Redbooks

ibm.com/redbooks

A Practical Guide to ICF Catalogs

Keith Winnard

Jose Gilberto Biondo Jr

Alvaro Salla

z Systems

IBM

Redbooks

International Technical Support Organization

A Practical Guide to ICF Catalogs

October 2016

SG24-8262-01

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

Second Edition (October 2016)

This edition applies to Version 2, Release 2, Modification 2 of DFSMS (product number 5650-ZOS).

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get personalized notifications of new content
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android

Promote your business in an IBM Redbooks publication

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

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

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication gives a broad understanding of integrated catalog facility (ICF) catalog environments. It includes suggestions for design, planning, and deployment tasks to help you create and maintain a balanced and efficient catalog environment.

Four scenarios are provided to illustrate sample implementations of typical activities that are associated with an organization's requirements.

Chapter 5, "Record-level sharing support for ICF catalogs" on page 87 describes Record Level Sharing (RLS) for Catalogs and shows the results of our tests in a controlled laboratory environment.

This version of the book is set at the IBM z/OS® V2R2 level.

This publication is for readers who want to gain an understanding of ICF catalogs and the considerations and practices that surround an ICF catalog environment deployment.

Authors

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

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

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

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

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

Stephen Branch (ICF Catalog and IDCAMS Development, IBM) for technical guidance and reviewing the publication.

Patricia Choi (DFSMS Catalog Development, IBM) for technical guidance and reviewing the publication.

Rich Lungerhausen (DFSMS Catalog Development, IBM) for technical guidance and reviewing the publication.

Terri Menendez (VSAM/RLS/Catalog Development, IBM) for technical guidance and reviewing the publication.

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

Thanks to the authors of the previous editions of this book.

Authors of the first edition, *A Practical Guide to ICF Catalogs*, published in June 2015:

- Keith Winnard
- Jose Gilberto Biondo Jr
- Gert Laumann
- Norbert Schlumberger
- Jeanne Vangsness
- Chao Xing CX Ma

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

Summary of changes

This section describes the technical changes that were made in this edition of the book and in previous editions. This edition might also include minor corrections and editorial changes that are not identified.

Summary of Changes
for SG24-8262-01
for A Practical Guide to ICF Catalogs
as created or updated on October 17, 2016.

October 2016, Second Edition

The second edition.

New information

- Information was added throughout the publication to include updates at the z/OS V2R2 level.
- Information was added throughout the publication to help the reader understand more about setting up and maintaining an ICF catalog environment.

1

ICF Technical overview

If you are relatively new to the integrated catalog facility (ICF) catalogs, or want to broaden your knowledge and appreciation of ICF catalog environments, this chapter aims to present you with a high-level understanding of why there is a need for catalogs and the roles that they play.

Four scenarios are used throughout this book to provide a consistent point of reference. Chapter 2, “Planning ICF environments” on page 19 describes the planning, Chapter 3, “Deploying ICF catalogs” on page 45 describes the deployment, and Chapter 4, “Maintaining ICF catalogs” on page 65 describes maintenance of the ICF catalog environments of each scenario.

Note: This IBM Redbooks publication does not cover the use of the IBM Tivoli® Advanced Catalog Management product, which is highly recommended when ICF catalogs are managed.

This chapter includes the following topics:

1.1 Introduction to data nature in commercial environment

The most unique feature of any organization is its data because the data defines the organization. Without it, the areas of marketing, order processing, logistics, human resources, sales, and so on, cannot exist. These areas need processes, and processes need data.

Where does data come from? For the purposes of this IBM Redbooks publication, consider that two general sources are responsible for data creation:

- The first source is *external* to the organization. Data is captured from transactions with clients, partners, suppliers, and anyone who interacts with the organization.
- The second source is *internal* to the organization, also called infrastructure data. Data is generated from processes as part of the transaction fulfillment. It also is generated for analytical and reporting tasks to gain a better understanding of the organization's well-being and opportunities.

Organizations typically rely on multiple applications. Each application features a discrete function in the organization, and often has many data sets associated with it. In addition, several versions or iterations of the used data sets might be available, such as for development, testing, pre-production, production, backup, migration, and archiving purposes.

An organization's data can be categorized and stored in separate data sets for use by different applications. Some data sets might be used by a single application only (dedicated), where others might be accessed by multiple applications (shared).

The voluminous nature of data (known as *big data*) in the modern world, including how it is categorized, has a direct effect on the number of data sets that is required to store it. For perspective, this number can range from tens of thousands to millions in a single one enterprise. These data sets are in I/O devices (such as 3390 disk) recognized by their volume serial (VOLSER) number. In a medium size installation, several thousands of such devices can be available. Therefore, several components and tools are required to control the creation and management of all of these data sets.

1.2 ICF catalogs overview

ICF catalogs are z/OS VSAM data sets that relate a data set name with its related information, such as its device location through its VOLSER, type, SMS constructs, I/O, and activity.

A major catalog function is to provide the VOLSER for a data set referenced (for example, at JCL) by its data set name only.

An ICF catalog is a key component in determining a data set's location. A well-structured multiple ICF catalog environment is essential to accommodate an organization's data sets. ICF catalogs are also a key cornerstone to data storage and management.

Although an ICF catalog is a single component, it has a strong relationship with the VSAM Volume Data Set (VVDS). Consider the following points:

- The Basic Catalog Structure (BCS) is the ICF catalog. It is a KSDS VSAM cluster and contains records that describe the logical information about a data set that is cataloged in the BCS.

- The VVDS contains records that describe the physical information about the data set that is cataloged in the BCS. It is an ESDS VSAM cluster containing dynamic information data for VSAM and SMS data sets.

The increasing rate of data growth presents challenges for the definition, availability, access, management, performance, and maintenance of ICF catalogs. Questions abound that relate to data, but the scope of this book aims to answer the following fundamental questions:

- How do I install an ICF catalog environment?
- What are the considerations for planning an ICF catalog environment?
- How might I deploy ICF catalogs?
- How do I maintain a balanced, safe, and efficient ICF catalog environment?

ICF catalogs are part of the IBM z/OS environment, so the scope of this book is focused on z/OS only.

This book also describes the basic components that are required to provide a solution to these questions. We explain the considerations for planning environments to accommodate the solution, suggest four linked scenarios and their relevant plans, demonstrate the deployment of those plans, and show how to maintain control of your ICF catalog environment.

Careful planning is essential to successfully create and maintain an ICF catalog environment. As part of the planning process, the ongoing monitoring and refinement of the ICF catalog environment must be regularly reported and reviewed. This process helps you to sustain optimum value to the organization and to help minimize the risk of data loss or outage.

The next section provides an overview of the related components that are necessary to build an ICF catalog environment.

1.3 ICF catalog-related components

Figure 1-1 shows the catalog-related components that can play a role in catalog processing, depending on your setup and the features you use. Consider Figure 1-1 as a checklist for which catalog components might be available.

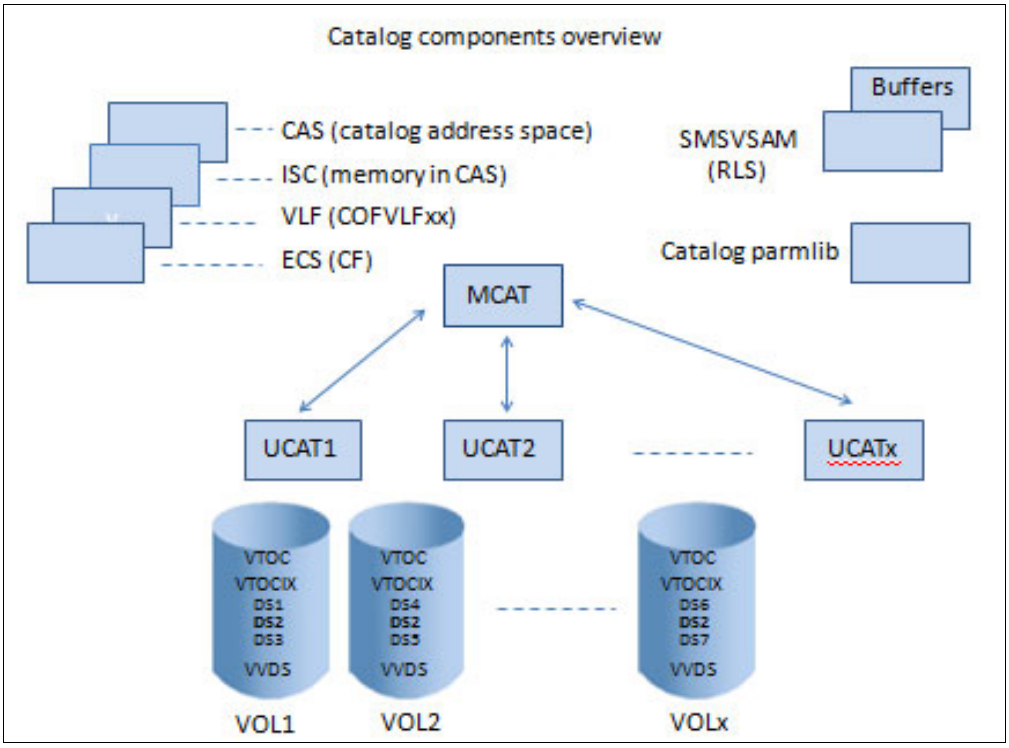


Figure 1-1 Catalog component overview

Alternatively, Figure 1-1 can serve as a guide for where to look if you want to change your ICF catalog environment, or if you are resolving unexpected situations. An ICF catalog is part of an integrated environment; therefore, any change to that environment might affect other components that are in or related to that environment.

ICF catalogs are divided logically into the following categories:

- The master catalog (MCAT). MCAT is the catalog for catalogs. All data set catalog searches (called *Locates*) that are done by catalog management start at the MCAT. It contains cataloged information about z/OS system data sets (high qualifier name is SYS1) and pointers to UCATs. A specific ICF Catalog is selected at IPL as the MCAT.
- The user catalog (UCAT) is connected to the MCAT. You can define many UCATs on z/OS systems. The actual number depends on how the organization decides to categorize its data sets. You want sufficient UCATs to balance the catalog load accesses, reduce risks that are related to damaging catalogs, ease catalog management, and help with creating standards for qualifying data set names.

Note: A UCAT does not point to another UCAT.

- An alias (not shown in Figure 1-1 on page 4) is one or more high-level qualifiers that are defined in the MCAT to associate it with any data set having the same high-level qualifiers with a particular UCAT. That is, an ALIAS provides a pointer for locating data sets. So, you can locate a data set by searching the master catalog to find the ALIAS and use that information to find the correct UCAT. The installation can decide how many qualifiers can be used in an ALIAS. In z/OS V2R2 (up to four qualifiers can be used).

There is no physical difference between an MCAT and UCATs. For example, an MCAT in one z/OS maybe the UCAT in another z/OS.

Caching (or buffering) is a performance feature that minimizes asynchronous I/Os by keeping data in virtual memory buffers. Then, catalog management keeps ICF catalog VSAM control intervals (CIs) in such buffers. In-storage Catalog Cache (ISC) and VLF cache BCS records, ECS caches the Catalog Self-describing VVR.

Figure 1-1 on page 4 show components that are related to ICF catalogs.

Figure 1-1 on page 4 also shows the basic catalog structure and environment from the MCAT, where all searches for a data sets start. From there, the searches are directed to a user catalog connector record (UCON), which is based on the alias that relates to the UCAT.

The UCON also points to the device volume VOLSER where the catalog is located. Having located the UCAT, the catalog points to the device volume VOLSER where the data set is located.

Catalog information can be cached in the following virtual storage location areas:

- ISC that is at the CAS private area.
- Virtual Lookaside Facility (VLF) data spaces. VLF is a z/OS component in charge of keeping objects in virtual storage data spaces. These objects can be catalog entries, IBM RACF® data, PDS directories, REXX procedures, and so on.
- Storage management subsystem (SMS) VSAM RLS buffer pool.

The next sections describe the related components and how they play a role in the catalog environment.

1.3.1 Catalog address space

Catalog management programs are mainly in the private area of the catalog address space (CAS), which is started at the time of initial program load (IPL). Communication between CAS and user address spaces (for example, issuing the Locate macro) happens through cross-memory services. A UCAT is opened by CAS when the first request is directed to it, and it remains open until an event requiring it to be closed occurs.

Alternatively, it remains open until CAS determines that something fundamental about the UCAT changed (for example, it was moved to a new volume), and the catalog control blocks must be rebuilt.

CAS holds control block information for open catalogs, alias information, and catalogs volume VOLSER numbers. The CAS status can be displayed and managed by using operator commands.

1.3.2 Master catalog

The MCAT is the highest level in the catalog structure, where information about UCATs and z/OS system-critical data sets are found (SYS1 high qualifier data sets are cataloged in MCAT). In addition, the MCAT contains an object that is called the alias (there can be many aliases). It is the alias that identifies the high-level qualifiers of the data set name and points to the UCAT that contains the entry for the requested data set matching this high qualifier data set name.

The MCAT must be identified at IPL time. The SYSCATxx member of `SYS1.NUCLEUS` contains information about the MCAT and initial parameters for the CAS. The LOADxx parmlib member can also be used to identify the MCAT and, if so, the information in the LOADxx member is used instead of the information that is in the SYSCATxx member.

The IGGCATxx parmlib member does not state which ICF catalog is the MCAT, but it does contain initial parameters for the CAS.

1.3.3 User catalogs

UCAT data sets are pointed to by an alias (data set name high qualifiers) in the MCAT. UCATs hold information about user and application data sets, such as the data set name and the corresponding devices that contain the data set (VOLSERs). When specifying the data set name, a search is made in the MCAT aliases. When located, the alias entry points to the UCAT that contains the data set information.

1.3.4 VTOC

The volume table of contents (VTOC) is an area on the DASD volume holding information about this volume data sets, and its free space. There is one VTOC per DASD volume. It includes, among other details, the location of every data set extents and the number of extents that are allocated to it. VTOC is created at DASD volume initialization by the IBM Device Support Facilities (ICKDSF) utility program. An extent is a set of contiguous DASD tracks free or allocated to data set.

1.3.5 VTOC index

The VTOC index is a key-sequenced data set register of all data sets on the volume. The purpose of the VTOC index is to speed up locating data sets on a VTOC by using an index rather than reading serially through the VTOC. The VTOC index is normally created at volume initialization, but can be built based on the VTOC with the **`BUILD IX`** function in ICKDSF at a later moment.

1.3.6 VSAM Volume data set

The VVDS holds more physical catalog information about certain data sets (VSAM and SMS) on the volume where the VVDS is located. This information includes such information as Data Facility Storage Management Subsystem (DFSMS) constructs, VSAM data set extents, and I/O activity. VVDS is a logical extension of the BCS catalog component.

1.3.7 In Storage caching

ISC is the virtual memory in CAS reserved for caching. The amount of memory that is used is fixed. When the cache fills, entries are removed from the cache based on a *least recently used* (LRU) basis. This caching is used (performance-wise) by MCAT control intervals only.

One key difference between MCAT and UCATs is that an MCAT is seldom modified. However, UCATs are altered often. The ISC LRU algorithm does not deal with data buffer modifications. If these modifications occur, the catalog management stops using buffers and accesses the catalog control interval by I/O operations only.

Ensure that ISC is the default caching location for ICF catalogs. For example, if VLF is not specified, UCATs are cached in ISC.

1.3.8 Catalog data space caching

As we already saw, catalog data space caching (CDSC), or VLF caching, is based in a data space that is owned by the VLF, which is a z/OS component. It is defined through the COFVLFxx member in `SYS1.PARMLIB`. Catalog entries are not limited by a piece of the data space storage size at a catalog level, but use whatever storage is assigned to the data space in total. At the point when the storage becomes saturated, an LRU algorithm starts removing entries from the VLF cache.

1.3.9 Enhanced Catalog Sharing

Catalog sharing was introduced before IBM Parallel Sysplex®. Write integrity was implemented through GRS global enqueues. The cache coherence, was implemented through a catalog record called VVR to ensure read integrity, where all the shared catalog updates were logged. Later the Enhanced Catalog Sharing (ECS) gives far better performance for shared catalogs. ECS uses a coupling facility structure resource to keep the VVR instead of in the 3390 volume catalog data set.

1.3.10 SMSVSAM and record-level sharing

SMSVSAM is the started task address space, which handles VSAM record-level sharing (RLS) requests. Recently, the benefits of RLS access were extended to ICF catalogs. SMSVSAM can switch RLS-managed catalogs in and out of RLS mode. RLS access removed the previous approach of enqueuing on the entire catalog (SYSIGGV2) and uses locks at the record level instead. This change is intended to improve performance and availability. It is advised to have a large enough CF structure for RLS-managed catalogs to ensure optimum caching in RLS mode. For more information about ICF RLS shared catalogs, see Chapter 5, “Record-level sharing support for ICF catalogs” on page 87.

1.3.11 Catalog parmlib IGGCATxx member

A catalog parmlib member IGGCATxx was introduced in z/OS V1.R13. Catalog parmlib can replace settings for the catalog, which was done in the LOADxx or SYSCATxx members. There are more than 20 keywords that the installation can customize the ICF catalog settings. Many settings previously required IPL, but now can be dynamically modified by restarting CAS.

1.3.12 ICF catalogs and tape located data sets

Tape data sets do not differ from disk data sets in terms of cataloging. Tape data sets also are cataloged in an ICF catalog. However, tape data sets do not need to be cataloged like DFSMS-managed disk data sets.

Tape data sets often are registered in a tape management system, such as DFSMS Removable Media Manager (DFSMSrmm). In this way, the tape data sets are protected and easily accessible without being registered in an ICF catalog.

Tape data sets are defined as non-VSAM data sets when cataloged in an ICF catalog. Therefore, all non-VSAM catalog handling applies to tape, such as deleting and redefining these data sets.

1.4 Understanding a data set catalog search

Catalog standard search order is a process that occurs when you attempt to locate a data set. The search might use the ALIAS object to go through the ICF catalog hierarchy to find a data set entry. A search initially enters the MCAT, and the ALIAS leads it to the UCAT.

Unless other parameters are set, such as using the **CAT** statement to point to a nonstandard search order catalog, attempts to locate the data set follows the standard search order.

A typical catalog search includes the following steps:

1. A catalog locate request comes from a batch job, often through a JCL DD statement, a transaction manager, or a TSO user. The search starts in the MCAT that is based on the ALIAS of the data set in scope.
2. The ALIAS points to a UCAT and the request is directed to that UCAT, where a key-sequenced search at the KSDS index component for the data set is performed.
3. The data set is in the UCAT with the VOLSERs of the data set for which is being searched. The catalog request is then completed.

The catalog requester proceeds to the VTOC and VSAM volume data set (VVDS) where the data set is located.

1.5 Catalog scenarios

This book uses four sample ICF catalog environment scenarios to help show different aspects of ICF catalog environments. These scenarios are intended to relate to requirements during an organization's system and data growth.

The four scenarios also show how MCAT and UCATs work, the common operations that are used, and the restriction or limitation for catalog processing. Each scenario introduces another layer of complexity.

In Chapter 3, "Deploying ICF catalogs" on page 45, we repeat the scenario diagrams, their respective plans, the job control language (JCL) commands, and the output that is associated with those commands.

1.5.1 Scenario 1: Single master catalog with multiple user catalogs

This scenario is the basic configuration of a simplified z/OS system. It shows a logical partition (LPAR) with one MCAT and four UCATs, as shown in Figure 1-2.

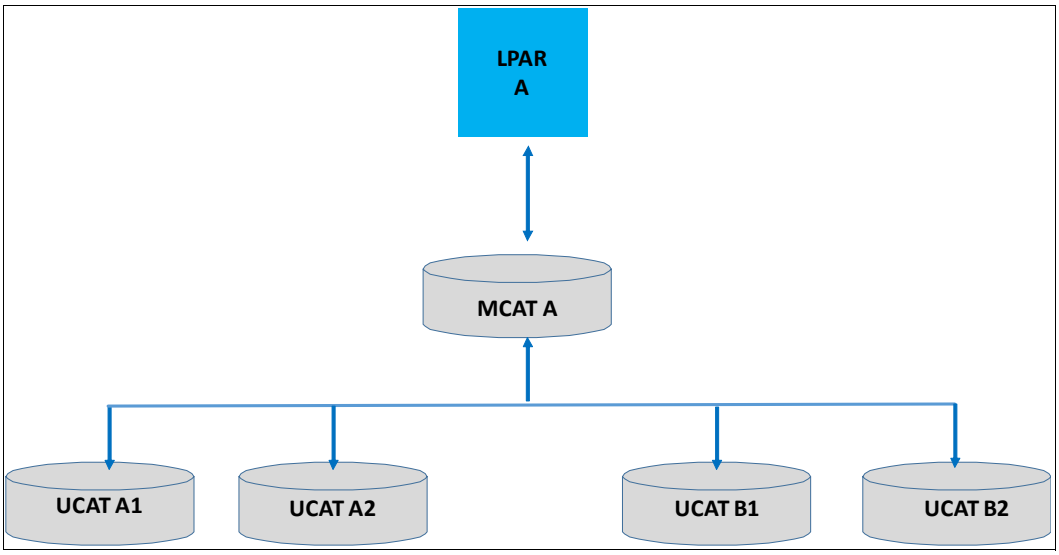


Figure 1-2 ICF catalog design in Scenario 1

LPAR A uses MCAT A as the MCAT to IPL, and all four UCATs connect to MCAT A.

The relationship between the catalogs is listed in Table 1-1.

