGGCAS class in the COFVLFxx parmlib member, and recycle VLF.

Besides moving a catalog from one cache to the other, you can simply remove the catalog from cache. For example, if you determine that a catalog is not effectively using the catalog data space cache, you can remove it from the cache. The MODIFY CATALOG,REPORT,CACHE command provides information which you can use to evaluate cache performance.

Six parameters of the MODIFY CATALOG command can be used to modify how a catalog uses cache: ISC and NOISC; VLF and NOVLF; and ALLOCATE with NOISC or NOVLF. Which version you use depends on whether the catalog is already allocated to CAS.

If the catalog is not allocated to CAS, then you can allocate it to CAS with the MODIFY CATALOG,ALLOCATE command. The NOISC or NOVLF parameter can be used to prevent the catalog from using the specified cache. Otherwise, the catalog uses the last cache assigned to it (during IPL or a previous MODIFY CATALOG command).

If the catalog is already allocated and using ISC, the NOISC parameter can be used to terminate the ISC. Similarly, NOVLF can terminate the catalog data space.

If you want to assign a catalog in the CDSC to use ISC, close or unallocate it and use the NOVLF option. NOISC must *not* have been previously specified.

Removing catalogs from the ISC cache also reduces the amount of storage used by the catalog address space, probably at the expense of catalog performance.

## Changing the Multilevel Alias Search Level

A catalog alias can have up to four qualifiers. However, the actual number of qualifiers used, or the "multilevel alias search level", is initialized in the SYSCAT*xx* member of SYS1.NUCLEUS or LOAD*xx* member of SYS1.PARMLIB. You can change this level without a system IPL by using the MODIFY CATALOG,ALIASLEVEL command. The change will remain in effect until the next time you IPL the system.

This command can be used to test various multilevel alias search levels. However, you should use this command with care. Changing the search level might result in some data sets becoming inaccessible, since the catalog searched might not be the catalog in which the data set is actually cataloged.

For example, the following aliases might be defined:

| Alias | For Catalog |
|---|---|
| APPLIC1.TESTING | SYS1.ICFCAT.TESTING |
| APPLIC1 | SYS1.ICFCAT.APPLIC |

In this case, if the original search level were 2, the data set APPLIC1.TESTING.DATA would be cataloged in SYS1.ICFCAT.TESTING. However, if you used MODIFY CATALOG,ALIASLEVEL(1) to change the search level to 1, subsequent requests for this data set would orient the catalog search to SYS1.ICFCAT.APPLIC. This results in a "data set not found" error.

See “The Multilevel Alias Facility” on page 20 for more information on the multilevel alias facility.

## Opening, Closing, Allocating, and Unallocating Catalogs

Using MODIFY CATALOG, you can specify that a catalog be closed, or that the catalog be allocated or unallocated from the catalog address space. You cannot, however, permanently close a catalog or prevent a user from accessing a catalog, nor can you explicitly open a catalog, with the MODIFY CATALOG command. To prevent users from accessing a catalog, use the access method services ALTER LOCK command.

Catalogs are allocated to the catalog address space when the first request for the catalog is processed. However, you can use the ALLOCATE parameter with the NOISC or NOVLF parameters to prevent the catalog from using the specified cache.

When you unallocate a catalog from CAS, the catalog is closed and all CAS private storage used by the catalog is freed. The device on which the catalog resides is also unallocated from CAS. However, CAS storage related to the catalog remains allocated. This space can only be freed by deleting the catalog. Common service area space for the catalog is not freed.

You can also close a catalog. When you use CLOSE, the catalog is closed but remains allocated to CAS. All the CAS private storage associated with the catalog is freed.

Closing or unallocating a catalog might be useful when the control blocks for a catalog become corrupted. When the catalog is opened again, new control blocks are built for the catalog.

If a catalog is using the CDSC, all CDSC space used by the catalog is freed when the catalog is closed or unallocated. This space is then available for the use of other catalogs in the CDSC.

If you alter a catalog's attributes (for example, the catalog's share options), you can close the catalog, and when it is reopened, the new values are recognized. Thus, catalog attributes can be changed without cancelling jobs or performing a system IPL.

If your installation has little or no catalog activity once your system is up and running, you might consider closing all the catalogs to free up CAS private storage associated with the closed catalogs.

You can also close a VVDS using the VCLOSE parameter. When you close a VVDS, the CAS private storage used by the VVDS is freed. The next request that uses the VVDS reopens the VVDS, and new control blocks are built for the VVDS.

Use VUNALLOCATE to unallocate all VVDSs. This might be necessary to allow the VARY command to vary a volume offline. VVDSs remain unallocated until you issue MODIFY CATALOG,NOVUNALLOCATE.

## Changing the Maximum Number of Catalogs and Tasks in CAS

When you IPL a system, the maximum number of catalogs that can be open in the catalog address space is set at 9999. The maximum number of CAS service tasks available for user requests is set to either 180 (the default) or 90% of the value

optionally specified in the SYSCAT*xx* member of SYS1.NUCLEUS. You can specify the number of catalogs and tasks as follows:

- Specify the service task lower limit in SYSCATxx (SYS1.NUCLEUS), or LOADxx (parmlib).
- Specify the maximum number of concurrent user service tasks in the TASKMAX parameter of the IGGCATxx parmlib member. The value of TASKMAX in IGGCATxx should be no more than 90% of the maximum number of concurrent Catalog requests specified in SYSCATxx.

You can also change these values temporarily using the CATMAX or TASKMAX parameters on the MODIFY CATALOG command. Changing these values can help you manage or limit the amount of storage used by CAS to perform catalog functions.

- When you use CATMAX to change the maximum number of catalogs that can be open in CAS and the new limit is lower than the previous limit, all open catalogs are closed. This does not unallocate catalogs. Catalogs remain allocated to CAS, but in restart status. All the storage associated with the catalogs that were closed is freed.

  If a request for a closed catalog must be processed after the limit for open catalogs is reached, the least-recently used catalog is closed and the required catalog is opened.

  Limiting the number of open catalogs affects catalog performance. However, if space is a primary consideration, you might need to set a maximum.
- When you use TASKMAX to change the maximum number of CAS service tasks, first determine the current number of CAS tasks. The REPORT parameter shows this value in the "# ATTACHED SERVICE TASKS" field.

  When the upper limit for tasks is reached, any new requests that require catalog resources wait until another task is finished. No user jobs fail because the limit is reached, but if the limit is set too low, it might cause a significant performance degradation for jobs on the system.

  Catalog management reserves 10% of the total number of tasks for its exclusive use, with the remaining tasks being available for both catalog management use and user requests. The reserved tasks are only allocated if all the other tasks are in use, and catalog management requires the use of a CAS service task. Thus, the highest number of service tasks listed for the catalog address space might be as high as 180 (if the default value is used) or 360 (if the largest possible value is specified in SYSCAT*xx*).

See also "The intersection of SYSCATxx, LOADxx, IGGCATxx, and MODIFY CATALOG command" on page 35.

## Enabling and Disabling Operator Prompts for Certain Functions

If you delete a catalog using the IDCAMS DELETE UCAT FORCE command in an SMS environment, all data sets in the catalog are deleted. If you issue DELETE VVDS RECOVERY, all NVRS and VVRS from the VVDS are deleted. To prevent inadvertent deletion of data sets or VVR entries, IDCAMS issues a WTOR to the master console before proceeding with a delete command. Use MODIFY CATALOG,ENABLE(DELFORCEWNG) to enable prompting; it is the default. If you want to suppress the prompts, use MODIFY CATALOG,DISABLE(DELFORCEWNG).

If the catalog address space is restarted by a MODIFY CATALOG,RESTART and the installation has suppressed prompting, it will attempt to preserve that setting

after the restart. Restarting the catalog address space might reset the option to prompt. This occurs when certain catalog control blocks have been damaged and must be rebuilt as part of the restart.

The current setting of the option is displayed in response to MODIFY CATALOG,REPORT.

The installation may choose to force compliance with the documented syntax rules for data set names that are to be cataloged, or they may allow names that violate the rules to be created. The default is that checking will be enabled; data set names that violate the syntax rules will result in a failed catalog request. If you have programs that must create names that do not meet the syntax rules you can disable this checking; however IBM utilities may not be able to remove these entries from the catalog.

### Enabling and Disabling Data Set Name Validity Checking

The installation will either force compliance with the documented syntax rules for data set names that are to be cataloged, or it will allow names to be created that violate these rules. Data set names that violate the syntax rules will result in a failed catalog request. The default setting enables validity checking. If you have programs that must create names that do not meet the syntax rules, you can disable validity checking. However, IBM utilities may not be able to remove these entries from the catalog.

To enable or disable data set name syntax checking, use the ENABLE(DSNCHECK) or DISABLE(DSNCHECK) parameters on MODIFY CATALOG. See "MODIFY CATALOG Command Syntax."

## MODIFY CATALOG Command Syntax

This section contains an explanation of the syntax and parameters of the MODIFY CATALOG command. Use this command to communicate with the catalog address space, in order to display information or request services.

When a system console operator issues any MODIFY CATALOG command except for MODIFY CATALOG,RESTART, messages return to that system console exclusively. If the system console operator issues the MODIFY CATALOG,RESTART, command, messages are returned to the master console and the console issuing the MODIFY CATALOG,RESTART. For information on system messages and abend codes, refer to *z/OS MVS System Messages, Vol 1 (ABA-AOM)* through *z/OS MVS System Messages, Vol 10 (IXC-IZP)*, and *z/OS MVS System Codes*.

| | |
|---|---|
| **{MODIFY\|F}** | **CATALOG**,<br>**[ABEND(***task***)[,FORCE]]**<br>**[ALIASLEVEL(***n***)]**<br>**[ALLOCATE(***catname***)[, {NOISC\|NOVLF}]]**<br>**[ALLOCATED[(***volser***)]]**<br>**[CATMAX(***nnnn***)]**<br>**[CLOSE(***catname***)]**<br>**[CONTENTION]**<br>**[CONTENTION[(***resource*[,*wait_time*][,*action-flags*])]]**<br>**[DISABLE(***feature***)]**<br>**[DUMPON[(***rc*,*rsn*,*mm*[,*cnt*])]]**<br>**[DUMPOFF]**<br>**[ECSHR(***value***)]** |

| | |
|---|---|
| | **[ENABLE(***feature***)]**<br>**[END(***id***)[,{REDRIVE \| NOREDRIVE}][,FORCE]]**<br>**[ENTRY[(***csectname***)]]**<br>**[{ISC \| NOISC}(***catname***)]**<br>**[LIST[(***task***)]]**<br>**[LISTJ(***jobname***)[,DETAIL]]**<br>**[MLA(***value***)]**<br>**[NOTIFYEXTENT[(***percent***)]]**<br>**[RECOVER,{LOCK \| UNLOCK \| SUSPEND \| RESUME}(***ucat***)]**<br>**[REPORT]**<br>**[REPORT,CACHE[(***catname***)]]**<br>**[REPORT,CATSTATS[(***catname***)]]**<br>**[REPORT,CATSTATX({***catname* \| *catprefix****})]**<br>**[REPORT,DUMP]**<br>**[REPORT,PERFORMANCE[(RESET)]]**<br>**[RESET,CACHE[(***catname***)]]**<br>**[RESET,CATSTATS[(***catname***)]]**<br>**[RESTART]**<br>**[{RLSENABLE \| RLSQUIESCE }{(***ucat***) \| ,SYSTEM}]**<br>**[{SYS%ON \| SYS%OFF}]**<br>**[TAKEDUMP]**<br>**[TAKEDUMP(SYSPLEX)]**<br>**[TASKMAX(***nnn***)]**<br>**[UNALLOCATE[(***catname***)]]**<br>**[VCLOSE(***volser***)]**<br>**[VDUMPON(***pdf,rc,compid,error***)]**<br>**[VDUMPOFF]**<br>**[{VLF \| NOVLF}(***catname***)]**<br>**[{VUNALLOCATE \| NOVUNALLOCATE}]**<br>**[VVDSSPACE(***primary,secondary***)]** |

where:

**ABEND(*task*)[,FORCE]**
specifies a CAS task to be abnormally ended with abend code C1A (for the SYSPCMD and SYSPQUI tasks) or abend code 91A (for all other tasks). Any catalog request in process at the time of the abnormal ending is redriven one time. The *task* identifier can be specified as:

*id* To abnormally end the task with this unique CAS identifier. If *id* is specified, the optional FORCE parameter may also be specified:

**FORCE**
To abnormally end an active service task, even if the service task is in recall.

**Restriction:** Do not use FORCE unless the address space or task that the service task is operating on behalf of has ended abnormally.

*hextaskaddr*
To abnormally end the task whose CAS service task has this four byte hexadecimal address.

**ALLOCATE**
To terminate the CAS allocation task (IGG0CLGE) and attach a new CAS allocation task.

**ANALYSIS**
To terminate the CAS analysis task (IGG0CLGG) and attach a new CAS analysis task.

**ASYNC**
To terminate the CAS asynchronous event task (IGG0CLSA) and attach a new CAS asynchronous event task.

**MODIFY**
To terminate the CAS modify task (IGG0CLGA) and attach a new CAS modify task.

**SYSPCMD**
To terminate the Sysplex command task with a C1A ABEND; a new sysplex command task will be attached.

**SYSPQUI**
To terminate the Sysplex quiesce task with a C1A ABEND; a new sysplex quiesce task will be attached.

**ALIASLEVEL(***n***)**
specifies the number of data set name qualifiers to be used in the multilevel alias facility catalog search. The alias level is initially set at IPL with the value specified in the SYSCAT*xx* member of SYS1.NUCLEUS. The value *n* can be an integer between 1 and 4, inclusive.

**ALLOCATE(***catname***)[,{NOISC|NOVLF}]**
specifies that the catalog is to be allocated to CAS. The catalog must be specified in *catname*. The optional parameters are:

**NOISC**
specifies that the in-storage catalog is to be deactivated for the catalog.

**NOVLF**
specifies that the catalog data space cache is to be deactivated for the catalog.

**ALLOCATED[(***volser***)]**
specifies that the name, volume serial number, current allocation count, and status flags for every catalog currently allocated on the system be displayed. This information is listed in message IEC348I.

If you specify a volume serial number (*volser*), only open catalogs that reside on the specified volume are listed.

**CATMAX(***nnnn***)**
specifies the maximum number of catalogs that can be opened concurrently in CAS. When the limit is exceeded, the least recently accessed catalog is closed, freeing the CAS storage it had occupied. Closed catalogs are not unallocated. They remain allocated, but in restart status with no CAS storage. If the new limit is less than the previous limit, all currently open catalogs are closed.

The minimum value is 1 and the maximum value is 9999.

The number specified for nnnn is in decimal.

**CLOSE(***catname***)**
specifies that the catalog named in *catname* is to be closed. All CAS storage for the catalog is released. The catalog is not permanently closed. The next job that requires the catalog opens it.

**CONTENTION**
displays the Catalog resources for which contention detection is available

(ALLOCLCK, SYSIGGV2, SYSZTIOT, and SYSZVVDS), along with the current wait time and associated actions for each.

**CONTENTION(***resource*,*wait_time*,*action-flags*)**
specifies a new wait time or action (or both) for one of the Catalog resources for which contention detection is available. See "Detecting Catalog Resource Contention" on page 136 for complete information.

*resource*
*resource* can be any of:

**ALLOCLCK**
ALLOCLCK is an internal CAS lock that serializes access to CAS allocation task responsible for most catalog allocation events.

**SYSIGGV2**
The SYSIGGV2 reserve is used to serialize access to associated Catalog records.

**SYSZTIOT**
The SYSZTIOT resource is used to control access to task input/output table resources.

**SYSZVVDS**
The SYSZVVDS reserve is used to serialize access to associated VVDS records. The SYSZVVDS reserve, along with the SYSIGGV2 reserve, provide a mechanism to facilitate cross system sharing of catalogs.

The value of *resource* is case insensitive.

*wait_time*
Specifies a new wait time, in minutes, that the system lets CAS tasks wait on the *resource* resource before taking the action specified by *action-flags*.

The default *wait_time* is 10 minutes.

The minimum *wait_time* value is 5 minutes. The maximum value is 9999 minutes.

Specifying a *wait_time* value of zero disables the resource contention checking for the specified resource.

*action-flags*
Specifies an action that the system is to take when a catalog service task with an active resource in contention detection is detected past the wait threshold specified for the resource. *action-flags* can be one of:

**N** Notification action only is specified. The notification action generates a contention summary message to the operator the first time a catalog service task with an active resource in contention detection is detected past the wait threshold specified for the resource. Additionally, when a service task initially passes the wait threshold a LOGREC message specific to the task and contention event is created.

**Note:** Notification cannot be turned off for a resource for which contention detection is active.

**NR** Notification and re-drive. In addition to the notification action, the re-drive action is specified. When re-drive is active, the first time a service task with an active resource passes the contention

threshold, the service task is abended and the request is resubmitted to catalog for processing. The abend is a 91A, which will produce a dump, unless suppressed, and a LOGREC with error code 246 is generated out of module IGG0CLA9 for the service task. When the service task is abended, all resources it currently holds are freed, and the contention caused due to this request is removed.

**NR** can also be specified as **R** or **RN**.

The default *action-flags* is "N".

If *resource*, *wait_time*, and *action-flags* are all omitted, the **MODIFY CONTENTION** command displays the Catalog resources for which contention detection is available (ALLOCLCK, SYSIGGV2, SYSZTIOT, and SYSZVVDS), along with the current wait time and associated actions for each.

**DISABLE(***feature***)**
disables a particular optional feature, where *feature* can be any **one** of the following:

**AUTOTUNING**
indicates that the Catalog Address Space should not attempt to improve performance of any catalog on this system by modifying the number of data and index buffers and VSAM strings on the current system.

**BCSCHECK**
disables verification of the BCS record structure before the record is written to ensure valid records are written to the catalog. An RC14 RSN46 dump will be taken and the operation failed if an invalid BCS record is detected.

**DELFORCEWNG**
disables the warning message IDC1997I or IDC1998I when attempting to use the DELETE VVDS RECOVERY or DELETE USERCATALOG FORCE command.

**DELRECOVWNG**
specifies that message IDC1999I should not be issued if a DELETE UCAT RECOVERY command is attempted.

DELRECOVWNG is disabled, by default.

**DSNCHECK**
disables syntax checking on names being added to a catalog.

**ENQCHECK**
disables the search for outstanding SYSIGGV2 enqueues at the end of a CAS service task request.

**EXTENDEDALIAS**
disables the ability to create extension records for user catalog aliases on the current system.

EXTENDEDALIAS is disabled, by default.

**SYMREC**
specifies that SYMREC records are not created. Use this option to temporarily disable the creation of SYMREC records. For example, if a problem is causing repeated creation of SYMREC records and this is disrupting how well you are able to manage of the SYMREC target data set, you can disable the SYMREC records.

**UPDTFAIL**
disables the message IEC390I when a VSAM update request against a catalog has been abnormally terminated. This message is intended to alert the installation that potential catalog damage may have resulted from the incomplete request. The default for this option is enabled.

**VVRCHECK**
disables enhanced VVR checking on VVDS I/O.

VVRCHECK is disabled, by default.

**{DUMPON|DUMPOFF}**
specifies whether or not CAS dynamic dumping is to occur. Dynamic dumping by CAS does not occur unless you specify DUMPON. You can use these DUMPON commands for problem solving:

```
MODIFY CATALOG,DUMPON
MODIFY CATALOG,DUMPON(rc,rsn,mm)
MODIFY CATALOG,DUMPON(rc,rsn,mm,cnt)
```

where:

**rc** Specifies the catalog return code in decimal format (one to three characters from 0 to 255), or * for wild card searches.

**rsn** Specifies the catalog reason code in decimal format (one to three characters from 0 to 255), or * for wild card searches.

**mm** Specifies the catalog module identifier in CAS, or ** for wild card searches.

**cnt** Specifies to capture a dump on the nth occurrence of the condition (one to three characters from 1 to 999). The default value is '1' to cause a dump to be captured on the first occurrence of the condition.

For example:

MODIFY CATALOG,DUMPON(8,42,**) creates a dump for any return code 8, reason code 42, and for any module that detected the error.

You can use the options in parentheses that follow the DUMPON parameter to create a dump whenever a given return code, reason code, and module identifier occur. The dump can prove valuable to service personnel for solving problems. Typically, the return code, reason code, and module identifier are available on return from CAS, and printed by IDCAMS. The module identifier corresponds to the last two characters in the catalog module name. For example, the module identifier is A3 for IGG0CLA3.

You can specify the return code, reason code, and module identifiers as a string of asterisks to indicate that any value encountered will match the value of that field. This is referred to as a generic match. All three fields cannot be specified simultaneously using asterisks. Whenever you specify a generic match for a field, it is assumed that field always matches the value that is returned by catalog for a catalog request. As an example:

```
MODIFY CATALOG,DUMPON(008,042,**)
```

creates a dump for return code 8, and reason code 42, regardless of the module that detects the error.

An option has been provided for a match count to obtain the *n*th occurrence of a return code, reason code, and module identifier. The match count is decremented by one each time a return code, reason code, and module identifier is set in the catalog address space. If this option is not specified or is set to 000, then the first occurrence causes a dump.

You can set only one set of return codes, reason codes, and module identifiers at a time. Each entry overwrites the previous information. After a match occurs, the information is cleared, and the original DUMPON status is maintained. If you enter DUMPON *without* the additional options, certain conditions produce dumps automatically. If you create a DUMPON with options, a match causes a dump and the return code, reason code, and module identifier are cleared. The DUMPON status remains on.

You can use MODIFY CATALOG,REPORT,DUMP to view the settings.

The header for the catalog dynamic dump contains the return code and reason code in hex. For example:

```
CAS DYNAMIC DUMP-IGG0CLA9 RCX'F6' RSNX'00'
```

**Note:** The DUMPON parameter will only work if the module detecting the return and reason codes are not in LPA.

**ECSHR(***value***)**

Specifies changes for enhanced catalog sharing (ECS) mode, where *value* may be one of the following:

**AUTOADD**
AUTOADD indicates whether ECS-eligible catalogs should be automatically added to the ECS structure. AUTOADD re-enables and causes ECS-eligible catalogs to be automatically added to the ECS structure on the next reference to the catalog. See the ENABLE parameter description above for the conditions that make a catalog eligible. AUTOADD is a sysplex-wide function.

**CONNECT**
CONNECT causes the system to connect to the ECS structure. The catalog address space issues an IXLCONN request to allocate and connect to the ECS structure or to connect to the already allocated structure.

**DISCONNECT**
DISCONNECT causes the system to disconnect from the ECS structure. The catalog address space issues an IXLDISC request. Any ECS catalogs in the structure are removed, and will subsequently be shared via the VVDS sharing mode.

**ENABLE,***catname*
ENABLE causes the named catalog (*catname*) that is temporarily ineligible to be enabled to the ECS structure if all the following conditions are true:
- The catalog has the ECSHARING attribute
- The catalog has shareoptions(3 4)
- The catalog resides on a volume that is defined as shared
- The system is connected to the ECS structure

If any of the above conditions are not true, the request is rejected. The issuer of this command should ensure that all current or potential sharers of the catalog are capable of ECS. Otherwise, manual intervention might be required to return to the VVDS sharing protocol.

**ENABLEALL**
ENABLEALL causes all temporarily ineligible catalogs known to CAS to be enabled for ECS.

**REMOVE,***catname*
REMOVE causes the named catalog (*catname*) to be removed from the ECS structure, thereby affecting all systems sharing this catalog. The sharing

protocol then reverts to the VVDS method. The catalog will not automatically be added to the ECS structure (even if AUTOADD is currently enabled) until the catalog is re-enabled for ECS activity by one of the following:
- The MODIFY CATALOG,ECSHR(ENABLE) command is issued for the catalog
- The MODIFY CATALOG,ECSHR(ENABLEALL) command is issued
- The MODIFY CATALOG,ECSHR(AUTOADD) command is reissued.

**STATUS**
STATUS causes the status of each catalog that has been referenced since the last IPL to be displayed in message IEC380I. The message includes the status of the CF connection as well as the ECS status of each catalog.

**STATUS,***catname*
STATUS causes the status for the named catalog (*catname*) to be displayed in message IEC380I. The message includes the status of the CF connection as well as the ECS status of the named catalog.

**ENABLE(***feature***)**
enables a particular optional feature, where *feature* can be any **one** of the following:

**AUTOTUNING**
Indicates that the Catalog Address Space should automatically attempt to improve performance of catalogs by modifying the number of data and index buffers and VSAM strings on the current system. This is the default value.

**BCSCHECK**
Enables verification of the BCS record structure before the record is written to ensure valid records are written to the catalog. An RC14 RSN46 dump will be taken and the operation failed if an invalid BCS record is detected.

**DELFORCEWNG**
enables issuance of messages IDC1997I and IDC1998I when a DELETE VVDS RECOVERY or DELETE USERCATALOG FORCE are performed.

**DELRECOVWNG**
specifies that message IDC1999I be issued if a DELETE UCAT RECOVERY command is attempted.

**DSNCHECK**
enables syntax checking of data set names being added to a catalog.

DSNCHECK is enabled by default.

**ENQCHECK**
enables the search for outstanding SYSIGGV2 enqueues at the end of a CAS service task request. Enabling this check will cause increased CPU time and degraded performance and is not recommended for general use. This request will cause CAS to set a DUMPON for an RC246 RSN244 error and will take a dump when it finds an outstanding SYSIGGV2 resource is owned by the finished request. Please send the dump to the IBM Service Center.

**EXTENDEDALIAS**
enables the ability to create extension records for user catalog aliases on the current system. Note that you should only enable this feature when all systems in the sysplex are V1R13 or greater.

**SYMREC**
specifies that SYMREC records are to be created. Use this option to reset the default value if you have disabled the creation of SYMREC records using the MODIFY CATALOG, DISABLE(SYMREC) command.

SYMREC is enabled by default.

**UPDTFAIL**
enables the message IEC390I to notify the operator when a VSAM request to update a catalog fails abnormally. This notification is designed to alert the installation that a catalog may be damaged by an incomplete update request against a catalog. The default is that this feature is enabled.

UPDTFAIL is enabled by default.

**VVRCHECK**
enables enhanced VVDS record validation during VVDS I/O.

**END(***id***)[,{REDRIVE|NOREDRIVE}][,FORCE]**
specifies that the task identified by *id* is to be ended. This is the preferred method of ending a CAS task. The subparameters are:

*id* specifies the unique task identification.

**REDRIVE**
specifies that any catalog request that is in process at the time of the abnormal ending is to be redriven. The CAS task is abnormally ended with abend code 91A. The **MODIFY CATALOG,END(***xx***),REDRIVE** command may be entered as many times as necessary for a given catalog request.

**NOREDRIVE**
specifies that any catalog request that is in process at the time of the abnormal ending fails with catalog return code 246. The CAS task is abnormally ended with abend code 71A. NOREDRIVE is the default.

**FORCE**
specifies the abnormal ending of an active service task, even if it is in recall.

**Restriction:** Do not use FORCE unless the address space or task that the service task is operating on behalf of has ended abnormally.

**ENTRY[(***csectname***)]**
specifies that the starting addresses, the FMIDs, and the PTF/APAR levels of all the modules in the catalog load modules IGG0CLX0 (resident in CAS) and IGG0CLHA (resident in the link pack area) are to be displayed. The information is displayed in message IEC349I.

You can specify an individual CSECT in *csectname*, or you can have information for all entry points displayed by omitting a CSECT name.

The output of this command is probably best viewed on the system log due to its size, if all entry points are requested.

**{ISC|NOISC}(***catname***)**
specifies whether the indicated catalog's records are to be held in the in-storage catalog (ISC). The catalog must already be allocated to use this command. **ISC** specifies the catalog is to use ISC; **NOISC** specifies the catalog is not to use ISC.

The objective of in-storage cache is to cache only those records that are read directly. Records will not be cached for other types of requests, and therefore in-storage cache is not then in effect. The ISC flag will be turned on the first

time a direct read is done. This flag will remain on until that record is no longer in cache. At that point, it will remain off until the next direct read to the catalog.

**LIST[(*task*)]**
specifies that currently active CAS service tasks, their related job names, elapsed times, and unique identifications, are to be listed. The information is listed in message IEC347I. The identifiers listed can then be used in other MODIFY CATALOG commands that require a CAS task ID.

All current tasks are listed unless *task* is specified. The *task* can be specified as:

*id* To list information about the task with this unique identification.

*hextaskaddr*
To list information about the task whose task control block (TCB) address is this four byte hexadecimal address.

*jobname*
To list information about all service tasks currently active for this job.

**LISTJ(*jobname*)[,DETAIL]**
Specifies that information should be displayed about the status of the catalog service task that is processing the catalog request for the specified *jobname*. If specified without the DISPLAY keyword, the output for LISTJ(*jobname*) is similar to that produced by LIST(*jobname*).

**DETAIL**
Requests additional optional information. This optional information is primarily internal information about the catalog request. However, some of the information can be useful for diagnosing real-time problems involving the catalog address space. For example, an entry could show that the request has been waiting for a long time for completion of an ENQ. Then the ENQ resource name shown in the detail information could be used with a D GRS,C to find out what task in the system might be causing the wait condition.

**MLA(*value*)**
allows the operator to selectively enable, disable, or rebuild the multi-level alias facility control blocks. This may be necessary if the MLA has disabled itself, which is usually indicated by one of the messages IEC369I, IEC370I, IEC374I, or IEC375I. *value* may be specified as:

**DISABLE**
to disable MLA processing. When the MLA is disabled, generic searches will fail with return code 194 and reason code 8.

**ENABLE**
to re-enable and rebuild the MLA control blocks. This may be used to counteract a previous DISABLE command or to activate the MLA after a previous failure during its initialization.

**REFRESH**
forces a complete rebuild of the MLA control blocks. This can be used when an error is suspected in the MLA structure, but the MLA logic detects no condition that it considers a trigger for a rebuild.

**NOTIFYEXTENT(*percent*)**
specifies the percentage of the maximum extents possible for a catalog that is currently allocated. *percent* is a percentage number from 0-99. (You can omit leading zeros.) A percentage value of zero indicates that normal monitoring is suppressed. The default is 80. The setting specified is retained across catalog

restarts, but not IPLs. If the allocated extent threshold exceeds the given threshold for any catalog, the system will issue message IEC361I for that catalog. If a catalog exceeds 90% utilization of the maximum extents, the system will issue message IEC361I even if the threshold has been set to zero (that is, no normal monitoring).

**[RECOVER,{LOCK | UNLOCK | SUSPEND | RESUME}(*ucat*)]**
specifies a dynamic reset of the catalogs LOCK/UNLOCK or SUSPEND/RESUME VVR setting.