Table 1-1 ICF catalog relationship in Scenario 1

User catalog	Connected Master catalog	Available for LPARs
UCAT A1	MCAT A	LPAR A
UCAT A2	MCAT A	LPAR A
UCAT B1	MCAT A	LPAR A
UCAT B2	MCAT A	LPAR A

1.5.2 Scenario 2: Starting new LPAR that shares the master catalog

As the organization grows, it might be advisable to expand from the single LPAR to a sysplex or multi-LPARs with data sharing by using a CF. This scenario introduces a new LPAR B that uses the same MCAT. LPAR B can be used as a production system in the same Sysplex or an application development and testing environment system. The catalog structure is shown in Figure 1-3.

Important: This section focuses on the catalog perspective only. All other system operations are *not* included.

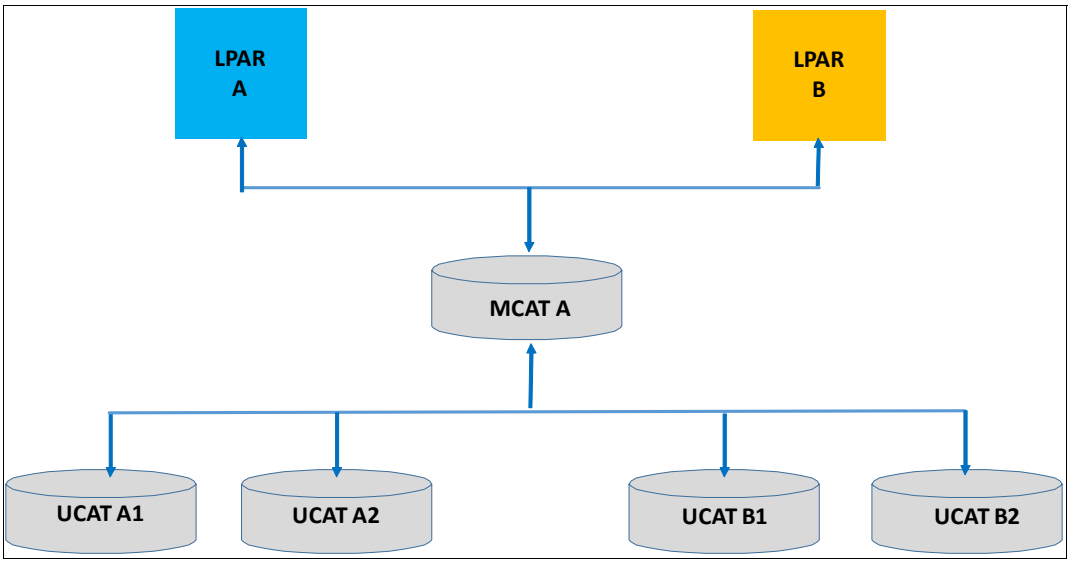


Figure 1-3 ICF catalog design in Scenario 2

The target structure of each catalog is listed in Table 1-2. All user catalogs are available for both LPARs.

Table 1-2 ICF catalog relationship in Scenario 2

User catalog	Connected Master catalog	Available for LPARs
UCAT A1	MCAT A	LPAR A, LPAR B
UCAT A2	MCAT A	LPAR A, LPAR B
UCAT B1	MCAT A	LPAR A, LPAR B
UCAT B2	MCAT A	LPAR A, LPAR B

1.5.3 Scenario 3: Splitting master catalog into two systems

In certain circumstances, the organization might want to split the sysplex into two independent systems. The new system features the following requirements:

- A new master catalog
- A move of a subset of user catalogs to the new separate catalog structures
- A connection of those user catalogs to the new master catalog

There might still be a requirement to share some but not all data sets; therefore, some of the catalogs must be shared (although others are dedicated to a specific LPAR).

As shown in Figure 1-4, `MCAT B` is the new Master catalog that was split from `MCAT A`. `LPAR B` uses `MCAT B` then `UCAT B2` dedicated connects to `MCAT B`. `UCAT A2` and `UCAT B1` are shared to `LPAR A` and `LPAR B`.

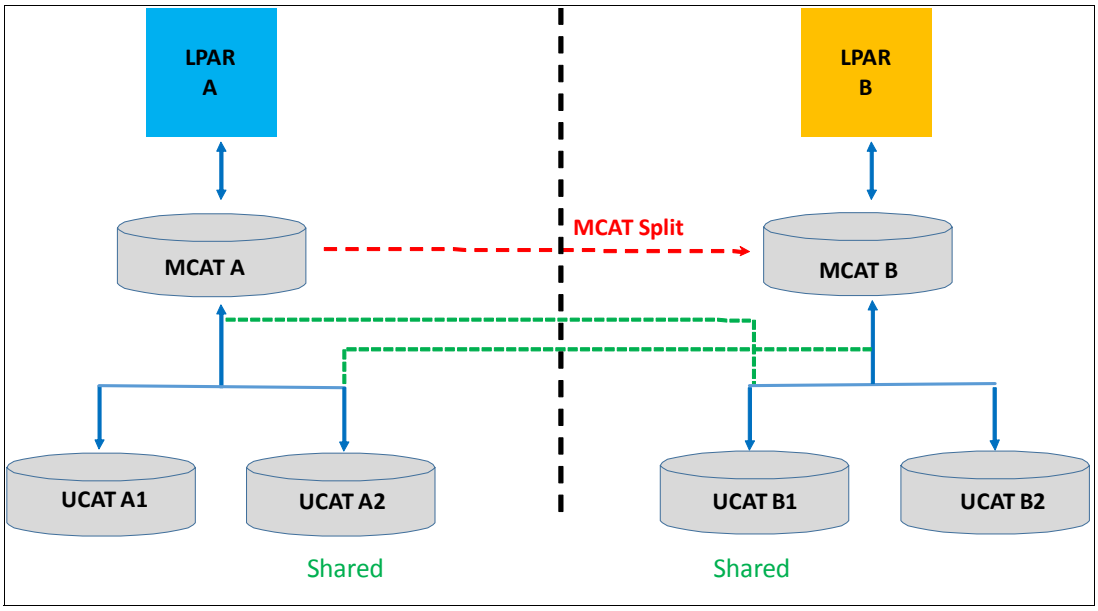


Figure 1-4 ICF catalog design in Scenario 3

The final status of each catalog after splitting is listed in Table 1-3. You must pay attention to non-shared user catalogs `UCAT A1` and `UCAT B2` because those catalogs cannot be used by the other LPAR.

Table 1-3 ICF catalog relationship after split in Scenario 3

User catalog	Connected Master catalog	Available for LPARs
`UCAT A1`	`MCAT A`	`LPAR A`
`UCAT A2`	`MCAT A`, `MCAT B`	`LPAR A`, `LPAR B`
`UCAT B1`	`MCAT A`, `MCAT B`	`LPAR A`, `LPAR B`
`UCAT B2`	`MCAT B`	`LPAR B`

1.5.4 Scenario 4: System consolidation

This scenario describes a consolidation and integration of ICF catalog environments. Suppose that our sample organization acquires another organization and gains another LPAR. In this case, we name this LPAR `LPAR C`. The organization might want to consolidate all of the LPARs onto its own mainframe, but initially keep the data separate. Therefore, the newly acquired is an independent system with its own MCAT and UCATs.

As shown in Figure 1-5, `LPAR C` is an independent system with its own master catalog (`MCAT C`), and two user catalogs (`UCAT C1` and `UCAT C2`).

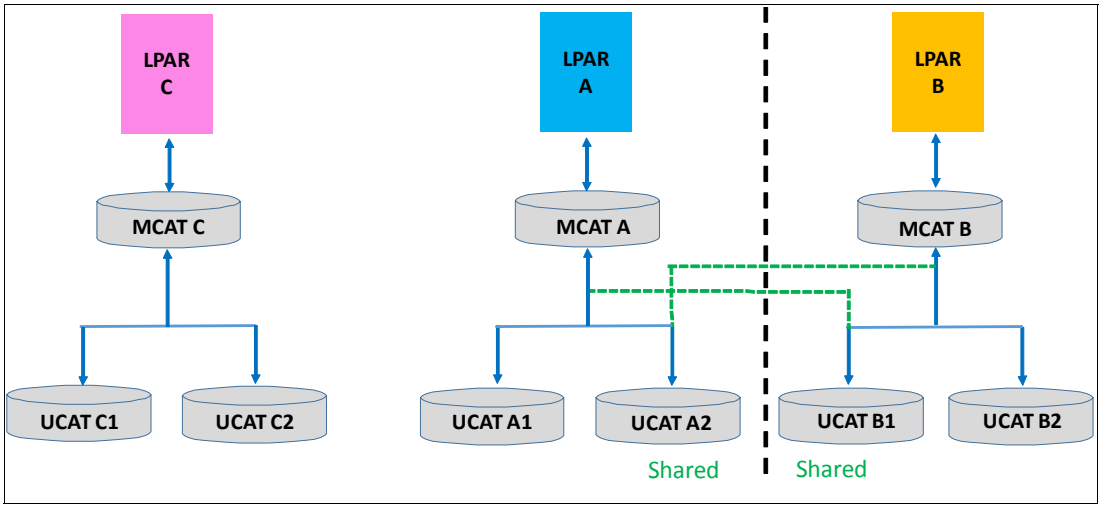


Figure 1-5 ICF catalog design before consolidation in Scenario 4

The relationship between catalogs before consolidation is listed in Table 1-4.

Table 1-4 ICF catalog relationship before consolidation in Scenario 4

User catalog	Connected Master catalog	Available for LPARs
UCAT A1	MCAT A	LPAR A
UCAT A2	MCAT A, MCAT B	LPAR A, LPAR B
UCAT B1	MCAT A, MCAT B	LPAR A, LPAR B
UCAT B2	MCAT B	LPAR B
UCAT C1	MCAT C	LPAR C
UCAT C2	MCAT C	LPAR C

After the new system is examined and established, the organization might want to restructure the catalog further so that the catalog runs more efficiently.

The aim might be to merge MCAT C to MCAT A, connect UCAT C1 and UCAT C2 to MCAT A, then load LPAR C with MCAT A. After this process is completed, MCAT C is no longer used. The final relationship of catalogs after MCAT A and MCAT C consolidation is shown in Figure 1-6.

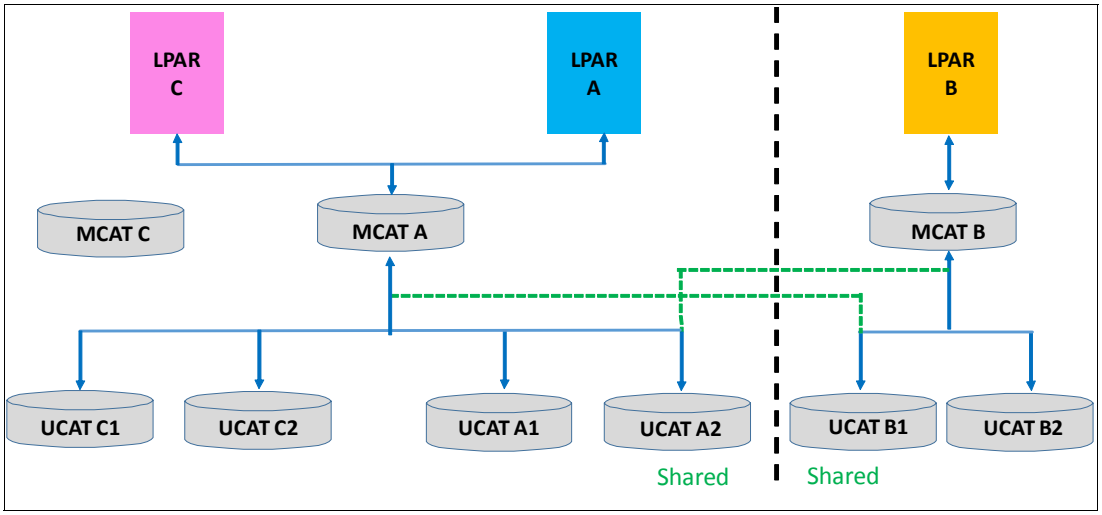


Figure 1-6 ICF catalog design after consolidation in Scenario 4

The final status of each catalog after system consolidation is listed in Table 1-5.

Table 1-5 ICF catalog relationship after consolidation in Scenario 4

User catalog	Connected Master catalog	Available for LPARs
UCAT A1	MCAT A	LPAR A, LPAR C
UCAT A2	MCAT A, MCAT B	LPAR A, LPAR B, LPAR C
UCAT B1	MCAT A, MCAT B	LPAR A, LPAR B, LPAR C
UCAT B2	MCAT B	LPAR B
UCAT C1	MCAT A	LPAR A, LPAR C
UCAT C2	MCAT A	LPAR A, LPAR C

1.6 Reporting

To understand how your data, applications, and business grow, it is important to generate reports that can show you the usual performance information that you need in a simple and efficient format.

Unlike alerts that are issued when an error is identified on the system, reports must consist of relevant data that is grouped into a format that is easy to understand and enable the readers to forecast issues, identify opportunities, and help guide projects.

Reports are valuable to technical and management teams, where one team can predict and prevent outages and the other identify business opportunities and application trends.

If reports can be used by different areas, it is necessary to identify what types of reports are available, what information they deliver, and to what groups this information is relevant.

1.6.1 Available reporting types

You can request the following statistic reports (which are defined by default) from the catalog:

- **REPORT** provides basic information about some of the current limits and installation-specified defaults that are selected for the catalog address space, as shown in Example 1-1.

Example 1-1 F CATALOG,REPORT

```
F CATALOG,REPORT
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT OUTPUT 877
*CAS*************************************************************
*  CATALOG COMPONENT LEVEL   = HDZ2220                           *
*  CATALOG ADDRESS SPACE ASN = 002F                              *
*  SERVICE TASK UPPER LIMIT  =  180                              *
*  SERVICE TASK LOWER LIMIT  =   60                              *
*  HIGHEST # SERVICE TASKS   =   21                              *
*  # ATTACHED SERVICE TASKS  =   21                              *
*  MAXIMUM # OPEN CATALOGS   = 1,024                             *
*  ALIAS TABLE AVAILABLE     = YES                               *
*  ALIAS LEVELS SPECIFIED    = 2                                 *
*  SYS% TO SYS1 CONVERSION   = OFF                               *
*  CAS MOTHER TASK           = 007AC680                          *
*  CAS MODIFY TASK           = 007FC870                          *
*  CAS ANALYSIS TASK         = 007FC410                          *
*  CAS ALLOCATION TASK       = 007FC640                          *
*  CAS ASYNC TASK            = 007FC1E0                          *
*  CAS SYSPLEX COMMAND TASK  = 00794C58                          *
*  CAS SYSPLEX QUIESCE TASK  = 00794E88                          *
*  VOLCAT HI-LEVEL QUALIFIER = SYS1                              *
*  NOTIFY EXTENT             =   80%                             *
*  DEFAULT VVDS SPACE        = ( 10, 10) TRKS                    *
*  ENABLED FEATURES          = DSNCHECK DELFORCEWNG SYMREC       *
*  ENABLED FEATURES          = UPDTFAIL GDGEXTENDED              *
*  DISABLED FEATURES         = VVRCHECK AUTOTUNING BCSCHECK      *
*  DISABLED FEATURES         = DELRECOVWNG EXTENDEDALIAS         *
*  DISABLED FEATURES         = ECS AUTOADD DUMPON GDGFIFO        *
*  DISABLED FEATURES         = GDGSCRATCH GDGPURGE               *
*  INTERCEPTS                = (NONE)                            *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

- **CATSTATS[(catname)]** lists the I/O statistics and BUFNI, BUFND, and STRNO for all catalogs that are active in the catalog address space. If you specify a catalog name(*catname*), statistics are listed for that catalog only, as shown in Example 1-2.

Example 1-2 F CATALOG,REPORT,CATSTATS(UCAT.BH6CAT)

```
F CATALOG,REPORT,CATSTATS(UCAT.BH6CAT)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG I/O STATS REPORT 885
*CAS*************************************************************
*     ADDS  UPDATES     GETS   GETUPD  DELETES  BUFNI  BUFND STRNO *
*                                                                 *
* UCAT.BH6CAT                                                     *
*       5        0    5,410        5        5       5     10     2 *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

- **CATSTATX[({catname | catprefix*})]** lists the same I/O statistics and settings that are produced by CATSTATS. It also provides CA-Reclaim and CA-Reuse statistics. Output is in the form of a table for each selected catalog, as shown in Example 1-3. The catname or catprefix* must be specified. If catprefix* is used, all open catalogs that match the prefix are reported on.

Example 1-3 F CATALOG,REPORT,CATSTATX(UCAT.BH6CAT)

```
f catalog,report,catstatx(ucat.bh6cat)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I EXTENDED CATALOG STATS
*CAS*************************************************************
*  CATALOG NAME     = UCAT.BH6CAT                                 *
*  INSERTS (ADDS)   =         6                                   *
*  UPDATES          =         0                                   *
*  RETRIEVES        =     5,590                                   *
*  RETRIEVES FOR UP =         6                                   *
*  ERASES (DELETES) =         6                                   *
*  CA-RECLAIMS      =         0                                   *
*  CA-REUSES        =         0                                   *
*  BUFNI SETTING    =         5                                   *
*  BUFND SETTING    =        10                                   *
*  STRNO SETTING    =         2                                   *
*  AVG ELAPSED TIME =     1.398 MSEC                             *
*  AVG CPU TIME     =   127.863 USEC                             *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Note: Use caution when the catprefix* option is used because it can generate a significant amount of write to operator (WTO) output.

- **PERFORMANCE:** Shows performance statistics, such as response time, for all the types of catalog events, as shown in Example 1-4 on page 16. This information can be used to identify possible catalog bottlenecks, which can affect application performance and the Service Level Agreements (SLA).

Example 1-4 F CATALOG,REPORT,PERFORMANCE

```
F CATALOG,REPORT,PERFORMANCE
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG PERFORMANCE REPORT 826
*CAS************************************************
*  Statistics since  9:56:00.70 on  09/27/2016       *
*  -----CATALOG EVENT----   --COUNT--  ---AVERAGE---  *
*  Entries to Catalog          6,181      5.658 MSEC  *
*  BCS ENQ Shr Sys             8,326      0.743 MSEC  *
*  BCS ENQ Excl Sys               66      0.009 MSEC  *
*  BCS DEQ                    12,183      0.003 MSEC  *
*  VVDS RESERVE CI             1,791      0.006 MSEC  *
*  VVDS DEQ CI                 1,791      0.005 MSEC  *
*  VVDS RESERVE Shr           13,022      0.153 MSEC  *
*  VVDS RESERVE Excl              22      0.321 MSEC  *
*  VVDS DEQ                   13,044      0.006 MSEC  *
*  SPHERE ENQ Excl Sys            54      0.478 MSEC  *
*  SPHERE DEQ                     54      0.005 MSEC  *
*  CAXWA ENQ Shr                 887      0.004 MSEC  *
*  CAXWA DEQ                     887      0.001 MSEC  *
*  VDSPM ENQ                   8,344      0.009 MSEC  *
*  VDSPM DEQ                   8,344      0.001 MSEC  *
*  BCS Get                    15,411      0.039 MSEC  *
*  BCS Put                        45      0.239 MSEC  *
*  BCS Erase                       4      0.217 MSEC  *
*  VVDS I/O                   14,842      0.230 MSEC  *
*  VLF Define Major                1      0.133 MSEC  *
*  VLF Identify                1,386      0.000 MSEC  *
*  RMM Tape Exit                   4      0.000 MSEC  *
*  RMM Tape Exit                   4      0.000 MSEC  *
*  OEM Tape Exit                   4      0.000 MSEC  *
*  BCS Allocate                   27      1.793 MSEC  *
*  SMF Write                     870      0.016 MSEC  *
*  CAS Cache Lock                 15      0.011 MSEC  *
*  CAS Alloc Lock                  3      3.454 MSEC  *
*  CAS MLA Lock                   15     10.334 MSEC  *
*  IXLCONN                         2      0.301 MSEC  *
*  MVS Allocate                    7      6.685 MSEC  *
*  Capture UCB                     4      0.005 MSEC  *
*  RACROUTE Auth                 351      0.134 MSEC  *
*  Get LatchShr #             16,143      0.000 MSEC  *
*  ENQ SYSZPCCB                2,474      0.001 MSEC  *
*  DEQ SYSZPCCB                2,474      0.000 MSEC  *
*  Release Latch #            16,163      0.000 MSEC  *
*  Capture to Actual              10      0.001 MSEC  *
*  Get LatchExc #                 20      1.127 MSEC  *
*  ENQ SYSIGGV1                    2      0.002 MSEC  *
*  DEQ SYSIGGV1                    2      0.001 MSEC  *
*  ENDREQ                          5      0.001 MSEC  *
*  SYSVSAM S ENQ Excl             28     16.871 MSEC  *
*  SYSVSAM S DEQ                  28      0.003 MSEC  *
*  SYSVSAM D ENQ Shr              27      0.000 MSEC  *
*  SYSVSAM D DEQ                  27      0.002 MSEC  *
*CAS************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

- **Cache**: Displays the catalog cache information (including cache hit percentage), as shown in Example 1-5. A low cache hit can lead to higher processor usage or performance degradation. It is an extension of the performance data.

Example 1-5 F CATALOG,REPORT,CACHE

```
F CATALOG,REPORT,CACHE
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG CACHE REPORT 837
*CAS*****************************************************************
*   HIT% -RECORDS- -SEARCHES --FOUND-- -DELETES- -SHR UPD- --PURGE-- *
*                                                                     *
* CATALOG.SHRICF1.VIODFPK                          (ISC)              *
*    50%        1          2          1         0         0         0  *
* UCAT.ZOSMF                                       (ISC)              *
*    88%        3         27         24         0         0         0  *
* UCAT.IODFPK                                      (ISC)              *
*    25%        3          4          1         0         0         0  *
* UCAT.BH6CAT                                      (ISC)              *
*     8%       66      5,271        443         0         0         0  *
* MCAT.BH6CAT                                      (ISC)              *
*    52%      268      7,563      3,967         0         0         0  *
*CAS*****************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

1.7 Need for reviews

An ICF catalog environment stands on the following pillars:

- Integrity

  The content must be valid.

- Availability

  All ICF catalogs and related components must be available to support the creation and retrieval of data. That is, single point of failures (SPOF) must be avoided and the business continuance project must consider the ICF catalogs.

- Effectiveness

  The environment must be fit for the purpose and contribute to all aspects of, data creation and access. It must not be perceived as a hindrance.

- Functionality

  The z/OS software must provide the functions and tools to perform access to data sets, and to monitor, measure, report, analyze, and modify the environment.

- Performance

  The environment must be configured in a manner that successfully contributes to meeting SLA.

- Flexibility

  The environment's structure must facilitate permanent and temporary changes to the catalog's configuration, as required.

- Security

  A catalog is a key infrastructure entity and plays a critical role in the operation of the business. It must be protected against loss or damage.

Today, data creation and retrieval are on an unprecedented scale as social media expands, collaborative initiatives grow, faster and deeper analytics are needed, and pressures are put on an organization to perform on demand. The most effective environments are planned and based on understanding variables, such as the requirements, circumstances, and goals of the business.

A defined and quantified tolerance ought to be built into the plan. These tolerance values represent the points that, if exceeded, result in the destabilization of the environment in one or more of the pillars described in this chapter.

This destabilization might be a result of a single instance (such as a company acquisition), or a slow and steady creep in several areas. Exceeding the thresholds unexpectedly causes reactive tactical behavior to minimize the effect of the issue. In the short term, this issue is unlikely to reestablish the environment's effectiveness.

The purpose of the reviews is to prevent destabilization and attempt to provide an ongoing improvement and contribution to the organization.

2

Planning ICF environments

In this chapter, we describe how integrated catalog facility (ICF) catalog environments might be planned. Considerations and suggestions are given to help improve the chances of producing a realistic plan that meets your organization's needs.

This chapter includes the following topics:

2.1 The plan

The planning of ICF catalog environments considers various aspects that are related to business needs, security, legal requirements, and technical capabilities. Other factors also might need to be considered. The goal is to create a plan for the design and processes that builds a balanced ICF catalog environment and ensures that its effectiveness is maintained.

2.1.1 Why plans are necessary

A structured ICF catalog environment enables ever-changing data to be accessed effectively and efficiently. Each of the related components plays an essential part and must be used in a cohesive manner to ensure that the environment's goals are met.

Meeting this goal does not happen by luck.

Each related component's functionality and configuration must be considered, and the best blend achieved. The planning phase is where the thinking and designing occurs. The following examples show why a plan is needed:

- The first implementation of a new system or logical partition (LPAR)
- The addition of a secondary LPAR to the original implementation and data sharing
- Separation of data sharing
- Acquisition of another business and merger of different systems
- As a result of a regular review
- A temporary configuration to ease the business through a transitional phase
- Planning for disaster recovery and business continuity

The plan provides the path to be followed to set objectives and metrics, meet the requirements, and provide the appropriate outcomes to satisfy the needs of the initiative. In addition, it provides a record of how a particular need was met and explains the approach that is taken. Previous plans are useful documentary evidence of why and how configurations were designed, defined, and maintained.

2.1.2 Technical plan expectations

Concise and clear plans are preferable. The objective must be clear and unambiguous. Discuss the plan thoroughly with all interested parties to set common expectation levels, and ensure that everyone understands both their own roles and the roles of others.

The plan contains a series of tasks. Expect to see some or all of the following points in the plan. The following tasks also can appear, depending on the activity, environment, and other local standards and guidelines:

- Objective of the plan
- Deliverables
- Task identifier and description
- Components to be changed
- Activities related to the tasks (for example, stopping and starting other components)
- Who is responsible for delivering the task
- Review points
- Backout instructions if the tasks fail or cause post-implementation issues

2.1.3 Effect on other areas

ICF catalogs are a key component of the operational configuration and are used by all of the online workloads and most batch/TSO activity. Therefore, there can be an effect on other areas when changes are made to the ICF catalog environment or a related component. The following immediate areas might be affected:

- Operational housekeeping
- Security administration
- Operational automation
- Performance management
- Continuous availability (24x7)
- Reporting

The following other areas are not so immediate, but must be considered:

- Disaster recovery
- Capacity planning
- Hardware configuration

2.2 Specifications

Recall that performance and scalability are key requirements when dealing with ICF catalog structures. A scalable system is one that minimizes the effect of the Law of Diminishing Returns (LDR). LDR states that the queue time decreases when the amount of servers is increased. However, the service time increases because of the management cost.

When relating the LDR with ICF catalogs scalability (if you correctly plan them), you can have many of those ICF catalogs without a clear performance degradation or a larger increase to the service time.

2.2.1 ICF technical restrictions

Consider the following technical restrictions:

- The basic catalog structure (BCS) is a VSAM key-sequenced (KSDS) cluster and VVDS is a VSAM entry-sequenced (ESDS) cluster.
- A BCS data set can have up to 123 extents, but can occupy space on a single volume only.
- A BCS is limited to 4 GB unless it is defined with extended addressability attribute. In this case, the size limit of the BCS is determined by the control interval size multiplied by 4 GB. Therefore, a BCS with a control interval size of 4 KB has a maximum data set size of 16 TB. To use extended addressability, the BCS must be storage management subsystem (SMS)-managed (it must have an associated storage class) and be defined as extended format (`DSNTYPE=EXT`) attribute.
- The BCS cannot be compressed or striped. The only extended format option that is available for a BCS is *extended addressability*.

- By default, the number of aliases that a catalog can have is limited by the maximum record size for the MCAT. A practical maximum number is 3000 aliases per catalog. For IBM z/OS V1.R13 or later, if the `EXTENDEDALIAS` option is enabled, you can increase the number of possible aliases to a theoretical limit of over 500,000 aliases (depending on the length of alias names).

Note: The previous technical limitations are based on z/OS V2.2. Those limitations might vary for future z/OS versions.

2.2.2 Estimating catalog size

Based on operational experience, excessive secondary extents can slightly decrease the catalog performance (although secondary extents of BCS do not cause any functional problem). Therefore, a catalog must be ideally defined with sufficient primary space to avoid excessive or unnecessary secondary extents.

The required space of a BCS depends on the type and number of data sets and objects that are cataloged in the BCS. The required amount of BCS space for each type of data set or object can vary according to the following factors:

- Length of the data set or component names (up to 44 characters)
- Number of volumes per data set
- Number of relationships between components
- Number of alternative indexes for a VSAM KSDS data set
- Number of paths
- Presence of security information
- Presence of SMS information

For more information about space estimation and calculation of BCS, tape volume catalog, and VVDS, see *z/OS DFSMS Managing Catalogs*, SC23-6853.

2.2.3 Setting the control interval and control area size of a catalog

As a VSAM KSDS, the performance of BCS is also related to the setting of control interval (CI) and control area (CA) sizes, When a new BCS with the IDCAMS multifunction utility is defined, you can specify the CI size in the job control language (JCL) or enable it to take the default value. For more information about the practical setting guideline of catalog CI and CA, see *z/OS DFSMS Managing Catalogs*, SC23-6853.

2.3 Planning considerations

Whether planning a full ICF catalog environment or modifying an environment, consider the implications of the intended changes. Are there other areas that must take actions? The following topics are areas to consider during the planning, implementation, and post-implementation phases.

2.3.1 Data Integrity

Data integrity is an important discipline in data processing. It consists of the following components:

- Write integrity that guarantees that no updates (in any memory) are lost or overlaid.

- Read integrity that ensures the access of the most current and coherent contents of data. Some applications do not require full read integrity, and in such a case the reads are known as “dirty”.

How can a plan include integrity? Unless the objective of the plan is to specifically restore integrity to a broken component, assume that integrity is at the start of the exercise. However, you must prove this integrity.

An example of losing integrity might be an installation configuration in which several catalog management components are in several z/OS LPARs, sharing catalogs by using the ECS technique. All the z/OS LPARs are in the same Sysplex with the exception of one LPAR.

Use whatever means or utilities to prove that the ICF catalogs or related components are intact. These means can include listing, assessing record counts, backing up, or any other action to prove that the entity is intact and operable. Also, devise a method to test the integrity of affected components when the implementation activities are complete, but before the components are made available for general use.

2.3.2 Continuous availability

Planning for availability (24x7) is the high priority. ICF catalogs and their related components are essential for data access in a z/OS environment. Data sets access is severely limited or even impossible without the ICF catalogs, depending on the cause and size of the outage.

The point is to minimize the single point of failures (SPOF) at ICF catalogs components.

ICF catalog availability planning includes designing your environment with the minimum risk of experiencing a partial or full outage. In addition, the design must include an error-free recovery in the shortest time possible, which requires a degree of flexibility. The following considerations are typical:

- Choose where to allocate your catalogs to minimize the exposure to unnecessary risk of deletion or corruption.
- Back up the catalogs regularly.
- Run regular utilities to check that the catalog is accessible and has integrity.
- Set up test catalogs to develop catalog recovery skills, and practice them appropriately.
- Utilities, such as ICFRU, and products, such as Tivoli Advanced Catalog Management, are available that can apply a forward recovery process to restore an ICF catalog from a backup by using SMF records.

Beware of the implications of enabling one ICF catalog to have too many applications accessing data through it. Be more granular in the design. Several catalogs that service multiple applications can reduce the chance of a large outage, minimize the effect of that outage, and provide the opportunity for a quicker and simpler recovery.

Simple structures and environments are easier to understand than complex configurations. Reduce the risk of a failing catalog from spreading to other catalogs or affecting multiple applications. Pay particular attention to documenting the environments and their structure, and the processes for maintaining those environments.

Sharing ICF catalogs across more than one system, or as part of a highly functional Sysplex, presents better recovery options. At a simpler system level, carefully plan how catalogs can be repaired from another system (through temporary or permanent sharing) to help increase availability levels.

ICF catalog environment portability enables a catalog to be moved (imported or exported) from one system to another or to a disaster recovery site, with minimum risk and a structured approach.

Availability levels can also be improved by considering what kinds of data sets are cataloged into each ICF catalog. Should production, development, and test data be in the same catalog? Is it wiser to separate different applications? The answers depend on the policies and guidelines of the organization, and the complexity levels of the applications.

2.3.3 Security

Discuss the proposed environment or new entity with security administration to avoid inadvertently creating exposures. Include tasks for security administration to set up, test changes, and approve the new environment status.

For example, having the RACF right to run **F CATALOG** commands, such as **F CATALOG,ALLOCATED** or **F CATALOG,REPORT**, can be helpful to operations and other teams to retrieve information about catalog status. However, this access also provides the following means and other commands that are not intended to be performed by these teams:

```
F CATALOG,ALLOCATED
F CATALOG REPORT
```

Starting in z/OS V2.R2, an RACF resource profile was created to be checked by catalog command processing to ensure that the issuer of a command is authorized to run that command. This function allows users that use SDSF to issue operator reporting commands, the results of which they can view at their computer. However, users are not permitted to issue commands that change the catalog configuration.

The only requirement for activating this facility is to define a new **OPERCMDS** resource profile, as shown in Example 2-1.

Example 2-1 New OPERCMDS resource profile

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

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

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

2.3.4 Operations

The storage management team must work closely with the operational support teams, because their areas are so closely connected. When either party plans a change, the other party must be included to help assess and resolve any potential risks. In particular, explore the following areas:

- Effect on the operational schedule
- Operational jobs access to data sets
- Changes to the automation requirements
- Operational procedures and supporting documentation
- Potential effect on service level agreements (SLAs)

2.3.5 Documentation

Documentation is most effective when it accurately reflects the subject and addresses relevant topics concerning that subject. It must also have a simple structure and be easy to follow. The documentation must be self-contained and portable. The plan must include a section to handle the change control of the documentation.

Be aware that some documentation is stored off-site and must be updated as appropriate. An important consideration is how a partial implementation of a change is handled in terms of documentation. Perhaps circumstances arise whereby an implementation is only partially achieved, or is planned to occur in several steps over several weeks. This issue can lead to documentation becoming inaccurate, and the situation can become more critical if one of the implementation steps is backed out.

Disaster recovery documentation should be kept in line with all concerned systems. Recovering a system with no documentation might not be possible, or might prolong the recovery period to a point beyond acceptable business continuity thresholds. This situation is not acceptable if the business is to continue.

The following suggestions are to help you plan your documentation:

- Create the documentation in a modular format.
- Keep multiple copies of the documentation in several locations, especially away from the system it is describing. Without this separation, the documentation might be unavailable if the system fails.
- Review the documentation regularly to ensure that it accurately describes the ICF environments and configurations.
- Align the documentation with the respective systems.
- Treat the recovery job repository in the same way as the documentation.
- Use whatever media suits your purpose. In addition to the traditional approach, consider the use of video or other suitable media. If the media adds value, use it.
- Ensure that the documentation is secure. Limit access to only those who need it.

2.3.6 ICF catalog performance

This section focuses on the setup and features that can affect ICF catalog performance, and improve it if they are not in place. The setup, from allocation of a new ICF catalog to caching, sharing, and monitoring, is covered.

To ensure the best possible performance on your ICF catalogs, focus on the setup of your environment, from allocation of a catalog to the use of features that can affect performance. ICF catalogs grow, data volumes increase, and applications change in response to day-to-day activities. What worked well at creation time might not work as effectively now.

Good reporting is essential. It provides an indication of how well the ICF catalog environment meets workload demands. Based on trends and preset thresholds, you might need to review your setup and perform selected tuning activities based on any performance anomalies.

Definition of ICF catalogs

When a new ICF catalog is defined, it is suggested that you follow a few guidelines as to parameters in your **`DEFINE USERCATALOG`** command. The `STRNO` (string number) option defines the number of parallel reads to the ICF catalog. Use `STRNO=3` as a good start (the default is `2`). Never define more than 6 - 7 reads, unless you have a specific reason to define more reads.

Rather than use the BUFFERSPACE option, use BUFNI and BUFND. BUFNI (number of buffers on the index) should be specified. The default is STRNO + 2. Specify this number as the number of index levels +1 (for example, 4). BUFND could be specified as STRNO + 1, in this case 4. For SHAREOPTIONS and RECORDSIZE, use defaults. FREESPACE(0 0) is suggested, because splits might happen immediately and you cannot control where the access goes to the ICF catalog in terms of the range of the keys.

After you define the ICF catalog, you might later choose to change the values that you initially specified. The previous values can be changed by using the IDCAMS **ALTER** command. This command also enables you to change your catalogs easily if they suffer from lack of buffers or you want to have a standardized setup.

How the **DEFINE USERCATALOG** might be coded is shown in Example 2-2.

Example 2-2 Example of defining a user catalog with tuning parameters

```
DEFINE USERCATALOG -
(NAME(YOUR.CATALOG) -
FREESPACE(0 0) -
CYLINDER(10 10) -
VOLUME(vvvvvv) -
EATTR(OPT) -
STRNO(3)) -
DATA(CISZ(4096) BUFND(4)) -
INDEX(CISZ(4096) BUFNI(4))
```

For indexes, CISZ (CISIZE) should be kept at 4 KB. For data, that size can vary. If you have frequent sequential reads on your catalogs, CISZ should be larger than 4 KB. If your ICF catalog is facing direct reads, a small CISZ might be a better option. For ICF catalogs, direct reads are more likely than sequential reads.

Allocate a secondary extent in more than one cylinder because creating extents takes time. When it extents are created, capacity is added for some time and each CA is prevented from splitting from requiring new extents. CA Reclaim option also are used to avoid some extents by reusing CAs that were freed up. For more information about CA Reclaim, see *DFSMS Managing Catalogs*, SC23-6853.

Consider allocating your ICF catalogs on dedicated volumes to minimize the effect of other workloads on your catalog performance and to reduce contention. However, this recommendation is not so important with 3390 PAV. You can allocate ICF catalogs on the same volume. However, limit this allocation to some catalogs and balance your catalog input/output (I/O) on more volumes. If this DASD subsystem has a solid-state drive (SSD) and IBM Easy Tier®, the hot extents might be moved to the SSD automatically.

Caching ICF catalogs

In this topic, we focus on when you should prioritize the use of one cache option rather than other options.

The following methods are available to use caching (also known as *buffering*):

- In Storage Caching (ISC)

  ISC caches ICF catalog records at CAS private area. It is the caching default for ICF catalogs. When this cache is full, the Least Recently Used (LRU) algorithm applies. For MCAT, all accessed records are cached (for sequential and random accesses). For UCATs, only the random accessed records are cached.

  When a shared catalog is modified by the other z/OS, ISC purges the cache and uses I/O operations to the ICF catalog instead, Although this purge is done for integrity reasons, it severely affects the performance, Then, ISC caching is recommended for MCAT only, where practically you do not have updates.

- Catalog Data Space Cache (CDSC)

  CDSC caches ICF catalog records in Virtual Lookaside Facility (VLF) data spaces. VLF is a z/OS component that keeps data in memory.

  This cache is a large buffer in which the individual UCATs use storage until the limit (`MAXVIRT` as specified in COFVLFxx member) is reached. When this cache is full, the LRU algorithm applies to create room.

  CDSC does not purge the cache when a shared catalog is modified by other z/OS. CDSC can also be used for non-ECS catalogs.

  CDSC caching is highly recommended for UCATs because they are modified frequently.

- Record-level sharing (RLS)

  Since DFSMS V2R1, VSAM RLS caches ICF catalog records at SMSVSAM address space (above the bar) and CF structures. Benchmark tests show significant improvement in ICF catalog tasks; therefore, it a tuning option. For more information about RLS, see Chapter 5, “Record-level sharing support for ICF catalogs” on page 87.

  You can use only one cache type at any time.

  Consider that a break-even point for cache hit ratio is approximately 20%. If the hit ratio falls below this limit, it is a good idea to take the catalog trashing the cache out of CDSC to increase the hit ratio for the remaining catalogs. Otherwise, the resource use is too high compared to the benefit of caching.

  How an ICF catalog is taken out of default caching is shown in Example 2-3.

Example 2-3 Example of how to disable an ICF catalog from default caching

```
F CATALOG,NOISC(cat)
```

ICF catalog sharing and Enhanced Catalog Sharing

Enhanced Catalog Sharing (ECS) is a catalog function that ensures the read and write integrity of ICF catalogs at the data set level (not at logical record level) as done by VSAM RLS. ICF catalog updates in a shared environment need extra steps and checks for integrity reasons. The effect on performance is also a consideration.

ECS uses the following functions to implement integrity:

- Reserve macros to serialize accesses (shared or exclusive) to a shared ICF catalogs. It is a requirement that the SYSIGGV2 reserve be converted when ECS is used; otherwise, the catalogs are at risk of becoming damaged. The Reserves must be converted to ENQ Systems through the use of the GRS RNL convention list. This recommendation applies to SYSIGGV2, SYSZVVDS, and SYSVTOC major resource names.

- The use of a ICF catalog record that is named VSAM volume record (VVR) to ensure cache coherency; that is, to read the most current data contents. All updates that are done in the shared ICF catalog are time stamp logged in a wraparound fashion at the VVR. By using this technique, each catalog management can correctly update the cache contents.

  Originally, the VVR was kept at the ICF catalog data set. With ECS, it is kept in the CF structure that is named SYSIGGCAS_ECS. The gain in performance is achieved by keeping the information in the CF, which eliminates the need for an I/O operation.

Monitoring

To help you with assessing ICF catalog performance, the extract and report jobs must be submitted regularly. CAS keeps many counters that describe all of the ICF catalogs performance metrics.These counters can be reset and displayed through **F CATALOG** type of commands.

Once a week (depending on your environment's activity levels), a display of the counters can be shown, captured, and then stored for further processing to create trend and exception reports. After the weekly extract, nullify the ICF catalog counters by using the following command:

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

By using this command, your reports are always based on a comparable period (one week), and trends and deviations can be calculated based on the start and end of the period.

A good overall report is the performance report, which reports for your entire environment and not by ICF catalog. To produce this report, use the **F CATALOG,REPORT,PERFORMANCE** command. This report summarizes all catalog events, and gives you an idea over time about how much your environment grows, as shown in Example 2-4.

Example 2-4 Partial output of the F CATALOG,RESPORT,PERFORMANCE command

```
F CATALOG,REPORT,PERFORMANCE
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG PERFORMANCE REPORT 826
*CAS************************************************
*  Statistics since  9:56:00.70 on  09/27/2016       *
*  -----CATALOG EVENT----   --COUNT--  ---AVERAGE---  *
*  Entries to Catalog          6,181      5.658 MSEC  *
*  BCS ENQ Shr Sys             8,326      0.743 MSEC  *
*  BCS ENQ Excl Sys               66      0.009 MSEC  *
*  BCS DEQ                    12,183      0.003 MSEC  *
*  VVDS RESERVE CI             1,791      0.006 MSEC  *
*  VVDS DEQ CI                 1,791      0.005 MSEC  *
*  VVDS RESERVE Shr           13,022      0.153 MSEC  *
*  VVDS RESERVE Excl              22      0.321 MSEC  *
*  VVDS DEQ                   13,044      0.006 MSEC  *
*  SPHERE ENQ Excl Sys            54      0.478 MSEC  *
*  SPHERE DEQ                     54      0.005 MSEC  *
*  CAXWA ENQ Shr                 887      0.004 MSEC  *
*  CAXWA DEQ                     887      0.001 MSEC  *
*  VDSPM ENQ                   8,344      0.009 MSEC  *
*  VDSPM DEQ                   8,344      0.001 MSEC  *
*  BCS Get                    15,411      0.039 MSEC  *
*  BCS Put                        45      0.239 MSEC  *
*  BCS Erase                       4      0.217 MSEC  *
*  VVDS I/O                   14,842      0.230 MSEC  *
```

```
*  VLF Define Major               1      0.133 MSEC  *
*  VLF Identify               1,386      0.000 MSEC  *
*  RMM Tape Exit                  4      0.000 MSEC  *
*  RMM Tape Exit                  4      0.000 MSEC  *
*  OEM Tape Exit                  4      0.000 MSEC  *
*  BCS Allocate                  27      1.793 MSEC  *
*  SMF Write                    870      0.016 MSEC  *
*  CAS Cache Lock                15      0.011 MSEC  *
*  CAS Alloc Lock                 3      3.454 MSEC  *
*  CAS MLA Lock                  15     10.334 MSEC  *
*  IXLCONN                        2      0.301 MSEC  *
*  MVS Allocate                   7      6.685 MSEC  *
*  Capture UCB                    4      0.005 MSEC  *
*  RACROUTE Auth                351      0.134 MSEC  *
*  Get LatchShr #            16,143      0.000 MSEC  *
*  ENQ SYSZPCCB               2,474      0.001 MSEC  *
*  DEQ SYSZPCCB               2,474      0.000 MSEC  *
*  Release Latch #           16,163      0.000 MSEC  *
*  Capture to Actual             10      0.001 MSEC  *
*  Get LatchExc #                20      1.127 MSEC  *
*  ENQ SYSIGGV1                   2      0.002 MSEC  *
*  DEQ SYSIGGV1                   2      0.001 MSEC  *
*  ENDREQ                         5      0.001 MSEC  *
*  SYSVSAM S ENQ Excl            28     16.871 MSEC  *
*  SYSVSAM S DEQ                 28      0.003 MSEC  *
*  SYSVSAM D ENQ Shr             27      0.000 MSEC  *
*  SYSVSAM D DEQ                 27      0.002 MSEC  *
*CAS*************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

The most important line in Example 2-4 is the first line in the Catalog Event list. Since the last counters reset, we had 6181 K requests (COUNT) to all ICF catalogs of this z/OS and the average response time (AVERAGE) was 5.658 milliseconds.