These commands are sysplex in scope and so should only be issued from one system in the sysplex. Each command must either fully complete or time out before another command can take effect. If you issue multiple commands simultaneously from multiple systems, the commands execute on a first come first serve basis with a sysplex scope.

To issue these commands, you must have read access to the IGG.CATLOCK profile, and ALTER authority to the catalog. If the IGG.CATLOCK profile is not defined or the FACILITY class is not active, you cannot use this command.

**LOCK**
specifies that the system issue a serialized LOCK and CLOSE of the catalog. The LOCK attribute is an existing, permanent attribute of a catalog. Use LOCK to lock a catalog that is being reorganized or recovered. Unauthorized LOCK requests fail with return code 186.

For more information about LOCK, see the IDCAMS DEFINE USERCATALOG LOCK and ALTER *ucatname* LOCK in *z/OS DFSMS Access Method Services Commands*.

**UNLOCK**
specifies that a locked catalog be unlocked for a catalog that was locked in one of the following ways:

- F CATALOG,RECOVER,LOCK command
- IDCAMS command DEFINE USERCATALOG
- IDCAMS command ALTER *ucatname* LOCK.

See *z/OS DFSMS Access Method Services Commands* for IDCAMS commands.

**SUSPEND**
specifies that the system:

1. Perform a serialized close of the catalog for RLS or non-RLS access across the sysplex.
2. Obtain serialization to suspend new (unauthorized) catalog requests.
3. Set the catalog SUSPEND attribute. The SUSPEND attribute is a new catalog state that you can set using one of the following:
   - This command, F CATALOG,RECOVER,SUSPEND
   - IDCAMS command DEFINE USERCATALOG
   - IDCAMS command ALTER *ucatname* SUSPEND.

**RESUME**
specifies that catalogs suspended either by the F CATALOG or IDCAMS commands resume processing. This command also resets the SUSPEND attribute.

*ucat*
Specifies the catalog name or names for the RECOVER command. You can specify multiple catalogs to recover in one command using wildcards.

The user performing the forward recovery must be authorized as follows: If you have READ access to the IGG.CATLOCK profile and ALTER authority to the catalog, you can lock, unlock, suspend, or resume a catalog. If you have READ access to the IGG.CATLOCK profile, you can access and repair a locked or suspended catalog. If the IGG.CATLOCK profile is not defined, or the FACILITY class is not active, you cannot lock, unlock, suspend, or resume a catalog. The new commands may time out after x minutes and will result in a reversal of the command. Issuing the LOCK or SUSPEND command cannot cancel (reverse) a executing UNLOCK or RESUME command or vice versa. Each command must fully complete or time out, before another command will take effect. Multiple commands may be simultaneously issued from multiple systems, however, the commands will execute on a first come first serve basis.

**REPORT**
provides basic information about some of the current limits and installation-specified defaults that are selected for the catalog address space.

**REPORT,CACHE[(*catname*)]**
causes general information on catalog cache status for all catalogs currently active in the catalog address space to be listed. The report generated shows information useful in evaluating the catalog cache performance for the listed catalogs. If you specify a catalog name (*catname*), performance information will be listed only for that catalog.

**REPORT,CATSTATS[(*catname*)]**
lists the I/O statistics and BUFNI, BUFND, and STRNO for all catalogs currently active in the catalog address space. If you specify a catalog name (*catname*), statistics will be listed for only that catalog.

**REPORT,CATSTATX[({*catname* | *catprefix**})]**
lists the same I/O statistics and settings produced by CATSTATS, and additionally provides CA-Reclaim and CA-Reuse statistics. Output is in the form of a table for each selected catalog. The catname or catprefix* must be specified. If *catprefix** is used, all currently open catalogs which match the prefix will be reported on. Caution should be used with the *catprefix** option as it can generate a significant amount of WTO output.

**REPORT,DUMP**
The DUMP option is used to display the current dump status of catalog address space. This dump status can be enabled or disabled by DUMPON and DUMPOFF forms of the MODIFY command.

**REPORT,PERFORMANCE**
PERFORMANCE lists information about events in CAS that invoke code outside of the catalog component. It shows the total number of occurrences of each event and the average time spent completing that event.

**REPORT,PERFORMANCE(RESET)**
Use the RESET keyword to reset all performance statistics being accumulated. This allows you to periodically reset the statistical information and gather data to create a profile of the performance of the Catalog Address Space in your environment. The RESET keyword is also helpful in problem determination by allowing you to set the current statistics to zero during a period when performance might be a problem, and gather data specifically for that period.

**RESET,CACHE[(*catname*)]**
resets the cache statistics for all catalogs currently active in the Catalog address space. If you specify a catalog name (*catname*), statistics will be reset only for that catalog.

You can use this command to reset the cache statistics and determine how much activity the specified catalog gets during a given time period. To see the current cache statistics report, use the F CATALOG,REPORT,CACHE option.

**RESET,CATSTATS[(*catname*)]**
resets the I/O statistics for all catalogs currently active in the Catalog address space. If you specify a catalog name (*catname*), statistics will be reset only for that catalog.

You can use this command to reset the I/O statistics and determine how much activity the specified catalog gets during a given time period. To see the current I/O statistics report, use the F CATALOG,REPORT,CATSTATS option.

**RESTART**
specifies that CAS is to be restarted in a new address space. The CAS mother task is ended with abend code 81A, and any catalog requests in process at the time are redriven. The RESTART command should *only* be used when the only other option is an IPL. You should try other versions of the MODIFY command first before you use RESTART to solve catalog or VVDS problems.

**RLSENABLE | RLSQUIESCE {(*ucat*) | ,SYSTEM}**
specifies that the specified catalog or catalogs be enabled or quiesced for VSAM record-level sharing (RLS).

These commands are sysplex in scope and result in a sysplex wide enable or quiesce of the catalogs, and so should only be issued from one system in the sysplex. Issuing the RLSENABLE command cannot cancel or reverse a executing RLSQUIESCE command or vice versa. Each command must either fully complete or time out before another command can take effect. If you issue multiple commands simultaneously from multiple systems, the commands execute on a first come first serve basis with a sysplex scope.

Catalog requests will be suspended during the process of enabling or quiescing RLS access for catalogs. No catalog requests will fail as a result of the enable or quiesce.

*ucat*
Specifies the catalog name or names for the RLSENABLE | RLSQUIESCE command. *ucat* can be specified as:

- *ucat*, specifying an individual catalog using a fully qualified catalog name. The catalog name specified must be connected to active master catalog.
- *ucat**, specifying a list of catalogs using a partially qualified catalog name. The system resolves the list of catalogs from the master catalog that's active on the system where you issue the command.

**SYSTEM**
Specifies that RLSENABLE or RLSQUIESCE impact all catalogs across the sysplex opened for the system the command is issued from.

- RLSENABLE,SYSTEM closes all non-RLS catalogs across the sysplex opened on the system the command is issued from. Once the non-RLS catalogs are closed, the RLS QUIESCE attribute is set to NO in the catalogs' VVR entry. The next reference will access the catalogs for RLS.
- RLSQUIESCE,SYSTEM closes all RLS catalogs across the sysplex opened for the system the command is issued from. Once the RLS catalogs are closed, the RLS QUIESCE attribute is set to YES in the catalogs' VVR entry. The next reference will access the catalogs for non RLS.

**{SYS%ON|SYS%OFF}**
specifies that SYS% conversion to SYS1 either is (SYS%ON) or is not (SYS%OFF) to occur. The default setting for SYS% conversion is set during system initialization.

**TAKEDUMP**
causes the Catalog Address Space to issue an SVCDUMP using the proper dump options to ensure all of the data needed to diagnose catalog problems is available. This eliminates the need for a user to issue an MVS DUMP command and potentially omit vital dump parameters needed for the problem.

**TAKEDUMP(SYSPLEX)**
causes the issuance of sysplex-wide dumps. The dump of each system includes the Catalog, Master, and SMSVSAM address spaces and the SMSVSAM data spaces. Because these dumps will be large, you should ensure the dump data sets are large enough to handle this much data.

**TASKMAX(*nnn*)**
specifies the maximum number of CAS service tasks that are attached to process (non-CAS) catalog requests. Once this limit has been reached, further requests for CAS services are delayed until a task control block becomes available.

The value *nnn* must be in the range 24-360.

The number specified for *nnn* is in decimal.

**UNALLOCATE[(*catname*)]**
specifies that all catalogs allocated to CAS are to be removed from CAS. The CAS storage used by the catalogs is freed and the devices on which these catalogs reside are also unallocated from CAS.

If you specify a catalog (*catname*), only the specified catalog is unallocated. If this is the only catalog that resides on the device, the device on which the catalog resides is unallocated from CAS.

**VCLOSE(*volser*)**
specifies that the VVDS that resides on the volume with the volume serial number *volser* is to be closed. The VVDS is reopened by the next request for the VVDS, and new control blocks are built.

Do not use VCLOSE to unallocate a VVDS to vary a volume offline. Use VUNALLOCATE.

**{VDUMPON|VDUMPOFF}**
specifies whether or not user-initiated VSAM dynamic dumping is to occur. You can use the VDUMPON parameter to create a dump whenever a given combination of PDF code, error code, component code, or return code is encountered. You would usually request such a dump in response to the codes returned in the request parameter list (RPL) feedback area. (See "Return Codes from the Record-Management (Request) Macros" in *z/OS DFSMSdfp Diagnosis*.) The dump can provide valuable information to IBM service personnel for solving problems.

The VDUMPON command is specified with the following format:

```
MODIFY CATALOG,VDUMPON(pdf,rc,compid,error)
```

where:

**pdf** Specifies the VSAM Problem Determination Function code (one to three characters from 0 to 255), or * (asterisk).

**rc** Specifies the VSAM return code in decimal format (one to three characters from 0 to 255), or * (asterisk).

**compid**
Specifies the component code (0 - 5), or * (asterisk).

**error** Specifies the VSAM error code in decimal format (one to three characters from 0 to 255), or * (asterisk).

**Note:**

1. Specifying an asterisk (*) for any parameter indicates that a wild card search is to be done for that value. Up to three asterisks (without intervening spaces) may be specified where a single asterisk is allowed, but the extra asterisks have no effect on the command's output.
2. If a parameter is to be omitted, it must be specified as an asterisk (*). For example, **VDUMPON(*,*,*,2)** is syntactically valid, while **VDUMPON(2)**, **VDUMPON( , , ,2)**, and **VDUMPON(2, ,2)** are all invalid.
3. At least one of the parameters must be specified with a value other than asterisks. For example, **VDUMPON(*,*,*,*)** is not allowed.
4. User-initiated VSAM dynamic dumping does not occur unless you specify VDUMPON.
5. You can set only one VDUMPON at a time. Each entry overwrites the previous information. After a match occurs, the information is cleared and no further user-initiated dumps will be taken.

For example:

```
MODIFY CATALOG,VDUMPON(*,8,*,20)
```

requests a dump for any return code 8, error code 20 (exclusive control conflict), with any PDF code and any component code.

z/OS acknowledges the command with the following messages:

```
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT VDUMP OUTPUT 503
*CAS*********************************************************
*    STATUS  FUNC CODE  RETURN CODE  COMPONENT  ERROR CODE      *
*       ON      ***         008         ***        020         *
*CAS*********************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Issuing MODIFY CATALOG,VDUMPOFF will clear the VDUMP options.

**{VLF|NOVLF}(*catname*)**
specifies that the catalog data space cache is to be activated (VLF) or deactivated (NOVLF) for the catalog named *catname*. To activate the catalog data space cache (CDSC) for a catalog, the catalog must already be defined as eligible for CDSC. To deactivate CDSC, the catalog must already be allocated to CDSC.

**VUNALLOCATE|NOVUNALLOCATE**
specifies that all dynamically allocated VVDSs are to be unallocated from CAS (VUNALLOCATE) when a request is completed, or left allocated to CAS (NOVUNALLOCATE) after a request has completed. By default, VVDSs are unallocated after processing a request. Performance can be improved if you specify NOVUNALLOCATE, because repeated requests for a volume do not result in repeated dynamic allocations for the VVDS.

If you specify NOVUNALLOCATE, VVDSs remain allocated until VUNALLOCATE is issued, CAS is restarted, or the system is IPLed.

VUNALLOCATE can be used to unallocate a VVDS from CAS to allow a volume to be varied offline.

**VVDSSPACE(***primary*,*secondary***)**
Indicates that the Catalog Address Space should use the values specified as the primary and secondary allocation amount in tracks for an implicitly defined VVDS. The default value is ten tracks for both the primary and secondary values. The specified values are preserved across a Catalog Address Space restart, but are not preserved across an IPL.

# Chapter 9. Integrated Catalog Forward Recovery Utility (ICFRU)

This appendix is for the person who is responsible for setting up the installation's diagnostic and recovery procedures for catalogs. This will normally be an individual who performs systems programming tasks or data and storage administration tasks. The purpose of this appendix is to enable the user of the Integrated Catalog Forward Recovery Utility to understand and execute the programs and to interpret their output. It also assists in establishing a good recovery environment by providing guidance on how to manage the data needed for catalog recovery.

To use ICFRU effectively, you will need experience in z/OS data and storage management and particularly in catalog management. Familiarity with the installation's backup and recovery tools and techniques and with the use of Access Method Services commands is assumed. You may also need to refer to other sections in this book or to one or more of the following books:

*z/OS DFSMS Access Method Services Commands*
*z/OS MVS JCL Reference*
*z/OS MVS JCL User's Guide*
*z/OS MVS System Management Facilities (SMF)*
*z/OS MVS System Messages, Vol 1 (ABA-AOM)*
*z/OS MVS System Messages, Vol 2 (ARC-ASA)*
*z/OS MVS System Messages, Vol 3 (ASB-BPX)*
*z/OS MVS System Messages, Vol 4 (CBD-DMO)*
*z/OS MVS System Messages, Vol 5 (EDG-GFS)*
*z/OS MVS System Messages, Vol 6 (GOS-IEA)*
*z/OS MVS System Messages, Vol 7 (IEB-IEE)*
*z/OS MVS System Messages, Vol 8 (IEF-IGD)*
*z/OS MVS System Messages, Vol 9 (IGF-IWM)*
*z/OS MVS System Messages, Vol 10 (IXC-IZP)*

## Introduction to ICFRU

The Integrated Catalog Forward Recovery Utility (ICFRU) assists z/OS users in recovering a damaged catalog to a correct and current status. All types of catalog entries that may exist in the basic catalog structure of the integrated catalog facility are supported, including those for VSAM, non-VSAM, and generation data groups. A catalog to be recovered may have been shared by multiple systems. A master catalog may be recovered, provided it is not in use as a master catalog.

**Note:** Within this appendix, the term "catalog" should be understood to mean "an integrated catalog facility catalog", that is, "a basic catalog structure or BCS" (not to include the VSAM volume data set or VVDS).

An error-free backup of an undamaged basic catalog structure is used as the base for recovery. Catalog changes logged as SMF (System Management Facilities) records are selected and applied to this backup to create a new image of the catalog as it should now exist. This image of the catalog can then be reloaded to produce a current basic catalog structure.

ICFRU:

- Helps improve system availability by reducing catalog recovery time
- Supports all data set types — VSAM, non-VSAM, and generation data groups
- Supports recovery of catalogs shared by multiple systems
- Provides extensive diagnostic facilities
- Permits advance assessment of recovery completeness
- Provides for catalog forward recovery without user-written code
- Avoids catalog repair techniques requiring great technical expertise
- Supports standardization and rehearsal of backup and recovery procedures
- Permits non-disruptive testing of catalog recovery procedures

Most z/OS installations today depend heavily on the availability of catalog facilities for continued operation of batch processing, online systems, and time-sharing systems. An extended outage of a catalog can be extremely disruptive. While there are a number of programs available that can reload a copy of a catalog as it existed at some previous point in time, normal catalog usage cannot resume until it has been resynchronized with the data sets as they currently exist. Thus the need is for a facility that can quickly, and without great technical expertise, recover a catalog to a current and correct status, that is, to the point of failure.

The ICFRU provides the capability to recover a catalog to current and correct status quickly and easily. Combined with a regular program of catalog diagnostics and backups and proper recording, dumping, and tracking of data from the Systems Management Facilities (SMF), ICFRU can help shorten the time of a catalog outage and reduce the need to maintain a high level of technical expertise in catalog management.

## How the ICFRU Works

The ICFRU relies on the fact that catalog management routines log each catalog change to SMF (if you are recording the appropriate SMF record types). These SMF records contain images of the catalog records that can be combined with the catalog records from an IDCAMS EXPORT copy of a catalog to build a data set functionally identical to an IDCAMS EXPORT copy of the catalog as if it had been taken after the changes logged to SMF were applied. The new "EXPORT data set" can then be reloaded using IDCAMS IMPORT, thus recovering the catalog to the point of failure (or to the specified time).

There are actually two programs in ICFRU: Integrated Catalog Forward Recovery Record Selection and Validation (CRURRSV) and Integrated Catalog Forward Recovery Record Analysis and Processing (CRURRAP). Record Selection and Validation processes SMF dump data sets, extracting appropriate records that are then sorted and processed, together with an EXPORT copy, by Record Analysis and Processing to produce a data set, to be imported. See Figure 9 on page 167.

### SMF Record Selection and Validation

Recognizing the importance of complete SMF data to the recovery process, the most significant function of CRURRSV is to assist in verifying that all required SMF data is present, that is, that no SMF data has been omitted or lost. Gaps in SMF data longer than a user-specified interval are noted, as are any SMF "lost data" records. SMF IPL, EOD, and SWITCH records are logged to aid in the resolution of gaps.

SMF dump data sets from all systems having access to the catalog, covering the user-specified recovery start and stop times are needed as input. In multi-system installations, a clock difference value supplied by the user is used to adjust the start and stop times to ensure that all SMF records needed for recovery are selected. The outputs from CRURRSV are a set of reports, a log of significant

events encountered and an output data set containing the SMF records for the catalog to be recovered. The log and the reports are used to assess the completeness of the SMF input data.

It is also possible to select records for all catalogs, as an alternate method of SMF catalog record collection or in the unlikely event that multiple catalogs are to be recovered.

### Sorting the SMF Records

Before processing by CRURRAP, the SMF catalog records must be sorted into the same sequence as the records in the EXPORT data set, ascending data set name sequence. The records must be further sorted into descending date and time sequence. CRURRAP demands that the most current version of a record be processed first so that older records can be logged should an error be encountered.

### SMF and EXPORT Record Analysis and Processing

Based on user specifications, Record Analysis and Processing accepts and inspects the SMF and EXPORT records. Only the most current version of a catalog record is actually processed, although older records are also analyzed. If the only record for a catalog entry is the one from the EXPORT input, that record is written to the new "EXPORT data set". If the most current SMF record is an insert or an update, it is written to the "EXPORT data set" (if necessary, after a common VSAM password is inserted to replace VSAM security fields blanked out in the SMF record). If the most current record is a deletion, that record is omitted from the new "EXPORT data set". As each record is processed, it is tallied in an appropriate category. Finally, reports of errors, anomalies, correct processing and a report of records by data set are produced to aid in error resolution and, if desired, in auditing the process.

In summary, CRURRAP needs start and stop dates and times and a multi-system clock difference as execution parameters. The sorted SMF catalog records covering the start/stop interval and an EXPORT copy of the catalog from the start date and time are supplied as input. A new "EXPORT" data set is produced as output.

Note: It is also possible to use programs other than IDCAMS EXPORT for regular backups, but additional recovery steps will be required. See "Alternate Backup Methods" on page 170.

## ICFRU System Flow

You can control the execution of ICFRU programs through the specification of execution parameters (the same parameters for both CRURRSV and CRURRAP) and by providing the appropriate SMF dump data sets and EXPORT data set. Final recovery through IDCAMS IMPORT can be executed when you are satisfied with the results of ICFRU.

To use the ICFRU, you will need:

- The name of the catalog to be recovered
- Recovery start date and time - the date and time at which the backup to be used for recovery was made
- Recovery stop date and time - the date and time that corresponds to the closing of the catalog (or the time after which no updates were otherwise possible)
- An SMF gap time - the (approximate) interval just smaller than the minimum time used to fill (or switch) an SMF recording data set

- A multi-system clock difference - the (approximate) maximum difference between the TOD clocks of any two systems sharing the catalog
- All SMF dump data sets spanning the recovery period
- A sort utility and appropriate sort control
- The IDCAMS EXPORT data set to be used as the basis for recovery

The following figure presents an overall view of the ICFRU.

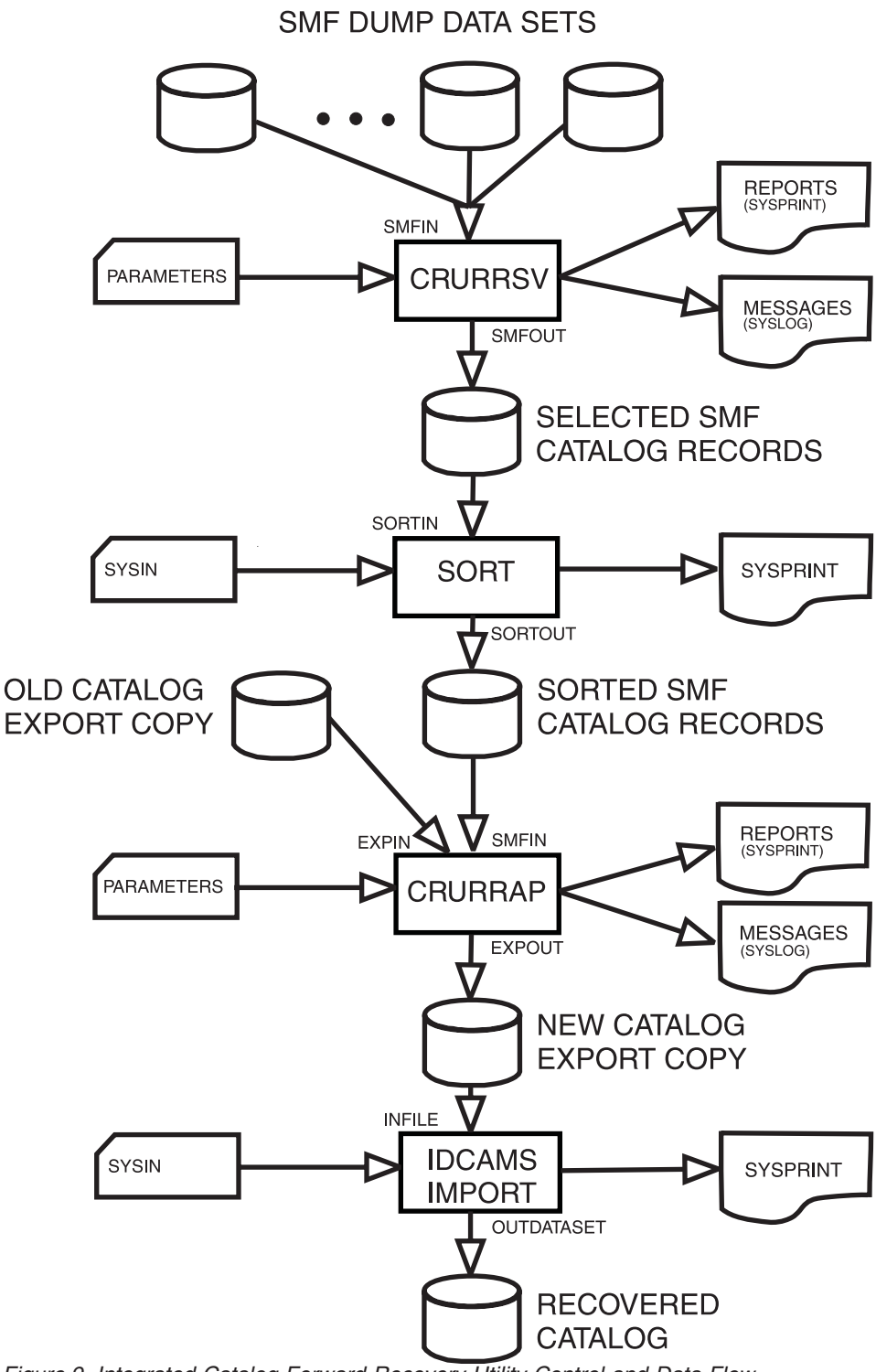



*Figure 9. Integrated Catalog Forward Recovery Utility Control and Data Flow*

## Specified Operating Environment

The Integrated Catalog Forward Recovery Utility is designed to operate with, and is supported for, any supported release of z/OS.

The ICFRU programs are designed to operate using the configurations supported by any supported release of z/OS.

The programs in ICFRU are dependent on the internal format of the EXPORT portable data set, the SMF record types 61/65/66, and the catalog records contained in these sources.

Approximate region size requirements for the programs are shown below. If additional buffers are specified by JCL, more storage may be required.
- CRURRSV - 256K
- CRURRAP - 512K

The programs of ICFRU execute below the 16M-byte virtual storage line. However, even with a large number of buffers, the required region size is less than 2 megabytes.

In addition, a sort facility, such as z/OS DFSORT or the equivalent, is required.

## Confirming Installation Readiness

To confirm that your systems are recording the correct SMF data and that other environmental factors will support the use of ICFRU in an actual recovery situation, we recommend that you execute ICFRU, using your own data as input. In general, the process will "simulate" a real recovery procedure (see "Executing Catalog Recovery" on page 173), with these exceptions:
- The actual recovery (that is, the IMPORT) need not be done.
- A later EXPORT should be created to compare with the new EXPORT created by CRURRAP.

In time sequence, the data should look as shown in Figure 10.

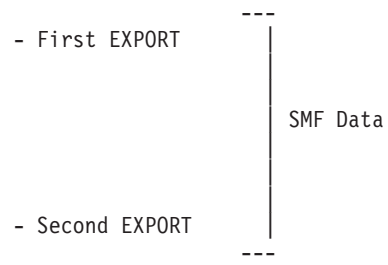



*Figure 10. Time Sequence for SMF Data Collection.* Both of the two EXPORTs must be inside the interval spanned by the SMF data.

In other words, the SMF data should start before the first EXPORT and continue after the second EXPORT, completely spanning the period between the EXPORTs.

To confirm installation readiness, use this somewhat simplified version of the general recovery procedure in "Executing Catalog Recovery" on page 173:
1. Do two EXPORTs. The amount of time between them is at your discretion. Bear in mind that in a real situation you are unlikely to want to create backups for catalogs less often than once a day, unless the catalog has low activity. Large quantities of SMF data can be time-consuming to process and difficult to manage.
2. Take note of the date and time information from the top of the SYSPRINT output for the EXPORTs (message IDC0594I).

3. Ensure that you have all the SMF data covering the intervening period. It may be necessary to switch and dump SMF data sets to accomplish this, depending on your installation procedures for handling SMF data. ICFRU cannot handle SMF data directly from the VSAM SMF data sets, nor can it handle concatenation of unlike devices, so all the SMF data must be gathered together onto similar media or batched and then concatenated after multiple passes with CRURRSV.
4. Run the CRURRSV, SORT and CRURRAP steps as shown in Figure 12 on page 178, Figure 13 on page 179 and Figure 14 on page 180. The specified start time should be the time of the first EXPORT; the specified stop time should be the time of the second EXPORT.
5. Compare the EXPORT data set created by ICFRU processing with the second IDCAMS EXPORT data set you created at the simulated failure point. See Figure 11 for suggested JCL.

```
//*********************************************************
//* INTEGRATED CATALOG FORWARD RECOVERY UTILITY           *
//* JCL EXAMPLE -                                         *
//* THIS JCL EXECUTES IEBCOMPR TO COMPARE THE NEW EXPORT *
//* PRODUCED BY CRURRAP WITH A SECOND EXPORT TAKEN AT     *
//* THE RECOVERY STOP TIME.                               *
//*                                                       *
//* THIS COMPARISON IS TO BE USED ONLY FOR TESTING OR     *
//* REHEARSAL SITUATIONS SINCE A SECOND EXPORT WILL       *
//* NOT EXIST IN A REAL RECOVERY SITUATION.               *
//*                                                       *
//* THIS STEP SHOULD ALSO BE EXECUTED AS A PART OF THE    *
//* INSTALLATION READINESS CONFIRMATION.                  *
//* MODIFY THE SYSUT1 DD STATEMENT TO POINT TO            *
//*        THE FIRST (OLDER) EXPORT DATA SET.             *
//* MODIFY THE SYSUT2 DD STATEMENT TO POINT TO            *
//*        THE SECOND (NEWER) EXPORT DATA SET.            *
//*********************************************************
//* COMPARE OUTPUT FROM CRURRAP WITH 'REAL' EXPORT.
//*********************************************************
//COMP  EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=Baplicat.NEW.CATALOG.EXPORT,DISP=SHR
//SYSUT2   DD DSN=Baplicat.SECOND.CATALOG.EXPORT,DISP=SHR
//SYSIN    DD DUMMY
```

*Figure 11. Comparing the Results of CRURRAP with an IDCAMS EXPORT.* Use this JCL (for tests or rehearsals) to compare the output of ICFRU with an EXPORT copy of the catalog created at the simulated point of failure.

6. Examine the output from the program runs including any miscompares from IEBCOMPR.

   In the IVP, the second EXPORT data set should compare exactly with the EXPORT data set created by CRURRAP. **This will not be true for all cases where this technique of comparing two EXPORTs is used.** The control records in the EXPORT data sets contain values representing the status of the catalog at the time of EXPORT. Since the EXPORTs are from different times, certain values may miscompare in records 3, 5, and 7. This is of no concern since the catalog will be redefined during the actual IMPORT and these values will not be used.

   There should be no miscompares after the last EXPORT control record (currently record number 8) except when VVR relative byte addresses (RBAs) have changed in the ecord. Again, this is of no concern and is expected as catalog management will reestablish the correct values when the catalog entry is first accessed.

See "Sample Reports from ICFRU" on page 204 for examples of correct output from ICFRU. Other sections in this appendix describe the logs, messages, and reports, including interpretation and suggested responses.

7. If you find other errors during program execution, or other miscompares, check the following:
   - SMFPRMxx member in SYS1.PARMLIB - Are you recording record types 61/65/66 for all jobs, including subsystems and system tasks?
   - SMF data - Are you sure you have all the SMF data covering the period between the two EXPORTs?
   - Execution parameters - Are you certain that the start and stop times correspond to the times of the two EXPORTs (see the SYSPRINT output from EXPORT)?
   - RBA values - Did the VVR RBA values in the record change? This is to be expected. This is not an error. If the record appears to be otherwise intact, do not be concerned.

# Operating ICFRU

## Planning for Catalog Recovery

To recover a catalog, the ICFRU needs three inputs:
- An IDCAMS EXPORT copy of the catalog
- The SMF data from all systems accessing the catalog
- Parameters describing the needed recovery action

Correct recovery requires that adequate records be kept for the catalog backup data sets and for the SMF dump data sets. It must be possible to:
1. Determine the date and time of the last valid catalog backup
2. Identify and locate the catalog backup data set
3. Identify and locate the SMF dump data sets covering the period between backup and failure

### Creating Catalog Backups

**Diagnostic Prerequisites:** Since the backup copy of the catalog is the basis for forward recovery, steps should be taken to ensure the integrity of the catalog when it is backed up. We suggest that you run IDCAMS EXAMINE, DIAGNOSE ICFCATALOG, and LISTCAT VOLUMES at backup time.

If the catalog fails one of these diagnostic tests, corrective action should be taken as soon as possible. Subsequent backups of the catalog cannot provide an adequate base for recovery. Errors in the backup copy are not corrected by the ICFRU.

**Running IDCAMS EXPORT:** If the diagnostic steps reveal no errors, the IDCAMS EXPORT step should be executed. If the EXPORT is successful, the backup data set should be cataloged along with the accompanying listing which gives the date and time of the catalog copy. See Figure 29 on page 211 and Figure 30 on page 212 for an example of the diagnostic and backup job that follows these recommendations.

We suggest that the backup job be run daily. If recovery is required, it should take only a few minutes to read the SMF records since the last backup.

**Alternate Backup Methods:** Even though the Record Analysis and Processing program requires an IDCAMS EXPORT copy of the catalog as input to recovery, it is still possible to use other programs, such as DFSMShsm and DFSMSdss, to backup your catalogs. However, if you do so, you will first have to recover or

restore the catalog to be recovered and then create an EXPORT copy using IDCAMS. Consequently, we recommend that IDCAMS be used as the regular backup program.

If you employ an automatic backup method or a physical dumping technique, do not neglect the need to run regular diagnostics for your catalogs. If you use an automated method, you will not be able to synchronize backup and diagnosis. With physical dumping, you will not have the record level copy that implies at least a minimal level of index- and record-sequence checking. Thus the diagnostic steps should not be omitted.

Just as with IDCAMS EXPORT, you must keep track of the date and time of catalog backups when other programs are used. DFSMShsm will record this information automatically. With DFSMSdss, you must use a technique similar to the one suggested in Figure 29 on page 211 and Figure 30 on page 212.

## Collecting the SMF Records

**Setting the SMF Parameters:** The ICFRU requires that all SMF type 61, type 65 and type 66 records be recorded. You should ensure that the SMF parameters specify recording for these record types for all jobs — batch, TSO and started tasks — and for all systems. Check the SMFPRM*xx* member in SYS1.PARMLIB.

**Dumping the SMF Data:** The Record Selection and Validation program does not have the capability to process the SMF VSAM recording data sets. Consequently, all SMF data must be dumped before it can be used. Your existing procedures for managing the SMF dump data sets can probably be used to supply input to the ICFRU.

**SMF Data Required for Recovery:** For complete analysis and reporting, Record Selection and Validation needs the set of all SMF records (of all types), from all systems having access to the catalog being recovered, from the time just preceding the backup time until just after the specified recovery stop time. (The minimum input requirement is actually the set of all SMF type 61, 65 and 66 records for the catalog being recovered, covering the interval from the time of the backup through the last catalog update.)

The input data sets containing the SMF records should be concatenated in date/time sequence (oldest first) but keeping the data sets for each system together if possible. It is not necessary to sort the records in strict date/time order (unless they have already been sorted in some other way). Likewise it is not necessary to keep all of the records for one system together. We suggest that you do, only because any messages for that system would be presented consecutively. If your procedures normally sort the SMF records from multiple systems together in date/time sequence (ascending), that input too, is acceptable to the Record Selection and Validation program.

**The SMF Dump Data Sets:** There are two ways of using the SMF dump data sets. You may be using a single data set to receive multiple dumps from SMF recording data sets (by specifying DISP=MOD). Alternately, a new dump data set may be used each time an SMF recording data sets is emptied. Either method produces dump data sets usable as input to the ICFRU.

*Device Types for the SMF Dump Data Sets:* The Record Selection and Validation program accepts SMF input through only one DD statement. Therefore, it is subject to operating system rules for the concatenation of data sets. This means that all of the SMF data to be processed by CRURRSV must reside on the same device type

and all input data sets should have the same block size (to avoid problems with having to put the data set with the largest block size first in the concatenation).

This is likely to be a consideration when dumping the most current SMF data for a recovery situation. Because of the urgency of the catalog recovery, it would be most desirable to use disk data sets and thereby avoid tape mounts.

If your installation maintains some current SMF dump data sets on disk for a period longer than the interval between catalog backups, then the current SMF data sets can be dumped to disk data sets and CRURRSV can run with only disk input.

If any of the SMF data since the last catalog backup is on tape, then you can either dump the current SMF data sets to tape also and run CRURRSV with only tape input, or you can run CRURRSV twice — once with tape input and again with disk input. The “two-run” approach has the distinct disadvantage that the two sets of resulting reports must be analyzed together to see whether all needed SMF data has been included.

For most rapid catalog recovery, we recommend that you:
- Keep at least one day's worth of SMF data “on-line”
- Back up the catalog daily after older SMF data sets have been moved “off-line”

Regardless of the technique used to manage the SMF dump data sets, you should be able to identify, for each system, the data set that contains records just prior to the backup date and time, that is, the recovery start date and time, and all data sets covering the subsequent interval through catalog closing or last update. No problem results from the inclusion of extraneous SMF data sets, but the run time for Record Selection and Validation will be extended by the time needed to pass the additional records.

## Timing Information

The amount of time required for the programs to process SMF data depends on such environmental considerations as the device type on which the SMF data resides, the processor speed, the operating system being used, and the nature of other work being run at the same time. The best estimate for your environment can probably be obtained by looking at run times for existing jobs that process SMF data in similar quantities (such as accounting routines).

We anticipate that CRURRSV will have considerably more data to process than CRURRAP and have therefore provided the following observations for optional use as a general guideline. These observations also show the benefit to be obtained from specifying additional buffers.

The main element in the run time is I/O activity. Therefore, if you are in a recovery situation and your objective is to minimize execution time with little consideration for virtual storage usage, you should specify additional buffers on the BUFNO DCB subparameter in the JCL for SMFIN and SMFOUT (CRURRSV), and for SMFIN, EXPIN, and EXPOUT (CRURRAP). Assuming cylinder allocation for DASD data sets, and availability of sufficient real storage, we suggest buffer specifications which will give around 60 buffers or 500K of buffers, whichever uses less virtual storage, for each of the DDnames above. This should minimize the run time. However, as you can see from the two previous examples, the run time is likely to be fairly short in any case.

The region size information in "Specified Operating Environment" on page 167 assumes default buffering (5 buffers for each of the DDnames above). Increasing the number of buffers may require an increase in the region size.

## Executing Catalog Recovery

This discussion of catalog recovery presumes that you have already completed a diagnosis of the catalog and have confirmed the need to recover the entire catalog (as opposed to recovering selected entries in the catalog).

There are many steps in a full catalog recovery procedure in addition to the actual *forward recovery* steps. To assist you in developing a full procedure for your installation, we will first describe the entire catalog recovery procedure and then, in the next section, describe the *forward recovery* steps involving ICFRU.

### Full Catalog Recovery Procedure

The principal steps required for a full catalog recovery follow below. Some steps will make reference to other, more specialized procedures.

1. List the catalog's aliases from each of the master catalogs to which it is connected. These aliases may have to be re-established later depending on the recovery technique used.
2. Determine whether the catalog is open on each system. Run the Catalog Display program.
3. Attempt to list the catalog from each system to which the catalog is connected using IDCAMS LISTCAT.
4. If LISTCAT is successful from systems that previously did not have the catalog open, it is likely that the control blocks associated with the catalog are in error on the system from which LISTCAT fails. In this case, cause the catalog to be closed and reopened on those systems that cannot access the catalog correctly.
5. If LISTCAT fails on all systems, the catalog will have to be recovered.
6. Deny access to the catalog from all systems except the system to be used for recovery.

   **Note:** In planning for recovery, remember that the recovery system must have access to a catalog (other than the damaged catalog) containing data sets needed for the recovery.

   You might be tempted to use IDCAMS DELETE UCAT with the RECOVERY option at this point. If you do, you will not be able to save, for future reference or diagnostics, a copy the damaged catalog including the most current changes. Additionally, without the following actions, it would be possible for users to begin using the recovered catalog before the diagnostics following recovery could be executed and possible corrective actions taken.

   Denying access may be accomplished by either of the following methods.

   a. Vary the catalog unit offline to prevent I/O to the catalog for the duration of the recovery. This is the preferred method since aliases associated with the catalog will not need to be re-established later as required by the next method. However, this frequently will not be possible because of other allocated data sets on the catalog volume.
   b. Disconnect the catalog from the master catalog. Use IDCAMS EXPORT DISCONNECT. This will also delete the aliases to the catalog. Note that the DISCONNECT command will function even if the catalog is open and its successful completion does not mean that catalog usage will cease.

      If necessary, make sure the catalog is closed after it is disconnected.

7. Deny access to the catalog from non-recovery jobs executing on the system to be used for recovery.

   It may be necessary to “hold” and “dry up” initiators on the recovery system.
8. Cause the catalog to be closed on the recovery system.

   It will be necessary to terminate all address spaces (including TSO) that have referenced the catalog and are causing it to be held open.
9. Record the date and time when it has been confirmed that the catalog is closed on all systems. This is the stop date and time needed as input to the Integrated Catalog Forward Recovery Utility. See “CRURRSV Parameters” on page 180.
10. Switch and dump the SMF data sets on all systems that have had access to the catalog. The SMF records for the catalog will be needed for forward recovery of the catalog. These dump data sets should be the last (by system) in the concatenation of data sets for SMFIN DD to the Record Selection and Validation program.
11. Save a copy of the damaged catalog for future use (for diagnostics, for example). Use DFSMSdss DUMP by data set, tracks or volume. If the catalog's VVR is not usable or is inaccessible because of problems with the VVDS, then the DFSMSdss DUMP by data set will not work.
12. Save the contents of the catalog in a readable format. Using DFSMSdss PRINT by tracks, save both the data and the index components.
13. Save the VVDS containing the catalog.

    Use DFSMSdss PRINT by tracks. If the VVDS is still usable, you can use IDCAMS PRINT.
14. Identify the EXPORT backup copy[1] of the catalog to be used as the basis for recovery. This will be the EXPIN data set for CRURRAP.

    Normally this will be the most current backup. If you use a versioning technique such as the one in Figure 30 on page 212, you can (from batch jobs only) refer to the “zero generation”: for example, Baplicat.CATALOG.BACKUP(0).
15. Establish a starting date and time for forward recovery. These values are needed as execution parameters for the programs in the ICFRU. See “CRURRSV Parameters” on page 180.

    With either IDCAMS EXPORT or DFSMSdss DUMP the first record in the output data set contains the date and time the copy was made. However, there is no utility to extract this information. (It is true that IDCAMS IMPORT will print out the date and time, but we need the information before actually running the IMPORT.)

    You can obtain the date and time in either case using IDCAMS PRINT COUNT(1). You will need to supply DCB information to read the DFSMSdss dump data set; the EXPORT format is VBS.

    Rather than interpreting the dumped data, you may prefer to save the messages from the job creating the copy. This will include the IDC0594I message giving the date and time of the EXPORT copy of the catalog. One technique is to use generation data groups for both the data set copies and the listings, thereby documenting each copy operation in a data set with related name and corresponding generation number.

1. With DFSMSdss or DFSMShsm backups, you will first have to RESTORE or RECOVER the catalog. Then you can create the necessary data set for recovery using IDCAMS EXPORT.

16. In a multi-system installation, determine the maximum difference in the TOD clock values among the systems. **This value is needed as an execution parameter for the programs in the ICFRU**. See "CRURRSV Parameters" on page 180.
17. Identify the SMF data needed for forward recovery of the old version of the catalog. **The concatenation of all these data sets is the SMF input (SMFIN DD) for CRURRSV (Record Selection and Validation).**

    You will need data from all systems having access to the catalog since the recovery start time identified above. Remember to include the recent data preceding the switch of SMF data sets included in the catalog recovery procedure.
18. Determine the interval to be used as the "significant gap time" for CRURRSV. **This value is needed as an execution parameter for the programs in the ICFRU**. The value should be the minimum number of minutes normally needed to fill an SMF recording data set. See "CRURRSV Parameters" on page 180.
19. Execute the CRURRSV program using the parameters and input data sets determined above.

    For details, see "Catalog Forward Recovery Steps with ICFRU" on page 177.
20. Review the condition code, reports and log messages from CRURRSV and determine whether an SMF lost-data condition exists or whether any necessary SMF data sets have been omitted. For guidance in interpreting the results, see "Reports from the Record Selection and Validation Program" on page 192.

    If one or more lost data conditions exists, save the dumped copies of the lost-data records for final error analysis. This record type tells you the period during which SMF records were not written because no SMF recording data set was available. It also tells you the number of SMF records lost. Also save any other messages that may indicate lost data.

    If one or more SMF dump data sets were omitted from the previous CRURRSV execution, run that program again supplying the previously omitted data sets. Be certain to include **all** such data sets, since the program will almost certainly give another set of error messages reflecting the fact that the data sets already processed are now absent and any data sets still missing in this second run will not be detected. The output may be added to the data set previously used (DISP=MOD) or written to a new data set to be concatenated with the previous one in the next sort step.

    **Note:** There is no way to determine conclusively whether all SMF records have been included. However, the program will detect suspiciously long intervals when no SMF records (not just catalog records) were written by a particular system. If SMF data has been truly lost, you may choose to continue with SMF recovery and perform additional forward recovery using other techniques.
21. Using DFSORT or similar facility, sort the output from CRURRSV by data set name (ascending) and date/time sequence (descending).

    For details, see "Sort Control Parameters" on page 182.
22. Execute the CRURRAP (Record Analysis and Processing) program using the parameters previously determined, the output from the previous sort as SMF input (SMFIN DD), and the EXPORT copy identified above as EXPORT input (EXPIN DD).
23. Review the results of CRURRAP for:
    a. Evidence of lost SMF data

b. A list of errors and anomalies to be investigated later

For guidance in interpreting the results, see “Reports from the Record Analysis and Processing Program” on page 196.

If there is evidence of lost SMF data, you should investigate again the results of CRURRSV to confirm whether data has, in fact, been lost. If so, attempt to identify the interval surrounding the lost data condition.

When you are satisfied with the results of CRURRAP, proceed as follows.

24. Use IDCAMS to DELETE the catalog for RECOVERY.

    This will prevent IMPORT failures due to the inability to perform integrity checking of a damaged catalog before internally deleting and redefining it as a part of IMPORT processing.

    This deletion will result in the removal of all associated aliases and the need to rebuild them later. You may wish to make a list of the aliases before deleting the related user catalog and then use the list to restore the aliases after IMPORTing the EXPORT copy.

25. Optionally, re-DEFINE the catalog using IDCAMS.

    Do this if you wish to preserve a current control area size that is less than one cylinder or if you choose to change the structure of the catalog in any way. Note that, with this procedure, the recovery must be to the same volume serial number and the same device type. Otherwise, you may use a different volume. Do not decrease the current maximum logical record size.

26. IMPORT the EXPORT copy produced by CRURRAP.

    Use the INTOEMPTY parameter if you have already redefined the catalog. Use the VOLUME sub-parameter if you are changing volume serial numbers.

    Note that the date and time of the EXPORTed copy given by message IDC0604I will be the date and time you specified as the stop time for the recovery.

    If the IMPORT is without the INTOEMPTY parameter, an existing copy of the catalog will be internally deleted (for recovery) and redefined. The deletion will result in the removal of all associated aliases and the need to rebuild them later.

27. Now that forward recovery is complete, again check the status of the catalog using IDCAMS DIAGNOSE, LISTCAT and EXAMINE.

    These diagnostics should not be steps within the job that recovered the catalog; the recovered catalog should be newly opened.

28. For any entries noted with error or anomaly messages from CRURRAP, make sure that the catalog entry now present represents reality.

    For VSAM entries, IDCAMS DIAGNOSE with the COMPARE option will perform the necessary checking, so you should review the results from the previous step.

    For questionable non-VSAM entries, you will have to resort to other methods. If the device type is DASD, check the VTOC for the indicated data set. If it is not present, remove the catalog entry. If the device type is for tape, check the tape management inventory (if you have one). With a small number of tape data sets to be investigated, it may also be possible to examine the tape volumes. If the data set is not on the indicated volume, remove the catalog entry.

29. If needed SMF data was lost, you may want to attempt to resolve discrepancies arising from the loss at this point (as opposed to waiting for users or production jobs to encounter a problem).

30. Rebuild the aliases for the catalog if the catalog was imported to a new volume or if the aliases have been deleted during the recovery.
31. If VSAM passwords are in use and if any were reset during the recovery (indicated by message CRU116I), security for these data sets should be redefined at this point. This may be done be security administration or by the data set owners.
32. Backup the catalog to start a new recovery cycle.

    Use IDCAMS EXPORT or use DFSMSdss DUMP by data set or other standard DFSMSdss dump procedure. Even if no corrections to the recovered catalog were necessary, you may want to perform the backup to make sure that the new backup is properly cataloged and tracked.
33. Restore access to the catalog from all other systems by either varying the catalog unit back online and mounting the catalog volume or reconnecting the catalog and redefining the aliases using IDCAMS IMPORT CONNECT and DEFINE ALIAS and by releasing any initiators held for the recovery process.

This completes the full catalog recovery procedure.

## Catalog Forward Recovery Steps with ICFRU

Since ICFRU program execution is not very visible in the rather lengthy catalog recovery procedure in the preceding section, this section presents only the steps that involved ICFRU programs.

The jobs below could be run as steps within a single job. There is no harm in executing all of the steps in this way since you can rerun them independently. However, this means that you should not use temporary data sets to pass between the job steps. It is especially important that you save the output data set from CRURRSV if there was a lot of SMF data that had to be read as input. The execution jobs or steps are shown in the following figures.

```
//*******************************************************
//* INTEGRATED CATALOG FORWARD RECOVERY UTILITY         *
//* JCL EXAMPLE -                                       *
//* THIS JCL EXECUTES RECORD SELECTION AND VALIDATION.  *
//* MODIFY THE STEPLIB DD STATEMENTS TO POINT TO THE    *
//* INSTALLATION LOAD MODULE LIBRARY CONTAINING CRURRSV. *
//* MODIFY THE SMFIN DD STATEMENT TO POINT TO THE       *
//* SMF INPUT DATA SETS.                                *
//* MODIFY THE SMFOUT DD STATEMENT TO CONFORM TO YOUR   *
//* INSTALLATION CONVENTIONS.                           *
//* SMFOUT MAY BE PASSED TO THE SORT STEP.              *
//* SPECIFY PARAMETERS APPROPRIATE TO THIS RECOVERY.    *
//*******************************************************
//* RUN CRURRSV TO EXTRACT APPROPRIATE DATA.
//*******************************************************
//RRSV EXEC PGM=CRURRSV,REGION=1024K,
// PARM=('catalog.name',
//       'mm/dd/yy','hh:mm:ss',
//       'mm/dd/yy','hh:mm:ss',
//       'mmmm',
//       'ssss')
//*PARM=('CATALOG.NAME',
//*      'STARTDATE','STARTTIME',
//*      'STOPDATE','STOPTIME',
//*      'GAPTIME',
//*      'CLOCKDIFFERENCE')
//STEPLIB DD DSN=USER.LOAD,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLOG DD SYSOUT=*
//SMFIN DD DSN=system1.smfdump(-n),DISP=SHR,DCB=BUFNO=60
//      DD DSN=system1.smfdump(-1),DISP=SHR,DCB=BUFNO=60
//      DD DSN=system1.smfdump(0),DISP=SHR,DCB=BUFNO=60
//      DD DSN=system2.smfdump(-n),DISP=SHR,DCB=BUFNO=60
//      DD DSN=system2.smfdump(-1),DISP=SHR,DCB=BUFNO=60
//      DD DSN=system2.smfdump(0),DISP=SHR,DCB=BUFNO=60
//SMFOUT DD DSN=SMF.CAT.RECS,DISP=(NEW,CATLG),
// UNIT=SYSDA,SPACE=(CYL,(10,1),RLSE),DCB=BUFNO=60
```

*Figure 12. Executing CRURRSV.* Use the skeleton JCL shown here for running Record Selection and Validation The output must next be sorted.

```
//*******************************************************
//* INTEGRATED CATALOG FORWARD RECOVERY UTILITY         *
//* JCL EXAMPLE -                                       *
//* THIS JCL EXECUTES THE NECESSARY SORT STEP.          *
//* THE SORTIN MAY BE PASSED FROM THE CRURRSV STEP.     *
//* MODIFY THE SORTOUT DD STATEMENT TO CONFORM TO YOUR  *
//* INSTALLATION CONVENTIONS.                           *
//* THE SORTOUT MAY BE PASSED TO THE CRURRAP STEP.      *
//* MODIFY THE SORT STEP AS APPROPRIATE, TO INVOKE      *
//* THE SORT PRODUCT AVAILABLE IN YOUR INSTALLATION     *
//* (THIS EXAMPLE INVOKES DFSORT.                       *
//*******************************************************
//* SORT THE OUTPUT FROM CRURRSV.
//*******************************************************
//SORT     EXEC PGM=ICEMAN
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DSN=SMF.CAT.RECS,DISP=SHR
//*        DD DSN=concatenations if necessary
//SORTOUT  DD DSN=SMF.SORTED.CAT.RECS,DISP=(NEW,CATLG),
// UNIT=SYSDA,SPACE=(CYL,(10,1),RLSE)
//SYSIN    DD *
OPTION DYNALLOC=SYSDA,FILSZ=E10000,EQUALS
 SORT FIELDS=(218,44,CH,A,262,1,BI,A,11,4,PD,D,7,4,BI,D)
/*
```

*Figure 13. Executing the Sort.* Use the skeleton JCL and sort control shown here for sorting the SMF records from CRURRSV. The sorted records will be processed by CRURRAP.

```
//*******************************************************
//* INTEGRATED CATALOG FORWARD RECOVERY UTILITY         *
//* JCL EXAMPLE -                                       *
//* THIS JCL EXECUTES RECORD SELECTION AND PROCESSING.  *
//* MODIFY THE STEPLIB DD STATEMENTS TO POINT TO THE    *
//* INSTALLATION LOAD MODULE LIBRARY CONTAINING CRURRAP. *
//* THE SMFIN DD STATEMENT MAY SPECIFY A DATA SET       *
//* PASSED FROM THE SORT STEP.                          *
//* MODIFY THE EXPIN DD STATEMENT TO POINT TO           *
//* THE EXPORT DATA SET TO BE USED AS THE RECOVERY BASE. *
//* MODIFY THE EXPOUT DD STATEMENT TO CONFORM TO YOUR   *
//* INSTALLATION CONVENTIONS.                           *
//* SPECIFY PARAMETERS APPROPRIATE TO THIS RECOVERY:    *
//* THEY SHOULD BE THE SAME AS THE ONES USED IN CRURRSV. *
//*******************************************************
//* RUN CRURRAP USING OUTPUT FROM SORT.
//*******************************************************
//RRAP EXEC PGM=CRURRAP,REGION=1536K,
// PARM=('catalog.name',
//       'mm/dd/yy','hh:mm:ss',
//       'mm/dd/yy','hh:mm:ss',
//       'mmmm',
//       'ssss')
//*PARM=('CATALOG.NAME',
//*      'STARTDATE','STARTTIME',
//*      'STOPDATE','STOPTIME',
//*      'GAPTIME',
//*      'CLOCKDIFFERENCE')
//STEPLIB DD DSN=USER.LOAD,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSLOG DD SYSOUT=*
//SMFIN    DD DSN=SMF.SORTED.CAT.RECS,DISP=SHR,DCB=BUFNO=60
//EXPIN    DD DSN=Baplicat.CATALOG.BACKUP(0),DISP=SHR,
//            DCB=BUFNO=60
//EXPOUT DD DSN=Baplicat.NEW.CATALOG.EXPORT,DISP=(NEW,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(10,1),RLSE),DCB=BUFNO=60
```

*Figure 14. Executing CRURRAP.* Use the skeleton JCL shown here for running CRURRAP. The data set produced will be used as input to an IDCAMS IMPORT.

# Syntax Reference for Executing ICFRU

## Execution Parameters

### CRURRSV Parameters

Execution parameters are provided on the PARM parameter of the EXEC statement. There are six required parameters and one optional parameter. An example showing all seven parameters is in Figure 15.

```
//S1 EXEC PGM=CRURRSV,
// PARM='UCAT.BROKEN,12/05/04,14:32:58,12/06/04,19:46:27,60,2'
```

*Figure 15. CRURRSV Execution Parameters - Example 1*

For ease of specification and alteration, you may wish to split the individual subparameters across multiple statements. An example of this is in Figure 16 on page 181.

```
//S1 EXEC PGM=CRURRSV,
// PARM=('UCAT.BROKEN',
//       '12/05/04','01:32:58',
//       '12/06/04','19:46:27',
//        60,
//         2)
```

*Figure 16. CRURRSV Execution Parameters - Example 2*

The individual subparameters are described below.

1. Name of catalog to be recovered, or '**' (all catalogs)
   - Length - 1 - 44 characters
   - Implications - Used as basis for SMF record selection. If all catalogs are indicated ('**'), then all type 61/65/66 records that fall within the start/stop range will be selected. This allows CRURRSV to be used to separate the SMF data required for catalog recovery on a periodic basis.
   - Restrictions - If a single catalogs is indicated, this must be the name specified when the catalog was DEFINEd, not an alias.

   This parameter is required.
2. Start date (Julian or Gregorian format)
   - Length - 6 characters (Julian) or 8 characters (Gregorian)
   - Format - YY.DDD (Julian - year.daynumber) or MM/DD/YY (Gregorian - month/day/year)
   - Implications - Used to determine the starting point for SMF record selection.
   - Restrictions - Leading zeros must be specified (for example, 12/05/04, not 12/5/04). To ensure correct synchronization of EXPORT and SMF data, this date should match the date of the EXPORT to be used as input to CRURRAP, if both programs are being run in series for recovery purposes. The EXPORT date will be found on the EXPORT SYSPRINT output, at the top of the first page. If a program other than EXPORT was used for backup, use the date that backup was taken. Such a procedure requires that the non-EXPORT backup be restored, and an EXPORT be done from that restored copy of the catalog. Hence the date of the EXPORT is not the correct starting point for use of SMF data. Expect message CRU104I when the start date/time do not match the EXPORT date/time. If CRURRSV is being used only to extract SMF data, any date may be specified.

   This parameter is required.
3. Start time
   - Length - 8 characters
   - Format - HH:MM:SS (hours:minutes:seconds)
   - Implications and Restrictions are the same as for start date. Leading zeros are also required for start time (for example, 01:32:58, not 1:32:58).

   This parameter is required.
4. Stop date (Julian or Gregorian format)
   - Length - 6 characters (Julian) or 8 characters (Gregorian)
   - Format - YY.DDD (Julian - year.daynumber) or MM/DD/YY (Gregorian - month/day/year)
   - Implications - Used to determine the stopping point for SMF record selection.
   - Restrictions - Leading zeros must be specified (for example, 12/05/04, not 12/5/04). To ensure that all required SMF data is processed, this should be the date on which the recovery is being done, or the date on which access to the catalog was stopped.

   This parameter is required.
5. Stop time
   - Length - 8 characters

- Format - HH:MM:SS (hours:minutes:seconds)
- Implications and Restrictions are the same as for stop date. Leading zeros are also required for stop time (for example, 01:32:58, not 1:32:58).

  This parameter is required.

6. Maximum acceptable gap between SMF records, in minutes
   - Length - 1 to 4 numeric characters
   - Format - MMMM
   - Implications - Used to trigger warning messages about possible missing SMF data. Specify a value slightly smaller than the usual time to fill an SMF data set. If you do not know what this value is, try specifying 30 minutes and check CRURRSV output for SWITCH SMF records, which will indicate the dates and times at which switches were actually done.
   - Restrictions - Leading zeros may be specified (for example, '0004') or omitted (for example, '4'). Specification of zero will cause the time between any two SMF records to be considered an unacceptable gap, and will create many warning messages.

     This parameter is required.
7. Maximum clock difference (multi-system operation only), in seconds
   - Length - 1 to 4 numeric characters
   - Format - SSSS
   - Implications - When the catalog to be recovered may have been updated by more than one system, specification of this parameter causes adjustment of the effective start and stop times to allow for potential differences in the system clocks on the involved systems. A clock difference value of '2', for example, would cause 2 seconds to be subtracted from the start time and added to the stop time to calculate effective start/stop times. These adjusted times are then used as the basis for SMF record selection.
   - Restrictions - Crossing midnight during this adjustment is not supported. Do not specify a start time which, if the clock difference is subtracted from it, would generate an effective start time on a different day than the specified start time. Likewise do not specify a stop time which, if the clock difference is added to it, would generate an effective stop time on a different day than the specified stop time. For example, a start time of 00:00:01 and a clock difference of 5 seconds would violate this restriction. If it is absolutely necessary to bypass this problem, specify a start time just before midnight (23:59:59) and the earlier start date, or a stop time just after midnight (00:00:01) and the later date, depending on whether your problem is with the start or stop adjustment.

     This parameter is required for multi-system operation, where the catalog to be recovered could have been updated by more than one system. For a single-system environment it can be omitted but should be specified as zero.

## Sort Control Parameters

The parameters given in this section assume that you are using z/OS DFSORT. However, the explanation should enable you to specify the necessary parameters for any other functionally equivalent sort program.

The output data set from Record Selection and Validation must be sorted as follows:

1. By data set name (ascending). The data set name begins at location 218[2]. It is 44 (alphameric or special) characters in length.

2. Locations are relative to 1 (not zero) and include the 4 bytes of length information for the variable length records from SMF.

2. By data set name extension number (ascending). The data set name extension is one binary byte at location 262[2].
3. By the date of the SMF record (descending) . The SMF date field is packed decimal, 4 bytes at location 11[2].
4. By the time of the SMF record (descending) . The SMF time field is 4 binary bytes at location 7[2].

   Note that because this field identifies the time to a granularity of 0.01 seconds, duplicate keys are possible. In order to maintain the original order of SMF records when the system encounters duplicate keys, you must specify the EQUALS sort parameter. However, if SMF records with matching time stamps were generated from different systems, ICFRU cannot know the proper order to apply these updates. This condition is indicated by the new message CRU117I, stating that ICFRU has encountered two or more SMF records from different systems referencing the same catalog entry. This message warns the user that the SMF records for the indicated Catalog entry may not have been applied in the correct order and that the user should verify the results.

The specification for DFSORT is as follows:

```
SORT FIELDS=(218,44,CH,A,262,1,BI,A,11,4,PD,D,7,4,BI,D)
```

Other parameters may be used to optimize the performance of DFSORT. In particular, CRURRSV will count the number of SMF records selected and this count can then be supplied to DFSORT.

## CRURRAP Parameters

Execution parameters are provided with the PARM parameter on the EXEC statement. There are six required parameters and one optional parameter. An example showing all seven parameters is in Figure 17.