For command-issued reports, consider the use of an automation package to issue these commands on a regularly defined interval, which gives the most valuable information that is based on your systems' behaviors and profile. Capture the output from these commands and then format and record it into a data set for later retrieval.

Another useful report to produce weekly is the cache report, which reports by ICF catalog and can identify tuning candidates by ICF catalog name. To create this report, run the **F CATALOG,REPORT,CACHE** command. An extract of this command output is shown in Example 2-5.

Example 2-5 Extract of output from F CATALOG,REPORT,CACHE command

```
*CAS*************************************************************
*   HIT% -RECORDS- -SEARCHES --FOUND-- -DELETES- -SHR UPD- --PURGE-- *
*                                                                   *
* UCAT.ZOSR1A                                  (ISC)              *
*    93%        1        15        14         0         0         0  *
* UCAT.VTFMTAPE                                (ISC)              *
*    66%        1         3         2         0         0         5  *
```

The cache report shows cache hit ratio. This ratio must be investigated per ICF catalog. It is suggested to be 20% and above. If ICF catalogs continually stay below 20%, they are candidates for removal from the cache or for moving to another caching option. A high purge might also affect performance and must be investigated and responded to. Purges are generated by the following activities:

- ISC by an update in a shared catalog. To avoid purge at ISC, move UCATs to VLF caching.
- VLF with ECS, when the VVR wraparound occurs. This wraparound means that update information was lost since the last time this catalog manager inspected the VVR. The VLF cache is purged to ensure the integrity.

A catalog statistics report that is issued by using the **F CATALOG,REPORT,CATSTATS** command gives an overview of the activity on the individual catalogs and of the buffering (`BUFND`, `BUFNI`, and `STRNO`). An extract of the output is shown in Example 2-6.

Example 2-6 Extract of output from F CATALOG,REPORT,CATSTAT

```
*CAS*************************************************************
*     ADDS  UPDATES     GETS   GETUPD  DELETES  BUFNI  BUFND STRNO *
*                                                                 *
* UCAT.ZOSR1A                                                     *
*       0        0       16        0        0      10     16     8 *
* UCAT.VTFMTAPE                                                   *
*      57      181      749      181        0      10     16     8 *
```

A similar report is created by using the **F CATALOG,REPORT,CATSTATX** command. This report includes information about CA reclaim and CA reuse. It now also reports on `CPU TIME` and `ELAPSE TIME` (these two items were added in z/OS V2.1).

Before z/OS V2.R2, it was difficult to collect accurate statistics about a ICF catalog when ICF catalogs were not closed cleanly because of errors. Information about how many CAs were reused were viewed only from the **CATSTATX** command, which collects information since the last initial program load (IPL) only.

However, this statistic problem was solved before DFSMS V2.R2 by issuing the following command:

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

The **CATSTATX** command output is shown in Example 2-7.

Example 2-7 CATSTATX command output

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

```
*  STRNO SETTING    =        8                                        *
*  AVG ELAPSED TIME =     0.397 MSEC                                 *
*  AVG CPU TIME     =    95.505 USEC                                 *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Also at z/OS V2.R2, the CAS command **MODIFY CATALOG,CLOSE** was enhanced and internally addresses this issue to keep statistics updated in VVR and close catalogs properly. The major benefits on this enhancement are that statistics are available from a **LISTCAT** as with any other KSDS.

System Management Facilities reports

System Management Facilities (SMF) data can also be used for catalog reporting. You can create your own reports based on Restructured Extended Executor (REXX) code or similar. Look into the following SMF record types to investigate ICF catalogs:

- Type 60: VVR updated
- Type 61: ICF define
- Type 62: VSAM Open
- Type 64: VSAM Close (new for ICF catalogs from z/OS V 1.13)
- Type 65: ICF delete
- Type 66: ICF alter

Resource Measurement Facility reports

For the RLS-managed catalogs, there are also RLS reports in IBM Resource Measurement Facility™ (RMF3) that you can run:

1. From the primary panel, select S for SYSPLEX.
2. In Sysplex Reports on the next panel, you will see these 3 RLS reports at the bottom of the panel, as shown in Example 2-8 on page 31.

Example 2-8 RMF3 reports as displayed in option 10, 11, and 12 in SYSPLEX option

```
01 SYSSUM   Sysplex performance summary               (SUM)
02 SYSRTD   Response time distribution                (RTD)
03 SYSWKM   Work Manager delays                       (WKM)
04 SYSENQ   Sysplex-wide Enqueue delays                (ES)
05 CFOVER   Coupling Facility overview                 (CO)
06 CFSYS    Coupling Facility systems                  (CS)
07 CFACT    Coupling Facility activity                 (CA)
08 CACHSUM  Cache summary                             (CAS)
09 CACHDET  Cache detail                              (CAD)
10 RLSSC    VSAM RLS activity by storage class        (RLS)
11 RLSDS    VSAM RLS activity by data set             (RLD)
12 RLSLRU   VSAM LRU overview                         (RLL)
```

2.3.7 Storage management subsystem

When planning catalog structure, consider how to use SMS settings to manage your catalogs. consider the following points about the close relationship between catalog and SMS:

- All SMS-managed permanent data sets must be cataloged.
- Catalogs can be defined as SMS-managed.

- The catalog data set does not need to be SMS-managed. However, it is suggested for catalogs containing entries of SMS-managed data also to be SMS-managed.
- SMS-managed and non-SMS-managed data sets can be cataloged to the same catalog.
- The extended format catalogs using extended addressability must be SMS-managed.

Therefore, considering the performance and availability requirements of different catalogs, it is important for you to work with Storage Management Team to define the appropriate SMS policy for catalogs, which includes the Data Class, Storage Class, Management Class, and Storage Group for SMS constructs and ACS routines. For more information about using catalogs with the SMS, see *z/OS DFSMS Managing Catalogs*, SC23-6853.

2.3.8 Features including parmlibs and caching considerations

Catalog setup offers granularity in settings that can improve the overall performance, automation, and monitoring of your catalogs. Implementing the appropriate settings can help you build a more robust and stable environment for the systems software and business applications running in your organization.

Important ICF catalogs tasks

The following areas will help to keep your ICF catalog environment effective and efficient.

Catalog address space

The CAS is the address space that services catalog requests. This address space is started at IPL, and holds all user catalog and alias names in control blocks within the address space. 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 the request or cause an abnormal end of task (abend) if, for any reason, the request is not satisfied. CAS has a specialty task called the *Analysis Task* that periodically checks CAS for errors.

CAS can be modified, diagnosed, and displayed by modify commands. The CAS address space can also be restarted, and is designed to have tasks timed out or restarted if errors occur. Parmlib parameter settings determine how CAS operates, such as how many concurrent tasks can run in CAS and how many user catalogs can be open at the same time.

In Storage Cache caching

ISC is one way of caching catalog records. Caching in ISC is caching that occurs within the CAS. ISC is the default catalog caching if no other caching is requested. If you do not want the catalog in ISC for some reason, you must remove this from cache by a command or by caching in CDSC instead.

For the MCAT, all records referred are cached in ISC except for alias records, which are stored in a separate table in storage.

For UCATs, each catalog is dedicated a certain amount of storage in ISC, and only accessed records are cached. When this limit is reached for an UCAT, records start being removed from the ISC cache based on an LRU algorithm.

CDSC caching

CDSC caching occurs in a VLF data space. This process is defined in the COFVLFxx member in SYS1.PARMLIB. CDSC uses VLF, so this process must be started to have the caching in effect. Adding catalogs to VLF requires a stop and start of VLF to have the added catalogs start using VLF.

Caching catalogs in VLF is not a constraint at a catalog level as in ISC. Catalog records are cached until the data space limit is reached (set by the **MAXVIRT** parameter in the COFVLFxx member). Only then do catalog records start being removed, again based on an LRU algorithm.

A sample of the COFVLFxx member is shown in Example 2-9. Here, you specify by name the catalogs that you want added to CDSC caching.

Example 2-9 Example of COFVLFxx member

```
CLASS NAME(IGGCAS)
EMAJ(UCAT.BCS1)
EMAJ(UCAT.BCS2)
…
MAXVIRT(4096|nnnn)
```

In this case, *nnnn* is the number of 4 KB blocks that you can specify. The minimum value is 256 MB, and the default value is 4096. A total of 15 4K blocks per ICF catalog is the suggested number. Therefore, if you have 50 ICF catalogs, this value should be (15*50) = 750 blocks.

Enhanced Catalog Sharing

ECS is an important performance feature. It is also an important sharing feature because VVDS time enqueued is reduced by ECS, and the potential of having ICF catalog hangs or lockouts can also be reduced by using ECS.

Next, we focus on the planning considerations for ECS and how to enable the feature.

ECS uses a structure in the CF for keeping the VVR record. The first step in implementing ECS is to size and allocate this structure. Sizing is done most easily by using the CFSizer, which is available at this website:

`http://www.ibm.com/systems/z/advantages/pso/index.html`

On the website, complete the following steps:

1. Choose **Supporting Products** and click **Coupling Facility Structure Sizer Tool**.
2. In this window, select the **Enhanced Catalog Sharing** option. The window that is shown in Figure 2-1 opens.

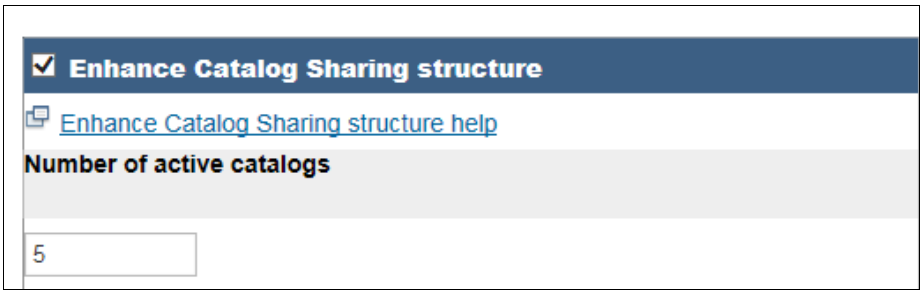


Figure 2-1 CF sizer pane for estimating ECS structure size

3. Enter the number of ICF catalogs that you expect to have under ECS control (in our example, 5 was chosen).

   When you submit this window, the results show you the size calculated at the top. Beneath that information is the JCL to create the structure.

Figure 2-2 shows an example of the sizing information in the window.

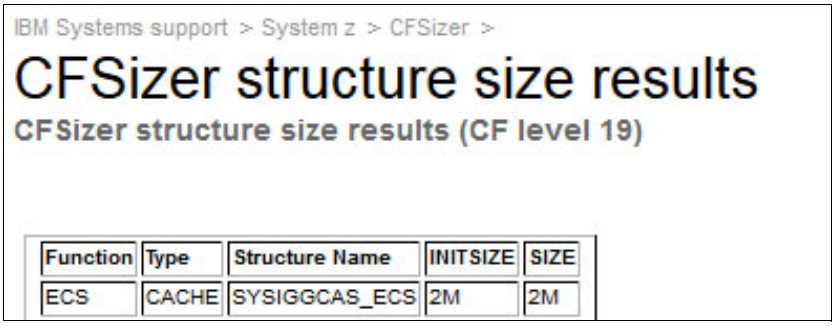


Figure 2-2 CF sizer result window with calculated structure size

4. Calculations that are done by CFSizer advise 2 MB in this case. In addition to performing the calculation, CFSizer lists the JCL that can be used for allocating the structure. Define the ECS structure in the coupling facility resource management (CFRM) policy and activate the policy.

   This action should connect all ECS-eligible systems to the ECS structure.

5. To allocate a new ICF catalog as an ECS candidate, you must use the IDCAMS keyword `ECSHARING`. This keyword is also supported on the IDCAMS **`ALTER`** command, which makes it possible to assign any ICF catalogs for ECS sharing.

6. In addition to the `ECSHARING` attribute on the ICF catalog, the ECS mode must be enabled by issuing the command that is shown in Example 2-10 on page 34.

   Example 2-10 Example of enabling ECS mode

   ```
   F CATALOG,ECSHR(AUTOADD)
   ```

7. `AUTOADD` connects the ECS eligible ICF catalogs to the structure at the next access to the ICF catalog. Activate `AUTOADD` automatically at IPL by setting the indicator in the SYSCATxx in `SYS1.NUCLEUS`. Alternatively, if you are using the IGGCATxx member, activate `AUTOADD` from there.

8. ECS mode and VVDS mode are mutually exclusive, but you can take an ICF catalog out of ECS mode on all systems by issuing the following command:

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

9. The `ECSHARING` attribute is still on the catalog, but the ICF catalog reverts to VVDS mode. You can return the ICF catalog to ECS mode by using the following command:

   ```
   F CATALOG ECSHR(ENABLE,catname)
   ```

10. If you want to stop ECS activity on one system, issue the following command:

    ```
    F CATALOG,ECSHR(DISCONNECT)
    ```

    This command disconnects ECS from the coupling facility structure on the system where the command is issued. All other systems stay connected to the CF structure, but quiesce their ECS activity.

ECS does not support the following scenarios:

- An ICF catalog cannot use ECS mode and VVDS mode protocols at the same time.
- A maximum of 1024 catalogs can be shared by using ECS from a single system.
- All systems sharing a catalog in ECS mode must be using the same CF and be in the same global resource serialization (GRS) complex.

For more information about ECS, see *z/OS V2R1.0 DFSMS Managing Catalogs*, SC23-6853.

SMSVSAM and record-level sharing

This section describes aspects of the SMSVSAM task.

Role of the SMSVSAM task

RLS is supported on ICF catalogs from DFSMS V2.1. SMSVSAM is the server task that handles RLS requests in general, and for the ICF catalogs with RLS enabled. For sharing and buffering, a storage class is required. The storage class is assigned with a cacheset and lockset identifying cache and lock structures in the CF, plus a data class with the buffering requirements. Additionally, for recovery and restart, define two sharing control data sets (SHCDSs) and one spare. These data sets hold critical restart information for SMSVSAM.

The SMSVSAM address space automatically starts at IPL if the `RLSINIT (YES)` keyword is specified in the IGDSMSxx member of `SYS1.PARMLIB`. If SMSVSAM is not started, you can start the address space after IPL by issuing the **`V SMS,SMSVSAM,ACTIVE`** command from the operator console.

Buffering

For catalogs in RLS mode, the catalog records are placed in RLS local buffer pools or CF cache structures. RLS obtains buffers dynamically, because they might be needed using system-managed buffering (SMB).

For more information about RLS for catalogs, see Chapter 5, “Record-level sharing support for ICF catalogs” on page 87.

Catalog parmlib

z/OS V1.13 introduced the catalog parmlib, a new IGGCATxx member in `SYS1.PARMLIB`, that can contain most ICF catalog settings. Using this parmlib member, it is possible to gain an understanding of the catalog settings just by looking into the parmlib member’s specifications. Only the *master catalog name* and *master catalog volume* need to go into the SYSCATxx member in `SYS1.NUCLEUS` or the LOADxx member in `SYS1.PARMLIB`.

Keywords that can be specified in the catalog parmlib are listed in Table 2-1. Most of these words can be changed dynamically by command and do not require an IPL to be updated. The table also shows the individual keywords, their defaults, and which keywords can be updated by command.

Table 2-1 Catalog parmlib overview

Keyword	Description	Default	Modify command
Aliaslevel	Multi-level alias number (1 - 4)	1	YES
ALLOCLCK	ALLOCLCK contention setting	10 minutes per notify	CONTENTION
AUTOADD	ECS AUTOADD (ON/OFF)	OFF	ECSHR
CATMAX	Maximum number of open catalogs (1 - 9999)	1024	YES
DELFORCEWNG	Warning MSG IDC1997I or IDC1998I	YES	ENABLE | DISABLE
DELRECOVWNG	Warning MSG IDC1999I	NO	ENABLE | DISABLE
DSNCHECK	Syntax check on names added	YES	ENABLE | DISABLE
DUMP	Dynamic dumps available?	OFF	YES

Keyword	Description	Default	Modify command
DUMPON	RC, RSN, MOD, CNT	1 (CNT)	YES
EXTENDEDALIAS	Extended alias can be created	NO	ENABLE | DISABLE
GDGFIFOENABLE	First-in first-out (FIFO) ordering on the generation data group (GDG)	NO	Not applicable (N/A)
NOTIFYEXTENT	ICFCAT extents exceeded (IEC361I)	80%	YES
SYMREC	Enabling of SYMREC creation	YES	ENABLE | DISABLE
SYS% ON OFF	SYS% to SYS1 conversion enabled	OFF	YES
SYSIGGV2	SYSIGGV2 contention setting	10 minutes per notify	CONTENTION
SYSZTIOT	SYSZTIOT contention setting	10 minutes per notify	CONTENTION
SYSZVVDS	SYSZVVDS contention setting	10 minutes per notify	CONTENTION
TAPEHLQ	High-level qualifier (HLQ) Tape Volume Catalog	SYS1	N/A
TASKMAX	Maximum number of user service tasks in CAS	180	YES
TASKMIN	Minimum number of user service tasks in CAS	60	YES
TASKTABLESIZE	Maximum number of catalog tasks	200	N/A
UPDTFAIL	IEC390I warning catalog update error	YES	ENABLE | DISABLE
VVDSSPACE	Default allocation size for VVDS	Tracks(10,10)	YES
VVRCHECK	Enhanced VVR checking on VVDS?	NO	ENABLE | DISABLE

It is advised to use catalog parmlib and, when enabled, add all of the keywords with the defaults, unless you want a different value on some keywords. Having the keywords that are specified in catalog parmlib enables you to dynamically change most of these keywords if you encounter an issue in which a changed value is needed.

2.3.9 System software considerations

The need for ever-improving ways to manage storage at the physical and logical level drives system software innovation to provide enhanced tools to meet modern demands to store and retrieve data. Each new release of z/OS and DFSMS continues to add functionality to this area. Other changes can be introduced by software maintenance.

Establish connections with the parties responsible for the installation, customizing, testing, and implementation of system software to assess any added or modified functionality. The inclusion of storage managers into the system software upgrade and maintenance process help prove that the ICF catalog environments were not negatively affected by the upgrade or maintenance application.

Also take the opportunity to discuss and evaluate new functionality introduced by the system software upgrade or maintenance implementation that adds value to business applications or environments.

2.3.10 Testing environments

A test system can be created and used by several resources that are available for support teams to create test scenarios. The scenarios are isolated or protected from other environments, and do not affect the production systems. These resources can vary from a simple set of direct access storage device (DASD) and tape volumes, to an entire LPAR or SYSPLEX environment.

You are highly advised to implement a test environment for your business. A test environment helps reduce the number of outages and performance degradation on production systems related to program code errors, gaps on implementation plan, incompatibility, and others.

If sufficient resources are not available to implement an LPAR or SYSPLEX test system, consider creating one of the following environments to enable your support team to perform the minimum necessary tasks before implementing projects on a production system:

- A set of DASD space

  Provide DASD space to help support teams test allocation of data sets and catalogs, fix a catalog in error, merge catalogs, test SMS policies, and other tasks.

- A set of DASD space with IPlable volumes

  This environment has all of the advantages of the previous option, plus the chance of installing products and fixes on these volumes. The test system can be started for testing on availability windows, and used to assist recover production if there is an outage.

- A copy of the production system with the main products installed

  In addition to the other advantages already mentioned, you can test upgrades on your main products, and back out the changes almost instantaneously when there is a failure.

2.4 Housekeeping

Whether you are planning to go through a system maintenance, consolidation, split, or only perform routine tasks, it is suggested to plan and run housekeeping tasks to maintain the system's performance and prevent errors. These tasks can be configured to run through automation or a job scheduler, and the output saved on data sets for reporting purposes, as listed in Table 2-2.

Table 2-2 Housekeeping in your environment

Threat	Prevention plan	Frequency
Catalog loss	You should always make sure that you back up a valid catalog, so you should run the **DIAGNOSE** and **EXAMINE** commands before each backup. You can run backups using different solutions to extend your recovery options.	At least once a day
Index and data structure errors	The IDCAMS **EXAMINE** option can be used to identify and fix index and data structure errors. IDCAMS checks if the data or index structure contains errors, and tries to fix it if requested. Also run **EXAMINE** before catalogs move or reorganize to ensure catalog integrity.	Weekly
Index and data integrity errors	IDCAMS also provides **DIAGNOSE** to identify and fix index and data integrity errors. It is advised to run **EXAMINE** before catalogs move or reorganize, to ensure catalog integrity.	Weekly

Threat	Prevention plan	Frequency
Catalog performance	The CAS provides plenty of reports on catalogs and cache performance. Define jobs to extract and process the data to create catalog performance reports, usage trends, and other information.	Monthly
Catalog growth	Set up a job or automation to collect catalog allocation information. This can include allocation extents and data set usage. Define plans to perform the necessary action.	Monthly
VVDS errors	Run IDCAMS **DIAGNOSE** against VVDSs to detect integrity errors.	Weekly
VVDS loss	Full-volume memory dumps are considered the best way to reduce the effects from a VVDS loss. Because this function is time-consuming and uses space, plan volume dumps based on volume importance and availability windows.	As required

Remember: All of the prevention plans and frequencies are suggestions and can vary from site to site, depending on your organization's needs.

2.5 Reporting

A well-structured reporting process is vital to a well-managed system. Each report must have a specific function, be informative, be easy to follow, have a purpose, and be effective in maintaining high-quality ICF environments. Reports can analyze your data growth, list application changes, identify bottlenecks, identify system trends, and so on.

Some of the reports can be fed into a process (such as REXX routines) and subsequent actions automatically deployed. Other reports can be issued for review purposes. Plan to create reports that are meaningful to the teams and the business, and store multiple versions of these reports, to enable identification of usage peaks, trends, and threats. Some reports that can be defined on your systems are listed in Table 2-3. You can also create your own reports based on customer needs.

Table 2-3 Types of reports and most common usage

Report type	Report description	Frequency
Catalog cache	Cache reports provide information related to cache searches and hits. A low cache hit can result in increased CPU usage for catalog task and low performance. Catalog cache report is not applicable for RLS catalogs.	Monthly
Catalog performance	This report contains information for each catalog event activity, including the count of events and average time. You can use this report to identify events that are taking too long to run, and plan corrective actions.	Monthly
Catalog statistics	Shows number of adds, updates, gets, and delete for a specified catalog. Use this report to analyze the number of requests processed for a catalog.	Monthly
VLF statistics	You can create jobs to extract VLF statistics from SMF records to define the best VLF cache values for catalogs.	On demand

Report type	Report description	Frequency
SMF data	There are other information available on SMF data to report on catalogs behavior. Plan creating specific jobs to extract and handle the information for use.	On demand

2.6 Monitoring and automation

This section focuses on built-in functions in the standard ICF catalog software. The need for automation goes beyond the options that are described here, but you should understand and consider these options.

2.6.1 Catalog contention detection

Catalog contention detection is a useful feature that monitors core catalog resources for time-outs. At time-out notification can be issued, or a redrive action can be initiated. Initially, only the SYSZTIOT (input/output table) resource could be notified on. But DFSMS V2.1 introduced the following extra resources that can be alerted on (in addition to SYSZTIOT):

- SYSZVVDS: Used to serialize access to associated VVDS records.
- SYSIGGV2: Used to serialize access to associated catalog resources.
- ALLOCLCK: A CAS internal lock that ensures serialization of access to most catalog allocation events.

Besides monitoring, catalog contention detection now also includes the possibility of triggering an action per resource monitored. Wait time and actions by resource must be set in the IGGCATxx member of the `SYS1.PARMLIB` catalog member, as shown in Example 2-11. When re-drive is active the first time a service task with an active resource passes the contention threshold, an abnormal end of task (abend) of the service task occurs. The request is then resubmitted to the catalog for processing.

Example 2-11 Catalog contention detection: Sample settings

```
SYSZTIOT(11,N)
SYSZVVDS(12,R)
SYSIGGV2(13,R)
ALLOCLCK(0,N)
```

The first parameter on the individual resource in Example 2-11 is the time-out value. If this parameter is set to `0`, the function is disabled for this resource. The second parameter is the action. The following actions are valid:

- `N`: Notification Only
- `R`: Redrive

The catalog detection function monitors resources looking into wait time, and issues alerts on those exceeding wait time. The following process starts based on the extended wait time:

1. A symptom record is written to LOGREC.
2. Message IEC393I is issued, which provides information about the task.
3. If the same task is still pending after another 5 minutes, message IEC393I is issued again.
4. Issuing message IEC393I next continues at 15-minute intervals until the wait ends.
5. At end, the status is updated in the next system check (within 30 seconds).

Catalog Contention time-out and re-drive settings can be displayed or changed by command, as shown in Example 2-12.

Example 2-12 Display and modify command for catalog contention

```
F CATALOG,CONTENTION
F CATALOG,CONTENTION(SYSZTIOT,,R)
```

The top command is the display command. The second command changes SYZTIOT to have the re-drive option. The output from the display command is shown in Example 2-13.

Example 2-13 F CATALOG CONTENTION output display

```
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG CONTENTION REPORT
*CAS*****************************************************
* RESOURCE        THRESHOLD(MIN)    ACTION(S)           *
*CAS*****************************************************
* SYSZTIOT           11             N                   *
* SYSZVVDS           12             NR                  *
* SYSIGGV2           13             NR                  *
* ALLOCLCK        INACTIVE          N                   *
*CAS*****************************************************
* ACTION KEY                                           *
* N = NOTIFY OPERATOR  R = REDRIVE REQUEST             *
*CAS*****************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

2.6.2 CA RECLAIM of VSAM KSDS data sets

The **CA RECLAIM** function (supported on VSAM KSDS only) enables reuse of empty internal (below HURBA) CAs by adding freed-up CAs to the free CA list. This makes what was previously unusable now eligible for immediate reuse. Before the **CA RECLAIM** function, the empty CA must wait until the next REORG of the VSAM KSDS. This maximizes the usage of your VSAM KSDS, and makes those more efficient by reducing the need for performing a REORG, which can improve the availability figures for key application data sets.

CA RECLAIM also works for ICF catalogs. ICF catalog maintenance can be particularly disruptive, because many applications can use the same ICF catalog. Therefore, the **CA RECLAIM** is useful, and is enabled by adding a new keyword, **CA_RECLAIM(DATACLAS)**, in the IGDSMSxx member in SYS1.PARMLIB. The default value is NONE.

Activate the **CA RECLAIM** function by refreshing your IGDSMSxx member by issuing the **SET SMS=*xx*** command. In this case, *xx* is the suffix of your active IGDSMS member. Alternatively, directly issue the activation command: **SETSMS CA_RECLAIM(DATACLAS)**.

Assign a DATACLAS to the data sets in scope, where the **CA RECLAIM** option is set to Y. For existing data sets, issue an IDCAMS **ALTER RECLAIMCA** command.

CA RECLAIM does not work under the following circumstances:

- Partially empty CAs. CAs must be freed up before reuse is possible.
- CAs that are empty before **CA RECLAIM** is enabled.
- CA with RBA 0 and the CA with the highest key of the KSDS.
- CAs in KSDS with the IMBED option active.
- Application that is opening a data set with global shared resources (GSR).

The **F CATALOG,REPORT,CATSTATX(catalog_name)** command displays reclaim activity and other ICF catalog information, as shown in Example 2-14 on page 41.

Example 2-14 CA-reclaim s and CA-reuses

```
*CAS*************************************************************
*  CATALOG NAME     = MY.CATALOG.EXAMPLE *
*  INSERTS (ADDS)   =   37,469                                   *
*  UPDATES          =   73,706                                   *
*  RETRIEVES        =    6,870K                                  *
*  RETRIEVES FOR UP =  133,816                                   *
*  ERASES (DELETES) =   29,661                                   *
*  CA-RECLAIMS      =        1                                   *
*  CA-REUSES        =       80                                   *
*  BUFNI SETTING    =        7                                   *
*  BUFND SETTING    =       11                                   *
*  STRNO SETTING    =        3                                   *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETE
```

Example 2-13 on page 40 and Example 2-14 show activity since the last reset of the catalog figures, or since the most recent IPL, where CAS was restarted. Only one reclaim was done, but 80 CAs became reusable. The following CA reclaim fields:

- CA-RECLAIMS

  The number of CA reclaims performed by VSAM on this catalog over the reported interval. For r RLS catalogs, this information is displayed as `N/A` because the values are unavailable.

- CA-REUSES

  The number of CA reuses performed by VSAM on this catalog over the reported interval. For RLS catalogs, this information is displayed as `N/A` because the values are unavailable.

2.6.3 Notify extents

`NOTIFYEXTENTS` is one of the IGGCATxx options that you might consider and enabled. With this parameter, you set the value for the maximum allowed number of extents for your ICF catalogs. If this is set to `80` (the default), you receive a notification in SYSLOG when the threshold is reached, as shown in Example 2-15.

Example 2-15 Notification in SYSLOG, when extent limit is reached on your ICF catalog

```
IEC361I CATALOG catalogname (data or index) HAS REACHED xx% OF THE MAXIMUM EXTENTS
```

You can use the warning message to trigger a notification from your automation package, and in this way be able to plan and prepare a REORG (or maybe a reallocation with a larger primary and secondary size) of your ICF catalog before the limit of 123 extents is reached.

2.6.4 Health checks done by IBM Health Checker for z/OS

The IBM Health Checker for z/OS also has warnings on ICF catalog-related setup. From the IBM Health Checker panel in IBM z/OS System Display and Search Facility (SDSF), you can monitor which health checks of all possible checks are enabled or disabled. To access SDSF,

type CF. Example 2-16 on page 42 shows an extract of the catalog-related and RLS-related health checks.

Example 2-16 IBM Health Checker: Current checks related to ICF catalog and RLS

```
NAME                              CheckOwner       State              Statu
CATALOG_IMBED_REPLICATE           IBMCATALOG       ACTIVE(ENABLED)    EXCEPTION-LOW
CATALOG_RNLS                      IBMCATALOG       ACTIVE(ENABLED)    SUCCESSFULL
VSAMRLS_CFCACHE_MINIMUM_SIZE      IBMVSAMRLS       ACTIVE(DISABLED)   GLOBAL-lpar
VSAMRLS_CFLS_FALSE_CONTENTION     IBMVSAMRLS       ACTIVE(DISABLED)   GLOBAL-lpar
VSAMRLS_DIAG_CONTENTION           IBMVSAMRLS       ACTIVE(ENABLED)    SUCCESSFUL
VSAMRLS_QUIESCE_STATUS            IBMVSAMRLS       ACTIVE(ENABLED)    SUCCESSFUL
VSAMRLS_SHCDS_CONSISTENCY         IBMVSAMRLS       ACTIVE(DISABLED)   GLOBAL-lpar
VSAMRLS_SHCDS_MINIMUM_SIZE        IBMVSAMRLS       ACTIVE(DISABLED)   GLOBAL-lpar
VSAMRLS_SINGLE_POINT_FAILURE      IBMVSAMRLS       ACTIVE(DISABLED)   GLOBAL-lpar
```

Health check limits are based on IBM suggested values, but can be changed by setting a policy in the IBM Health Checker HZSPRMxx parmlib, or by a **MODIFY** command.

From the IBM Health Checker panel, you can activate or deactivate checks through line commands. As shown in Example 2-17, we enabled a health check of VSAMRLS. An **ACTIVATE** command is issued to the IBM Health Checker task in the background.

Example 2-17 IBM Health Checker ACTIVATE command

```
F HZSPROC,ACTIVATE,CHECK=(IBMVSAMRLS,VSAMRLS_SINGLE_POINT_FAILURE)
```

z/OS V2.R2 provides a new Catalog Health Check to inspect all of the catalogs that are defined in the environment for share options and volume status (shared or non-shared) and reports any inconsistencies between the two. System programmers can decide to redefine or alter the inconsistent catalogs with the correct share options during system downtime.

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

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

For more information, see *IBM Health Checker for z/OS User's Guide*, SC23-6843.

2.7 Suggested practices summary

ICF catalog best practices depend on your setup and your organization's requirements, but general guidelines can be set as a rule to assist you in your considerations on how to manage the ICF catalogs environment.

In this IBM Redbooks publication, we have already made quite a few suggestions. To summarize this advice, and to give an overview of leading practices, the following list describes them:

- Use **FREESPACE(0 0)** at allocation time and let splits happen.
- Use cylinder allocation and secondary allocation of more than one cylinder.
- Also use CA Reclaim on your ICF catalogs.
- Only reorganize your ICF catalogs if the ICF catalog grows in too many extents.
- In general, keep your control interval (CI) size (CISIZE or CISZ) on the data and index portion at 4 KB (unless you have many sequential reads).
- Given the following definitions, set `BUFNI=TI - (HURBA/CASZ) +1`. Extra index buffers improve direct access:
  - `BUFNI` is the number of buffers on the index.
  - TI is the total number of index records.
  - HURBA is the high usage relative byte address number.
  - `CASZ = CISZ x CI/CA`.
- Set `BUFND=2`.
- Set `STRNO=3`.
- Do not code `BUFFERSPACE`.
- For a shared ICF catalog use share option `(3 4)`.
- CDSC cache is suggested, but also consider ECS or RLS on your ICF catalogs.
- Remove Imbed and Replicate.
- Monitor your ICF catalogs and keep them free of errors.
- Have your backup and recovery procedures in place.
- Always apply current maintenance.
- Involve other teams in your plans and gain their approval.

3

Deploying ICF catalogs

In Chapter 1, “ICF Technical overview” on page 1, integrated catalog facility (ICF) environments scenarios were described. These configurations and environments are much simpler than the typical organization. However, they are illustrative, and for continuity purposes are used for the sample plan extracts and deployments covered in this chapter.

This chapter includes the following topics:

3.1 About the deployment

We take a common approach to each scenario, which consists of the following subtopics:

- The scenario
- A task list related to the catalog environment
- Sample job control language (JCL), commands, and output

When implementing a change on a system, perform all of the actions necessary to make sure that you can recover to a valid point in time if the changes fail for any reason. We strongly suggest that you back up your catalogs before implementing changes, such as the ones described in this chapter. It is also advised to use different backup tools for performing the backup, so you have a second recovery plan in case the first plan fails.

Some backup techniques include, but are not limited to, the following methods:

- IBM z/OS Data Facility Storage Management Subsystem (DFSMS) Access Method Services utility (IDCAMS) export
- Logical data set dump
- Full volume memory dump

3.2 Scenario 1: Simple ICF catalog environment structure

This section describes how to create the ICF catalog environment for a single logical partition (LPAR), single master catalog (MCAT), and four user catalogs (UCATs). The activities shown are related to the ICF catalogs. Additional considerations and steps might be required depending on your environment.

3.2.1 Target Environment

Figure 3-1 shows the target environment.

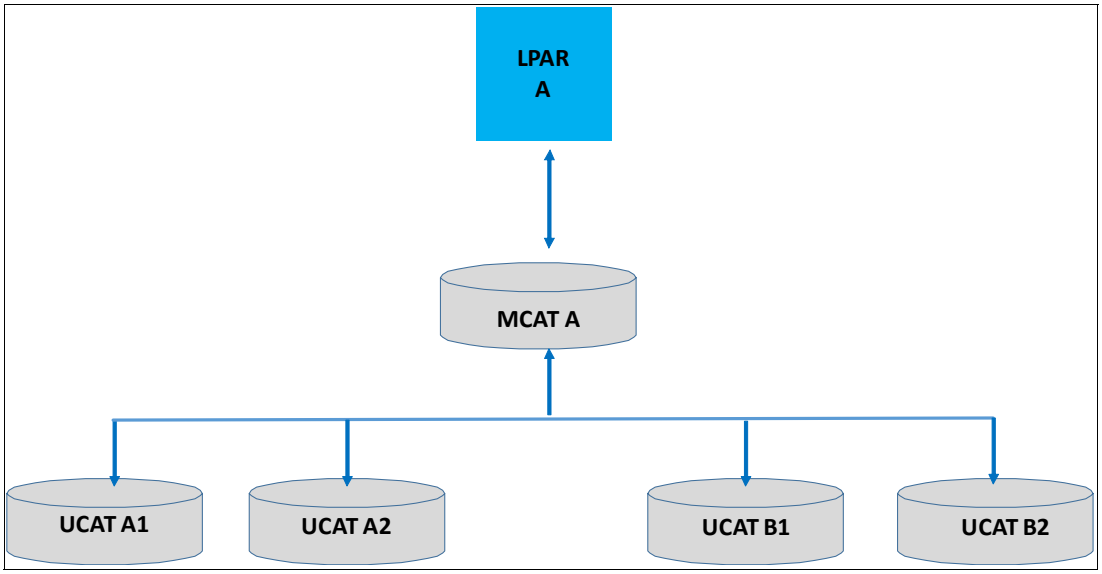


Figure 3-1 Single LPAR environment

3.2.2 Task list

The steps for creating a single environment are listed in Table 3-1. Upon completion of this plan, the system has one master catalog and four user catalogs. None of the user catalogs in this plan are shared.

Table 3-1 Scenario 1 activities

Step	Description
Define a master catalog.	Consider the following points when you define the system MCAT: ► The catalog size ► The volume to allocate the catalog ► The shareoptions ► The definitions for the sys1 data sets necessary to start the system from a powered off state
Define user catalogs.	Consider the following points when you define the system UCATs: ► The catalog size ► The volume to allocate the catalog ► The shareoptions ► Volume catalog (VOLCAT) planning considerations to include
Define the ALIASes and connect them to catalogs.	When defining the ALIASes, plan for the following aspects: ► ALIAS growth ► Grouping application ► Making sure that only required data sets are cataloged on the MCAT
Update the IGGCATxx and LOADxx members.	Before updating IGGCATxx or LOADxx members, define the following information: ► Where to place catalog parameters ► The default values used by the system
Create a disaster recover (DR) plan.	To create a DR plan, include the following information: ► Recovery point objective (RPO) ► Backup method ► Backup and recovery procedures
Create monitoring, automation, and report procedures.	The monitoring and report planning include defining the following tasks: ► What reports to extract ► The frequency the reports run ► How to store reports ► Error messages and alerts tracked by automation ► How to respond to alerts and errors
Create a maintenance plan.	Include the following tasks on your maintenance plan: ► What frequency the maintenance should run ► Negotiated availability windows for performing REORGs ► What reports to use to decide if maintenance is required

3.2.3 Deployment

The first step to implement Scenario One consists of creating the MCAT to be used by the system to be run in LPAR A. Complete the following steps:

1. It is suggested to dedicate a volume to the MCAT to avoid contention between the catalog and other data sets on the volume. Example 3-1 shows a sample user catalog **DEFINE** command, and the output from the command.

   It is assumed that you have a system running that you can use to define the catalog for the new LPAR A. The name of the new catalog to be used later by LPAR A is SYS1.MCATA. The IDCAMS parameters that define the master catalog are shown in Example 3-1.

   Important: Do not be confused by the **DEFINE USERCATALOG** statement. The new catalog that you are defining is a user catalog on the driving system but is the master catalog on LPAR A.

   Example 3-1 Define the MCAT

```
DEFINE USERCATALOG -
          (NAME(SYS1.MCATA) -
           ICFCAT -
           STORAGECLASS(NONSMS) -
           VOLUME (VOLMC1) -
           CYLINDERS (25 10) -
           FREESPACE(0 0) -
           SHR(3 3) -
           STRNO(3)) -
     DATA (CISZ(4096) -
           BUFND(4)) -
     INDEX(BUFNI(4))
IDC0510I CATALOG ALLOCATION STATUS FOR VOLUME VOLMC1 IS 0
IDC0512I NAME GENERATED-(I) SYS1.MCATA.CATINDEX
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

2. Next, define the following UCATs to be used in the LPAR A environment:
   - SYS1.UCATA1
   - SYS1.UCATA2
   - SYS1.UCATB1
   - SYS1.UCATB2

   It is suggested that UCATs are placed on separate volumes to provide a more balanced environment in terms of availability, clear structure, and performance. In addition, consider application usage, possible contention issues, and data growth.

   Example 3-2 shows the IDCAMS parameters to build the SYS1.UCATB2 user catalog. The other three catalogs use the same parameters, but have the appropriate name.

   Example 3-2 Define user catalog

```
DEFINE USERCATALOG -
          (NAME(SYS1.UCATB2) -
           ICFCAT -
           STORAGECLASS(NONSMS) -
           VOLUME(VOLUC2) -
           CYLINDERS(25 10) -
```

```
           FREESPACE(0 0) -
           SHR(3 3) -
           STRNO(3)) -
     DATA (CISZ(4096) -
           BUFND(4)) -
     INDEX(BUFNI(4)) -
     CAT(SYS1.MCATA)
IDC0510I CATALOG ALLOCATION STATUS FOR VOLUME VOLUC2 IS 0
IDC0512I NAME GENERATED-(I) SYS1.UCATB2.CATINDEX
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

3. After creating the catalogs, use the **IMPORT CONNECT** command to connect the UCATs into MCATA. This process establishes a relationship and structure between the MCAT and the UCATs, as shown in Example 3-3.