```
//S1 EXEC PGM=CRURRAP,
// PARM='UCAT.BROKEN,12/05/04,14:32:58,12/06/04,19:46:27,60,2'
```

*Figure 17. CRURRAP execution parameters*

For ease of specification and alteration, you may wish to split the individual subparameters across multiple statements. An example of this is in Figure 18.

```
//S1 EXEC PGM=CRURRAP,
// PARM=('UCAT.BROKEN',
//       '12/05/04','01:32:58',
//       '12/06/04','19:46:27',
//        60,
//         2)
```

*Figure 18. CRURRAP Execution Parameters - Example 2*

If you are using CRURRSV and CRURRAP together to do catalog recovery, you should specify the same parameters for both programs.

The individual subparameters are described below.

1. Name of catalog to be recovered
   - Length - 1 - 44 characters
   - Implications - Used as basis for SMF record selection.
   - Restrictions - This must be the name specified when the catalog was DEFINEd, not an alias. '**' (all catalogs) is not permitted by CRURRAP.

     This parameter is required.
2. Start date (Julian or Gregorian format)

- Length - 6 characters (Julian) or 8 characters (Gregorian)
- Format - YY.DDD (Julian - year.daynumber) or MM/DD/YY (Gregorian - month/day/year)
- Implications - Used to determine the starting point for SMF record selection.
- Restrictions - Leading zeros must be specified (for example, 12/05/04, not 12/5/04). To ensure correct synchronization of EXPORT and SMF data, this date should match the date of the EXPORT to be used as input to CRURRAP. If a program other than EXPORT was used for backup, use the date that backup was taken. Such a procedure requires that the non-EXPORT backup be restored, and an EXPORT be done from that restored copy of the catalog. Hence the date of the EXPORT is not the correct starting point for use of SMF data. Expect message CRU104I when the start date/time do not match the EXPORT date/time. The EXPORT date will be found on the EXPORT SYSPRINT output, at the top of the first page.

  This parameter is required.

3. Start time
   - Length - 8 characters
   - Format - HH:MM:SS (hours:minutes:seconds)
   - Implications and Restrictions are the same as for start date. Leading zeros are also required for start time (for example, 01:32:58, not 1:32:58).

     This parameter is required.
4. Stop date (Julian or Gregorian format)
   - Length - 6 characters (Julian) or 8 characters (Gregorian)
   - Format - YY.DDD (Julian - year.daynumber) or MM/DD/YY (Gregorian - month/day/year)
   - Implications - Used to determine the stopping point for SMF record selection.
   - Restrictions - Leading zeros must be specified (for example, 12/05/04, not 12/5/04). To ensure that all required SMF data is processed, this should be the date on which the recovery is being done, or the date on which access to the catalog was stopped.

     This parameter is required.
5. Stop time
   - Length - 8 characters
   - Format - HH:MM:SS (hours:minutes:seconds)
   - Implications and Restrictions are the same as for stop date. Leading zeros are also required for stop time (for example, 01:32:58, not 1:32:58).

     This parameter is required.
6. Maximum acceptable gap between SMF records, in minutes
   - Length - 1 to 4 numeric characters
   - Format - MMMM
   - Implications - None, for CRURRAP. This parameter is used by CRURRSV only but is accepted by CRURRAP for compatibility purposes.
   - Restrictions - Leading zeros may be specified (for example, '0004') or omitted (for example, '4').

     This parameter is required.
7. Maximum clock difference (multi-system operation only), in seconds
   - Length - 1 to 4 numeric characters
   - Format - SSSS
   - Implications - When the catalog to be recovered may have been updated by more than one system, specification of this parameter causes adjustment of the effective start and stop times to allow for potential differences in the system clocks on the involved systems. A clock difference value of '2', for example, would cause 2 seconds to be subtracted from the start time and

added to the stop time to calculate effective start/stop times. These adjusted times are then used as the basis for SMF record selection.

In addition, CRURRAP also uses the clock difference when examining catalog changes for the same entry within the clock difference interval. For example, if the clock difference is specified as 3 seconds and two SMF records indicate two catalog changes for the catalog entry for data set ABC less than 3 seconds apart and from different systems, then it is not possible to be certain of the order in which these changes really occurred. Messages will be written to SYSLOG when such conditions are found, providing more information about the condition and the action taken. If this parameter is omitted or specified as zero, no start/stop adjustment is done and the synchronization checking just described is also not done.

- Restrictions - Crossing midnight during this adjustment is not supported. Do not specify a start time which, if the clock difference is subtracted from it, would generate an effective start time on a different day than the specified start time. Likewise do not specify a stop time which, if the clock difference is added to it, would generate an effective stop time on a different day than the specified stop time. For example, a start time of 00:00:01 and a clock difference of 5 seconds would violate this restriction. If it is absolutely necessary to bypass this problem, specify a start time just before midnight (23:59:59) and the earlier start date, or a stop time just after midnight (00:00:01) and the later date, depending on whether your problem is with the start or stop adjustment.

  This parameter is required for multi-system operation, where the catalog to be recovered could have been updated by more than one system. For a single-system environment it can be omitted or specified as zero.

## Execution JCL Statements

This section describes the JCL required for the programs in the ICFRU.

### JCL for CRURRSV

See Figure 12 on page 178 for an example of JCL for CRURRSV. For information on execution parameters on the EXEC statement, see the section "CRURRSV Parameters" on page 180. The DD statements are described below.

- SYSPRINT - the output data set for reports, normally SYSOUT. LRECL, BLKSIZE, and RECFM DCB information is provided by CRURRSV and will therefore be ignored if provided on the DD statement. You may specify additional buffers with the BUFNO DCB subparameter in the JCL if you wish, but in view of the small amount of output this does not appear to be worthwhile.
- SYSLOG - the output data set for log messages, normally SYSOUT. DCB information is provided by CRURRSV and will be ignored if provided on the DD statement. You may specify additional buffers with the BUFNO DCB parameter in the JCL if you wish, but in most cases you are unlikely to get enough SYSLOG output to make this worthwhile.
- SMFIN - the SMF input data set. No LRECL/BLKSIZE/RECFM information is provided by the program. You do not need to provide DCB information in the JCL unless this data set is a non-labelled tape. In other cases, (standard labelled tape, DASD data set) the DCB information will be acquired from the data set label. Performance can be improved when large amounts of SMF data are processed by provision of additional buffers (with the BUFNO DCB subparameter). The default is 5 buffers. Virtual storage requirements for additional buffers must be added to the region size requirements for the program (see "Specified Operating Environment" on page 167).

- SMFOUT - the SMF output data set. The program copies the LRECL, BLKSIZE, and RECFM information for the SMFIN data set to the SMFOUT data set's DCB. Therefore if these DCB subparameters are specified in the JCL they will be ignored. Performance can be improved when large amounts of SMF data are processed by provision of additional buffers (with the BUFNO DCB subparameter). The default is 5 buffers. Virtual storage requirements for additional buffers must be added to the region size requirements for the program (see "Specified Operating Environment" on page 167).

## JCL for Sort

This section assumes that you are using z/OS DFSORT. However, the explanation should enable you to specify the necessary JCL for any other functionally equivalent sort program. See Figure 13 on page 179 for an example.

**SORTIN::** The data to be sorted is created by Record Selection and Validation and placed in the data set referred to by the SMFOUT DD statement. The DCB attributes of this data set were copied from the input data set supplied to CRURRSV by the SMFIN DD. (See "JCL for CRURRSV" on page 185.) Therefore, the DCB attributes of the data sets to be sorted will match those of the SMF dump data sets that you originally supplied. Multiple data sets may be concatenated to the SORTIN DD statement for DFSORT if multiple runs have been made with CRURRSV.

**SORTOUT::** The sorted data from DFSORT is placed in the data set identified by the SORTOUT DD statement. The DCB attributes of this data set may be copied from the input data set supplied to DFSORT by the SORTIN DD statement. In this case, the DCB attributes of the output data set will match those of the SMF dump data sets that you originally supplied. The SORTOUT data set will be used as input by CRURRAP to be read using QSAM. CRURRAP has no special DCB attribute requirements. See "JCL for CRURRAP."

## JCL for CRURRAP

See Figure 14 on page 180 for an example of JCL for CRURRSV. For information on execution parameters on the EXEC statement, see the section "CRURRAP Parameters" on page 183. The DD statements are described below.

- SYSPRINT - the output data set for reports, normally SYSOUT. LRECL, BLKSIZE, and RECFM DCB information is provided by CRURRSV and will therefore be ignored if provided on the DD statement. You may specify additional buffers with the BUFNO DCB subparameter in the JCL if you wish, but in view of the small amount of output this does not appear to be worthwhile.
- SYSLOG - the output data set for log messages, normally SYSOUT. DCB information is provided by CRURRSV and will be ignored if provided on the DD statement. You may specify additional buffers with the BUFNO DCB parameter in the JCL if you wish, but in most cases you are unlikely to get enough SYSLOG output to make this worthwhile.
- SMFIN - the SMF input data set. No LRECL/BLKSIZE/RECFM information is provided by the program. You do not need to provide DCB information in the JCL unless this data set is a non-labelled tape. In other cases, (standard labelled tape, DASD data set) the DCB information will be acquired from the data set label. Performance can be improved when large amounts of SMF data are processed by provision of additional buffers (with the BUFNO DCB subparameter). The default is 5 buffers. Virtual storage requirements for additional buffers must be added to the region size requirements for the program (see "Specified Operating Environment" on page 167).

- EXPIN - the EXPORT input data set. No LRECL/BLKSIZE/RECFM information is provided by the program. You do not need to provide DCB information in the JCL unless this data set is a non-labelled tape. In other cases, (standard labelled tape, DASD data set) the DCB information will be acquired from the data set label. Performance can be improved when large amounts of EXPORT data are processed by provision of additional buffers (with the BUFNO DCB subparameter). The default is 5 buffers. Virtual storage requirements for additional buffers must be added to the region size requirements for the program (see "Specified Operating Environment" on page 167).
- EXPOUT - the EXPORT output data set. The program copies the LRECL, BLKSIZE, and RECFM information for the EXPIN data set to the EXPOUT data set's DCB. Therefore if these DCB subparameters are specified in the JCL they will be ignored. Performance can be improved when large amounts of EXPORT data are processed by provision of additional buffers (with the BUFNO DCB subparameter). The default is 5 buffers.

# Codes Used by ICFRU

## ABENDs

As a general rule, the programs will not intentionally ABEND unless a suspect program logic error is encountered that could make further processing risky. The ABEND codes used are shown below with their meanings.

1. U0001 - An SMF record is being processed that is not a type 61/65/66 at a place in the logic where other record types should already have been excluded.

   Issued by module CRURRSV, CSECT CRURRSV.
2. U0002 - Field SMFxxSUB in a type 61/65/66 SMF record is neither 'IN', 'DE', or 'UP'. This field should have been either supplied by catalog management in the SMF record, or 'fabricated' by CRURRAP if missing. There is no way to determine the correct disposition of the record without this information. It is not sufficient to merely check whether the record in question is a DELETE, ALTER, OR DEFINE record. The pertinent question is which VSAM record management function is represented by this SMF record: an insert of a new record, an update of an existing record, or a delete. The record type (DEFINE, DELETE, ALTER) is merely an indication of the type of request to catalog management. For example, an IDCAMS DELETE could cause catalog record updates as well as deletions.

   Issued by module CRURRAP, CSECT CRURRAP.
3. U0003 - A calculated sequence value used to index a table of sequence values and sequence error values is greater than 16, the maximum acceptable value.

   Issued by module CRURRAP, CSECT CRURRAP.
4. U0004 - One of the following errors occurred during logging:
   a. Calculated message table offset greater than 32, the maximum acceptable value.
   b. SMF record to be logged is not type 61/65/66.
   c. Calculated action message table offset greater than 17, the maximum acceptable value.
   d. Index value not found in action message table.

   Issued by module CRURRAP, CSECT CRURRAP.
5. U0005 - The program was unable to locate the data name cell in a cluster record during 'fabrication' of the subtype field (SMFxxSUB). This prevented determination of the correct disposition for the record. See the discussion of

ABEND U0002 above for more on the importance of this field. This problem could be caused by either a logic error in the program or a change in the format of catalog records.

Issued by module CRURRAP, CSECT CRURRAP.

## Condition Codes and Return Codes

### Condition/Return Codes from CRURRSV

The condition code reported by CRURRSV is the maximum condition code encountered for any situation during the execution of the program. This maximum condition code will be set as the step completion code (also called the return code) so that the execution of subsequent steps in the same job can be contingent on the success of CRURRSV.

**Condition Code Zero:** is set by CRURRSV if the events represented by the SMF input records show no evidence of lost or omitted SMF data. The reports should still be reviewed for completeness and reasonableness.

**Condition Code Four:** is set by CRURRSV to indicate the presence of events that may accompany lost SMF data, even though no direct evidence of lost or missing SMF data is recognized. Review the log data set for messages associated with this condition, CRU1*nn*I.

**Condition Code Eight:** is set by CRURRSV to indicate direct evidence of lost or missing SMF data, that is, an SMF lost-data record or a gap in SMF record timestamps from a single system. Review the log data set for messages associated with this condition, CRU2*nn*I.

**Condition Code Twelve:** from CRURRSV indicates that either a non-SMF record was found in the input data set or that the maximum number of system identifiers (16) has been exceeded. Program execution terminates but reports for the records processed to that point are generated to assist you in determining the source of the problem. A log message, CRU3*nn*I, identifies the error. Supply the proper input data sets and rerun the job. If more than sixteen systems actually had access to the catalog (for example, because test systems were used), you may run Record Selection and Validation multiple times and concatenate the input to the subsequent sort step or job.

**Condition Code Sixteen:** is used by CRURRSV to indicate an error in parameter specification or an input data set that is empty or contains no qualifying records. Program execution terminates and no reports are produced. In the log data look for a message of the form CRU4*nn*I to explain this condition code. Correct the parameters or the input data (or both) and rerun CRURRSV.

### Condition/Return Codes from CRURRAP

The condition code reported by CRURRAP is the maximum condition code encountered for any situation during the execution of the program. This maximum condition code will be set as the step completion code (also called the return code) so that the execution of subsequent steps in the same job can be contingent on the success of CRURRAP.

**Condition Code Zero:** is set by CRURRAP if the events represented by the SMF input records and the records from the EXPORTed copy:

- Form logical sequences of changes to the catalog

- Are not ambiguous with respect to the order of the last updates for a single data set from multiple systems (Ambiguities may still exist with respect to the order of intermediate updates.)
- Require no user action to make the data sets accessible as they were at the point of failure.

The reports should still be reviewed for completeness and reasonableness.

**Condition Code Four:** is set by CRURRAP to indicate the presence of events that:
- Form illogical sequences of changes to the catalog when the the most current record for a data set is not involved
- Are ambiguous with respect to the order of the last update for a single data set from multiple systems
- Require user action to reset the VSAM security information as it was at the point of failure.
- Indicate a questionable specification of either the clock difference parameter or the recovery start date and time.

All of these conditions should be investigated and resolved based on information from the following ANOMALY REPORT and the log data set for messages CRU1*nn*I associated with this condition.

**Condition Code Eight:** is set by CRURRAP if the events represented by the SMF input records and the records from the EXPORTed copy form illogical sequences of changes to the catalog. Review the following ERROR REPORT and the log data set for messages CRU2*nn*I associated with this condition.

**Condition Code Twelve:** is used by CRURRAP to indicate a problem with one of the records from the EXPORT copy being used as input. Review the following ERROR REPORT and the log data set for messages CRU3*nn*I associated with this condition. Program execution continues only because this might be the last or only available copy of the catalog. Use the resulting output data set only as a last resort. It almost certainly contains errors that must be manually detected and resolved.

**Condition Code Sixteen:** is used by CRURRAP to indicate a problem in parameter specification, input data or the combination of parameters and input data. Program execution terminates and no reports are produced. In the log data look for a message of the form CRU4*nn*I to explain this condition code. Correct the parameters or the input data (or both) and rerun CRURRAP.

## ICFRU Messages

All messages from the ICFRU have one format: **CRUnnnI text**. CRU is the single message prefix used by both CRURRSV, Record Selection and Validation, and CRURRAP, Record Analysis and Processing. All messages are informational as indicated by the "**I**" in the final position; no operator action is required.

The first digit of the message serial number, nnn, indicates the severity of the condition described:

**CRU0*nn*I**
: Condition code 0, informational message; The condition will not affect the results.

**CRU1*nn*I**
: Condition code 4, warning message; The condition will not directly affect the results, but the situation should be examined for possible indirect effects on the output.

**CRU2*nn*I**
Condition code 8, error message; The condition directly affects on the results. Some records or catalog entries may be in error. The records or entries must be diagnosed for errors. Manual corrections may be required.

**CRU3*nn*I**
Condition code 12, serious error message, CRURRSV reports and terminates. CRURRAP continues but the resulting output should be used only if absolutely no other alternative is available. Extensive diagnosis will be required if the output data is used. Many entries may have to be corrected manually.

**CRU4*nn*I**
Condition code 16, terminating error message, The program cannot continue with the parameters and input supplied.

All messages are written to the SYSLOG data set in the order in which the events are encountered. Many messages are multiple-line messages, almost like small reports, supplying enough information so that you should need to refer to this messages section only infrequently. The explanatory text and user response are very similar for many messages. However, the complete text and response is given with each message to avoid the need for references to other messages.

ICFRU messages are documented in *z/OS MVS System Messages, Vol 4 (CBD-DMO)*.

## Message Logs from ICFRU

Each program of the ICFRU has a message log data set that is written to SYSLOG DD. This log is separate from the "reports" data set to allow you flexibility in viewing and printing the reports and logs separately. Message log pages are handled only by your Job Entry Subsystem if the log is actually printed. We anticipate that most reviewing of the log will be from a terminal. Therefore the messages are constructed to fit (generally) within 80 columns. The dumps will be wider than this, which means that the interpreted portions will be off the right side of the screen and you must scroll to them.

### The Message Log for CRURRSV

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRSV SYSLOG       12/17/04 (04.351)  10:39:44
```

The message log from Record Selection and Validation begins with these lines, identifying first the name, release and modification level numbers of the product writing the reports. The second line identifies the program by name and the DD name being used. The execution date is given in both Gregorian and Julian formats followed by the time at which program execution began.

If there is an error in start up processing, the error will be logged (but no reports are produced). Thus the program uses the log data set to communicate errors in input parameters or input data sets.

Events are logged sequentially as they are encountered. If all SMF data sets for one system are presented together, then all messages for that system's data will appear together.

The principal use of the log is to determine whether any SMF data has been omitted or lost. Therefore, gaps in SMF recording from a single system will be

logged as will "data lost" SMF records. To assist you in checking out gaps, the SMF SWITCH, HALT EOD, and IPL records are also logged.

The record that is being processed when one of these events is encountered will also be dumped. The dump will contain the relative addresses on the left side, the storage locations from which the record was dumped, the dumped data in hexadecimal and, on the right side, an EBCDIC interpretation of the dumped data.

## The Message Log for CRURRAP

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRAP SYSLOG       12/17/04 (04.351)  10:39:44
```

The message log from Record Analysis and Processing begins with these lines, identifying first the name, release and modification level numbers of the product writing the reports. The second line identifies the program by name and the DD name being used. The execution date is given in both Gregorian and Julian formats followed by the time at which program execution began.

If there is an error in start up processing, the error will be logged (but no reports are produced). Thus the program uses the log data set to communicate errors in input parameters or input data sets.

Events are logged sequentially as they are encountered. Thus all messages for a given catalog entry will presented sequentially with the newest record first.

The principal use of the log is to determine whether any SMF data has been omitted or lost or is out of sequence. All records for a catalog entry are examined in pairs. If the pair of events could not have happened in that order, a sequence error is recognized. A message is then produced and the records involved are dumped.

If two systems update the same catalog record within the user-specified clock difference interval, a synchronization check is recognized. A message is produced only if the check involves the most current record and both records of the pair are dumped.

When either a sequence error or a synchronization check involves the most current record for an entry, then all events for that entry are analyzed and logged with appropriate messages. The accompanying dumps mean that each record after the newest one will be dumped twice, once for each pair to which it belongs.

The dump will contain the relative addresses on the left side, the storage locations from which the record was dumped, the dumped data in hexadecimal and, on the right side, an EBCDIC interpretation of the dumped data.

# Reports from ICFRU

The ICFRU produces a number of reports designed to tell you:
- What execution parameters you specified
- What condition code resulted from program execution
- About any error situations encountered
- About any unexplained or unusual events
- A summary of correct processing
- Various totals you may cross-check to verify the results.

There is considerable detailed information contained in the reports. We anticipate that this program may be run very infrequently and perhaps in stressful conditions, so we have included just about everything we expected you might need. Again because it may be a long time since you have studied the reports, this section is quite detailed and structured to lead you through an analysis of the reports.

## Reports from the Record Selection and Validation Program

For complete examples of the reports from Record Selection and Validation, see Figure 19 on page 204 and Figure 20 on page 205. In this section we will repeat various lines from these examples with explanatory notes.

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRSV SYSPRINT     12/17/14 (14.351)  10:39:44    PAGE 01

                 RECORD SELECTION AND VALIDATION REPORT
```

Each page of reports from Record Selection and Validation begins with these lines, identifying first the name, release and modification level numbers of the product writing the reports. The second line identifies the program by name and the DD name being used. The execution date is given in both Gregorian and Julian formats followed by the time at which program execution began. Report pages are numbered consecutively. The report identification appears on each page.

```
                        EXECUTION PARAMETERS

CATALOG NAME               CRURCAT.VUSER01
RECORD SELECTION START     11/21/14 (14.325)  16:28:30
RECORD SELECTION STOP      11/21/14 (14.325)  16:48:38
SIGNIFICANT GAP TIME       0030 MINUTES
MAXIMUM CLOCK DIFFERENCE   NONE SECONDS
```

Each page of report information from Record Selection and Validation lists the execution parameters. Because the results must be viewed relative to these values, review the parameters to make sure that:

1. The correct catalog name was specified
2. The start date and time correspond to the date and time of the backup data set to be used for recovery
3. The stop date and time correspond to the date and time when it was confirmed that the catalog was closed in preparation for recovery
4. The gap specification (in minutes) is smaller than the shortest time interval spanned by a single SMF data set
5. The multi-system clock difference (specified in seconds) is at least as large as the maximum difference between the time-of-day clocks on any two systems.

```
REPORT FOR ALL SYSTEMS
```

or

```
REPORT FOR SYSTEM T81J
```

Record Selection and Validation produces first a summary report for all systems and then individual reports for each system encountered in the SMF data. You should first review the report for all systems, and then only if further information or investigation is needed, proceed to the individual system reports. The report analysis is basically the same for all reports, except that the first report will tell you about the different systems encountered and provides an overall summary.

```
RECORD SELECTION AND VALIDATION CONDITION CODE IS 04
```

This is the maximum condition code encountered during program execution, for all systems or for an individual system, depending on the report being viewed.

```
0 ANOMALIES (LOST DATA, GAPS) DETECTED
```

If the count of GAPS AND/OR LOST DATA RECORDS is not zero, you should analyze the following data to locate the cause.

```
31 RECORDS SELECTED FOR CRURCAT.VUSER01
```

If the count of RECORDS SELECTED is zero, your subsequent analysis should try to determine if this could be correct. Since the numbers will vary by catalog, by elapsed time and by activity, it is not possible to present guidelines on what these numbers should be. This line should be the total of the counts below of (TYPE 6x) RECORDS SELECTED.

```
 2 DEFINE (TYPE 61) RECORDS SELECTED
10 DELETE (TYPE 65) RECORDS SELECTED
19 ALTER  (TYPE 66) RECORDS SELECTED
```

Normally you might expect somewhat equal counts for DEFINE and DELETE with a smaller count for ALTER. This example is not very representative of what you should expect to see.

```
01 SYSTEM(S) RECORDED CHANGES TO THIS CATALOG
   T81J
```

These lines appear only on the report for all systems. Look at the count for SYSTEM(S) that RECORDED CHANGES TO THIS CATALOG and beneath it, the list of system identifiers (normally more than one line). Make sure that all systems having access to the catalog are in the list. However, if a system that had access to the catalog did not update the catalog it will not be listed here. There will be a system report for such a system and you should look there to be sure that that system's data was not omitted.

```
FOR CATALOG CRURCAT.VUSER01
```

Note that the next four lines concern only the SMF records for the catalogs named here.

```
11/21/14 (14.325)  16:12:46.57 OLDEST SMF CATALOG RECORD FOUND
```

The date and time of the OLDEST SMF CATALOG RECORD FOUND should normally precede the specified recovery start time. It should also precede the next entry.

```
11/21/14 (14.325)  16:28:31.74 OLDEST SMF CATALOG RECORD SELECTED
```

The date and time of OLDEST SMF CATALOG RECORD SELECTED should usually coincide fairly closely with the specified recovery start time. If EXPORT is being used for backup, there should be an SMF record a few seconds after the specified start time, resulting from the catalog update to turn on the temporary EXPORT indicator.

If the OLDEST SMF CATALOG RECORD SELECTED precedes the specified recovery start date and time, it should be within the interval specified as the multi-system clock difference (which is used to establish an effective start time). This condition may account for spurious error messages, if records in this interval result in reprocessing changes already represented in the EXPORTed copy of the catalog.

```
11/21/14 (14.325)  16:47:27.94 NEWEST SMF CATALOG RECORD SELECTED
```

The date and time of the NEWEST SMF CATALOG RECORD SELECTED for this catalog should usually be the same as that of the NEWEST SMF CATALOG RECORD FOUND unless intermediate record collection is being done.

The date and time of each should precede the specified recovery stop time. Even though the recovery stop time is adjusted by the multi-system clock difference, the time needed to confirm that the catalog was closed for recovery should be well after the last update to the catalog.

```
11/21/14 (14.325)  16:47:27.94 NEWEST SMF CATALOG RECORD FOUND
```

The date and time of the NEWEST SMF CATALOG RECORD FOUND for this catalog should usually be the same as that of the NEWEST SMF CATALOG RECORD SELECTED, unless intermediate record collection is being done. You would normally want the most current data.

```
FOR ALL SMF RECORD TYPES
```

Note that the remainder of the report concerns SMF records of all types, not just the SMF catalog records.

```
11/21/14 (14.325)  16:00:03.37 OLDEST SMF RECORD FOUND (ANY TYPE)
```

The date and time of the OLDEST SMF RECORD FOUND should normally precede the specified recovery start time. If it does not, older SMF data has probably been omitted.

```
11/21/14 (14.325)  16:59:58.54 NEWEST SMF RECORD FOUND (ANY TYPE)
```

The date and time of the NEWEST SMF RECORD FOUND should normally be later than the specified recovery stop time, otherwise the most recent SMF data has probably been omitted.

The next several lines of the report summarize the events that you should investigate for omitted or lost SMF data.

```
1 SYSTEM IDENTIFIERS WERE FOUND
```

This count includes all systems from which SMF data was read, not just those which updated the catalogs named above.

```
665 TOTAL SMF RECORDS WERE READ
```

Based on the experience of your installation, determine whether this count is reasonable considering the number of systems and the period covered by the SMF data.

```
1 SMF SWITCH (TYPE 90, SUBTYPE 6) RECORDS WERE FOUND
```

This count is not very interesting on the “all systems” report. However, on an individual system report, it tells you how many times the SMF recording data set was switched, and therefore gives you some idea of whether all appropriate SMF data sets have been included. In addition, in the log data set you will find a message for each switch, CRU023I. The time difference between the switch records tells you how often the SMF data sets are filling up (or being switched). Based on this value, you can determine whether the SMF gap time you specified is sufficiently small to detect the absence of an SMF input data set.

```
0 SMF EOD (TYPE 90, SUBTYPE 7) RECORDS WERE FOUND
```

Again, this value is primarily of interest for a single system. The intention is that you should be able to spot periods of deliberate system inactivity and thereby resolve corresponding gaps. Look for message CRU022I in the log. Normally it will precede the message for the next event.

```
0 SMF IPL (TYPE 0) RECORDS WERE FOUND
```

This value too is primarily of interest for a single system. The IPL may be a planned one or it may be the result of an interruption. Look for message CRU021I in the log. The SMF record for a planned IPL will normally be preceded by a record for EOD. The accompanying gap in SMF recording (if any) is thus explained and can be disregarded.

If no EOD record precedes the IPL record, there is a good chance that there has been a system interruption and SMF records may have been lost. If the time between the interruption and the IPL exceeds the gap time, an accompanying gap message should be expected. Otherwise, this IPL record may be the only indication of lost SMF data. Consequently, each IPL should be understood, referring to your problem management records and to the system log as necessary to establish the time of any accompanying interruption.

```
0 SMF LOST DATA (TYPE 7) RECORDS WERE FOUND
```

The next line gives the count of SMF LOST DATA RECORDS FOUND. This value is primarily of interest for a single system. Look for message CRU202I for that system in the log. The dumped record accompanying that message contains the beginning date and time of the resulting gap in SMF records and also the number of records lost. You will normally want to proceed with the ICFRU, using the SMF data you do have.

```
  0 FORWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
    0030 MINUTES WERE FOUND
```

Any gaps that appear should be resolved from the single system reports. A FORWARD GAP condition is recognized when the date and time for the current record is later than the previous record for that system by an interval longer than the gap time you specified. Each such event is signaled by message CRU201I in the log. It is most frequently not associated with a lost data condition, but rather with a normal or planned period of system inactivity.

If accompanied by a normal EOD and IPL sequence or if the interval is during a time when this system is normally inactive, the gap is accounted for an may be ignored. If the gap is not accounted for in this way, look for omissions in the SMF input data. Even if all input data is present, it may not have been presented in sequence by data set. In this case, there should be an accompanying BACKWARD GAP (see below).