Example 3-3 IMPORT CONNECT catalogs

```
IMPORT CONNECT -
            OBJECTS( SYS1.UCATB2  -
            DEVICETYPE(3390) -
            VOLUMES(VOLUC2)) -
            CATALOG(SYS1.MCATA)

IDC0603I  CONNECT FOR USER CATALOG  SYS1.UCATB2 SUCCESSFUL
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

When you create the UCAT from the driving system, you might want to use the **EXPORT DISCONNECT** command to disconnect the catalogs from the current MCAT after catalogs are defined to prevent users from inadvertently updating other system's catalogs.

The next step adds ALIAS entries to the new master catalog to direct searches and actions to the appropriate user catalog.

4. Use a **DEFINE ALIAS** command to create all of the necessary ALIASes, and point to the correct catalogs. Consider also running **DEFINE ALIAS** commands on other LPARs if they should have access to the data, as shown in Example 3-4.

Example 3-4 Define ALIAS

```
DEFINE ALIAS(NAME(DB2)      RELATE(SYS1.UCATA1)) -
   CAT(SYS1.MCATA)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

   DEFINE ALIAS(NAME(CICS)     RELATE(SYS1.UCATA2)) -
   CAT(SYS1.MCATA)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

   DEFINE ALIAS(NAME(TIVOLI)   RELATE(SYS1.UCATB1)) -
   CAT(SYS1.MCATA)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

   DEFINE ALIAS(NAME(GDPS)     RELATE(SYS1.UCATB2)) -
   CAT(SYS1.MCATA)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

5. Before starting the system LPAR A by using the new MCAT, it is also necessary to catalog the minimum system data sets to start the system. Example 3-5 shows the **DEFINE** command for three system data sets on SYSV01.

Example 3-5 Define system data sets before system start

```
DEFINE NONVSAM (NAME('SYS1.PARMLIB') -
   DEVT(3390)          -
   VOLUMES(SYSV01)) -
   CAT(SYS1.MCATA)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 DEFINE NONVSAM (NAME('SYS1.PROCLIB') -
   DEVT(3390)          -
   VOLUMES(SYSV01)) -
   CAT(SYS1.MCATA)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 DEFINE NONVSAM (NAME('SYS1.LOADLIB') -
   DEVT(3390)          -
   VOLUMES(SYSV01))   -
   CAT(SYS1.MCATA)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

The initial set up of your ICF catalog environment is complete. Consider the other tasks that are not described here but are listed in Table 3-1 on page 47, and customize them to your needs, policies, and standards.

3.3 Scenario 2: Adding an LPAR

As the organization grows, it can be advisable to expand from the single LPAR to a SYSPLEX or multi-LPARs with data sharing using a coupling facility (CF).

3.3.1 Target environment

This scenario introduces a new LPAR using the same master catalog. The new LPAR B can be used as an application development and testing environment. LPAR A could be the production environment. The catalog structure is shown in Figure 3-2 on page 51.

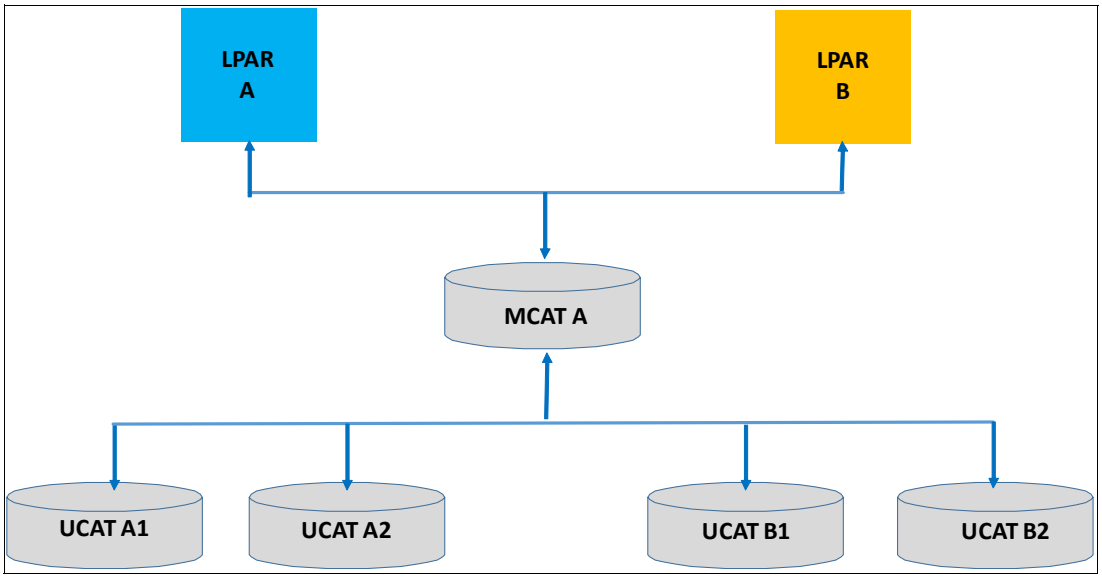


Figure 3-2 Catalog structure

3.3.2 Task list

LPAR B is to be added to an existing LPAR A environment connected to the existing master catalog MCAT A. An initial program load (IPL) is planned to be done from the same IPL volume with existing parameters. Input/output (I/O) generation (IOGEN) is planned to be the same as for the existing LPAR. The activities are Scenario 2 are listed in Table 3-2.

Table 3-2 Scenario 2 activities

Step	Description
Prepare the existing environment (LPAR A) for sharing.	Use the symbolic name &SYSNAME wherever possible, for example, LOGREC=SYS1.LOGREC.&SYSNAME. You should use &SYSNAME for the IEASYSxx parameters **VIODSN=**, **PAGE=**, **SWAP=**, **DUPLEX=**, and **NONVIO=**. You should also use &SYSNAME for the SMFPRMxx parameters **DSNAME=** and **SID=**.
Make changes to the existing MCAT.	Add any alias that might be unique for the new LPAR.
Prepare existing user catalogs for sharing.	User catalogs that are going to be shared should have these changes: ► Change the share option to SHR(3 4) if not already in place. ► Move shared catalogs to the virtual lookaside facility (VLF) for most efficient access. ► Ensure that catalogs are on a device defined as a shared unit control block (UCB). ► Consider using Enhanced Catalog Sharing (ECS). ► RLS is a strong option to consider. For more information about RLS, see Chapter 5, “Record-level sharing support for ICF catalogs” on page 87.
Make changes to the DR plan.	► Add a new LPAR to the DR plan. ► Define RPO and recovery time objective (RTO) for the new LPAR. ► Document recovery demands for the new LPAR.
Make changes to catalog monitoring, automation, and report procedures.	Consider the effects that this will have on the environments, and determine what areas you want to monitor and report on to ensure smooth operational running. Changes to your automation routines might be necessary to accommodate the shared environments.
Make changes to the catalog maintenance plan.	The documentation for recovery procedures should be updated, because the LPARs can now be used as backup for each other. Recovery can be done from the other LPAR if one is down.

3.3.3 Deployment

Complete the following steps to check if the current catalogs include the correct shareoptions values set:

1. Issue a **LISTC ENT(/) CAT(/) ALL** command next to the catalog name on TSO ISPF option 3.4. A sample of the output is shown in Example 3-6.

Example 3-6 Displaying catalog shareoptions

```
DATA ------- SYS1.MCATB
      HISTORY
        DATASET-OWNER-----(NULL)     CREATION--------2014.321
        RELEASE----------------2     EXPIRATION------0000.000
        ACCOUNT-INFO-----------------------------------(NULL)
      PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
      ASSOCIATIONS
        CLUSTER--0000000000000000000000000000000000000000000000
      ATTRIBUTES
        KEYLEN----------------45     AVGLRECL------------4086
BUFSPACE------
-----11776     CISIZE--------------4096
        RKP--------------------9     MAXLRECL-----------32400
EXCPEXIT------
----(NULL)     CI/CA----------------180
        BUFND------------------4     STRNO------------------3
        SHROPTNS(3,3)      SPEED     UNIQUE           NOERASE     INDEXED
```

2. The necessary value to share the catalog is `3,4`. If you already have this value set up, you can skip to the next step. You can alter the shareoptions value with the IDCAMS **ALTER** command, as shown on Example 3-7.

Example 3-7 Alter catalog shareoptions

```
ALTER   SYS1.MCATA SHAREOPTIONS(3,4)
IDC0531I ENTRY SYS1.MCATA ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATA1 SHAREOPTIONS(3,4)
IDC0531I ENTRY SYS1.UCATA1 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATA2 SHAREOPTIONS(3,4)
IDC0531I ENTRY SYS1.UCATA2 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATB1 SHAREOPTIONS(3,4)
IDC0531I ENTRY SYS1.UCATB1 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATB2 SHAREOPTIONS(3,4)
IDC0531I ENTRY SYS1.UCATB2 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

3. It is also advised to use ECS to improve catalog performance. Alter the catalog attribute to use ECS on your environment if possible. It is also required to define a cache structure and update the COFVLFxx member, as shown in Example 3-8.

Example 3-8 Define Enhanced Catalog Sharing

```
ALTER   SYS1.MCATA ECSHR
IDC0531I ENTRY SYS1.MCATA ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATA1 ECSHR
IDC0531I ENTRY SYS1.UCATA1 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATA2 ECSHR
IDC0531I ENTRY SYS1.UCATA2 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATB1 ECSHR
IDC0531I ENTRY SYS1.UCATB1 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

 ALTER   SYS1.UCATB2 ECSHR
IDC0531I ENTRY SYS1.UCATB2 ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

4. For new changes to take effect, it is necessary to close the catalog on CATALOG address space. Because the **CLOSE** command removes only the data from CATALOG memory and the data is reloaded on the next catalog request, there is no affect to the system or applications, as shown in Example 3-9.

Example 3-9 Close catalogs to activate new settings

```
3 //COMMAND  EXEC PGM=IEFBR14
4 // COMMAND 'F CATALOG,CLOSE(SYS1.MCATA)'
5 // COMMAND 'F CATALOG,CLOSE(SYS1.UCATA1)'
6 // COMMAND 'F CATALOG,CLOSE(SYS1.UCATA2)'
7 // COMMAND 'F CATALOG,CLOSE(SYS1.UCATB1)'
8 // COMMAND 'F CATALOG,CLOSE(SYS1.UCATB2)'
ICH70001I GILBION  LAST ACCESS AT 16:05:39 ON MONDAY, NOVEMBER 17, 2014
IEF142I REFCAT COMMAND - STEP WAS EXECUTED - COND CODE 0000
```

5. Before starting the systems, extra steps might be necessary, including updating parmlibs, copying data sets, and changing backup polices.

3.4 Scenario 3: Multiple MCATs and UCATs

In our third scenario, we create an MCAT to be used by LPAR B and move application data between the LPARs. Some of this data is shared by both LPARs, although other data sets are exclusive to each LPAR.

3.4.1 Target environment

We create `MCAT B` on `LPAR B`, and share `UCATS A2` and `B1` between both LPARs, as shown in Figure 3-3.

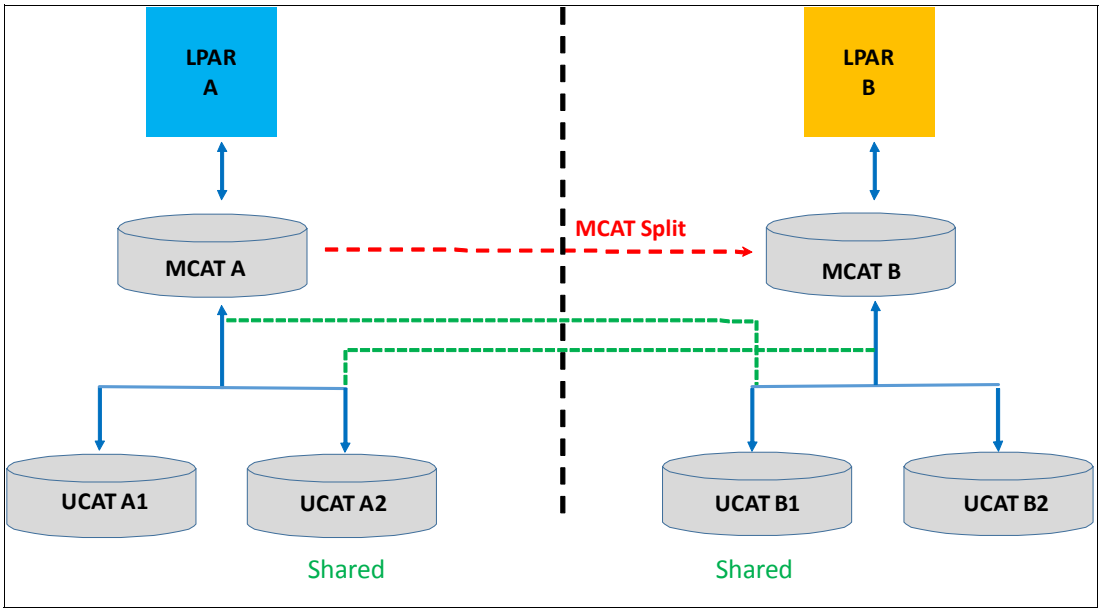


Figure 3-3 Create a new master catalog for LPAR B and introduce sharing

3.4.2 Task list

This scenario describes splitting one MCAT into two separate MCATs and moving several UCATs that are connecting to the new MCAT. At the same time, some UCATs can be shared by two MCATs. The steps for the suggested plan for this scenario are listed in Table 3-3.

Table 3-3 Scenario 3 activities

Step	Description
Examine and Diagnose the old MCAT and UCATs.	Make sure that the current MCAT and UCATs are error-free: ► Run IDCAMS **`VERIFY`** and **`EXAMINE`** commands on the current MCAT. ► Diagnose the integrity of the current MCAT.
Make changes to the existing MCAT.	Add any alias that might be unique for the new LPAR.
Prepare existing UCATs for sharing.	UCATs that are going to be shared should have these changes made: ► Change the share option to `SHR(3 4)` if it is not already in place. ► Move shared catalogs to VLF for most efficient access. ► Ensure that catalogs are on a device defined as shared UCB. ► For most optimum sharing, consider using ECS.
Make changes to the DR plan.	► Add a new LPAR to the DR plan. ► Define RPO and RTO for the new LPAR. ► Document recovery demands for the new LPAR.
Make changes to catalog monitoring, automation, and report procedures.	No changes.
Make changes to the catalog maintenance plan.	Recovery procedures (documentation) should be updated, because the LPARs can now be used as backup for each other (recovery can occur from the other LPAR if one is down).

3.4.3 Deployment

Ensure that all catalogs are error-free before creating the catalogs' backup and starting the split project. Complete the following steps:

1. Create a valid recovery point if there is a failure. The job output from **VERIFY**, **EXAMINE** and **DIAGNOSE** commands against a user catalog are shown in Example 3-10.

Example 3-10 VERIFY, EXAMINE, and DIAGNOSE catalogs

```
VERIFY  DATASET(SYS1.UCATA1)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

EXAMINE NAME(SYS1.UCATA1) -
    INDEXTEST NODATATEST ERRORLIMIT(1000)
IDC0170OI INDEXTEST BEGINS
IDC11773I             731 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KE
IDC11773I               7 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KE
IDC11774I CURRENT INDEX CISIZE IS  3584, RECOMMENDED MINIMUM INDEX CISI
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

DIAGNOSE ICFCATALOG INDATASET(SYS1.UCATA1)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

2. We also ran a **LISTC** job to list catalog entries (and the output directed to data sets) to compare the data set and ALIAS definition after the system split, as shown in Example 3-11.

Example 3-11 Output from the Listcat catalogs

```
IDCAMS  SYSTEM SERVICES
                            LISTING FROM CATALOG -- SYS1.UCATA1
         THE NUMBER OF ENTRIES PROCESSED WAS:
                   AIX ------------------0
                   ALIAS ----------------0
                   CLUSTER --------------2
                   DATA -----------------2
                   GDG ------------------0
                   INDEX ----------------1
                   NONVSAM ----------15968
                   PAGESPACE ------------0
                   PATH -----------------0
                   SPACE ----------------0
                   USERCATALOG ----------0
                   TAPELIBRARY ----------0
                   TAPEVOLUME -----------0
                   TOTAL ----------15973
```

3. Define a new master catalog for LPAR B. It is not recommended to place the new MCAT on the same volume as the current MCAT, to reduce the exposure to disk failures and contention. You can use your current MCAT as a model, or use specific allocation attributes, depending on your needs.

Example 3-12 shows the IDCAMS parameters.

Example 3-12 Define a new MCAT for LPAR B

```
DEFINE USERCATALOG -
          (NAME(SYS1.MCATB) -
           ICFCAT -
           STORAGECLASS(NONSMS) -
           VOLUME (VOLMC2) -
           CYLINDERS (25 10) -
           FREESPACE(10 10) -
           SHR(3 3) -
           STRNO(3)) -
     DATA (CISZ(4096) -
           BUFND(4)) -
     INDEX(BUFNI(4))

IDC0510I CATALOG ALLOCATION STATUS FOR VOLUME VOLMC2 IS 0
IDC0512I NAME GENERATED-(I) SYS1.MCATB.CATINDEX
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

4. Create a backup copy of your catalogs. This process can be done by using the IDCAMS **EXPORT** command, an ADRDSSU logical or physical memory dump, or other tools that are available on your environment. Check the output to confirm that the backup job ran with no errors. Create at least two backup copies of your catalogs, using different tools, to provide an alternative restore option, as shown in Example 3-13.

Example 3-13 Back up your catalogs

```
DUMP  DS(INCL(                  -
    SYS1.UCATB2                    -
          )) -
          OUTDD(OUT1) COMPRESS OPT(4) ALLDATA(*) ALLEXCP
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP

ADR109I (R/I)-RI01 (01), 2014.321 16:49:57 INITIAL SCAN OF USER CONTROL
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2014.321 16:49:57 EXECUTION BEGINS
ADR801I (001)-DTDSC(01), 2014.321 16:49:58 DATA SET FILTERING IS COMPLE
                         AND 0 FAILED FOR OTHER REASONS
ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROC
                          CLUSTER NAME   SYS1.UCATB2
                          CATALOG NAME   MCAT.SANDBOX.Z1C.SBOX00
                          COMPONENT NAME SYS1.UCATB2
                          COMPONENT NAME SYS1.UCATB2.CATINDEX
ADR006I (001)-STEND(02), 2014.321 16:49:58 EXECUTION ENDS

EXPORT SYS1.UCATB2    -
         OUTFILE(EXPDD) -
         TEMPORARY
IDC0005I NUMBER OF RECORDS PROCESSED WAS 5971
IDC0594I PORTABLE DATA SET CREATED SUCCESSFULLY ON 11/17/14 AT 16:49:59
IDC1147I IT IS RECOMMENDED THAT DIAGNOSE AND EXAMINE BE RUN BEFORE
IDC1147I IMPORT OF CATALOG
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

Attention: Avoid the use of a high-level qualifier that might be disconnected from your system. If you disconnect the ALIAS from the catalog, you might not be able to access your data set unless you catalog it again.

5. Use **REPRO NOMERGECAT** to copy the necessary information from MCAT A to MCAT B. **NOMERGECAT** requires the target catalog to be empty to complete the copy.

Attention: If you use the **MERGECAT** option, the data sets being moved are *deleted* from the source catalog. This issue can lead to system failures and outages to your system if system data sets are removed from the MCAT. The parameters that are used in this process are shown in Example 3-14.

Example 3-14 REPRO NOMERGECAT

```
REPRO -
        INDATASET(SYS1.MCATA) -
        OUTDATASET(SYS1.MCATB) -
        NOMERGECAT
IDC11468I NVR/VVR NOW POINTS TO TARGET CATALOG.
IDC0005I NUMBER OF RECORDS PROCESSED WAS 110
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Tip: If you are running z/OS 2.1 or later, all catalog and ALIAS information is also copied. You might need to **EXPORT DISCONNECT** the UCATs that are not accessed by the new MCAT.

6. Use the **EXPORT DISCONNECT** command to remove the catalog that is not accessed from LPAR A, as shown in Example 3-15. Ensure that *all* of the ALIASes on the catalogs that are exported are not used on LPAR A. If necessary, you can run **REPRO** commands to move these data set to other catalogs.

Example 3-15 Export disconnect catalog

```
EXPORT -
          'SYS1.UCATB2'      -
           CAT('SYS1.MCATA')   -
           DISCONNECT
IDC0144I VOLUME SERIAL FOR SYS1.UCATB2 DURING EXPORT DISCONNECT WAS
IDC0144I VOLUC2
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

If you are running z/OS V2.1 or later, the **REPRO NOMERGECAT** step also imports and connects the ALIAS to the new master catalog.

7. If you are running earlier versions of z/OS or are not sure if the ALIASes also were copied, you can run the **IMPORT CONNECT** command to connect user catalogs, as shown in Example 3-16.

Example 3-16 Connect catalogs to new master catalog

```
IMPORT CONNECT -
           OBJECTS( SYS1.UCATB2  -
           DEVICETYPE(3390) -
           VOLUMES(VOLUC2))  -
           CATALOG(SYS1.MCATB)
IDC0603I  CONNECT FOR USER CATALOG  SYS1.UCATB2 SUCCESSFUL
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

Tip: If the user catalog was imported, you receive a duplicate data set name error message.

8. Define the ALIAS used on LPAR B, as shown in Example 3-17. If the ALIAS information was copied during the **REPRO NOMERGECAT** step, you receive a duplicate data set name for the existing ALIAS.

Example 3-17 Define ALIAS to new master catalog

```
DEFINE ALIAS(NAME(CICS)     RELATE(SYS1.UCATA2)) -
  CAT(SYS1.MCATB)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DEFINE ALIAS(NAME(TIVOLI)   RELATE(SYS1.UCATB1)) -
  CAT(SYS1.MCATB)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DEFINE ALIAS(NAME(GDPS)     RELATE(SYS1.UCATB2)) -
  CAT(SYS1.MCATB)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

9. Check the integrity of your catalogs after the split by running a new **VERIFY**, **EXAMINE**, and **DIAGNOSE** against the catalogs, as shown in Example 3-18. Analyze the output to identify and correct any errors encountered.