If the gap cannot be explained by any of the previous reasons, you should assume that SMF data has been lost. You will normally want to proceed with the ICFRU, using the SMF data you do have.

```
  0 BACKWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
    0030 MINUTES WERE FOUND
```

A BACKWARD GAP condition is recognized when the date and time for the current record is earlier than the previous record for that system by an interval longer than the gap time you specified. The event is signaled by message CRU200I in the log. It typically results from concatenating newer SMF dump data sets in front of older ones. Thus the principal use of this count and its companion message is the resolution of forward gaps.

It could also be caused by reusing a partially filled SMF recording data set before it has been dumped (or emptied). Thus it might accompany a system interruption, which resulted in the use of an SMF recording data set other than the one active at the time of interruption. In this last case you might expect to find an IPL record after the backward gap.

# Reports from the Record Analysis and Processing Program

## General Description of CRURRAP Reports

The Record Analysis and Processing program produces four reports:

- ERROR REPORT
- ANOMALY REPORT
- REPORT OF RECORDS PROCESSED WITHOUT ERROR OR ANOMALY
- REPORT OF RECORDS BY DATA SET

You will probably want to review the reports in this order, using the CRURRAP final condition code to anticipate which reports should contain non-zero values.

The reports have a width less than 80 characters to make it easy to review them from the terminal. In general, the output lines within a report are arranged hierarchically with lower levels of detail being progressively indented to the right. Thus, the sum of the values at a given level is the value of the line above and to the left. The exceptions to this are the Error and Anomaly Reports that contain a report total at the same indentation as the next subordinate level.

The text of the report lines is also arranged hierarchically in the sense that the text of a lower level describes a more specific condition within the scope described at the higher levels. The lowest level detail lines also contain message identifiers that should appear in the log data set for error or anomaly conditions. The descriptive phrase used in a detail report lines is expanded in the explanation of the companion message. The messages issued by ICFRU are described in *z/OS MVS System Messages, Vol 4 (CBD-DMO)* The text of the summary lines is explained in the following discussion of the individual reports.

For complete examples see "Reports from Record Analysis and Processing" on page 207. In this section we will repeat various lines from these examples with explanatory notes.

**Common Header for all CRURRAP Reports:**

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRAP SYSPRINT     12/17/14 (04.351)  10:39:44    PAGE 1
```

Each page of reports from Record Analysis and Processing begins with these lines, identifying first the name, release and modification level numbers of the product writing the reports. The second line identifies the program by name and the DD name being used. The execution date is given in both Gregorian and Julian formats followed by the time at which program execution began. Report pages are numbered consecutively.

**Restatement of CRURRAP Execution Parameters:**

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY V2R1
CRURRAP SYSPRINT                02/04/13 (13.035) 11:25:16     PAGE 1

                RECORD ANALYSIS AND PROCESSING REPORT
                        EXECUTION PARAMETERS
```

```
CATALOG NAME               CATEI001.UCAT1
RECORD SELECTION START     02/04/13 (13.035)  11:24:39
RECORD SELECTION STOP      02/04/13 (13.035)  11:25:02
SIGNIFICANT GAP TIME       0030 MINUTES
MAXIMUM CLOCK DIFFERENCE   0000 SECONDS
CATALOG WAS EXPORTED ON    02/04/13 AT 11:24:39
```

The first page of the reports from Record Analysis And Processing lists the execution parameters. Because the results must be viewed relative to these values, review the parameters to make sure that:

1. The correct catalog name was specified
2. The start date and time correspond to the date and time of the backup data set to be used for recovery
3. The stop date and time correspond to the date and time when it was confirmed that the catalog was closed in preparation for recovery
4. The gap specification (in minutes) is smaller than the shortest time interval spanned by a single SMF data set
5. The multi-system clock difference (specified in seconds) is at least as large as the maximum difference between the time-of-day clocks on any two systems.
6. The date and time that the catalog was exported correspond to the date and time of the portable data set to be used for recovery was created.

**CRURRAP Error Report:** Record Analysis and Processing produces first a summary report of all errors encountered that result in a condition code of either 8 or 12.

0 TOTAL ERRORS

The Record Analysis and Processing program recognizes as errors incorrect records found in the EXPORT input and certain sequences of SMF and EXPORT records.

```
0 RECORDS REJECTED FROM EXPIN
   0 RECORDS WITH INVALID LENGTHS              (CRU302I)
   0 RECORDS WITH INVALID CATALOG RECORD TYPES (CRU303I)
```

Records from the EXPORT data set used as input are not processed if they have an incorrect length field or are not of a recognized type. Any such records indicate a serious problem with the catalog copy being used as the basis for recovery. Review the log data set for messages CRU3*nn*I associated with this condition. Program execution continues only because this might be the last or only available copy of the catalog. **Use the resulting output data set only as a last resort.** It almost certainly contains errors that must be manually detected and resolved.

0 ERRORS IN EVENT SEQUENCE INVOLVING THE MOST CURRENT RECORD

Certain sequences of SMF and/or EXPORT records cannot occur with correct processing (deletion of a nonexistent record, for example). These are referred to as “errors in event sequence” or simply as “sequence errors” because there must be a missing record or a record that is actually newer bears an earlier date/time stamp.

The “most current record” for a catalog entry is either its SMF record with the highest date/time stamp or, if there is no SMF record, its record from the EXPORT input data set.

See the next section of the report for more specific identification and breakdown of the errors.

```
0 SEQUENCE ERRORS, BUT NO SYNCHRONIZATION CHECK
   0 SMF UPDATE FOR A NON-EXISTENT RECORD     (CRU203I)
   0 SMF DELETE FOR A NON-EXISTENT RECORD     (CRU204I)
   0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU205I)
   0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU206I)
   0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU207I)
   0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU208I)
   0 SMF INSERT PRECEDED BY EXPORT RECORD     (CRU209I)

0 SEQUENCE ERRORS, WITH A SYNCHRONIZATION CHECK
   0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU205I)
   0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU206I)
   0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU207I)
   0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU208I)
```

A “synchronization check” is the condition which exists when there are SMF records for the same catalog entry from different systems within the interval you specified as the multi-system clock difference. The absence of a synchronization check indicates a potentially more serious problem in that there is no likely explanation for the sequence error as there might be by reversing the sequence of the records when justified by the accompanying synchronization check.

For each error, review the log data set to find the message CRU2*nn*I associated with that condition. Save the listings of this report and the log data set to be used in with the diagnostics following the IDCAMS IMPORT of the new catalog. Each condition is explained more fully in the text of the message whose number appears at the end of the line.

**CRURRAP Anomaly Report:** Record Analysis and Processing produces a summary report of all anomalies encountered that result in a condition code of 0 or 4.

0 ANOMALIES(CONDITION CODES 4 AND 0)

The Record Analysis and Processing program recognizes as anomalies, synchronization checks not accompanied by a sequence error and sequence errors that do not involve the most current record for a catalog entry.

```
0 SYNCHRONIZATION CHECKS INVOLVING THE MOST CURRENT RECORD
                  BUT WITH NO EVENT SEQUENCE ERROR (CRU113I)

   0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU003I)
   0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU004I)
   0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU005I)
   0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU006I)
   0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU007I)
```

A “synchronization check” is the condition which exists when there are SMF records for the same catalog entry from different systems within the interval you specified as the multi-system clock difference.

The “most current record” for a catalog entry is either its SMF record with the highest date/time stamp or, if there is no SMF record, its record from the EXPORT input data set.

The main purpose of performing synchronization checking is to help resolve sequence errors. However, even if no sequence errors are noted, a large number of synchronization checks may indicate that the specification for the clock difference was too large.

ERRORS IN EVENT SEQUENCE INVOLVING SUPERSEDED RECORDS

Certain sequences of SMF and EXPORT records These are referred to as “errors in event sequence” or simply as “sequence errors” because there must be a missing record or a record that is actually newer bears an earlier date/time stamp.

A “superseded record” for a catalog entry is either an SMF record that bears a lower date/time stamp than at least one other SMF record for that data set (and so is not processed in favor of the newer record), or it can be a record from the EXPORT input data set of a catalog entry that is also represented by an SMF record. The SMF record is presumed to be the more current representation of the catalog entry.

```
0 SEQUENCE ERRORS, BUT NO SYNCHRONIZATION CHECK
   0 SMF UPDATE FOR A NON-EXISTENT RECORD     (CRU106I)
   0 SMF DELETE FOR A NON-EXISTENT RECORD     (CRU107I)
   0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU108I)
   0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU109I)
   0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU110I)
   0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU111I)
   0 SMF INSERT PRECEDED BY EXPORT RECORD     (CRU112I)
```

“Sequence errors” and “synchronization checks” are explained above.

These errors should be investigated carefully because they may indicate an otherwise undetected lost SMF data condition.

Review the log message for each error, noting the date and time. The most common explanation will be that there were SMF records within the interval between the *effective* start time (specified start time minus the clock difference) and the *specified* recovery start time,that is, the time the backup copy of the catalog was made.

If the records are not within this “start-up interval”, note the date and time of the records in the companion message and investigate the system log and other sources for possible indications of lost data.

```
0 SEQUENCE ERRORS, WITH A SYNCHRONIZATION CHECK
   0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU108I)
   0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU109I)
   0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU110I)
   0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU111I)
```

“Sequence errors” and “synchronization checks” are explained above.

Errors in this category are most frequently explained by a difference in the clocks in a multi-systems environment, so that the SMF date/time stamps do not reflect the true order of the catalog updates. If reversing the order of updates makes the sequence a logical one, this is most likely the situation. Otherwise, proceed as for errors in the preceding category.

```
0 SYNCHRONIZATION CHECKS INVOLVING SUPERSEDED RECORDS
                  BUT WITH NO EVENT SEQUENCE ERRORS (CRU020I)

   0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU013I)
   0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU014I)
   0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU015I)
   0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU016I)
   0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU017I)
```

"Sequence errors", "synchronization checks" and "superseded records" are explained above. Since no errors are involved in this situation, the most likely explanation is that the clock difference specification is too large (or that the clocks are not synchronized closely enough).

**CRURRAP Report of Records Processed without Error:** For completeness, Record Analysis and Processing produces a summary report of records processed without error or anomaly. The report is really for information only. However, if you review some number of reports for your installation you will develop some feeling for the kinds of numbers you should expect to see when everything has gone as expected.

66 TOTAL RECORDS PROCESSED

The "records processed" without error or anomaly are catalog records only, the EXPORT control records are not included in this count. Since correct processing implies omissions, the number of records processed will not equal the number of records in the output data set, nor the sum of the records from the input data sets.

This report can also tell you about the update activity to your catalogs. Given the interval covered by the SMF data you can calculate the rate of update activity.

```
39 MOST CURRENT RECORDS PROCESSED W/OUT ERROR OR ANOMALY
    4 SMF INSERT FOR A NEW RECORD              (CRU002I)
    0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU003I)
    0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU004I)
    1 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU005I)
    2 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU006I)
    0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU007I)
    8 SMF UPDATE PRECEDED BY EXPORT RECORD     (CRU008I)
    9 SMF DELETE PRECEDED BY EXPORT RECORD     (CRU009I)
   15 EXPORT RECORD CARRIED FORWARD            (CRU001I)
```

These are the records that really matter since only these records determine the content of the "new catalog"that is, the one to be built from the new EXPORT data set.

```
27 SUPERSEDED RECORDS PROCESSED WITHOUT ERROR OR ANOMALY
    0 SMF INSERT FOR A NEW RECORD              (CRU012I)
    0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU013I)
    0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU014I)
    4 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU015I)
    0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU016I)
    0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU017I)
    3 SMF UPDATE PRECEDED BY EXPORT RECORD     (CRU018I)
    0 SMF DELETE PRECEDED BY EXPORT RECORD     (CRU019I)
   20 EXPORT RECORD SUPERSEDED                 (CRU011I)
```

A record, from either SMF or from EXPORT, is superseded when there is at least one newer SMF record for that same catalog entry. These records are not involved in the processing which determines the content of the new catalog but their count is included here to give you some history of changes to the catalog.

**CRURRAP Report of Records by Data Set:** As an aid to auditing the functions of CRURRAP, the Record Analysis and Processing program produces a summary report showing the source and disposition of all input and all output records. Certain values are developed independently of others, so that summing the subtotals produces a cross-check of the totals.

36 TOTAL RECORDS IN THE NEW EXPORT DATA SET (EXPOUT)

The count of records actually written out is maintained directly by the program at the time the write occurs.

The next major headings identify the source of the records in the new EXPORT data set.

23 RECORDS FORWARDED FROM THE OLD EXPORT DATA SET (EXPIN)

As records are written, their source is identified. Those from the old EXPORT are summed here.

8 CONTROL RECORDS

The EXPORT control records must come from this source; there are no such records written to SMF. As a consequence of this, the IMPORTed catalog will have the same structural definition as the EXPORTed catalog (unless you define a new catalog and IMPORT INTOEMPTY).

15 CATALOG RECORDS

Apart from the control records mentioned above, this accounts for all of the remaining records in the EXPORT data set.

13 CATALOG RECORDS SELECTED FROM THE SMF DATA SET (SMFIN)

As records are written, their source is identified, those from the SMF data are summed here. Not all SMF catalog records in the input stream are included in this count; some may be excluded because they do not meet the selection criteria.

43 TOTAL RECORDS FROM THE OLD EXPORT DATA SET (EXPIN)

This total accounts for all records read from the old EXPORT data set. The next three major headings account for the disposition of these records.

23 RECORDS CARRIED FORWARD TO THE NEW EXPORT DATA SET

These are records of catalog entries for which no corresponding SMF records were found. They are simply written from the old EXPORT to the new EXPORT without change.

8 CONTROL RECORDS

All of the control records should be carried to the new EXPORT.

15 CATALOG RECORDS

This is the count of the records in the old, input EXPORT data set excluding the control records. They were written to the new EXPORT.

20 RECORDS SUPERSEDED OR DELETED (BASED ON SMF DATA)

This is a count of the records for which corresponding SMF records were found. The catalog entries for these records were taken from the SMF data.

0 RECORDS REJECTED BECAUSE OF ERRORS

It is extremely unlikely that there are records in the old EXPORT data set that are recognized as being in error. Only the following items are checked by the program:

- 0 INVALID LENGTH

  The record length is part of the logical content of a catalog record. If this length does not actually match the length of the record read, the program recognizes an error and rejects the record.
- 0 UNRECOGNIZED CATALOG RECORD TYPE

  To protect against bad input data and to recognize invalid records, the program checks that the catalog record type is one that is recognized, (nonVSAM or cluster, for example).

31 TOTAL RECORDS FROM THE SMF DATA SET (SMFIN)

This is the count of all records read from the SMF input data sets. The following lines give the disposition of these records.

13 RECORDS CARRIED FORWARD TO THE NEW EXPORT DATA SET

As records are written to the new EXPORT data set, their source is identified. Those from SMF are counted here.

18 RECORDS SUPERSEDED OR DELETED BY NEWER SMF DATA

These are the SMF records applicable to the recovery in progress that were either deletion records, causing the catalog entry to be omitted from the new EXPORT data set, or superseded records, that is, those for which SMF records with higher date/time stamps were found.

0 RECORDS REJECTED

Unrecognizable records or records not applicable to this recovery are categorized as follows.

- 0 NOT AN MVS SMF RECORD

  The indicator for an MVS SMF record was not on. The record may not be an SMF record at all.
- 0 NOT AN SMF CATALOG RECORD

  The MVS SMF record type was not a 61, 65 or 66.
- 0 NOT AN SMF CATALOG RECORD FOR THIS CATALOG

  The MVS SMF catalog record contained a name that did not match the name of the catalog being recovered.
- 0 DATE/TIME EARLIER THAN EFFECTIVE START TIME

  The SMF record was for the catalog being recovered, but its date and time preceded the specified start date and time less the multi-system clock difference.
- 0 DATE/TIME LATER THAN EFFECTIVE STOP TIME

  The SMF record was for the catalog being recovered, but its date and time followed the specified stop date and time plus the multi-system clock difference.

As a final audit point, the program accounts for all records encountered.

36 TOTAL OF ALL OUTPUT RECORDS

Output records are only those written to the new EXPORT data set. Consequently, this number should match the number of records in that data set.

38 TOTAL OF ALL RECORDS DISCARDED

Discarded records are all those records that were:

1. Rejected because they were in error, unrecognized or not applicable to this recovery
2. Superseded by more current SMF records
3. Most current SMF deletion records resulting in the omission of a catalog entry from the new EXPORT

74 TOTAL OF ALL INPUT RECORDS

This number is the sum of the records read from the SMF input data sets and the EXPORT input data set. It should be the sum of the two lines above.

# Samples and Examples

## Sample Reports from ICFRU

### Reports from Record Selection and Validation

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRSV SYSPRINT                12/17/14 (14.351)  10:39:44    PAGE 01

                RECORD SELECTION AND VALIDATION REPORT

                        EXECUTION PARAMETERS

CATALOG NAME               CRURCAT.VUSER01
RECORD SELECTION START     11/21/14 (14.325)  16:28:30
RECORD SELECTION STOP      11/21/14 (14.325)  16:48:38
SIGNIFICANT GAP TIME       0030 MINUTES
MAXIMUM CLOCK DIFFERENCE   NONE SECONDS

REPORT FOR ALL SYSTEMS

RECORD SELECTION AND VALIDATION CONDITION CODE IS 04

         0 ANOMALIES (LOST DATA, GAPS) DETECTED
        31 RECORDS SELECTED FOR CRURCAT.VUSER01
         2 DEFINE (TYPE 61) RECORDS SELECTED
        10 DELETE (TYPE 65) RECORDS SELECTED
        19 ALTER  (TYPE 66) RECORDS SELECTED

     01 SYSTEM(S) RECORDED CHANGES TO THIS CATALOG
        T81J

FOR CATALOG CRURCAT.VUSER01
11/21/14 (14.325)  16:12:46.57 OLDEST SMF CATALOG RECORD FOUND
11/21/14 (14.325)  16:28:31.74 OLDEST SMF CATALOG RECORD SELECTED
11/21/14 (14.325)  16:47:27.94 NEWEST SMF CATALOG RECORD SELECTED
11/21/14 (14.325)  16:47:27.94 NEWEST SMF CATALOG RECORD FOUND

FOR ALL SMF RECORD TYPES
11/21/14 (14.325)  16:00:03.37 OLDEST SMF RECORD FOUND (ANY TYPE)
11/21/14 (14.325)  16:59:58.54 NEWEST SMF RECORD FOUND (ANY TYPE)

         1 SYSTEM IDENTIFIERS WERE FOUND
       665 TOTAL SMF RECORDS WERE READ
         1 SMF SWITCH (TYPE 90, SUBTYPE 6) RECORDS WERE FOUND
         0 SMF EOD (TYPE 90, SUBTYPE 7) RECORDS WERE FOUND
         0 SMF IPL (TYPE 0) RECORDS WERE FOUND
         0 SMF LOST DATA (TYPE 7) RECORDS WERE FOUND
         0 FORWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
           0030 MINUTES WERE FOUND

         0 BACKWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
           0030 MINUTES WERE FOUND

RECORD SELECTION AND VALIDATION CONDITION CODE IS 04
```

*Figure 19. Record Selection and Validation Report (all systems)*

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRSV SYSPRINT                12/17/14 (04.351)  10:39:44    PAGE 02

                RECORD SELECTION AND VALIDATION REPORT

                        EXECUTION PARAMETERS

CATALOG NAME               CRURCAT.VUSER01
RECORD SELECTION START     11/21/14 (14.325)  16:28:30
RECORD SELECTION STOP      11/21/14 (14.325)  16:48:38
SIGNIFICANT GAP TIME       0030 MINUTES
MAXIMUM CLOCK DIFFERENCE   NONE SECONDS
REPORT FOR SYSTEM T81J

RECORD SELECTION AND VALIDATION CONDITION CODE IS 04

         0 ANOMALIES (LOST DATA, GAPS) DETECTED
        31 RECORDS SELECTED FOR CRURCAT.VUSER01
         2 DEFINE (TYPE 61) RECORDS SELECTED
        10 DELETE (TYPE 65) RECORDS SELECTED
        19 ALTER  (TYPE 66) RECORDS SELECTED

FOR CATALOG CRURCAT.VUSER01
11/21/14 (14.325)  16:12:46.57 OLDEST SMF CATALOG RECORD FOUND
11/21/14 (14.325)  16:28:31.74 OLDEST SMF CATALOG RECORD SELECTED
11/21/14 (14.325)  16:47:27.94 NEWEST SMF CATALOG RECORD SELECTED
11/21/14 (14.325)  16:47:27.94 NEWEST SMF CATALOG RECORD FOUND

FOR ALL SMF RECORD TYPES
11/21/14 (14.325)  16:00:03.37 OLDEST SMF RECORD FOUND (ANY TYPE)
11/21/14 (14.325)  16:59:58.54 NEWEST SMF RECORD FOUND (ANY TYPE)

       665 TOTAL SMF RECORDS WERE READ
         1 SMF SWITCH (TYPE 90, SUBTYPE 6) RECORDS WERE FOUND
         0 SMF EOD (TYPE 90, SUBTYPE 7) RECORDS WERE FOUND
         0 SMF IPL (TYPE 0) RECORDS WERE FOUND
         0 SMF LOST DATA (TYPE 7) RECORDS WERE FOUND
         0 FORWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
           0030 MINUTES WERE FOUND
         0 BACKWARD GAPS IN SINGLE-SYSTEM SMF RECORDS LONGER THAN
           0030 MINUTES WERE FOUND

RECORD SELECTION AND VALIDATION CONDITION CODE IS 04
```

*Figure 20. Record Selection and Validation Report (one system)*

## SYSLOG from Record Selection and Validation

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRSV SYSLOG                   12/17/04 (04.351)  10:39:44


CRU100I CLOCK DIFFERENCE PARAMETER NOT PROVIDED, CLOCK SYNCHRONIZATION ASSUMED


CRU023I SWITCH SMF RECORD FOUND FOR SYSID T81J
        11/21/04 (04.325)  16:51:17.98 RECORD BEING PROCESSED - DUMP FOLLOWS

+0000 009F80  004C0000 065A005C 96560085 325FE3F8   *.<...!.*o..e..T8
              F1D10000 00000024 000C0001 00000030    1J..............*
+0020 009FA0  001C0001 0006F0F1 E2D4C640 40404040   *......01SMF
              E2E8E2F1 4BD4C1D5 F240E2E8 E2F14BD4    SYS1.MAN2 SYS1.M*
+0040 009FC0  C1D5F140 007A1BD0 0085324F            *AN1 .:...e.|
```

*Figure 21. Record Selection and Validation Report SYSLOG.* Dump lines have been folded to fit within the page.

## Output from DFSORT

```
ICE143I 0 BLOCKSET     TECHNIQUE SELECTED
ICE000I 0 --- CONTROL STATEMENTS/MESSAGES ---- 5964-A01 REL 6.0
                                          ---- 10.39.47 DEC 17, 2004 ----

       SORT FIELDS=(218,44,CH,A,262,1,BI,A,11,4,PD,D,7,4,BI,D)

ICE088I 0 HEMINGU .SORT  , INPUT LRECL = 32767, BLKSIZE = 4096, TYPE = VS
ICE093I 0 MAIN STORAGE = (MAX,2097152,586496), NMAX = 0
ICE128I 0 OPTIONS: SIZE=2097152,MAXLIM=2097152,MINLIM=204800,EQUALS=Y,
                   LIST=Y,ERET=RC16 ,MSGDDN=SYSOUT
ICE129I 0 OPTIONS: VIO=N,EXCPVR=N,RESDNT=NONE,SMF=NO   ,WRKSEC=Y,
                   OUTSEC=Y,VERIFY=N,CHALT=N,DYNALOC=(SYSDA   ,01)
ICE130I 0 OPTIONS: RESALL=12288,RESINV=0,SVC=109,CHECK=N,WRKREL=Y,
                   OUTREL=Y,CKPT=N,STIMER=Y
ICE084I 0 EXCP ACCESS METHOD USED FOR SORTOUT
ICE084I 0 EXCP ACCESS METHOD USED FOR SORTIN
ICE055I 1 INSERT 0, DELETE 0
ICE054I 1 RECORDS - IN: 31, OUT: 31
ICE134I 1 NUMBER OF BYTES SORTED: 11164
ICE098I 0 AVERAGE RECORD LENGTH = 360 BYTES
ICE052I 1 END OF SORT/MERGE
```

*Figure 22. DFSORT SYSPRINT.* Some lines have been folded to fit within the page.

## Reports from Record Analysis and Processing

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRAP SYSPRINT                12/17/04 (04.351) 10:39:52     PAGE 1

                RECORD ANALYSIS AND PROCESSING REPORT

                        EXECUTION PARAMETERS

CATALOG NAME               CRURCAT.VUSER01
RECORD SELECTION START     11/21/04 (04.325)  16:28:30
RECORD SELECTION STOP      11/21/04 (04.325)  16:48:38
SIGNIFICANT GAP TIME       0030 MINUTES
MAXIMUM CLOCK DIFFERENCE   NONE SECONDS

 RECORD ANALYSIS AND PROCESSING CONDITION CODE IS 04


                           ERROR REPORT

         0 TOTAL ERRORS (CONDITION CODES 12 AND 8)

         0 RECORDS REJECTED FROM EXPIN               (LOGGED, DUMPED, CC=12)

               0 RECORDS WITH INVALID LENGTHS              (CRU302I)
               0 RECORDS WITH INVALID CATALOG RECORD TYPES (CRU303I)

         0 ERRORS IN EVENT SEQUENCE INVOLVING THE MOST CURRENT RECORD
                                                 (LOGGED, DUMPED, CC=8)

               0 SEQUENCE ERRORS, BUT NO SYNCHRONIZATION CHECK
                     0 SMF UPDATE FOR A NON-EXISTENT RECORD     (CRU203I)
                     0 SMF DELETE FOR A NON-EXISTENT RECORD     (CRU204I)
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU205I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU206I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU207I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU208I)
                     0 SMF INSERT PRECEDED BY EXPORT RECORD     (CRU209I)

               0 SEQUENCE ERRORS, WITH A SYNCHRONIZATION CHECK
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU205I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU206I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU207I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU208I)
```

*Figure 23. Record Analysis and Processing Error Report*

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRAP SYSPRINT                12/17/04 (04.351) 10:39:52     PAGE 2


                         ANOMALY REPORT

         0 ANOMALIES (CONDITION CODES 4 AND 0)

         0 SYNCHRONIZATION CHECKS INVOLVING THE MOST CURRENT RECORD
                         BUT WITH NO EVENT SEQUENCE ERROR  (CRU113I)
                                                 (LOGGED, DUMPED, CC=4)
               0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU003I)
               0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU004I)
               0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU005I)
               0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU006I)
               0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU007I)

         0 ERRORS IN EVENT SEQUENCE INVOLVING A SUPERSEDED RECORD
                                                           (LOGGED, CC=4)

               0 SEQUENCE ERRORS, BUT NO SYNCHRONIZATION CHECK
                     0 SMF UPDATE FOR A NON-EXISTENT RECORD     (CRU106I)
                     0 SMF DELETE FOR A NON-EXISTENT RECORD     (CRU107I)
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU108I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU109I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU110I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU111I)
                     0 SMF INSERT PRECEDED BY EXPORT RECORD     (CRU112I)

               0 SEQUENCE ERRORS, WITH A SYNCHRONIZATION CHECK
                     0 SMF INSERT PRECEDED BY AN SMF INSERT     (CRU108I)
                     0 SMF INSERT PRECEDED BY AN SMF UPDATE     (CRU109I)
                     0 SMF UPDATE PRECEDED BY AN SMF DELETE     (CRU110I)
                     0 SMF DELETE PRECEDED BY AN SMF DELETE     (CRU111I)

         0 SYNCHRONIZATION CHECKS INVOLVING A SUPERSEDED RECORD
                         BUT WITH NO EVENT SEQUENCE ERROR  (CRU020I)
                                                    (NOT LOGGED, CC=0)
               0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU013I)
               0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU014I)
               0 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU015I)
               0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU016I)
               0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU017I)
```

*Figure 24. Record Analysis and Processing Anomaly Report*

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRAP SYSPRINT                12/17/04 (04.351) 10:39:52     PAGE 3


         REPORT OF RECORDS PROCESSED WITHOUT ERROR OR ANOMALY

        66 TOTAL RECORDS PROCESSED (NO ERROR/NO ANOMALY, CONDITION CODE 0)

             39 MOST CURRENT RECORDS PROCESSED WITHOUT ERROR OR ANOMALY
                    4 SMF INSERT FOR A NEW RECORD              (CRU002I)
                    0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU003I)
                    0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU004I)
                    1 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU005I)
                    2 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU006I)
                    0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU007I)
                    8 SMF UPDATE PRECEDED BY EXPORT RECORD     (CRU008I)
                    9 SMF DELETE PRECEDED BY EXPORT RECORD     (CRU009I)
                   15 EXPORT RECORD CARRIED FORWARD            (CRU001I)

             27 SUPERSEDED RECORDS PROCESSED WITHOUT ERROR OR ANOMALY
                    0 SMF INSERT FOR A NEW RECORD              (CRU012I)
                    0 SMF UPDATE PRECEDED BY AN SMF INSERT     (CRU013I)
                    0 SMF DELETE PRECEDED BY AN SMF INSERT     (CRU014I)
                    4 SMF UPDATE PRECEDED BY AN SMF UPDATE     (CRU015I)
                    0 SMF DELETE PRECEDED BY AN SMF UPDATE     (CRU016I)
                    0 SMF INSERT PRECEDED BY AN SMF DELETE     (CRU017I)
                    3 SMF UPDATE PRECEDED BY EXPORT RECORD     (CRU018I)
                    0 SMF DELETE PRECEDED BY EXPORT RECORD     (CRU019I)
                   20 EXPORT RECORD SUPERSEDED                 (CRU011I)
```

*Figure 25. Record Analysis and Processing Report of Records Processed without Error*

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRAP SYSPRINT                12/17/04 (04.351) 10:39:52     PAGE 4


                  REPORT OF RECORDS BY DATA SET

        36 TOTAL RECORDS IN THE NEW EXPORT DATA SET (EXPOUT)
                    8 CONTROL RECORDS
                   28 CATALOG RECORDS

              23 RECORDS FORWARDED FROM THE OLD EXPORT DATA SET (EXPIN)
                    8 CONTROL RECORDS
                   15 CATALOG RECORDS

              13 CATALOG RECORDS SELECTED FROM THE SMF DATA SET (SMFIN)

        43 TOTAL RECORDS FROM THE OLD EXPORT DATA SET (EXPIN)

              23 RECORDS CARRIED FORWARD TO THE NEW EXPORT DATA SET
                    8 CONTROL RECORDS
                   15 CATALOG RECORDS

              20 RECORDS SUPERSEDED OR DELETED (BASED ON SMF DATA)

               0 RECORDS REJECTED BECAUSE OF ERRORS
                    0 INVALID LENGTH
                    0 UNRECOGNIZED CATALOG RECORD TYPE

        31 TOTAL RECORDS FROM THE SMF DATA SET (SMFIN)

              13 RECORDS CARRIED FORWARD TO THE NEW EXPORT DATA SET

              18 RECORDS SUPERSEDED OR DELETED BY NEWER SMF RECORDS

               0 RECORDS REJECTED
                    0 NOT AN MVS SMF RECORD
                    0 NOT AN SMF CATALOG RECORD
                    0 NOT AN SMF CATALOG RECORD FOR THIS CATALOG
                    0 DATE/TIME EARLIER THAN EFFECTIVE START TIME
                    0 DATE/TIME LATER THAN EFFECTIVE STOP TIME

        36 TOTAL OF ALL OUTPUT RECORDS

        38 TOTAL OF ALL RECORDS DISCARDED

        74 TOTAL OF ALL INPUT RECORDS
```

*Figure 26. Record Analysis and Processing Report of Records by Data Set*

### SYSLOG from Record Analysis and Processing

```
INTEGRATED CATALOG FORWARD RECOVERY UTILITY R1M0

CRURRAP SYSLOG                   12/17/04 (04.351)  10:39:52


CRU100I CLOCK DIFFERENCE PARAMETER NOT PROVIDED, CLOCK SYNCHRONIZATION ASSUMED
```

*Figure 27. Record Analysis and Processing SYSLOG*

### Output from Compare Utility

```
                          COMPARE UTILITY                    PAGE 0001


    END OF JOB-TOTAL NUMBER OF RECORDS COMPARED = 00000036
```

*Figure 28. IEBCOMPR SYSPRINT*

## Examples for Catalog Diagnosis, EXPORT and IMPORT

```
//*************************************************************
//* THE GDG'S AND MODEL DSCB'S FOR THE DATA SETS ASSOCIATED WITH *
//* EXPORTING CATALOG applic.catalog ARE DEFINED.  BY CONVENTION *
//* ALL BACKUP DATA SETS WILL START WITH 'Bapplic' .             *
//*************************************************************
//SETUPDS  EXEC PGM=IDCAMS
//LIST     DD  DSN=IDCAMS.LIST.DCB,DISP=(NEW,CATLG),
//             VOL=SER=catvol,UNIT=devtyp,SPACE=(0,0),
//             DCB=(RECFM=VBA,LRECL=125,BLKSIZE=4250)
//EXPORT   DD  DSN=IDCAMS.EXPORT.DCB,DISP=(NEW,CATLG),
//             VOL=SER=catvol,UNIT=devtyp,SPACE=(0,0),
//             DCB=(RECFM=VBS,LRECL=32404,BLKSIZE=devblk)
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DEF ALIAS ( NAME(Bapplic) REL(RECOVERY.CATALOG) )
 DEF GDG   ( NAME(Bapplic.CATALOG.LISTING) LIM(7) NEMP SCR -
             OWNER(stormgmt) FOR(9999) ) CAT(RECOVERY.CATALOG)
 DEF GDG   ( NAME(Bapplic.CATALOG.BACKUP)  LIM(7) NEMP SCR -
             OWNER(stormgmt) FOR(9999) ) CAT(RECOVERY.CATALOG)
 DEF GDG   ( NAME(Bapplic.PROBLEM.LISTING) LIM(7) NEMP SCR -
             OWNER(stormgmt) FOR(9999) ) CAT(RECOVERY.CATALOG)
 LISTCAT LEVEL(Bapplic) ALL
/*
```

*Figure 29. Setup the Backup Data Sets for Catalog Export.* EXPORT copies and companion listings are to be saved as companion generation data sets of the same generation number.

```
//*************************************************************
//* DIAGNOSE THE CATALOG,                                     *
//* LIST ITS ALIASES,                                         *
//* LIST ITS SELF-DESCRIBING ENTRY COMPLETELY,                *
//* LIST ITS ENTRIES WITH JUST THE VOLUME INFORMATION         *
//*************************************************************
//DIAGLIST EXEC PGM=IDCAMS
//SETBKDS  DD  DSN=Bapplic.CATALOG.BACKUP(+1),DISP=(NEW,PASS),
//             DCB=IDCAMS.EXPORT.DCB,
//             VOL=SER=bckvol,UNIT=devtyp,SPACE=(TRK,(15,15),RLSE)
//SYSPRINT DD  DSN=Bapplic.CATALOG.LISTING(+1),DISP=(MOD,PASS),
//             DCB=IDCAMS.LIST.DCB,
//             VOL=SER=lstvol,UNIT=devtyp,SPACE=(TRK,(15,15),RLSE)
//SYSIN    DD *
 DIAG  ICFCAT IDS(applic.CATALOG)
 LISTC ENTRY(applic.CATALOG) ALL
 LISTC ENTRY(applic.CATALOG) ALL CAT(applic.CATALOG)
 LISTC VOL CAT(applic.CATALOG)
/*
//*************************************************************
//* EXPORT THE CATALOG IF THE DIAGNOSTICS WERE OKAY           *
//*************************************************************
//EXPCAT   EXEC PGM=IDCAMS,COND=(8,LE)
//CATBACK  DD  DSN=Bapplic.CATALOG.BACKUP(+1),DISP=(OLD,PASS),
//             DCB=IDCAMS.EXPORT.DCB,
//             SPACE=(TRK,(15,15),RLSE)
//SYSPRINT DD DSN=Bapplic.CATALOG.LISTING(+1),DISP=(MOD,PASS),
//             DCB=IDCAMS.LIST.DCB,
//             SPACE=(TRK,(15,15),RLSE)
//SYSIN    DD  *
 EXP applic.CATALOG OFILE(CATBACK) TEMP
/*
//*************************************************************
//* CATALOG THE BACKUP AND THE LISTING IF EXPORT WAS OKAY     *
//*************************************************************
//CATAL    EXEC PGM=IEFBR14,COND=(8,LE)
//CATBACK  DD  DSN=Bapplic.CATALOG.BACKUP(+1),DISP=(OLD,CATLG)
//CATLIST  DD  DSN=Bapplic.CATALOG.LISTING(+1),DISP=(OLD,CATLG)
//*************************************************************
//* COPY THE LISTING AND DISCARD THE BACKUP IF EXPORT FAILED  *
//*************************************************************
//REPOUT   EXEC PGM=IDCAMS,COND=(0,EQ,CATAL)
//SYSPRINT DD  SYSOUT=*
//CATBACK  DD  DSN=Bapplic.CATALOG.BACKUP(+1),DISP=(OLD,DELETE)
//CATLIST  DD  DSN=Bapplic.CATALOG.LISTING(+1),DISP=(OLD,DELETE)
//CATPROB  DD  DSN=Bapplic.PROBLEM.LISTING(+1),DISP=(NEW,CATLG),
//             VOL=SER=lstvol,UNIT=devtyp,SPACE=(TRK,(15,15),RLSE)
//SYSIN    DD *
 REPRO INFILE(CATLIST) OUTFILE(CATPROB)
/*
```

*Figure 30. Catalog Diagnose and Backup.* The catalog is diagnosed, its aliases and self-describing information are saved, along with its entries. If this is successful, the catalog is EXPORTed. The EXPORTed copy and companion listings are kept in corresponding generation data sets, when the EXPORT step is successful. Otherwise, they are discarded and the listing is saved in a “problem” data set.

# Program Descriptions

## CRURRSV — Record Selection and Validation

### Purpose

The purpose is to select and validate SMF data according to user-specified parameters, for presentation to CRURRAP. The program attempts to determine whether a complete and correct set of SMF data is available for catalog recovery purposes. A report and a log are produced to assist you in verifying that the right SMF data has been supplied.

### Input

Input consists of two items:

1. SMF data (DDname SMFIN)
2. User-specified parameters (the PARM parameter on the EXEC statement)

### Output

Output consists of three items:

1. Selected and validated SMF data (DDname SMFOUT)
2. A report (DDname SYSPRINT)
3. A log (DDname SYSLOG)

See the section "JCL for CRURRSV" on page 185 for more information on JCL requirements.

### Processing

The program opens input and output data sets, then processes user-specified parameters. If a clock difference value was supplied, it then calculates effective start/stop times by subtracting the clock difference from the start time and by adding it to the stop time. The main processing loop is then entered. In this loop, an SMF record is read, then the following conditions are checked:

- Non-SMF record supplied as input
- Excessive time gap between SMF records

Log messages and SMF record dumps are produced for these conditions. Counters and tables are maintained (overall and by SYSID) in preparation for reporting. These record types are also logged and dumped:

- IPL
- Halt EOD
- Switch SMF
- Lost data

Type 61/65/66 records that meet selection criteria - correct catalog (unless all catalogs specified) and within effective start/stop range - are written to the output SMF data set. When EOF is reached on SMFIN, a check is made for excessive time gap between specified start time and oldest SMF record found, or between newest SMF record found and specified stop time. Log messages are produced for these conditions. Then the report is formatted, data sets are closed, the program establishes a return code, and terminates.

# CRURRAP — Record Analysis and Processing

## Purpose

The purpose is to analyze SMF data selected and validated by CRURRSV and to process it with a catalog EXPORT to produce a new and current catalog EXPORT, to be used as input to IDCAMS for catalog recovery. A report and a log are produced to assist you in verifying that the right SMF data has been supplied and that consistent results have been achieved.

## Input

Input consists of three items:

1. SMF data (DDname SMFIN)
2. Catalog EXPORT data (DDname EXPIN)
3. User-specified parameters (the PARM parameter on the EXEC statement)

## Output

Output consists of three items:

1. A new EXPORT (DDname EXPOUT)
2. A report (DDname SYSPRINT)
3. A log (DDname SYSLOG)

See the section "JCL for CRURRAP" on page 186 for more information on JCL requirements.

## Processing

The program opens input and output data sets, then processes user-specified parameters. If a clock difference value was supplied, it then calculates effective start/stop times by subtracting the clock difference from the start time and by adding it to the stop time. GETMAINs are issued and addressability established for four buffers of 32K each:

1. SMF 'I' buffer (input buffer for SMF records)
2. SMF 'S' buffer (secondary SMF buffer)
3. SMF 'C' buffer (buffer for most current SMF record encountered so far for a particular data set)
4. EXPORT 'X' buffer (input buffer for EXPORT records)

The SMF 'I' and 'S' buffers are primed. Then the EXPORT control records are read. The following errors are checked for during this processing:

- No data supplied (immediate EOF on either SMFIN or EXPIN)
- Data in EXPIN is not in EXPORT format
- Data is in EXPORT format but is not an integrated catalog facility EXPORT
- Specified start time does not match time of EXPORT
- First non-control record in EXPORT is not a basic catalog structure (BCS) cluster record
- EXPORT is not for the catalog specified with input parameters

During control record processing the user-specified stop date and time are used to replace the EXPORT date and time in one of the EXPORT control records, so that the subsequent IMPORT will reflect catalog currency correctly. All control records are written to EXPOUT.

The main processing loop is then entered. 'Promotion' of records between buffers takes place based on the relative positions of the entryname + pad byte values of the records (the 45-byte BCS record keys). The purpose of this is to move the most current record one step up the buffer hierarchy (from 'X' or 'S' to 'C', and from 'X' or 'I' to 'S'). In the buffer hierarchy, 'C' is the most current, then 'S', then 'X' or 'I'. Movement takes place as follows:

- From 'X' to 'C' if key in 'S' > key in 'X'
- From 'S' to 'C' if key in 'S' >= key in 'X'
- From 'X' to 'S' if key in 'I' > key in 'X'
- From 'I' to 'S' if key in 'I' >= key in 'X'

When a record is 'promoted' from the 'I' buffer, the GETSMF subroutine is called to read another SMF record into 'I'. When a record is 'promoted' from 'X', the GETEXP subroutine is called to read another EXPORT record into 'X'. This process continues until EOF is reached on both SMFIN and EXPIN.

When the most current record for an entry is in the 'C' buffer, it is assigned a set of values that represent its status relative to other records for the same entry. This setting also involves the subtype field (SMFxxSUB), which indicates whether the catalog activity represented by the SMF record was an insert, a delete, or an update. The assigned values are used later for logging purposes.

If the records in 'C' and 'S' have the same key and are from different systems, a synchronization check is done. This is where the user-specified clock difference is used. A synchronization error is defined as a situation where more than one system changed the catalog within the user-specified clock difference value. For example, if one system changed the catalog at 14:33:57 and another system changed it at 14:33:58 and the clock difference specified was 2 seconds, then it is not possible to be certain in which order these changes occurred, since the clock difference of 2 seconds means that in fact the 14:33:58 change might have actually happened first if the clocks were not synchronized.

Now we are ready to decide whether or not to carry forward the record in 'C' to the new EXPORT (EXPOUT). If the record is not a delete (according to SMFxxSUB) then it is carried forward. If the source of a record to be carried forward is SMF, it is checked for security fields. VSAM password and related fields are set to blanks when SMF records are written by catalog management, since writing them to the SMF data set would constitute a security exposure. Leaving these fields as blanks would make resetting these difficult (or perhaps even impossible, since some of the fields should be numeric). The program resets these fields as the records are written to EXPORT, as indicated below. The resetting simplifies re-establishing of VSAM passwords and related fields after recovery.

- Passwords - Set to 'Z's (all four passwords)
- Password Prompting Code - Set to 'Z's
- Attempts - Set to '2' (the IDCAMS default)
- USVR module name - Set to 'Z's
- USVR string length - Calculated and corrected
- USVR string - Left as blanks

A table of counters (the record processing table) is updated based on the set of values previously assigned. Logging and dumping of records follows, also based on the set of values, for cases where something suspicious was found. Control then goes back to the top of the main loop.

When both input data sets have been processed, reporting is done by calling CSECT CRURREPO. The record processing table and other counters are used to produce reports on errors, anomalies, and records processed without error or anomaly. The final report page summarizes processing by data set. Data sets are then closed, the program establishes a return code, and terminates.

## Capacities and Limitations

### Size of Counters and Numeric Report Fields

The only data areas likely to be affected by processing large volumes of data are counters. All counters in the programs are a fullword in size, and therefore can handle numbers up to and including 2,147,483,647. However, numeric fields in the reports can handle a maximum of 7 digits (that is, 9,999,999).

### Records from SMF or EXPORT

Records up to 32K bytes in length can be processed from either source.

### Multi-system operation

Data from up to sixteen different SYSIDs can be processed by CRURRSV. If for some reason the user has SMF data from more than sixteen systems merged together, even though fewer than sixteen systems could have updated the catalog being recovered, then the data for just the systems that could have updated the catalog should be extracted (using IFASMFDP or another utility) before the data is passed to CRURRSV. If more than sixteen systems could have updated the catalog (due to intermittent use of test systems, for example), use IFASMFDP (SMF dump) or another utility to create more than one data set, each of which has sixteen or fewer SYSIDs represented. Run each batch through CRURRSV separately, then concatenate the outputs from the CRURRSV runs as input to the sort.

### Concatenation of Unlike Input

Neither program supports concatenation of unlike input (for example, data sets on DASD and tape). Data sets on different devices similar in type are acceptable (for example, 3350 and 3380). If the input data sets to be concatenated are similar in device type but have differing blocksizes, then the normal concatenation rules still apply (the largest blocksize must be first in the concatenation). Note that the programs will not process SMF data directly from the SMF system data sets. The circumvention for all of these restrictions is the same: copy the appropriate portion of the SMF data so that all inputs are in similar format. IFASMFDP can be used to do this, unless reblocking is required (IFASMFDP always forces its output to the same LRECL and BLKSIZE). To reblock, use IDCAMS or IEBGENER and specify DCB information for the output data set.

### Recovery Scope

ICFRU provides forward recovery for the basic catalog structure (BCS) of the integrated catalog facility. It does not handle VSAM catalogs. The ICFRU does not manage recovery of VSAM volume data sets (VVDS); that can be accomplished only by recovering the VSAM data sets represented in the damaged VVDS.

# Chapter 10. Catalog Diagnostic Information

This appendix contains diagnosis, modification, or tuning information. See also "Diagnosing a Catalog Performance Problem" on page 31.

## The Basic Catalog Structure (BCS)

The BCS is a VSAM key-sequenced data set and contains volume, data set security, ownership, and association information for VSAM and non-VSAM data sets.

A BCS can also point to a volume on which only non-VSAM data sets and generation data sets reside. If the volume is SMS-managed, the VVDS also contains information about the non-VSAM and generation data sets. If the volume is not SMS-managed, the information is only maintained in the BCS.

Related information in the BCS is grouped into logical, variable-length, spanned records related by key. The BCS uses keys that are the data set names (plus 1 character for extensions). A control interval can contain multiple BCS records. To reduce the number of I/Os necessary for catalog processing, logically related data is consolidated in the BCS.

The cell is the smallest block of information in the BCS and might contain the name, volume, owner, and association information for a catalog entry.

Entries for SMS-managed data sets contain an SMS subcell. This subcell contains the names for the storage, data, and management classes for the data set.

VSAM components must have the same SMS attributes as their associated base cluster. These components must also be cataloged in the same catalog as the base cluster, and this catalog must be the one SMS defines as the default catalog. (The default catalog is the catalog the system chooses using only the catalog aliases and the data set names, that is, the catalog chosen when you let the system choose the catalog, instead of directing the search yourself.)

Temporary SMS-managed VSAM data sets have VVRs in the VVDS, but they do not have BCS entries.

### BCS Records

There are two types of BCS records: the sphere record and the nonsphere record. A sphere record contains one or more components. The key length of any record is 45 bytes, consisting of a 44-byte data set or object name and a 1-byte pad character to indicate an extension record.

Records and components are a logically related group of cells. These logical groupings are physically adjacent in a sphere record. An example of a component is the index of a VSAM data set.

#### Sphere Records

Sphere records might have related extension records. Extension records are created when the maximum record size of the BCS cannot contain a new component entry for a VSAM data set or alternate DEFINE USERCATALOG command. The default is 32400 bytes.

An alternate index or generation data group must be able to fit within an extension record.

An extension record is created when:

- An alternate index or generation data group is defined and does not fit in the current sphere record.
- A path is defined and the entry in the association cell does not fit in the sphere record.

  An extension cell will be created if there is a component entry for an alternate index or generation data set which can be moved into the extension record. The path entry must remain in the primary sphere record.
- Volumes are added to a cluster or alternate index and the volume cell does not fit in the sphere record.

The key of the extension record is the base cluster or generation data group name and the pad character.

The pad character is a binary number. The first extension is X'01', the second extension is X'02', and so on, up to the 255th extension which is X'FF'. The maximum number of extensions allowed is 255.

A component level entity is moved to the new extension record whether it is the component being updated or the last component on the current sphere record. For a VSAM sphere record, this is an alternate index. For a generation data group sphere record, the generation data set component is moved. Only one component resides in each extension record.

## Association Cells

Certain types of BCS entries can be paired with other BCS entries. This pairing of records is called an “association”. Associated entries are connected by name and are indicated by an association cell in an entry.

Figure 31 illustrates an example of an association and its logical connections.

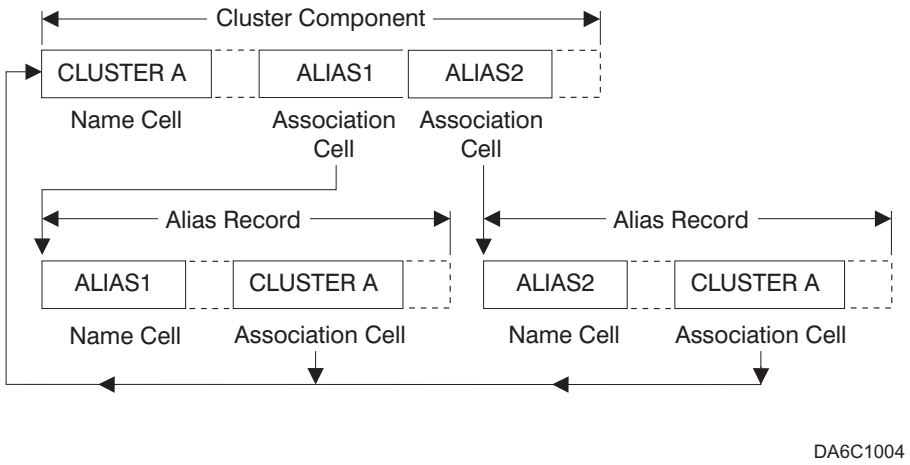



*Figure 31. Example of an Association and Its Logical Connections*

The following associations can occur:

- Alias entries with catalog connector, non-VSAM data set, or generation data set entries.

- Catalog connector, non-VSAM data set, or generation data set entries with alias entries.
- Path entries with cluster or alternate index entries.
- Cluster or alternate index entries with path entries.
- Cluster truenames with data, index, and alternate index components.

### BCS Record Types

The first cell in each record has a cell type field, which is also the record type (or ID). The record type is identified by the DIAGNOSE command if DIAGNOSE lists the entry. The following are the possible record types and their one character identifiers:

| ID | Record Type |
|---|---|
| A | non-VSAM data set |
| B | generation data group |
| C | cluster |
| D | data component of a cluster |
| E | VSAM extension record |
| G | alternate index |
| H | generation data set |
| I | index component of a cluster |
| J | generation data group extension cell |
| L | library |
| R | path |
| T | true name |
| U | user catalog connector |
| V | user catalog connector extension record |
| W | volume |
| X | alias |

## Initial Contents of a BCS

When a catalog is defined on a volume, a VSAM sphere record is built for the VVDS which resides on the volume with the BCS. This record has the name of the VVDS as its key (SYS1.VVDS.V*volser*).

A catalog has a sphere record similar to other VSAM key-sequenced data sets. This is a self-describing sphere record for the catalog, and contains information about the catalog itself. This sphere record is given a key of binary zeros to ensure it is the first record in the catalog. The data component name for the BCS is the catalog name as you have defined it. A true name record is created for the data and index components. The true name records are related with a key to the catalog name. A record is created to relate the index component to the catalog name of binary zeros, in the same way as for the data component.

## Allocation and Non-VSAM Catalog Entries

In non-VSAM data set catalog entries, there is a pointer to the data set's DSCB. This allows the system to locate the data set more quickly than can be done through a VTOC search. When a non-VSAM data set is allocated, if the pointer does not point to the correct DSCB, allocation marks the data set so that the data set can be recataloged when it is deallocated. When the data set is deallocated, it is recataloged with the information gathered at allocation time.

Normally, this maintains correct information in the catalog. However, if you move a data set or otherwise update the catalog entry for a data set, and allocation has

marked the data set to be recataloged at deallocation, the updates you made to the catalog entry are lost. Deallocation does not recognize that you have changed the entry.

For example, if you move a system data set, and allocation had marked that data set to be recataloged at deallocation time, the data set is recataloged with the volume serial number of the *old* volume, not the volume you moved it to.

There is no way to prevent this problem. To repair the catalog entry, simply delete the entry with DELETE NOSCRATCH, and recatalog the data set with DEFINE NONVSAM.

## The VSAM Volume Data Set (VVDS)

The VVDS is a VSAM entry-sequenced data set that has a 4KB control interval size. It contains information about the VSAM and SMS-managed non-VSAM data sets residing on the volume with the VVDS.

A VVDS is recognized by the restricted data set name SYS1.VVDS.V*volser*, where *volser* is the volume serial number of the volume on which the VVDS resides.

The VVDS is composed of a minimum of two records:
1. A VSAM volume control record (VVCR)
2. A VVDS self-describing volume record.

The first logical record in a VVDS is the VSAM volume control record (VVCR). It contains information for management of DASD space and the BCS names which currently have VSAM or SMS-managed non-VSAM data sets on the volume. It might have a pointer to an overflow VVCR.

The second logical record in the VVDS is the VVDS self-describing VVR (VSAM volume record). This self-describing VVR contains information that describes the VVDS.

The remaining logical records in the VVDS are VVRs for VSAM objects or non-VSAM volume records (NVRs) for SMS-managed non-VSAM data sets. The hexadecimal RBA of the record is used as its key or identifier.

Figure 32 shows the general structure of a VVDS.

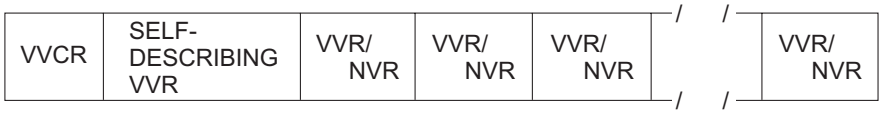



*Figure 32. VSAM Volume Data Set (VVDS) Structure*

## VSAM Volume Record (VVR)

VSAM volume records contain information about the VSAM data sets residing on the volume with the VVDS. If more than one VVR is associated with a component, the first (primary) VVR contains information pertaining to the data set as a whole. The other (secondary) VVRs do not repeat the primary information in their records but contain information for their own component, such as extents, RBAs, and allocation quantities, and most of the information needed to open a VSAM data set.

The number of VVRs for VSAM data sets varies according to the type of data set and the options specified for the data set. Table 18 contains the number of primary VVRs for each type of data set.

*Table 18. Number of Primary VVRs for Data Set Types*

| Data Set Type | Number of Primary VVRs |
|---|---|
| Entry-sequenced data set | 1 per volume for the data component |
| Key-sequenced data set with the NOIMBED option | 1 per volume for the data component; 1 for the index |
| Key-sequenced data set with the IMBED option | 1 per volume for the data component; 1 for the index; 1 per volume for the sequence set |
| Key range key-sequenced data set with the NOIMBED option | 1 per key range per volume; 1 for the index |
| Key range key-sequenced data set with the IMBED option | 2 per key range per volume; 1 per volume for the index |

The size of a VVR depends on its type (primary or secondary), and is determined by the combined lengths of VVR cells. Figure 33 shows the primary and secondary VVRs and the cells that constitute each VVR. These VVR cells are:

- VVR header cell
- VVR data set information cell
- VVR AMDSB cell
- VVR volume information cell

The VVRLEN field of the VVR header cell contains the length of the entire VVR. The VVRTYPE field of the same cell contains the VVR type code, which is either "Z" (for primary) or "Q" (for secondary).

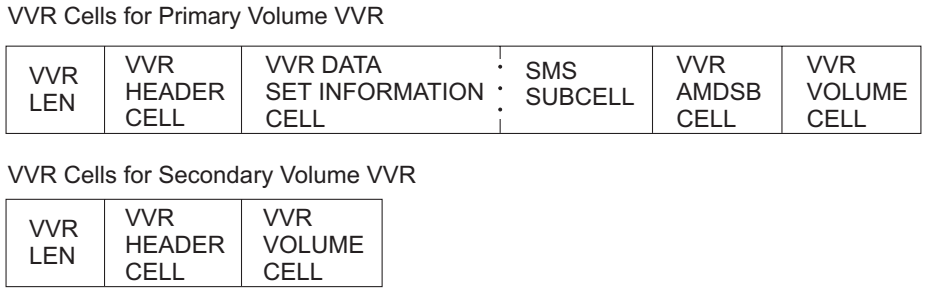



*Figure 33. VSAM Volume Record (VVR) Structure*

Figure 34 on page 222 shows examples of the information contained in each type of VVR cell.

VVR Header Cell

| LEN | TYPE | FLAG | KRQ | COMP NAME LEN | COMP NAME | CLS NAME LEN | CLS NAME | CAT NAME LEN | BASE CLS NAME LEN | BASE CLS NAME |
|---|---|---|---|---|---|---|---|---|---|---|

VVR Data Set Information Cell

| LEN | TYPE | ATTR1 | ATTR2 | OPEN IND | BUF SZ | PRI SPC | SEC SPC | SPC OPT | DS HU RBA | DS HA RBA | LRECL | RESV | EXPECT | EXIT | DS HK RBA | CLS ATT | TIME STMP | AL TAB TIME STMP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

VVR Volume Information Cell

| LEN | TYPE | VOLFG | #EXT | HKRBA | HURBA | HARBA | BLKSZ | BLK PER TRK | TRK PER AU | TYP EXT | TRK PER CYL | BYT PER TRK | BYT PER AU | LOW KEY LEN | LOW KEY | HI KEY LEN | HI KEY | EXT LEN | EXT |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

*Figure 34. Examples of VVR Cell Information*

## Non-VSAM Volume Record (NVR)

The non-VSAM volume record (NVR) is equivalent to a VVR record, but the NVR record is for SMS-managed non-VSAM data sets. The NVR contains SMS-related information.

If an SMS-managed non-VSAM data set spans volumes, only the first volume contains an NVR for that data set. See Figure 35.

VVR Cells for Primary Volume VVR

| NVR LEN | NVR HEADER CELL | NVR DATA SET INFORMATION CELL | SMS SUBCELL |
|---|---|---|---|

*Figure 35. Non-VSAM Volume Record (NVR) Structure*

# Chapter 11. Catalog Search Interface User's Guide

Catalog Search Interface (CSI) is a read-only general-use programming interface that is used to obtain information about entries contained in catalogs. The catalog entries are selected using a generic filter key provided as input. The generic filter key can be a fully-qualified entry name, in which case one entry is returned, or the generic filter key can contain "wild cards" so that multiple entries can be selected on a single invocation. The type or types of entries desired can also be specified. For instance, all non-VSAM entries that begin with "ABC" could be selected.

Field information for each entry is requested by specifying field names. This eliminates the need for the caller to know whether the information is in the Basic Catalog Structure (BCS) or in the VSAM Volume Data Set (VVDS).

A work area, whose address and size is provided on input, is used to return the selected entries and the field information for those entries. If all of the entries cannot be contained in the work area provided, then as many as possible are returned in one invocation, and an indicator is set to reflect that more entries exist. Subsequent invocations can specify that the request is being resumed and the additional entries can be obtained. Resumes can be repeated until all entries have been returned. The resume process allows the delivery of large amounts of information to the user program without impacting catalog resources.

## CSI Invocation

CSI can be invoked in either 24-bit or 31-bit addressing mode. CSI is reentrant and reusable. CSI can be invoked in any protection key and in either supervisor or problem state.The module name of the CSI program is IGGCSI00; it is stored in SYS1.LINKLIB.

The invocation can be done by any of the following methods:

- An assembler language CALL statement that results in a V-type address constant
- An assembler language LINK macro
- Assembler language LOAD and CALL macros
- A high-level language that results in one of these types of invocation.

When CSI is invoked:

- general-purpose register (GPR) 1 points to a parameter list
- GPR 13 points to a standard register save area of length 72 bytes
- GPR 14 contains the return address in the caller's program
- GPR 15 contains the entry point of CSI.

The parameter list pointed to by GPR 1 has three fullword entries:

- The first word of the parameter list contains a pointer to a 4-byte reason area used to return error or status information.
- The second word points to a list of selection criteria fields used to communicate with CSI.
- The third word points to a work area that contains data on return from CSI.

On return, a return code is passed in GPR 15 to indicate whether an error has occurred and the nature of the error. The first address in the parameter list points to a 4-byte reason area that contains information concerning the return code.

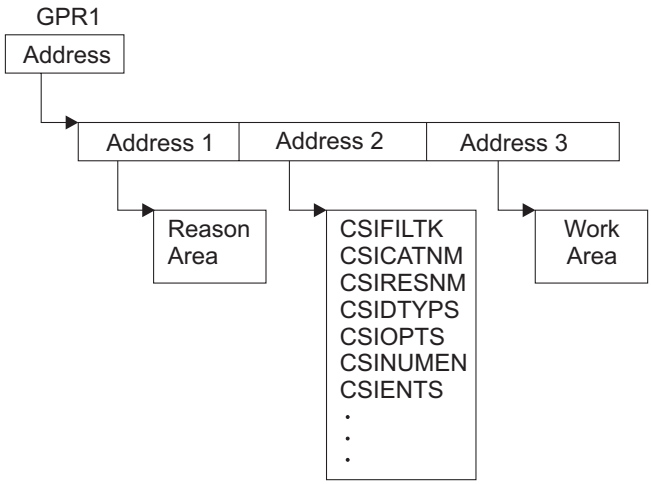
## The Parameter List

On invocation, general-purpose register 1 points to the CSI parameter list.

The first word in the parameter contains an address that points to a 4-byte reason area. On return from CSI, the reason area contains a module identification, reason code and return code. See the section on Output for further discussion of this parameter.

The second word contains the address of the selection criteria fields. The selection criteria data fields supply information to CSI on invocation, and contain information for a resume if there is more information to be returned than can fit in the user-provided work area.

The third word contains the address of a user work area in which entry information will be returned. The caller must place the size of the work area in the first word of the work area. Although this area has no fixed size, it must be between 1024 bytes and 1,048,575 bytes, inclusive. On an initial call, the CSI will return the minimum size necessary to contain one entry, its catalog and fields included, and return code 32 will be set if the minimum size required is not specified. On a resume call, the required length field (CSIREQLN) from the previous call should always be checked to ensure that the return area size is large enough for the next entry.




## Selection Criteria Fields

Table 19 shows the selection criteria fields.

*Table 19. Selection Criteria Fields*

| Offset | Type | Length | Field Name | Description |
|---|---|---|---|---|
| 0(0) | | VL | CSIFIELD | CSI selection criteria fields |
| 0(0) | Character | 44 | CSIFILTK | Generic filter key |
| 44(2C) | Character | 44 | CSICATNM | Catalog name or blanks |

*Table 19. Selection Criteria Fields (continued)*

| Offset | Type | Length | Field Name | Description |
|---|---|---|---|---|
| 88(58) | Character | 44 | CSIRESNM | Resume name or blanks |
| 132(84) | Character | 16 | CSIDTYPD | Entry types |
| 132(84) | Character | 1 | CSIDTYPS | Entry types to be returned. All types = blanks |
| 148(94) | Character | 4 | CSIOPTS | CSI Options |
| 148(94) | Character | 1 | CSICLDI | Return data or index, Y or blank |
| 149(95) | Character | 1 | CSIRESUM | Resume, Y or blank |
| 150(96) | Character | 1 | CSIS1CAT | Search 1 catalog only, Y or blank |
| 151(97) | Character | 1 | CSIOPTNS | An F entry means to use fullword lengths; any other entry means use halfword lengths. |
| 152(98) | Fixed | 2 | CSINUMEN | Number of entries in table |
| 154(9A) | Character | VL | CSIENTS | Variable length table containing field names |
| 154(9A) | Character | 8 | CSIFLDNM | Field name (1st one) |

A PL/I mapping is provided in IGGCSINP. In assembler language you can use the IGGCSINA macro to define the DSECT. The fields are described in the following sections.

## CSIFILTK, Generic Filter Key

CSI uses a generic filter key supplied in CSIFILTK. A generic filter key is a character string that describes the catalog entry names for which you want information returned. The generic filter key can contain the following symbols and are interpreted as follows:

***** A single asterisk by itself indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.

****** A double asterisk indicates that zero or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a period or a blank.

**%** A single percent sign by itself indicates that exactly one alphanumeric or national character can occupy that position.

**%%...** One to eight percent signs can be specified in each qualifier.

If an absolute GDS name is specified followed by a period and one of the previous symbols, the GDS name (if it exists) will not be returned from the search. However, it is possible that longer data set names that start with the same prefix as the absolute GDS name will be found. See the CSDFILTK examples that follow.

## CSIFILTK Examples