Example 3-18 VERIFY, EXAMINE, and DIAGNOSE catalogs

```
VERIFY  DATASET(SYS1.UCATB2)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

EXAMINE NAME(SYS1.UCATB2) -
    INDEXTEST NODATATEST ERRORLIMIT(1000)
IDC0170OI INDEXTEST BEGINS
IDC11773I             275 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KE
IDC11773I               3 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KE
IDC11774I CURRENT INDEX CISIZE IS  3584, RECOMMENDED MINIMUM INDEX CISI
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

EXAMINE NAME(SYS1.UCATB2) -
    NOINDEXTEST DATATEST ERRORLIMIT(1000)
IDC01701I DATATEST BEGINS
IDC01709I DATATEST COMPLETE - NO ERRORS DETECTED
```

```
IDC01708I 275 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 5971 RECORDS
IDC01711I DATA COMPONENT CONTAINS 0 DELETED CONTROL INTERVALS
IDC01712I MAXIMUM LENGTH DATA RECORD CONTAINS 221 BYTES
IDC01722I 96 PERCENT FREE SPACE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

DIAGNOSE ICFCATALOG INDATASET(SYS1.UCATB2)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

10. Run a new backup of your catalogs and update any documentation and processes that are changed by the catalog split.

3.4.4 Scenario 4: Acquisition and integration

This scenario deals with a consolidation and integration of ICF catalog environments. Suppose that our sample organization acquires another organization, and therefore gains another LPAR. In this case, we call this `LPAR C`. The organization might want to consolidate all of the LPARs onto its own mainframe, but initially keep the data separate. So, the newly acquired organization would be an independent system with its own MCAT and UCATs.

3.4.5 Target environment

The ICF catalog environment for `LPAR C` appears as a separate system initially, as shown in Figure 3-4.

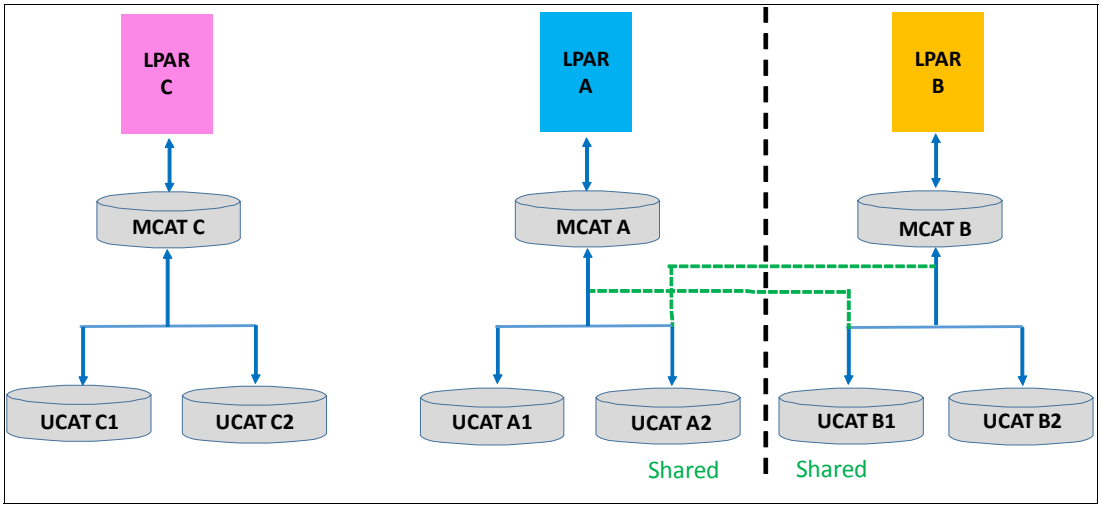


Figure 3-4 Acquisition of LPAR C

The goal to integrate the LPAR C ICF catalog environment into the LPAR B structure might look as shown in Figure 3-5. UCATs C1 and C2 are now connected to MCAT A. MCAT C is no longer required.

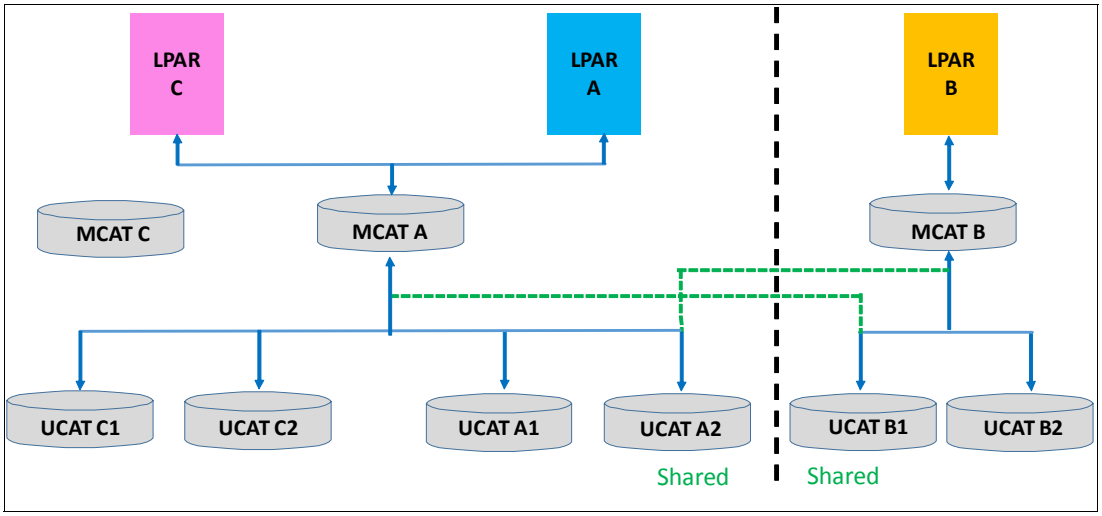


Figure 3-5 Integration of LPAR C into the full ICF catalog environment

3.4.6 Task list

The tasks that are listed in Table 3-4 describe merging MCAT C into MCAT A. It also includes consideration steps for identifying data set names conflicts.

Table 3-4 Scenario 4 activities

Step	Description
Examine and diagnose catalogs.	Make sure that all catalogs are error-free before performing backups and system merge: ► Run IDCAMS **VERIFY** and **EXAMINE** commands on all of the current catalogs. ► Diagnose the integrity of the catalogs.
Back up the catalogs.	Back up all of the catalogs involved in the merge process, to protect system resiliency.
Check your resources.	Confirm that all necessary resources (direct access storage device (DASD) and tape) are configured and accessible on the receiving system.
Analyze data set conflicts.	During a system merge, it is common that users, applications, and even user catalog names match on both systems. To prevent errors related to duplicate data set names during your merge, consider the following points: ► Analyze your systems to identify all data set and ALIAS conflicts. ► Define if any ALIASes or data sets should be renamed. ► If user catalogs will also be merged, consider catalog availability and sizing.
Prepare user catalogs for sharing.	Check if user catalogs have the necessary settings for sharing: ► Change shareoptions to SHR(3 4) if not already in place. ► Move shared catalogs to VLF for more efficient access. ► For most optimum sharing, consider using ECS.

Step	Description
Merge master catalog and define ALIASes.	During the merge window, shut down the sending system, and perform the following tasks: ► Merge catalogs with the **MERGECAT** option. ► Check the job output to identify and solve any remaining data set names conflicts. ► Define the ALIASes on the receiving MCAT.
Examine and diagnose catalogs.	Run new **VERIFY**, **EXAMINE**, and **DIAGNOSE** commands against the catalogs to make sure that they have no errors after system merge.
Back up your new configuration.	Create a new backup of your catalogs to protect the system from errors and outages.

3.4.7 Deployment

This sections describes the steps to consolidate two MCATs into a single MCAT used by both LPARs. It is assumed that LPAR A and LPAR C have access to each other's storage devices. Complete the following steps:

1. Before performing the catalog merge, check the integrity of your catalogs. We advise you to run at least the **VERIFY**, **EXAMINE**, and **DIAGNOSE** commands against the catalogs. You can also use more tools to check catalog integrity, as shown in Example 3-19.

Example 3-19 VERIFY, EXAMINE, and DIAGNOSE catalogs

```
VERIFY  DATASET(SYS1.UCATB1)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

EXAMINE NAME(SYS1.UCATB1) -                                          0016500
    INDEXTEST NODATATEST ERRORLIMIT(1000)                              0016600
IDC01700I INDEXTEST BEGINS
IDC11773I             275 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY LENGTH
IDC11773I               3 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KEY LENGTH
IDC11774I CURRENT INDEX CISIZE IS  3584, RECOMMENDED MINIMUM INDEX CISIZE IS  3
IDCAMS  SYSTEM SERVICES                                          TIME: 09:44:2
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

DIAGNOSE ICFCATALOG INDATASET(SYS1.UCATB1)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

2. Analyze and fix any errors encountered before proceeding to next steps. Example 3-20 shows a sample error that was detected during **DIAGNOSE** processing.

Example 3-20 Sample error detected by diagnose

```
IDC21364I ERROR DETECTED BY DIAGNOSE:
  ICFCAT ENTRY: SYS1.MCATA (D)
  RECORD: 00000000000000000000000000000000000000000000 /00
  OFFSET: X'004A'
  REASON: 11 - UNCOMPLETED DELETE DETECTED
IDC21365I ICFCAT RECORD DISPLAY:
  RECORD: 00000000000000000000000000000000000000000000 /00
000000  00DB0034 C3004800 2D000000 00000000   00000000 00000000 00000000 000000
000020  00000000 00000000 00000000 00000000   00000000 00000014 01FFFFFF FFFFFF
000040  FF481432 1F00000F 01000012 C4004480   0BE2E8E2 F14BD4C3 C1E3C100 001401
000060  FFFFFFFF FFFFFF08 14321F00 000F0100   001E0400 E5D6D3D4 C3F13010 200F82
000080  00003000 00000000 00000000 0000001B   C9004D00 14E2E8E2 F14BD4C3 C1E3C1
0000A0  C3C1E3C9 D5C4C5E7 00001401 FFFFFFFF   FFFFFFFF 0814321F 00000F01 00001E
0000C0  00E5D6D3 D4C3F130 10200F82 00000020   00000000 00000000 000000
IDC21363I THE FOLLOWING ENTRIES HAD ERRORS:
  SYS1.MCATA (D) - REASON CODE: 11
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
```

3. Create a **LISTCAT** copy of your catalogs to compare the data prior to and after the merge, as shown in Example 3-21.

Example 3-21 Sample LISTCAT output

```
IDCAMS  SYSTEM SERVICES                                          TIME:
09:50:46
                          LISTING FROM CATALOG -- SYS1.UCATC1
         THE NUMBER OF ENTRIES PROCESSED WAS:
                   AIX ------------------0
                   ALIAS ----------------0
                   CLUSTER --------------2
                   DATA -----------------2
                   GDG ------------------0
                   INDEX ----------------1
                   NONVSAM -----------1000
                   PAGESPACE ------------0
                   PATH -----------------0
                   SPACE ----------------0
                   USERCATALOG ----------0
                   TAPELIBRARY ----------0
                   TAPEVOLUME -----------0
                   TOTAL -------------1005
         THE NUMBER OF PROTECTED ENTRIES SUPPRESSED WAS 0
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

4. Create backup copies of your catalogs. It is suggested to create at least two backup copies by using different techniques for extra recovery protection, as shown in Example 3-22.

Example 3-22 Sample catalog backup

```
EXPORT SYS1.MCATC     -
         OUTFILE(EXPDD) -
         TEMPORARY
IDC0005I NUMBER OF RECORDS PROCESSED WAS 12
IDC0594I PORTABLE DATA SET CREATED SUCCESSFULLY ON 11/17/19 AT 16:49:58
IDC1147I IT IS RECOMMENDED THAT DIAGNOSE AND EXAMINE BE RUN BEFORE
IDC1147I IMPORT OF CATALOG
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0


  EXPORT SYS1.MCATA     -
         OUTFILE(EXPDD) -
         TEMPORARY
IDC0005I NUMBER OF RECORDS PROCESSED WAS 12
IDC0594I PORTABLE DATA SET CREATED SUCCESSFULLY ON 11/17/19 AT 16:50:43
IDC1147I IT IS RECOMMENDED THAT DIAGNOSE AND EXAMINE BE RUN BEFORE
IDC1147I IMPORT OF CATALOG
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

5. Ensure that all other required changes are in place before performing the **REPRO MERGECAT**. These changes might include renaming data sets to avoid data set name duplication, using system variables for data set and member names, and updating system parmlibs.
6. During your change window, shut down LPAR C and perform the **REPRO MERGECAT** to move the necessary entries to the LPAR A MCAT. Check the output and correct any errors identified. Example 3-23 shows some data sets moved, and errors related to data set name conflict.

Example 3-23 REPRO MERGECAT

```
IDC0639I SPHERE CONVERSION STARTED FOR SYS1.TEST.V180
IDC01402I SPHERE CONVERSION COMPLETED FOR SYS1.TEST.V180
IDC0639I SPHERE CONVERSION STARTED FOR SYS1.UCATC2
IDC01402I SPHERE CONVERSION COMPLETED FOR SYS1.UCATC2
IDC0639I SPHERE CONVERSION STARTED FOR SYS1.VVDS.VVOLMC3
IDC21409I CONVERSION FAILED FOR SPHERE SYS1.TEST.V97
IDC0639I SPHERE CONVERSION STARTED FOR SYS1.TEST.V98
IDC3013I DUPLICATE DATA SET NAME
IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS
IGGOCLEH-38
```

Tip: If you are running earlier than z/OS version 2.1, you should also **IMPORT CONNECT** the UCATs, and define the ALIASes on the correct MCAT. For more information about the **IMPORT CONNECT** and **DEFINE ALIAS** commands, see 3.4, “Scenario 3: Multiple MCATs and UCATs” on page 53.

7. After starting the systems, run a new catalog integrity check and analyze any possible errors, as shown in Example 3-24.

Example 3-24 VERIFY, EXAMINE, and DIAGNOSE after MERGECAT

```
VERIFY  DATASET(SYS1.UCATC2)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
  EXAMINE NAME(SYS1.UCATC2) -
00169001
    INDEXTEST NODATATEST ERRORLIMIT(1000)
00169101
IDC01700I INDEXTEST BEGINS
IDC11773I              47 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY
LENGTH:
IDC11774I CURRENT INDEX CISIZE IS  3584, RECOMMENDED MINIMUM INDEX CISIZE IS
35
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
  EXAMINE NAME(SYS1.UCATC2) -
00291001
    NOINDEXTEST DATATEST ERRORLIMIT(1000)
00292001
IDC01701I DATATEST BEGINS
IDC01709I DATATEST COMPLETE - NO ERRORS DETECTED
IDC01708I 47 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 1004 RECORDS
IDC01711I DATA COMPONENT CONTAINS 0 DELETED CONTROL INTERVALS
IDC01712I MAXIMUM LENGTH DATA RECORD CONTAINS 221 BYTES
IDC01722I 99 PERCENT FREE SPACE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
  DIAGNOSE ICFCATALOG INDATASET(SYS1.UCATC2)
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

8. Create a backup of your catalog environment and keep it for an extended period. Also, update any documentation, monitoring, reporting, and automations that are running on your systems.

4

Maintaining ICF catalogs

This chapter describes maintaining integrated catalog facility (ICF) catalogs and includes the following topics:

4.1 Maintenance

This section describes various aspects of ICF maintenance.

4.2 Backing up ICF catalogs

The backup of ICF catalogs is essential to recovering an ICF catalog environment from a failure. An ICF catalog backup is at least a recovery point that you can go back to rather than having to re-create all catalog pointers manually. This manual task can be unrealistic, especially if you have hundreds of thousands of data sets on thousands of disk volumes.

A forward recovery procedure should also be prepared in addition to having your backup, based on ICFRU or Tivoli Advanced Catalog Management product, for example, as described later in this chapter.

Daily backup is advised unless you have ICF catalogs with extremely low activity.

Backup of your ICF catalogs might depend on more criteria, such as the following items:

- Criticality of the individual ICF catalog
- Frequency of updates to the ICF catalog
- Size (you do not want to lose a big catalog with many data set pointers)
- Customer service level agreement (SLA) to meet recovery time objective (RTO) and recovery point objective (RPO)

Back up once a day is typical, but also consider if you should back up more often, or if you should also differentiate your backup based on criticality or other criteria. If you choose to differentiate, your recovery procedure will of course be more complex, because you must consider which ICF catalog is in scope, and an individual backup cycle for each catalog.

Consider taking more than one backup and maybe use different tools, such as the IDCAMS **EXPORT** command, DFSMS Data Set Services (DFSMSdss), or DFSMS Hierarchical Storage Manager (DFSMShsm). This can increase your recovery potential if you have a severe ICF catalog issue.

Check your procedures for implementing ICF catalog backup to ensure that you have all of the ICF catalogs backed up. Whenever you create an ICF catalog, a backup should be scheduled. In addition, cross-reference the ICF catalogs and the corresponding backup to ensure that they are always in sync.

Before z/OS V2R2, when a catalog was to be restored from a DFSMSdss backup it had to be restored to the same VOLSER from which it was backed up. With z/OS V2R2, a catalog backup can be restored to any volume. To perform this restoration, the device capacity must match and a logical restore must be performed. The restriction in past releases for a physical data set restore still exists.

4.3 Reorganizing ICF catalogs

The reorganization of ICF catalogs is disruptive, and can mean outage for business-critical applications while the reorganization goes on. Often, this means that this activity must happen in maintenance windows where the application can be closed down, but because

many applications today run continuously, this is becoming more and more difficult. Reorganization of ICF catalogs should be minimized due to its disruptive nature, and also due to the risk you always have when working with ICF catalogs.

As a guideline, only reorganize when your ICF catalog has reached a high number of extents. A limit could be 80 extents, for example, leaving you time to schedule a reorganization in time before the maximum limit of 123 extents is reached. Before the reorganization, also consider extending the primary and secondary allocation of the ICF catalog to prevent future requirements for reorganization.

Do not do reorganization to prevent control interval (CI) and control area (CA) splits. Let these happen, because they will reoccur anyway after you have done the reorganization. Using **CA Reclaim** can reduce the need for reorganization, and should also be implemented on your ICF catalogs.

If one high-level qualifier thrashes one of your ICF catalogs due to high activity, consider moving that qualifier to a dedicated ICF catalog. Moving the qualifier isolates this problem and avoids affecting other high-level qualifiers and applications due to reorganization.

4.4 REPRO MERGECAT as a maintenance tool

The IDCAMS **REPRO MERGECAT** function is primarily a migration tool that is used for moving data to a different ICF catalog at one of the following levels:

- Full ICF catalog level
- Data set level
- Alias level

From a maintenance perspective, the **REPRO MERGECAT** function can also be used to balance ICF catalogs across the entire ICF catalog environment. This is appropriate if the input/output (I/O) rate on one ICF catalog grows too high. You might also do this if the number of entries in the ICF catalog brings it to a size where having that many entries for one application in one user catalog increases your risk.

Another reason for moving selected catalog entries to a different ICF catalog also can be a high-activity application that increases the number of ICF catalog's extents, which causes the frequent need for reorganization. You can then off load the ICF catalog, perhaps by using a multilevel alias on your target ICF catalog. Also, move only one critical application to its own ICF catalog. Example 4-1 shows how multilevel alias `aaaa.bbbb` is moved to a different ICF catalog.

Example 4-1 Example of moving data sets for a multilevel alias to a different ICF catalog

```
REPRO -
INDATASET(your source catalog ) -
OUTDATASET(your target catalog -
LEVEL(aaaa.bbbb ) -
MERGECAT
/*
```

If this alias is a new multilevel alias, you must create this alias by using an IDCAMS **DEFINE ALIAS** statement. Relate this information to the new ICF catalog to access data there.

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

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

Changes that were introduced in z/OS V2R2 enhance IDCAMS to provide a new optional keyword, **MESSAGELEVEL(*ALL* | *SHORT*)**, for **REPRO MERGECAT**. The **SHORT** option reduces the number of lines for each entry. The **ALL** option is default and is the same as the current output, as shown in Example 4-2.

Example 4-2 New REPRO MERGECAT keyword, MESSAGELEVEL

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

When **MESSAGELEVEL(SHORT)** is used, **IDC0639I** and **IDC01402I** messages are replaced by the new **IDC01401I** message, as seen in Example 4-3.

Example 4-3 REPRO Output reduced by using MESSAGELEVEL(SHORT)

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

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

4.5 Other types of ICF catalog maintenance

You might need to work with your ICF catalogs for other reasons than already described in this chapter. Catalogs can grow in size over the years, and might need to be extended. Alternatively, features change, as for **IMBED** and **REPLICATE**, where new disk technology makes parameters obsolete. It is advised to remove the **IMBED** and **REPLICATE** parameters for performance reasons. Do this by backing up the ICF catalog and import it again. When you redefine the ICF catalog, **IMBED** and **REPLICATE** are removed.

Many changes to the ICF catalog are dynamic, either by command or using IDCAMS **ALTER** statements. An example of this is CA Reclaim, which can be implemented using an IDCAMS **ALTER** command.

4.6 Recovering ICF Catalogs

ICF catalog recovery procedures should be well-documented and prepared. After the procedures have been outlined, you should build the jobs and prepare the commands needed to perform the recovery. The final step is to test the recovery procedure, and to do so on a regular schedule.

The intent of the following topics is to describe the current features that can be used in catalog recovery, to make the best possible and least disruptive recovery.

Suspending catalog access while performing maintenance and recovery

A significant enhancement that was added in DFSMS V2.1 for ICF catalogs is the **MODIFY RECOVER** command to support ICF catalog recovery procedures. This command works for both Virtual Storage Access Method (VSAM) record-level sharing (RLS)-managed and non-RLS-managed catalogs. It can be used for locking and unlocking the ICF catalog while maintenance or recovery goes on, to ensure integrity.

The **RECOVER LOCK** command is equivalent to the IDCAMS **LOCK** function, but also has a corresponding **SUSPEND** option, both of which are explained in this chapter. See the **RECOVER LOCK** command issued in Example 4-4 on page 69.

Example 4-4 Example of locking an ICF catalog using the RECOVER command