```
VSAM.DATA.SET%
will return entry names:
                    VSAM.DATA.SET1
                    VSAM.DATA.SET2
    will not return:
                    VSAM.DATA.SET30
```

```
VSAM.DATA.SET%%
will return entry names:
                     VSAM.DATA.SET30
                     VSAM.DATA.SET31
    will not return:
                     VSAM.DATA.SET1
                     VSAM.DATA.SET2
VSAM.*.SET
will return entry names:
                     VSAM.DATA1.SET
                     VSAM.DATA2.SET
    will not return:
                     VSAM.DATA.SET.KSDS
VSAM.*A
will return entry names:
                     VSAM.A
                     VSAM.BA
                     VSAM.BBA
    will not return:
                     VSAM.B
                     VSAM.AB
VSAM.DATA.*
will return entry names:
                     VSAM.DATA.SET1
                     VSAM.DATA.SET2
    will not return:
                     VSAM.DATA.SET.KSDS
VSAM.DATA*
will return entry names:
                     VSAM.DATA1
                     VSAM.DATA23
    will not return:
                     VSAM.DATA.SET
VSAM.**
will return entry names:
                     VSAM
                     VSAM.DATA.SET1
                     VSAM.DATA.SET2
                     VSAM.DATA.SET.KSDS
    will not return:
                     VSAM1.DATA.SET
VSAM.DATA.SET
will return entry name:
                     VSAM.DATA.SET  only
**.DATA
will return entry names whose low level qualifier is DATA:
                     VSAM.DATA
                     NONVSAM.WORK.DATA
**
will return every entry name in a catalog:
```

For a GDG base named DATASET.GDG containing the following GDS entries:

```
DATASET.GDG.G0001V00
DATASET.GDG.G0002V00
DATASET.GDG.G0003V00
```

and for the non-VSAM data set named DATASET.GDG.G0001V00.XYZ, the following keys will return the following results:

```
DATASET.GDG.**
     DATASET.GDG.G0001V00
     DATASET.GDG.G0002V00
     DATASET.GDG.G0003V00
     DATASET.GDG.G0001V00.XYZ

 DATASET.GDG.G0001V00
```

```
    DATASET.GDG.G0001V00

DATASET.GDG.G0001V00.**
    DATASET.GDG.G0001V00.XYZ

DATASET.GDG.G000%V00
    DATASET.GDG.G0001V00
    DATASET.GDG.G0002V00
    DATASET.GDG.G0003V00

DATASET.GDG.G000%V00.**
    DATASET.GDG.G0001V00
    DATASET.GDG.G0002V00
    DATASET.GDG.G0003V00
    DATASET.GDG.G0001V00.XYZ
```

**Exception:** The entries returned by the data set name are not necessarily returned in ascending order.

## CSICATNM, Catalog Name

CSICATNM is used for catalog selection. The following paragraphs describe how CSI performs catalog selection.

CSI will use the catalog name supplied in CSICATNM to search for entries if CSICATNM is not blanks. If CSICATNM is blanks, Catalog Management will attempt to use the high-level qualifier of CSIFILTK to locate an alias that matches. If an alias is found, the user catalog for that alias will be searched; then, the master catalog will be searched. The master catalog will not be searched if CSIS1CAT is set to Y. If no alias is found, only the master catalog will be searched.

Caution should be exercised in using high-level qualifiers that contain generic filters because multiple catalogs can be searched. A high-level of "**" will cause a search of every user catalog in the system.

CSIS1CAT can be used to limit the search to one catalog only. This is useful if the filter key could cause catalog management to select more than one catalog for searching. Also if not set, the master catalog will be searched in addition to one selected user catalog. In cases where the catalog name is supplied as input and a resume is done, CSIS1CAT will cause only that catalog whose name is supplied as input to be searched. Otherwise, on resume, catalog management cannot tell whether this is a search across many catalogs and the resume caused the CSICATNM to be set or if the name was supplied by the caller.

If a tape volume catalog library entry type or a tape volume catalog volume entry type is specified in CSIDTYPS, a tape volume catalog will be searched. Tape volume catalog library entry types and tape volume catalog volume entry types should not be mixed with catalog entry types.

## CSIRESNM, Resume Name

CSIRESNM is the name of the catalog entry on which the search will resume if the work area space for return information is used up. If the request can be resumed, CSIRESUM will be set to Y on return from the call to CSI, CSIRESNM will contain the name of the entry on which to resume, and CSICATNM will contain the name of the catalog in which the entry to be resumed was found. Normally, the application program will test CSIRESUM, and if it is Y, the application program will reissue the call to CSI without changing CSIRESNM and CSICATNM.

## CSIDTYPS, Entry Types

CSIDTYPS determines what type of catalog entries will be returned.

Valid types for CSIDTYPS are:

| Type | Description |
|---|---|
| **A** | non-VSAM data set |
| **B** | Generation data group |
| **C** | Cluster |
| **G** | Alternate index |
| **H** | Generation data set |
| **L** | Tape volume catalog library entry |
| **R** | VSAM path |
| **U** | User catalog connector entry |
| **W** | Tape volume catalog volume entry |
| **X** | Alias |

VSAM components, data and index, are returned with the cluster. Thus, there are no type specifications for them, however, "D" and "I" types will appear in the output information.

The valid types can be mixed and in any order. Blanks cannot separate the types. For instance, “ABH” might be specified to get only the non-VSAM, generation data group and generation data set entries. All other positions in CSIDTYPS must be blanks (X'40') when the types are specified.

All blanks for CSIDTYPS can be set and will get types A, B, C, G, H, R, U, X. These are the catalog types. L and W must be explicitly specified in order to get the Tape Volume Catalog entries.

## CSIOPTS, Options

### CSICLDI

CSICLDI determines whether information will be returned if the data and index names of a cluster do not match the filter key, but the cluster name does. If CSICLDI is set toY then components are returned if the cluster name matches the filter key.

### CSIOPTNS

If this value is set to F, all fields used to describe the output data for each entry will be fullword (4 bytes) in length. If the value is set to blank, all fields will be halfword (2 bytes) in length. Note that using 2–byte fields might mean that certain types of entries that return more than 65,535 characters for that catalog entry will not be processed.

### CSIRESUM

CSIRESUM is set to Y if CSI detects that there are more catalog entries that meet the search criteria. This field must be blank on the first invocation. CSIRESUM can be tested, and CSI can be called again to obtain more entries. CSIRESUM will be set to blank (X'40') if no more entries meet the search criteria.

### CSIS1CAT

CSIS1CAT causes only one catalog to be searched. It is used in conjunction with CSICATNM to determine which catalogs to search. Refer to the section on CSICATNM for more details concerning catalog selection.

## CSINUMEN, Number of Field Names

CSINUMEN is a binary number indicating the number of field names following in the subscripted area CSIFLDNM.

There is a limit of 100 field names per invocation of CSI. CSINUMEN cannot be greater than 100.

## CSIFLDNM, Field Names

CSIFLDNM is a list of 8-byte field names. If the field name is not eight characters long, then it must be padded on the right with blanks to make eight characters.

Valid field names that can be used in the list and the information returned for each field name is described in section "Field Name Directory" on page 235.

# Return Codes for General Purpose Register 15

On return, general-purpose register 15 can contain the following:

| Return Code | Description |
|---|---|
| **0** | No errors or return messages |
| **4** | Information was returned from Catalog Management processing. Further information is returned in the 4-byte reason area pointed to by the first address in the parameter list. See Table 20 on page 230 for the interpretation of this information. |
| **8** | Failure in Catalog Search Interface routine. Further information is returned in the 4-byte reason area pointed to by the first address in the parameter list. See "Return Code 8" on page 230 for the interpretation of this information. |
| **C** | Error in Catalog Search Interface routine parameter list - check for zero entries. |
| **10** | Parameter list pointer in general-purpose register 1 is zero. |

## Return Codes 4 and 8

When general-purpose register 15 contains 4 or 8, the first address in the parameter list points to a 4-byte reason area. The pattern of the information in the reason area is:

| Description | Length in Bytes |
|---|---|
| **Module ID** | 2 |
| **Reason code** | 1 |
| **Return code** | 1 |

### Return Code 4

When general-purpose register 15 contains 4, the reason area information is passed back from Catalog Management.

The module identification is a two-character EBCDIC code that can be used by IBM Service Personnel to determine which Catalog Management module set the reason and return code. The module identification information is not generally useful for applications.

Reason and return codes returned by Catalog Management are found in LookAt under message IDC3009I. For the description of a particular return and reason code, see *z/OS MVS System Messages, Vol 6 (GOS-IEA)*. For Return codes 100 (X'64') and 122 (X'7A'), see Table 20

*Table 20. Return Codes 100 and 122*

| Return Code | Description | Programmer Response |
|---|---|---|
| 100 | Catalog Management has detected an error while processing the request.<br>• Reason Code 4: at least one data set entry is returned with an error.<br>• Reason Code 8: at least one catalog entry is returned with an error. | • Reason Code 4: locate the entry or entries with the CSIENTER flag set and inspect the CSIRETN field for further information.<br>• Reason Code 8: inspect the return code and reason code for the catalog entry for further information. |
| 122 | An invalid filter key was provided. | For all reason codes: fix filter key and resubmit.<br>**Note:** This return code may also indicate an invalid data set name in the catalog being searched. |

### Return Code 8

When general-purpose register 15 contains 8, the reason area information is passed back from Catalog Search Interface routine. The module identification is always set to X'FFFF'. The reason codes have the following meanings:

**01** Insufficient storage for Getmain, increase region size

**02** Invalid entry type in CSIDTYPS

**03** Invalid data/index option in CSICLDI, should be Y or blank

**04** Invalid resume option in CSIRESUM, should be Y or blank

**05** Invalid search one catalog option in CSIS1CAT, should be Y or blank

**06** Invalid number of fields value in CSINUMEN, should be between zero and 100, inclusive

**07** Invalid work area length, should be between 1024 and 1048575, inclusive

**08** The CSIOPTNS value is not F or blank

The return code is set to 8 for all reason codes.

## Return Work Area Format

Entry information retrieved will be returned in the user-provided work area. Resume information is returned in the Selection Criteria Fields. If on return CSIRESUM is set to Y, then CSIRESNM will be set to the next entry to be processed, and CSICATNM will be the catalog in which that entry will be found. When the program has processed returned entry information, it should re-invoke CSI to resume at the next entry. Make sure to check the CSIREQLN field before

issuing the resume call to be sure the return area size needed has not increased; if it has increased, you must pass in a work area of the new required size.

The caller must set CSIUSRLN prior to CSI invocation. CSIUSRLN is the size in bytes of the work area including itself. CSIUSRLN must be a fixed fullword value between 1024 and 1,048,575, inclusive. Although CSIUSRLN can be 1,048,575, large sizes may impact Catalog Management resources. A size of 64,000 is generally recommended.

All other data is returned by CSI in the user-provided work area.

## Work Area Format Table

Upon return to the caller, the work area will be in the format shown in Table 21.

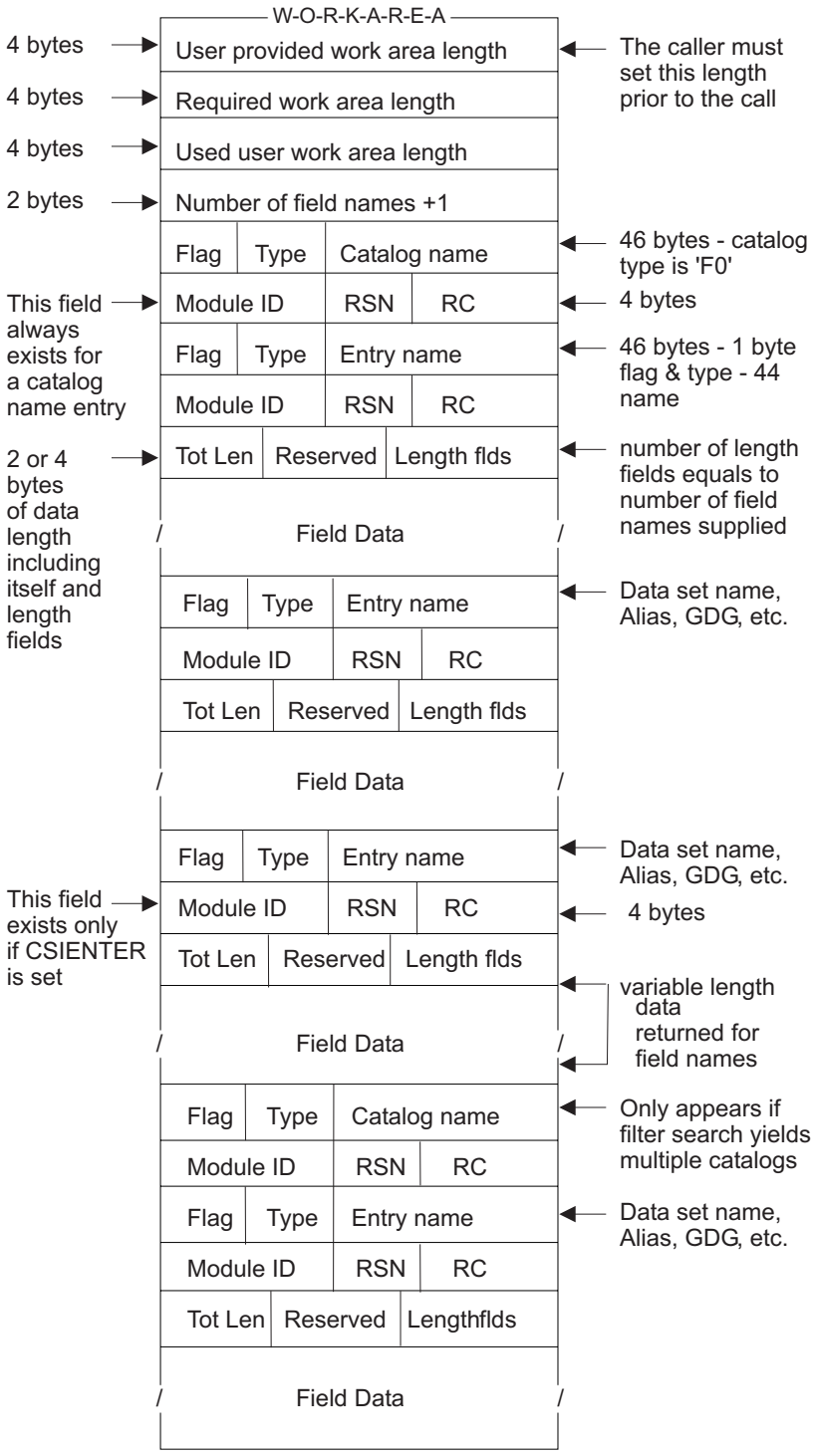
*Table 21. Work Area Format Table*

| Offset | Type | Length | Name | Description |
|---|---|---|---|---|
| **Information returned for work area** | | | | |
| 0(0) | Character | VL | CSIRWORK | CSI Return Work Area |
| 0(0) | Fixed | 4 | CSIUSRLN | Total length of work area. User provided. |
| 4(4) | Fixed | 4 | CSIREQLN | Minimum required work area for 1 catalog name entry and 1 data entry entry. |
| 8(8) | Fixed | 4 | CSIUSDLN | Total length of work area used in returning entries. |
| 12(C) | Fixed | 2 | CSINUMFD | Number of field names plus 1. |
| **Information returned for each catalog** | | | | |
| 0(0) | Bitstring | 1 | CSICFLG | Catalog flag information |
| | 1... .... | | CSINTICF | Not supported. |
| | .1.. .... | | CSINOENT | No entry found for this catalog. |
| | ..1. .... | | CSINTCMP | Data gotten for this catalog is not complete. |
| | ...1 .... | | CSICERR | Whole catalog not processed due to error. |
| | .... 1... | | CSICERRP | Catalog partially processed due to error. |
| | .... .111 | | | Reserved. |
| 1(1) | Character | 1 | CSICTYPE | Catalog type. X'F0' |
| 2(2) | Character | 44 | CSICNAME | Catalog name |
| 46(2E) | Character | 0 | CSICRETN | Return information for Catalog. |
| 46(2E) | Character | 2 | CSICRETM | Catalog return module ID |
| 48(30) | Fixed | 1 | CSICRETR | Catalog return reason code |
| 49(31) | Fixed | 1 | CSICRETC | Catalog return code |
| **Information returned for each entry** | | | | |
| 0(0) | Bitstring | 1 | CSIEFLAG | Entry flag information. |
| | 1... .... | | CSIPMENT | Primary entry. |
| | 0... .... | | | This entry associates with the preceding primary entry. |
| | .1.. .... | | CSIENTER | Error indication is set for this entry and error code follows CSIENAME. |
| | ..1. .... | | CSIEDATA | Data is returned for this entry. |
| | ...1 1111 | | | Reserved. |
| 1(1) | Character | 1 | CSIETYPE | Entry Type.<br><br>A Non-VSAM data set<br>B Generation data group<br>C Cluster<br>D Data component<br>G Alternate index<br>H Generation data set<br>I Index component<br>L ATL Library entry<br>R Path<br>U User catalog connector entry<br>W ATL Volume entry<br>X Alias |

*Table 21. Work Area Format Table (continued)*

| Offset | Type | Length | Name | Description |
|---|---|---|---|---|
| 2(2) | Character | 44 | CSIENAME | Entry name. |
| 46(2E) | Character | 0 | CSIRETN | Error return information for entry. Only exists if CSIENTER is 1. |
| 46(2E) | Character | 2 | CSIERETM | Entry return module ID |
| 48(30) | Fixed | 1 | CSIERETR | Entry return reason code |
| 49(31) | Fixed | 1 | CSIERETC | Entry return code |
| 46(2E) | Character | VL | CSIEDATA | Returned data for entry. Only exists if CSIENTER is 0. |
| **If CSIOPTNS is not F:** | | | | |
| 46(2E) | Character | 2 | CSITOTLN | Total length of returned information including this field and length fields. The next entry begins at this offset plus this length. |
| 48(30) | | 2 | | Reserved. |
| 50(32) | Character | VL | CSILENFD | Length of fields. There is one length field returned for each field name passed on input. |
| 50(32) | Character | 2 | CSILENF1 | First length field. |
| **If CSIOPTNS is F:** | | | | |
| 46(2E) | Character | 4 | CSITOTLN | Total length of returned information including this field and length fields. The next entry begins at this offset plus this length. |
| 50(32) | | 4 | | Reserved. |
| 54(36) | Character | VL | CSILENFD | Length of fields. There is one length field returned for each field name passed on input. |
| 54(36) | Character | 4 | CSILENF1 | First length field. |
| **Information returned for each field name** | | | | |
| 0(0) | Character | VL | CSIFDDAT | Field data. For each field name passed on input, there will be a data item corresponding to its length. |
| | | | | The next entry would begin here if more than 1 entry is returned. |
| | | | | The next catalog entry would begin after all of the entries, if more than 1 catalog is searched. |

## Work Area Format Picture




## Work Area Format Description

The first field is the length of the work area and is set by the user to tell CSI how much data can be returned.

The second field is the minimum required length for a catalog entry and one entry's worth of returned data. If the minimum length is greater than the work

area length, then the work area length must be increased to at least as much as the minimum length. If this is not done, and a resume condition occurs, the user program will appear as if in an endless loop because the same information will be returned for each resume until the first length is increased to contain the entire entry. A generation data group (GDG) with all of its associated generation data sets (GDS), or a nonVSAM data set or user catalog connector with thousands of associated aliases is seen by the CSI as one record. Thus the required length for these entries is apt to be large.

The third field is the amount of space that was used in the work area. CSI always returns a full entry. If the last entry will not fully fit in the remaining work area space, the resume flag is set and the space at the end of the work area is unused. The unused space will usually be small.

Next follows the number of field names plus 1.

A catalog name entry is returned for every catalog processed. A catalog entry may be identified because its type is x'F0'. This is an artificial type invented so that the next catalog entry can be found. The catalog entry is always followed by return information. The return code portion will be zeroes if no problems were encountered while processing the catalog during the call.

Following the catalog entry is one or more entries contained in the catalog that match the search criteria (filter key). Each entry has flags, followed by its type and name. If the flags indicate, a module id, reason code and return code follow the entry name; otherwise, the field information for the entry follows.The flag byte in front of the entry type will also indicate possible errors that were encountered.

The "MODULE ID / RSN / RC" returned in the work area is returned when an error is detected by Catalog Management. This field only exists when the flag CSIENTER is set for this entry; otherwise, it is not present. See the previous section entitled "Return Codes", subsection "Return Code 4", for a description of this information.

If no errors or messages occurred, then field information for the entry is returned as a set of lengths and then the data corresponding to the lengths.

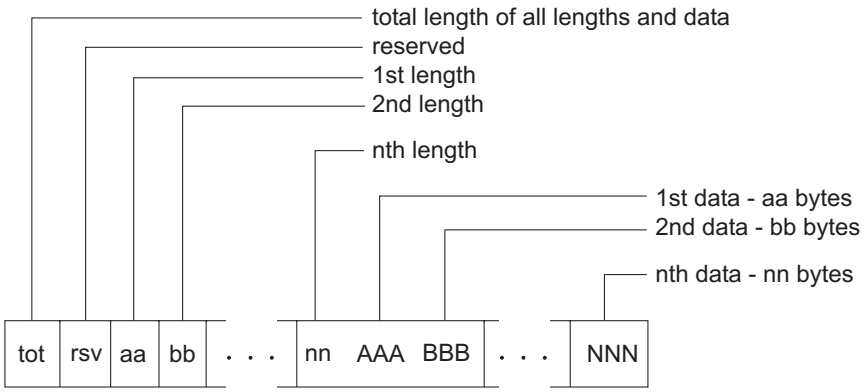
The first length field is the length of all of the returned data for this entry. It is the total of the length of the field itself, all of the length fields following it (one per fieldname supplied in CSIFLDNM), and the length of the actual data returned for this entry. The total length field is two bytes long, unless CSIOPTNS was specified as F, in which case this length field (and all remaining length fields) are four bytes long. The reserved field following this total length is also two or four bytes long.

Next is a set of lengths corresponding to the number of fields passed in.The length fields are two bytes long if CSIOPTNS is blank and four bytes long if CSIOPTNS is F. Each length is used to determine the length and position of the returned data for the entry.

For example, if three field names were supplied on input, then there will be three field lengths. Each length will be for the data immediately following the lengths. If the lengths had values 4, 6, and 8, then following the last length, there would be 4-bytes worth of data for the first field, 4-bytes from the last length field would be 6 bytes of data for the second field, and 10 bytes from the last length would be 8 bytes worth of data for the third field. Each length is set to:

- -1: If the data to be retrieved is suppressed; security data will be suppressed if the caller does not have the proper RACF authority.
- 0: If no data was found for this entry. This can happen, for instance, if the data does not apply to the particular entry type being supplied.
- 2 or 4: If the data field to be retrieved has variable length and does not exist. Data with variable length is always returned with two- or four-byte length information preceding the data (depending on the setting of CSIOPTNS) and is included as part of data. Therefore, when the field to be retrieved does not exist, this preceding length information is set to 0 and the total length of the data returned will then be the length of this preceding length information.
- n: If the data field to be retrieved has fixed length and does not exist. n is the length of the fixed length data and the data is set to be all 'FF'X.
- n: If the data is retrieved and is the total length of the data retrieved.

The following illustration shows the general relationship of the length fields and their corresponding data.




**Restriction:** Total information returned for an entry cannot exceed 65,535 characters unless CSIOPTNS is set to F (use fullword lengths). If CSIOPTNS is not set to F, any attempt to retrieve information greater than 65,535 characters results in setting the CSIENTER flag in that entry and the CSI request will continue with the next data set (if any). When the CSI request is complete, the return code will be 100 and the reason code will be 4, unless a more serious error occurs. For the entry with the CSIENTER flag set, the CSIRETN value is returned; the return code is 44 and the reason code is 14. For example, a request for the NAME fieldname of a user catalog (or data set) that has more than 1489 aliases defined would result in this error.

## Field Name Directory

These are valid field names that can be used in CSIFLDNM. The information returned for each field name is given in the description.

The REP column refers to fields that can repeat when returned by CSI.

### Catalog Field Names

Table 22 shows the catalog field names.

*Table 22. Catalog Field Names*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Character | 36 | ACTOKEN | Active compression dictionary token |

*Table 22. Catalog Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Fixed | 2 | AKEYPOS | The relative position in the data record of this AIX® key. Only applicable for catalog entry types of AIX. Note that the field is only valid if the component type is "D" and the record type is for a alternate index. |
| no | Fixed | 8 | AMDCIREC | Control interval size for 4 bytes and maximum record size for 4 bytes |
| no | Fixed | 4 | AMDKEY | Relative position of KSDS key for 2 bytes and key length of KSDS key for 2 bytes |
| yes | Character | 45 | ASSOC | A repeating list of catalog records associated with this entry. Consists of a 1-byte value similar to field name ENTYPE, followed by the 44-byte name of the association. If the name contains system symbolics, the name and type are resolved before being returned. |
| no | Bitstring | 1 | ASSOCSYB | Indicates whether the entry is a symbolic.<br>• X'00' - The entry is not a symbolic<br>• X'80' - The entry is a symbolic |
| yes | Character | 45 | ASSOCSYM | A repeating list of catalog records associated with this entry. Consists of a 1-byte value similar to field name ENTYPE, followed by the 44-byte name of the association. If the name contains system symbolics, they are not resolved. |
| no | Bitstring | 1 | ATTR1 | Attributes: |
| | 1... .... | | | Speed |
| | .1.. .... | | | Unique |
| | ..1. .... | | | Reusable |
| | ...1 .... | | | Erase |
| | .... 1... | | | ECSHARING - ICF catalogs only |
| | .... .1.. | | | Inhibit update |
| | .... ..1. | | | Temporary export |
| | .... ...1 | | | Track overflow |
| no | Bitstring | 1 | ATTR2 | Share attributes |
| | 11.. .... | | | Region (00 = 1, 01 = 2, 10 = 3, 11 = 4) |
| | ..11 .... | | | System (00 = 1, 01 = 2, 10 = 3, 11 = 4) |
| | .... xxxx | | | Reserved |
| no | Fixed | 2 | BUFND | The number of buffers requested for Data |
| no | Fixed | 2 | BUFNI | The number of buffers requested for Index |
| no | Fixed | 4 | BUFSIZE | Maximum buffer size |
| no | Bitstring | 1 | CATTR | Attributes for pagespace and swapspace |
| | xxxx xx.. | | | Reserved |
| | .... ..1. | | | Swap=1, noswap=0 |
| | .... ...1 | | | Data set is a pagespace |
| no | Bitstring | 1 | COMPIND | Compression indicator. |
| | x..x xxxx | | | Reserved |

*Table 22. Catalog Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| | .1.. .... | | | Data set is extended format |
| | ..1. .... | | | Data set is compressible |
| no | Fixed | 8 | COMUDSIZ | Compressed user data size |
| no | Character | VL | DATACLAS | SMS data class |
| yes | Fixed | 4 | DEVTYP | UCB device type |
| yes | Fixed | 3 | DSCBTTR | TTR of format-1 DSCB for non-VSAM data set |
| no | Mixed | 4 | DSCRDT2 | Creation date. Packed decimal YYDDDF for 3 bytes appended with one byte century indicator. If the century byte is 00 then add 1900 to get the year; if 01, add 2000. |
| no | Mixed | 4 | DSEXDT2 | Expiration date. Packed decimal YYDDDF for 3 bytes appended with one byte century indicator. If the century byte is 00 then add 1900 to get the year; if 01, add 2000. |
| no | Binary | 1 | EATTR | Data set attribute for controlling allocation of VSAM data sets (note that EATTR for Non-VSAM data sets is not carried in the catalog information for such data sets). The value of EATTR is as follows:<br>• X'00' - EATTR not specified. Defaults for EAS eligibility should apply.<br>• X'01' - EATTR=NO specified. The data set **cannot** have extended attributes have and cannot reside in EAS.<br>• X'02' - EATTR=OPT specified. The data set **can** have extended attributes and can optionally reside in an EAS.<br>• X'03' - Not used. EATTR value treated as not specified. |
| no | Character | 44 | ENTNAME | The name of the entry |
| no | Character | 1 | ENTYPE | Entry type, ex., 'C' is cluster, 'A' is non-VSAM, etc. |
| no | Character | 8 | EXCPEXIT | Exception exit |
| yes | Fixed | 2 | FILESEQ | File sequence number |
| no | character | 1 | FSDSFLAG | File System Data Set Flag<br><br>X'00' - Not a zFS data set<br>X'80' - zFS data set<br>X'FF' - Attribute not applicable to entry |
| no | Mixed | 4 | GDGALTDT | Last alteration date. Packed decimal YYDDDF for 3 bytes appended with one byte century indicator. If the century byte is 00 then add 1900 to get the year; if 01, add 2000. |
| no | Bitstring | 1 | GDGATTR | Generation data group attributes |
| | 0... .... | | | Delete oldest GDS when GATLIMIT exceeded |
| | 1... .... | | | Delete all GDSs when GATLIMIT exceeded |
| | .0.. .... | | | Do not scratch data set when rolled off |
| | .1.. .... | | | Scratch data set when rolled off if volume mounted |

*Table 22. Catalog Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Fixed | 1 | GDGLIMIT | Maximum number of generation data sets allowed in the GDG |
| yes | Character | 4 | GENLEVEL | GDG generation level — 1 for each active generation in EBCDIC format c'0000' |
| yes | Fixed | 4 | HARBA | High-allocated RBA |
| no | Fixed | 4 | HARBADS | Data set high-allocated RBA |
| yes | Character | VL | HIKEYV | High Key on volume |
| no | Fixed | 4 | HILVLRBA | RBA of High Level Index Record |
| yes | Fixed | 4 | HKRBA | RBA of data control interval with high key |
| yes | Fixed | 4 | HURBA | High-used RBA for the volume requested |
| no | Fixed | 4 | HURBADS | Data set high-used RBA |
| no | Fixed | 2 | INDXLVLS | Number of Index Levels |
| yes | Bitstring | 1 | ITYPEXT | Type of extent |
| | 1... .... | | | Sequence set with data, IMBED |
| | .1.. ... | | | Extents not preformatted |
| | ..1. .... | | | Converted VSAM data set volume |
| | ...x xxxx | | | Reserved |
| no | Binary | 1 | LOGPARMS | Value of LOG parameter set by IDCAMS DEFINE/ALTER<br>X'00' - never set<br>X'01' - LOG(NONE)<br>X'02' - LOG(UNDO)<br>X'03' - LOG(ALL) |
| no | Character | 26 | LOGSTRID | Value of LOGSTREAMID parameter set by IDCAMS DEFINE/ALTER |
| yes | Character | VL | LOKEYV | Low Key on volume |
| no | Fixed | 4 | LRECL | Average logical record size |
| no | Fixed | 8 | LTBACKDT | Last backup date in TOD format. |
| no | Character | VL | MGMTCLAS | SMS management class |
| yes | Character | 44 | NAME | The name of an associated entry |
| yes | Fixed | 2 | NOBLKTRK | Number of physical blocks per track. This is the value reported by IDCAMS LISTCAT as PHYRECS/TRK |
| yes | Fixed | 4 | NOBYTAU | Number of bytes per allocation unit |
| yes | Fixed | 4 | NOBYTTRK | Number of bytes per track |
| yes | Fixed | 1 | NOEXTNT | Number of extents. This is the value reported by IDCAMS LISTCAT as EXTENTS. |
| yes | Fixed | 2 | NOTRKAU | Number of tracks per allocation unit. This is the value reported by IDCAMS LISTCAT as TRACKS/CA. |
| no | Character | 1 | NVSMATTR | Non-VSAM attribute information |
| | c'H' | | | Active GDS |
| | c'N' | | | Deferred GDS |

*Table 22. Catalog Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| | c'M' | | | Rolled-off GDS |
| | c'L' | | | Extended partitioned data set (PDSE) |
| | c'P' | | | POSIX data set |
| | x'00' | | | Simple non-VSAM data set |
| | c'I' | | | Active GDS-PDSE |
| | c'J' | | | Deferred GDS-PDSE |
| | c'K' | | | Rolled-Off GDS-PDSE |
| no | Bitstring | 1 | OPENIND | Open indicator |
| | x... .... | | | Open =1, not open=0, 1 may mean that the data set was not closed properly |
| | .xxx xxxx | | | Reserved |
| no | Character | 8 | OWNERID | Owner of the data set |
| no | Fixed | 2 | PASSATMP | Number of attempts to prompt for password |
| no | Character | 8 | PASSPRMT | Password prompt code name |
| no | Character | 32 | PASSWORD | Four 8-byte passwords (VSAM data sets only) |
| yes | Fixed | 4 | PHYBLKSZ | Physical blocksize. This is the value reported by IDCAMS LISTCAT as PHYREC-SIZE . |
| no | Fixed | 3 | PRIMSPAC | Primary space allocation |
| no | Binary | 8 | RECVTIME | Recovery time, TOD value, local |
| no | Binary | 8 | RECVTIMG | Recovery time, TOD value, GMT |
| no | Bitstring | 1 | RGATTR | Alternate index/path attributes |
| | 1... .... | | | Upgrade=1, noupgrade=0 |
| | .1.. .... | | | Entry is an alternate index |
| | ..xx xxxx | | | Reserved |
| no | Bit | 1 | RLSBWO | Value of BWO parameter set by IDCAMS DEFINE/ALTER<br>xxxx 0000 - undefined or not set<br>xxxx 0001 - BWO(TYPECICS)<br>xxxx 0010 - reserved<br>xxxx 0011 - BWO(TYPEIMS)<br>xxxx 0100 - reserved |
| no | Bit | 1 | RLSFLAGS | xxxx 0xxx - Recovery not required<br>xxxx 1xxx - Recovery required<br>xxxx x0xx - Catalog is Enabled for RLS<br>xxxx x1xx - Catalog is Quiesced for RLS<br>xxxx xx0x - Catalog is not being used in RLS mode<br>XXXX XX1X - Catalog is being used in RLS mode |
| no | Fixed | 4 | SEQSTRBA | RBA of First Sequence Set Record. RRDS maximum record number if RRDs. |
| no | Fixed | 3 | SCONSPAC | Secondary space allocation |
| no | Bitstring | 1 | SECFLAGS | Security flag information x'80' means the data set has a discrete RACF profile |

*Table 22. Catalog Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Bitstring | 1 | SMSSFLAG | SMS FLAGS |
| | 1... .... | | | VSAM extended format |
| | .1.. .... | | | VSAM compressed format |
| | ..1. .... | | | RLS in use |
| | ...1 .... | | | RLS VSAM quiesced |
| | .... xxxx | | | Reserved |
| no | xx.. .... | | SPACOPTN | Equals '01' for record allocation, '10' for track allocation, and '11' for cylinder |
| | ..xx xxxx | | | Reserved |
| no | Character | VL | STORCLAS | SMS storage class |
| no | Fixed | 2 | STRIPCNT | Striping counts for striped data sets |
| no | Fixed | 1 | STRNO | Number of concurrent requests |
| yes | Fixed | 4 | TRACKS | Total number of tracks per volume. This field pertains only to VSAM data sets. |
| yes | Character | 1 | TYPE | The type of an associated entry |
| no | Fixed | 8 | UDATASIZ | User data size<br>**Restriction:** This field is only valid for extended format VSAM and non-VSAM data sets. |
| no | Character | VL | USERAREC | User authorization record |
| no | Character | 8 | USVRMDUL | User security verification module |
| yes | xxx. .... | | VOLFLG | '100' is the primary volume with space allocated, '010' is the candidate volume with no space allocated, '001' is the overflow volume (keyrange data set only) with no space allocated |
| yes | Character | 6 | VOLSER | Volume serial number. A VOLSER of all asterisks is the IPL volume. For a symbolic value (for example, "&xxxxx"), use the ASASYMBM service to convert the symbolic value to a valid character string. |
| no | Bitstring | 1 | VSAMREUS | VSAM data set information |
| | 1... .... | | | Data set has RACF discrete profile |
| | .1.. .... | | | Index component data set |
| | ..1. .... | | | Reusable data set |
| | ...1 .... | | | Erase specified (cluster only) |
| | .... xx.. | | | Reserved |
| | .... ..1. | | | Swap space (cluster only) |
| | .... ...1 | | | Page space (cluster only) |
| no | fixed | 46 | VSAMSTAT | Statistics infomation for VSAM components. |
| | | 1 | | Percentage of free CIs in CA |
| | | 1 | | Percentage of bytes free in CI |
| | | 2 | | Number CIs/ CA |
| | | 4 | | Free CIs/ CA |
| | | 2 | | Free bytes/CI |

*Table 22. Catalog Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| | | 4 | | Number of logical records |
| | | 4 | | Number of deleted records |
| | | 4 | | Number of inserted records |
| | | 4 | | Number of updated records |
| | | 4 | | Number of retrieved records |
| | | 4 | | Bytes of free space in component. For non-extended addressability data sets, this field represents the actual amount of free space in the component. If the data set is extended addressable, the value in the field is the number of free CIs; the bytes of freespace can be obtained by multiplying the field value by the CI size. |
| | | 4 | | Number of CI splits |
| | | 4 | | Number of CA splits |
| | | 4 | | Number of EXCPs |
| no | Bitstring | 2 | VSAMTYPE | VSAM data set type information |
| | First Byte: | | | |
| | 1... .... | | | KSDS=1, not KSDS=0 |
| | .1.. .... | | | Write check |
| | ..1. .... | | | Imbed |
| | ...1 .... | | | Replicate |
| | .... x... | | | Reserved |
| | .... .1.. | | | Key-range data set |
| | .... ..1. | | | RRDS |
| | .... ...1 | | | Spanned records allowed |
| | Second Byte: | | | |
| | 1... .... | | | Non-unique or unique keys allowed |
| | ...x x.x. | | | Reserved |
| | .x.. .... | | | • 0=CA-RECLAIM(YES),<br>• 1=CA-RECLAIM(NO) |
| | ..1. .... | | | The data set was not closed properly and the recorded statistics are not accurate. |
| | .... .1.. | | | LDS |
| | .... ...1 | | | VRRDS |
| no | Bitstring | 2 | VVRNFLGS | Extended format flags |
| | First Byte: | | | |
| | 1... .... | | | COMUDSIZ and UDATASIZ are invalid |
| | .1... .... | | | Block level compression |
| | ..xx xxxx | | | Reserved |
| | Second Byte: | | | |
| | 1111 1111 | | | Not defined |
| no | Bitstring | 1 | XACIFLAG | Extended attribute flags |

*Table 22. Catalog Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| | x... .... | | | Reserved, may be on |
| | .1.. .... | | | Data set can be greater than 4 GB |
| | ..xx .... | | | Reserved, may be on |
| yes | Fixed | 8 | XHARBA | High-allocated RBA |
| no | Fixed | 8 | XHARBADS | Data-set high-allocated RBA |
| yes | Fixed | 8 | XHKRBA | RBA of data control interval with high key |
| yes | Fixed | 8 | XHURBA | High-used RBA for the volume requested |
| no | Fixed | 8 | XHURBADS | Data-set high-used RBA |

**Note:** If you attempt to retrieve a 4-byte RBA value (such as, HARBA, HURBA, HARBADS, HURBADS, or HKRBA) and the value will not fit in the 4-bytes provided, that length of that returned data will be zero as shown under "Work Area Format Description" in Appendix D. You can either change to always request the extended fields shown above, or request the setting of XACIFLAG and inspect bit 1 to determine whether or not RBAs can be greater than 4 bytes. If so, then request the fields with the names given here.

## Library Entry Field Names

These names are only valid for tape volume catalogs in DFSMS/MVS.

The REP column refers to fields that can repeat when returned by CSI.

*Table 23. Library Entry Field Names*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Character | 8 | LCBCONID | Library console identification |
| no | Character | 120 | LCBDESCR | Library Description |
| no | Character | 8 | LCBDEVTP | Library device type |
| no | Fixed | 4 | LCBEMPTY | Number of empty slots |
| no | Character | 5 | LCBLIBID | Library Identification |
| no | Flag | 1 | LCBLOGIC | Library logic type |
| no | Fixed | 1020 | LCBSCRTH | Number of scratch volumes for all 255 media types |
| no | Fixed | 4 | LCBSLOTS | Number of slots |
| no | Fixed | 1020 | LCBTHRES | Library scratch threshold for all 255 media types |

## Volume Entry Field Names

These names are only valid for tape volume catalogs in DFSMS/MVS.

The REP column refers to fields that can repeat when returned by CSI.

*Table 24. Volume Entry Field Names*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Character | 1 | VCBCHKPT | Volume checkpoint |
| | | | "Y" | Yes |
| | | | "N" | No |
| | | | " " | Unknown |
| no | Character | 10 | VCBCRDT | Volume creation date, YYYY-MM-DD |

*Table 24. Volume Entry Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Group item | 4 | VCBDEVTP | Volume device type |
| | Fixed | 1 | | Recording Technology<br><br>- Not defined (0)<br>- 18 tracks (1)<br>- 36 tracks (2)<br>- 128 tracks (3)<br>- 256 tracks (4)<br>- 384 tracks (5)<br>- EFMT1 (6) |
| | Fixed | 1 | | Media Type<br><br>- Not defined (0)<br>- Media 1 (1)<br>- Media 2 (2)<br>- Media 3 (3)<br>- Media 4 (4)<br>- Media 5 (5)<br>- Media 6 (6)<br>- Media 7 (7)<br>- Media 8 (8) |
| | Fixed | 1 | | Compact Type<br><br>- Not Defined (0)<br>- No Compaction (1)<br>- IDRC (2) |
| | Fixed | 1 | | Special Attribute<br><br>- Not Defined (0)<br>- Read Compatible (1) |
| no | Character | 10 | VCBEDATE | Volume entry/eject date, YYYY-MM-DD |
| no | Bitstring | 2 | VCBERRST | Volume error status |
| no | Character | 10 | VCBEXPDT | Volume expiration date, YYYY-MM-DD |
| no | Character | 44 | VCBLIBNM | Volume library name |
| no | Character | 1 | VCBLOC | Volume location |
| | | | "L" | Library |
| | | | "S" | Shelf |
| | | | " " | Unknown |
| no | Character | 10 | VCBMOUNT | Volume last mount date, YYYY-MM-DD |
| no | Character | 64 | VCBOWNER | Volume owner information |
| no | Character | 8 | VCBSGRP | Volume storage group |
| no | Character | 32 | VCBSHELF | Volume shelf location |
| no | Flag | 1 | VCBUATTR | Volume user attribute |
| | | | "P" | Private |
| | | | "S" | Scratch |
| | | | " " | Unknown |
| no | Flag | 1 | VCBWPROT | Volume write protection status |

*Table 24. Volume Entry Field Names (continued)*

| Rep | Type | Length | Name | Description |
|---|---|---|---|---|
| no | Character | 10 | VCBWRITE | Volume last written date, YYYY-MM-DD |

## Sample Programs

Three sample assembler programs and one sample REXX EXEC come with the IGGCSI00 module. These sample programs are intended to let you try out CSI without having to write your own program. You should be able to get CSI up and running using these programs with only a few JCL changes. You may want to modify them to fit your specific needs.

### IGGCSILC

IGGCSILC is an assembler program that produces output similar to Access Method Services (IDCAMS) LISTCAT NAME. Try running it against an IDCAMS LISTCAT NAMES.

Change the Binder/Linkage Editor SYSLIN DD statement to point to the linkage library for your installation.

The input to IGGCSILC is an 80-byte SYSIN DD record. The catalog name of the catalog to be listed should be left justified in column 1 of the record.

A listing of all entry names in the catalog is printed along with a summary of the number each entry type found and the total of all entries found.

### IGGCSIVG

IGGCSIVG is an assembler program that identifies unused space at the end of a VSAM data set. Basically, it computes the difference of the high-used and the high-allocated relative byte addresses (RBAs) for each VSAM data set in a given catalog. This program is useful in identifying over-allocated space.

Change the Binder/Linkage Editor SYSLIN DD statement to point to the linkage library for your installation. IGGCSIVG will run only with releases of DFSMS.

The input to IGGCSIVG is an 80-byte SYSIN DD record. The catalog name of the catalog to be processed must be left justified and start in column 1 of the input record. Multiple input records can be supplied.

IGGCSIVG prints a summary for each catalog name showing the total unused space by VSAM type. A total for all catalog names supplied is printed after the last catalog name is processed.

### IGGCSIVS

IGGCSIVS is an assembler program that identifies which data sets on a given volume reside in a particular catalog. In the event of a disk drive failure, this program would be useful in identifying which entries in a catalog need to be cleaned up if the data sets were recovered to a different volume serial number.

Change the Binder/Linkage Editor SYSLIN DD statement to point to the linkage library for your installation.

The input to IGGCSIVS is an 80-byte SYSIN DD record. The six- character volume serial number should be left justified and starts in column 1 of the input record. The catalog name of the catalog to be searched starts in column 7. Multiple input records can be supplied.

The output is a listing of all data sets that reside in the given catalog and are on the given volume.

## IGGCSIRX

IGGCSIRX is a REXX EXEC that uses CSI. Move this EXEC to a REXX EXEC library and ensure that the linkage library for IGGCSI00 is accessible by the TSO session. IGGCSIVS will run with any release of DFSMS supported by CSI.

When executed, IGGCSIRX will prompt the user for a filter key. This should be a partially qualified data set name as described for the selection criteria field CSIFILTK. The data set name, its type, and volume serial number(s) are returned to the user's TSO session.

# Chapter 12. Detecting Obsolete Catalog Attributes with IBM Health Checker for z/OS

You can use the CATALOG_IMBED_REPLICATE check running in the IBM Health Checker for z/OS framework to detect instances of the obsolete IMBED and REPLICATE attributes for user and master catalogs. No supported releases of z/OS honor either the IMBED or REPLICATE attributes for new catalogs, they are obsoleted by newer, cached DASD devices. These attributes were obsoleted because they waste DASD space and degrade performance, in some cases causing unplanned outages. In addition, servicing catalogs with these attributes is very difficult.

If the check finds instances of IMBED or REPLICATE attributes, the system issues exception message IGGHC104E and generates a report in message IGGHC106I in the message buffer to describe the check's findings. IBM suggests that you use the EXPORT/IMPORT command to remove the attributes:

- Use the EXPORT command to create a back up and later to recover.
- Use the IMPORT command for the exported copies.

Ideally, you should do this during system down time, when the catalogs cannot be accessed by any users.

This check is shipped as active and, by default, runs once a day. However, once you have identified all the catalogs you need to redefine without IMBED and REPLICATE attributes, IBM suggests that you turn the check off as soon as possible. This is recommended because:

- Users can no longer define catalogs with the obsolete IMBED or REPLICATE attributes. That means that once you have identified any existing catalogs that were defined with IMBED and REPLICATE and redefined them, it is no longer useful to run this check.
- Leaving the check on after you have identified and/or redefined any catalogs defined with IMBED or REPLICATE attributes, the check can cause a performance issue. the check does a sequential search of the active Master Catalog, which means it reads all the records in the master catalog. This I/O to the master catalog to read all the records can impact performance.

You can turn the check off temporarily using F HZSPROC command or permanently using the HZSPRMxx parmlib member.

Note that catalog requests that result in sequential processing of a master catalog will not add or update records in VLF. Catalog requests that kick off direct processing of a master catalog however, **will** add or update records in VLF.

**Related information**:

- To see a full description of the check, see CATALOG_IMBED_REPLICATE in *IBM Health Checker for z/OS: User's Guide*.
- To set up and start using IBM Health Checker for z/OS, see Setting up IBM Health Checker for z/OS in *IBM Health Checker for z/OS: User's Guide*.

  Once the IBM Health Checker for z/OS is up and running, an exit routine automatically adds the check to the system and the check will run once every 24 hours or at another interval you specify.

To modify check attributes, such as the interval or severity for the check, do one of the following:

- Make temporary, dynamic changes using either SDSF or the `F hzsproc,UPDATE` command.
- Make permanent changes for the check in an HZSPRMxx parmlib member.

See Managing checks in *IBM Health Checker for z/OS: User's Guide*.

- To use the EXPORT and IMPORT commands, see *z/OS DFSMS Access Method Services Commands* for information on the EXPORT and IMPORT commands.

# Chapter 13. Accessing Catalogs for Record Level Sharing (RLS)

## Activating the SMSVSAM Address Space

RLS exploitation requires a Parallel Sysplex®, and the SMSVSAM address space available on each LPAR where RLS access is required. Refer to *z/OS DFSMSdfp Storage Administration* for details on enabling the SMSVSAM address space.

## Enabling a Catalog in RLS Mode

Only SMS-managed catalogs residing on SMS-managed volumes can be used for RLS. To make a catalog RLS-eligible, a LOG parameter is available on the IDCAMS DEFINE and ALTER commands. Specify LOG(NONE) to make a catalog eligible for using the RLS protocol, as opposed to the VVDS or the ECS protocol. However, the system will not actually use the RLS protocol unless:

- RLSENABLE attribute is specified AND
- The STORCLAS and its cache set information are specified AND
- The SMSVSAM address space is active on the system AND
- The catalog is not currently open in either VVDS mode or ECS mode from any system in release 1.13 of z/OS or below.

You can alter the LOG parameter of a catalog at any time. If you nullify the parameter or specify RLSQUIESCE and the catalog is currently using RLS mode, it will be converted back to either VVDS or ECS mode on all systems that are sharing it. This is convenient if you find that the catalog must be accessed by a system that does not support the RLS protocol.

## Restrictions on RLS Mode Usage

There are several restrictions on using RLS mode:

- RLS protocol cannot be used simultaneously with either ECS or VVDS protocol for a catalog. This is enforced by the SMSVSAM address space. If you attempt to use a catalog currently in RLS mode from a non-RLS system in the sysplex, the associated catalog request will fail with return code RC4 and reason code RSN124. If a catalog is currently in either ECS mode or VVDS mode and is shared by a system that cannot support RLS for catalogs, the attempt to use the catalog in RLS mode will also fail with return code RC4 and reason code RSN124.

  **Note:** If you attempt to use a catalog that is currently RLS-active from a system outside the sysplex, the request might break the catalog.
- No more than 1024 catalogs can currently be opened from any single system using RLS.
- All systems sharing the catalog in RLS mode must be on the release that supports RLS for catalogs and have the SMSVSAM address space available.

## Activating RLS

To activate RLS, perform the following steps:

1. Define SHCDS data sets. Define CF cache structure(s) and CF lock structure(s) in the CFRM policy and activate the policy.
2. Activate the SMSVSAM address space by either specifying RLSINIT(YES) in SYS1.PARMLIB(IGDSMSxx) or issuing V SMS,SMSVSAM,ACTIVE command.
3. Specify the STORCLAS attribute and its cache set for the desired catalogs to be in RLS mode.
4. Specify the LOG(NONE) parameter for the desired catalog to be in RLS mode.
5. Specify RLSENABLE from IDCAMS DEFINE or ALTER, or MODIFY CATALOG,RLSENABLE(catname)

**Note:** RLSENABLE and the LOG parameter may be specified together, but RLSENABLE cannot be specified without first specifying the LOG parameter.

## Operational Considerations

Although a catalog's VSAM share options are ignored when it is in RLS mode, they are still used when the catalog is quiesced from RLS use. Therefore, it is necessary to ensure that correct share options are specified. For example, a catalog to be shared across the sysplex in RLS mode needs to be defined with SHAREOPTIONS(3 4) so that it can continue to be shared across the sysplex correctly if it is ever switched out of RLS mode and back to either ECS mode or VVDS mode.

- It is recommended that prior to enabling catalogs in RLS mode, all systems in the sysplex (including pre z/OS 2.1 systems) have enabled SMSVSAM. The SMSVAM address space must be available in a fallback situation in order to switch catalogs out of RLS mode. It is also recommended that all systems in the sysplex have migrated to z/OS 2.1 prior to implementing RLS for Catalog to prevent accidentally accessing a catalog from a release that does not support RLS mode. If a system that does not support RLS for catalogs must remain in the sysplex, be aware that if a catalog in RLS mode is ever RLS-quiesced for any reason, the non-RLS system could start accessing the catalog in either ECS or VVDS mode. In this case, you will need to explicitly close the catalog from the non-RLS system before enabling the catalog back to RLS mode.
- It is necessary to stop RLS mode for a catalog if access is needed by a system that does not support the RLS protocol. This can be temporarily accomplished by using the MODIFY CATALOG,RLSQUIESCE(catname) command. This does not nullify the LOG parameter, but does remove the catalog from RLS mode on all systems. It can only be returned to RLS mode by issuing a MODIFY CATALOG,RLSENABLE(catname) command. To switch all catalogs used by this system from RLS to non-RLS mode, issue MODIFY CATALOG,RLSQUIESCE,SYSTEM. To switch all RLS-eligible catalogs from non-RLS back to RLS mode, issue MODIFY CATALOG,RLSENABLE,SYSTEM.
- The RLSENABLE and RLSQUIESCE interfaces are intended for you to enable your catalogs to RLS mode and only fallback to non-RLS mode during an emergency. They are not intended to be used to switch a catalog back and forth between RLS and non-RLS on a regular basis.
- It is recommended that you dedicate a separate CF cache structure and lock structure for any catalog in RLS mode for better performance and failure isolation.
- To protect catalog's integrity, the SYSIGGV2 BCS resource will be held share by each SMSVSAM address space while a catalog is opened for RLS access. This is done to ensure that applications that explicitly raise the ENQ exclusive in order to have the exclusive access to the catalog can no longer do so. The SYSIGGV2 BCS resource will be released when the catalog is closed for RLS access.

- If you plan to terminate the SMSVSAM address space for a long time, and your catalog requests cannot tolerate the prolonged down time, consider quiescing the catalogs from the RLS access before terminating the SMSVSAM address space. Otherwise, if the SMSVSAM address space is down, an IEC365D message will be displayed on the console to indicate that some catalog requests are still waiting for the SMSVSAM address space to be available. The catalog requests will be waiting on SYSZIGG5 BCS ENQ in the user's address space as long as the SMSVSAM is not available. You may decide to ignore the message if the SMSVSAM address space is going to be available soon, or issue RLSQUIESCE to switch all catalogs used by this system out of RLS mode. In this case once the SMSVSAM address space is available, you may issue MODIFY, CATALOG,RLSENABLE,SYSTEM to switch all RLS-eligible catalogs back to RLS mode.
- During a catalog hang, it is essential to capture the dumps from CAS, SMSVSAM, and MASTER address spaces from all systems in the sysplex. Simply issue MODIFY CATALOG,TAKEDUMP(SYSPLEX) to capture all dumps. You may need to increase your dump data sets to accommodate the addition of dumps from the SMSVSAM address space and its dataspaces, especially if there are buffers placed above the bar.
- To identify which job(s) may be contributing to a catalog hang, use the combinations of the following console commands:

  ```
  MODIFY CATALOG,LIST
  DISPLAY GRS,C
  DISPLAY SMS,SMSVSAM,DIAG(CONTENTION)
  ```

  If a diagnosis is impossible, restart both the CAS and SMSVSAM address space from all involved systems.
- Although a catalog's VSAM share options are ignored when it is in RLS mode, they are still used when the catalog is quiesced from RLS use. Therefore, it is necessary to ensure that correct share options are specified. For example, a catalog to be shared across the sysplex in RLS mode needs to be defined with SHAREOPTIONS(3 4) so that it can continue to be shared across the sysplex correctly if it is ever switched out of RLS mode and back to either ECS mode or VVDS mode.

# Appendix. Accessibility

Accessible publications for this product are offered through the z/OS Information Center, which is available at www.ibm.com/systems/z/os/zos/bkserv/.

If you experience difficulty with the accessibility of any z/OS information, please send a detailed message to mhvrcfs@us.ibm.com or to the following mailing address:

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

## Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

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

## Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

## Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

## Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing the z/OS Information Center using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually

exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

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

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

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

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

The following words and symbols are used next to the dotted decimal numbers:

- ? means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- ! means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1!

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

- * means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

  **Note:**

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

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

# Notices

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

This information might contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

## Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and 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 (http://www.ibm.com/software/support/systemsz/lifecycle/)
- For information about currently-supported IBM hardware, contact your IBM representative.

## Programming Interface Information

This book documents intended Programming Interfaces that allow the customer to write programs to obtain services of z/OS.

## Trademarks

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

# Index

# D



# E



# F



# G



# I



# K

# L



# M



# N



# O



# P



# R

## S



## T



## U



## V