```
F CATALOG,RECOVER,LOCK(ucat name)
```

This command locks and closes the ICF catalog sysplex-wide. The catalog appears as **LOCKED** in a catalog display. Trying to access the catalog, you get return code 186, with the message UNABLE TO ACCESS LOCKED CATALOG. A **LOCK** is generally used when the catalog is being reorganized or recovered. **LOCK** fails any user request while the **LOCK** is active.

You can release the **LOCK** status again by issuing the **UNLOCK** command shown in Example 4-5.

Example 4-5 Example of unlocking an ICF catalog using RECOVER command

```
F CATALOG,RECOVER,UNLOCK(ucat name)
```

This command releases a **LOCK** set by a previous **RECOVER LOCK** command, or by an IDCAMS **ALTER LOCK** command.

Restriction: *Only* UCATs can be locked. It is not possible to lock an MCAT. Also note that if a catalog is locked during initial program load (IPL), it is not accessible until an **UNLOCK** is issued.

As an alternative to the **RECOVER LOCK** and **UNLOCK** command pair, a **SUSPEND** and **RESUME** option is also available on the **RECOVER** command. You perform a **SUSPEND** on an ICF catalog, as shown in Example 4-6.

Example 4-6 Example of suspending an ICF catalog using the RECOVER command

```
F CATALOG,RECOVER,SUSPEND(ucat name)
```

This command performs a serialized close of the ICF catalog across the sysplex. The **SUSPEND** command prevents unauthorized requests to the catalog. In a catalog display, the catalog appears as in a suspended state.

To resume catalog activity after maintenance or recovery, you can release catalog activity by issuing the **RECOVER** command with a **RESUME** option, as shown in Example 4-7.

Example 4-7 Example of resuming an ICF catalog using the RECOVER command

```
F CATALOG,RECOVER,RESUME(ucat name)
```

The user catalog name can be specified generically, or as a fully qualified catalog name.

The **SUSPEND** and **RESUME** option might be preferable to the **LOCK** and **UNLOCK** option, because catalog requests during maintenance and recovery are *queued and restarted* using the **SUSPEND** and **RESUME** option, but they are *failed* using the **LOCK** and **UNLOCK** option.

Also, Example 4-8 shows some of the messages returned when you **LOCK** an ICF catalog using **RECOVER**.

Example 4-8 Messages returned when performing a RECOVER LOCK command

```
F CATALOG,RECOVER,LOCK(UCAT.RLSTST)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I MODIFY CATALOG UCAT.RLSTST TO STATE LOCK SUCCESSFUL
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Example 4-9 shows messages returned from a **RECOVER UNLOCK** command.

Example 4-9 Messages returned when doing a RECOVER UNLOCK command

```
F CATALOG,RECOVER,UNLOCK(UCAT.RLSTST)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I MODIFY CATALOG UCAT.RLSTST TO STATE UNLOCK SUCCESSFUL
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Example 4-10 and Example 4-11 show a comparison of the messages that you receive when you issue corresponding **RECOVER SUSPEND** and **RECOVER RESUME** commands. The messages related to the **SUSPEND** are shown in Example 4-10.

Example 4-10 Messages received when issuing a RECOVER SUSPEND command

```
F CATALOG,RECOVER,SUSPEND(UCAT.RLSTST)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I MODIFY CATALOG UCAT.RLSTST TO STATE SUSPEND SUCCESSFUL
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

The messages from the **RECOVER RESUME** look like those shown in Example 4-11.

Example 4-11 Messages received when issuing a RECOVER RESUME command

```
F CATALOG,RECOVER,RESUME(UCAT.RLSTST)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I MODIFY CATALOG UCAT.RLSTST TO STATE RESUME SUCCESSFUL
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Important: You need to ensure appropriate access to the IGG.CATLOCK resource to be able to perform the **LOCK** and **UNLOCK**, and the **SUSPEND** and **RESUME**, commands.

It is also important to mention that the **ALTER LOCK** and **SUSPEND** command now works sysplex-wide, ensuring full integrity.

DELETE USERCATALOG and DEFINE USERCATALOG by using NODISCONNECT and RECONNECT

DFSMS V2.1 introduced new options for the **DELETE USERCATALOG** and **DEFINE USERCATALOG** statements, to keep and maintain alias information about the deletion and creation of an ICF catalog.

With **DELETE USERCATALOG NODISCONNECT** and **DEFINE USERCATALOG RECONNECT** parameters, you no longer have to maintain the ALIAS pointers yourself. **NODISCONNECT** retains alias information at the deletion of a user catalog. If you use this parameter, **RECONNECT** can be used on the **DEFINE USERCATALOG** command to have the newly defined catalog connected to the existing alias information, so that you do not have to rebuild this information manually.

RECONNECT also supports rebuilding alias information, even if the ICF catalog is defined on another volume or device type.

NODISCONNECT and RECONNECT subparameters scenario

For visualization of how the new **NODISCONNECT** and **RECONNECT** subparameters work, a **DELETE** of an ICF catalog is done by using the keyword **NODISCONNECT**. The ICF catalog is defined again with the **RECONNECT** parameter.

An example of deleting an ICF catalog by using the **NODISCONNECT** keyword is shown in Example 4-12.

Example 4-12 Example of deleting an ICF catalog using the NODISCONNECT parameter

```
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE UCAT.RLSTST USERCATALOG RECOVERY NODISCONNECT
/*
```

Next, the user catalog was defined again with the **RECONNECT** keyword to connect to the existing ALIAS pointers, as shown in Example 4-13.

Example 4-13 Example of defining an ICF catalog using the RECONNECT parameter

```
//IDCAMS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE USERCATALOG -
(NAME(UCAT.RLSTST ) ICFCATALOG -
VOLUME(MHL1A0) TRK(5 1) -
STORCLAS(SCRLS) -
DATACLAS(WELCHRLS) -
LOG(NONE) RECONNECT -
FREESPACE(0 0) -
NOIMBED NOREPLICATE) -
DATA (CISZ(4096))
/*
```

The only ALIAS in this test catalog was RLSTST, with one data set created. When we looked up the ALIAS using IBM z/OS Interactive System Productivity Facility (ISPF) after defining the UCAT, the ALIAS was visible. At this point, there are no data sets.

DFSMSdss awareness of LOCK and SUSPEND keywords

DFSMSdss also has the awareness of **LOCK** and **SUSPEND** when an ICF catalog is restored. On the **BCSRECOVER** keyword, you can specify either **LOCK** or **SUSPEND**, and DFSMSdss changes the status of the ICF catalog being restored to either of these, whichever is selected.

When the ICF catalog has been successfully recovered, DFSMSdss performs the corresponding **UNLOCK** or **RESUME** to reestablish normal ICF catalog access.

In the following scenario, we tried to restore an ICF catalog without having the required access to the IGG.CATLOCK resource. We did this so that we could demonstrate that DFSMSdss tries to do a **LOCK** during recovery. We used these DFSMSdss parameters to perform the recovery, as shown in Example 4-14.

Example 4-14 Recovering an ICF catalog using DFSMSdss with keyword BCSRECOVER(LOCK)

```
RESTORE DATASET(                    -
         INCLUDE(                   -
                UCAT.RLSTST  ))     -
         INDDNAME(TAPE)             -
         OUTDDNAME(DASD)            -
         BCSRECOVER(LOCK)
```

The **RESTORE** job was submitted and failed as expected, as shown in Example 4-15.

Example 4-15 DFSMSdss fails on BSCRECOVER(LOCK) due to missing access

```
ICH408I USER(MHLRES1 ) GROUP(SYS1 ) NAME(MHLRES1/ name )
IGG.CATLOCK CL(FACILITY)
INSUFFICIENT ACCESS AUTHORITY
ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
```

Example 4-15 shows that DFSMSdss does an implicit **LOCK** (or **SUSPEND**) of an ICF catalog before restoring the ICF catalog from a backup. To correct the error, READ access to the IGG.CATLOCK facility class was provided.

The restore was resubmitted (using the **BCSRECOVER** and **SUSPEND** keywords). The job ended successfully this time, as shown in Example 4-16.

Example 4-16 Restore of ICF catalog using DFSMSdss and BCSRECOVER keyword

```
ADR442I (001)-FRLBO(01), DATA SET UCAT.RLSTST PREALLOCATED, IN CATALOG MCAT.....
ADR360I (001)-TDVSM(02), PROCESSING SUSPENDED USER CATALOG UCAT.RLSTST
ADR963I (001)-TDLOG(02), CLUSTER UCAT.RLSTST WAS DUMPED USING RECORD LEVEL SHARI
GMT TIMESTAMP IS: 2014.275 00:04:50.8
ADR489I (001)-TDLOG(02), CLUSTER UCAT.RLSTST WAS RESTORED
CATALOG MCAT.SANDBOX.Z1C.SBOX00
COMPONENT UCAT.RLSTST
COMPONENT UCAT.RLSTST.CATINDEX
ADR372W (001)-TDNVS(03), ALIAS RLSTST NOT RESTORED FOR USER CATALOG UCAT.RLSTST,
ADR480W (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE NOT PROCESSED FROM THE LOG
RLSTST
ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
```

Before processing the restore, DFSMSdss performs a **SUSPEND** of the catalog. The catalog is then protected against concurrent updates. DFSMSdss also has the awareness that the memory dump was performed by using RLS access (implicit QUIESCE before doing the memory dump, and **RESUME** after the **DUMP** finished). DFSMSdss finds the user catalog pre-allocation and does not restore the alias because this information is in place. The **SUSPEND** issued by DFSMSdss is resumed again after a successful **RESTORE**.

Catalog recovery by using the current features

Recovery of a catalog might be changed to use the newest DFSMS V2.1 enhancements. You still need backups as your recovery checkpoint, and System Management Facilities (SMF) data, to perform forward recovery from this checkpoint to a current point in time.

Backup of ICF catalogs can be done by using the following methods:

- IDCAMS **EXPORT**
- DFSMShsm
- DFSMSdss

Using DFSMSdss as the backup tool, the RLS-managed ICF catalogs are quiesced. For the non-RLS-managed ICF catalogs, DFSMSdss enqueues on SYSIGGV2, as done in DFSMSdss before V2.1. This ensures the integrity of the backups. The following steps were used for a catalog recovery based on DFSMSdss and the features that were introduced in DFSMS V2.1:

1. In this scenario, we assume that we need to delete the current ICF catalog and allocate it on a different volume. Start by issuing the **F CATALOG,RECOVER,SUSPEND(*ucat name*)** command. This queues catalog requests against this catalog without failing them. Most importantly, the **SUSPEND** also closes the catalog across the sysplex, ensuring integrity.
2. We are now ready to delete the catalog from its current location (having verified the we have a valid backup). Delete should occur with the **NODISCONNECT** parameter to maintain alias pointers. Only data set control blocks (DSCBs) and VSAM volume records (VVRs) for the ICF catalog are being deleted.
3. Next, we redefine the catalog using the **RECONNECT** parameter (counterpart to **NODISCONNECT**). Use the **VOLSER(*vvvvvv*)**, **LOG(NONE)**, and **SUSPEND** parameters, in addition to **RLSQUIESCE** or **RLSENABLE**, for an RLS-managed ICF catalog. Using the **VOLSER** parameter updates across the sysplex, and **RECONNECT** connects existing alias information, while the **SUSPEND** parameter ensures integrity throughout the recover.
4. Restore can now happen using DFSMSdss. DFSMSdss detects the suspended status and restores the empty catalog without using additional serialization, except from the **SUSPEND**.
5. When the restore has been successfully completed, you need to do forward recovery from the time that the ICF catalog backup was taken until the current point in time. You can accomplish this by using the ICFRU tool that uses selected SMF records for the forward recovery. See the following section, “ICF catalog forward recovery using ICFRU” on page 73. ICFRU performs the forward recovery from this point in time to the current point in time.
6. After the ICFRU forward recovery of the ICF catalog has been completed, the only outstanding action is to release the **SUSPEND** of the catalog. Perform this action with the **F CATALOG,RECOVER,RESUME(*ucat name*)** command.

ICF catalog forward recovery using ICFRU

This section describes the ICFRU functionality.

Introduction to ICFRU

ICFRU is the recovery tool that can perform forward recovery of your ICF catalogs based on a point-in-time IDCAMS **EXPORT** backup and SMF records. Read this section to see how ICFRU can complement the recovery scenario described in the previous section, “Catalog recovery by using the current features” on page 72.

How ICFRU works

ICFRU needs a successful backup of your ICF catalog so that is can be recovered based on IDCAMS **EXPORT**. Note that ICFRU can only do forward recovery based on this. Additionally, to bring the ICF catalog to the current point in time, you need the ICF catalog-related SMF records in the time span from the last successful IDCAMS **EXPORT** of the ICF catalog to the current point in time, to be able to apply later changes to the ICF catalog (**DELETES**, **DEFINES**, and **RENAMES**).

ICFRU performs the forward recover based on two programs in ICFRU:

- Integrated Catalog Forward Recovery Record Selection and Validation (CRURRSV)
- Integrated Catalog Forward Recovery Record Analysis and Processing (CRURRAP)

The record selection and validation step validates the SMF records and issues alerts on time gaps that might indicate missing SMF data. You must use ICF catalog-related SMF records from all logical partitions (LPARs) having access to the ICF catalog in scope for recovery.

Before being processed in the analysis and processing step, SMF records are sorted in data set name order, and in descending date and time order.

Next, the analysis and processing step uses the IDCAMS **EXPORT** copy as input, and only looks at the most recent processing of the individual data set as seen in the SMF data. If a data set is seen on the IDCAMS **EXPORT**, but not in the SMF records, it is written to the output file. If a data set occurs in SMF as a deletion, it is not written to the output file. Of course, any creates are added to the output file.

The CRURRAP program ends up creating an `IDCAMS EXPORT` file. Even though it was not created by an IDCAMS **EXPORT** command, it can be used as input in an IDCAMS **IMPORT**, and has all of the ICF catalog updates up to the current point in time.

ICFRU processing steps

ICFRU is explained in much more detail in *z/OS V2R1.0 DFSMS Managing Catalogs, SC*23-6853. However, the processing steps give you an understanding of how ICFRU works:

1. Beneath the job control language (JCL) for the selection and validation step (the CRURRSV program), the input is the SMF records for the time period since the most recent IDCAMS **EXPORT** up to the current point in time.

   You must also provide the following information as input parameters:

   – ICF catalog name
   – Start date and time for the forward recovery
   – End date and time for the forward recovery
   – Maximum gap time in minutes in your SMF records
   – Maximum clock difference in seconds (between your LPARs)

   The JCL for the selection and validation step is shown in Example 4-17 on page 75.

Example 4-17 JCL example for the ICFRU selection and validation step

```
//*******************************************************
//* 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
```

2. Sort the SMF records extracted by the CRURRSV program in data set order, and in descending date and time order. The input for this sort is your SMF data from the time span between your most recent IDCAMS **EXPORT** backup and the current point in time, as shown in Example 4-18.

Example 4-18 Sort of SMF records before using these in ICFRU analysis and processing step

```
//*******************************************************
//* 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)
/*
```

3. Build an up-to-date IDCAMS EXPORT file for recovery of the catalog. The input for this file is your sorted SMF records and the IDCAMS **EXPORT** of your ICF catalog. You must provide the same input parameters as for the selection and validation step. Example 4-19 shows a sample of the analysis and processing step.

Example 4-19 JCL example of ICFRU analysis and processing step (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
```

Preparations for ICFRU forward recovery

ICFRU will use SMF record types 61,65,66. It is suggested that you run daily jobs to extract these record types into "ICFRU-ready" SMF files, so that you need less processing time when an ICF catalog failure occurs. Also, you should have jobs prepared that can extract from current SMF data sets. In addition to having the ICFRU jobs ready with good documentation, your procedures should also have been tested in a *controlled* test system or environment.

Time is critical when you encounter ICF catalog errors. During the extract of ICF catalog SMF data from a large volume of SMF data, struggling with building JCL and potential JCL errors can cost you valuable time and complicate the recovery.

This short introduction focuses on the ICFRU procedure. Any ICF catalog issue requires other considerations that are related to your environment and to the individual case.

New catalog search interface fields

Catalog Search Interface (CSI) is a read-only general-use programming interface that is used to obtain information about entries that are contained in catalogs or data about catalogs. The catalog entries are selected by 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” can be selected.

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

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

4.7 ICF catalog diagnostics

The following IDCAMS commands are available to check your VSAM data sets:

- `EXAMINE`

  This command analyzes and reports on the structural integrity of the index and data components of a key-sequenced data set cluster (KSDS) and of a variable-length relative record data set cluster (VRRDS). Any problems with the VSAM data set is reported by one of the IDCxxxxx messages.

- `VERIFY`

  This command causes a catalog to correctly reflect the end of a VSAM data set after an error occurs while closing a VSAM data set. The error might cause the catalog to be incorrect.

- `DIAGNOSE`

  This command can be used to scan a basic catalog structure (BCS) or a VSAM volume data set (VVDS) to validate the data structures and detect structure errors.

Diagnosing ICF catalog errors is an important discipline because ICF catalog errors that grow in extent can effect the information technology (IT) operations and the business. It is strongly advised to detect any early warnings on any errors of your ICF catalogs.

Frequent IDCAMS **`DIAGNOSE`** jobs should be scheduled because they provide warnings through nonzero return codes back to your job scheduling package. These warnings alert you to potential problems, so that you can handle them promptly before they escalate.

Use an automation tool to pick up the most possible ICF catalog error messages, such as `ICF catalog full` messages, at the earliest stage possible.

Performing ICF catalog diagnosis and recovery requires good skills and experience in ICF catalog handling. In addition, it requires a well-prepared toolbox (JCL library with prepared jobs for any kind of ICF catalog operation or recovery).

It is imperative to always understand any ICF catalog problem and root cause fully and to have a good backup of the current status before starting the recovery.

To quickly access the VVR/NVR that is associated with the catalog entry, the catalog record at BCS has a relative byte address (RBA) pointer to the VSAM volume data set (VVDS). When VVR/NVR are moved and VVR RBA is not valid, the catalog management identifies this condition and updates the catalog record. Catalog management then performs a sequential search of the VVDS to re-establish the RBA. Previous releases of DIAGNOSE identified this mismatch as an error and reports “not found” for the VVDS entry.

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

4.7.1 VERIFY command

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

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

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

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

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

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

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

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

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

IDC11728I DATA FOUND IN EMPTY CI
```

This enhancement works only when **EXAMINE** and **VERIFY RECOVER** are run in the same IDCAMS job step, as shown in Example 4-20 on page 80.

Example 4-20 EXAMINE/IDCAMS VERIFY RECOVER job example

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

Some of the new messages alerts are shown in Example 4-21.

Example 4-21 EXAMINE/IDCAMS VERIFY RECOVER output

```
EXAMINE NAME(SYSPLEX.VSMEVR03.KSDS01) DATATEST NOINDEXTEST 04870042
IDC01701I DATATEST BEGINS
IDC11728I DATA FOUND IN EMPTY CI
IDC01713I DATA CONTROL INTERVAL DISPLAY AT RBA 419 FOLLOWS
000000 C1C2C3C4 C5C6C7C8 C9D1F0F0 F0F0F2F5 F1F8D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 *ABCDEFGHIJ00002518KLMNOPQRSTUVWX*
000020 E8E98182 83848586 87888991 92939495 96979899 A2A3A4A5 A6A7A8A9 C1C2C3C4 *YZ.........................ABCD*
000040 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E98182 83848586 87888991 *EFGHIJKLMNOPQRSTUVWXYZ..........*
000060 92939495 96979899 A2A3A4A5 A6A7A8A9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 *................ABCDEFGHIJKLMNOP*
000080 D8D9E2E3 E4E5E6E7 E8E98182 83848586 87888991 92939495 96979899 A2A3A4A5 *QRSTUVWXYZ......................*
...lines removed to fit example on page...
000760 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
000780 00000000 00000000 00F70701 13F60701 12000AF5 070111F4 07011000 0AF30701 *.........7...6.....5...4.....3..*
0007A0 0FF20701 0E000BF1 F106020D F1F00602 03000A07 0004F807 0105000A F7070106 *.2.....11...10........8.....7...*
0007C0 F6070107 000AF507 0108F407 0109000A F307010A F207010B 000CF5F0 F105030C *6.....5...4.....3...2.....501...*
0007E0 06000200 0FF0F0F0 F0F2F4F8 000701F0 F0F0F0F2 F4F70007 000007F9 07F90000 *.....0000248...0000247.....9.9..*
IDC01714I ERROR LOCATED AT OFFSET 00000018
IDC21702I MINOR ERRORS FOUND BY DATATEST
IDC01708I 749 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 749 RECORDS
IDC01711I DATA COMPONENT CONTAINS 0 DELETED CONTROL INTERVALS
IDC01712I MAXIMUM LENGTH DATA RECORD CONTAINS 2000 BYTES
IDC01722I 99 PERCENT FREE SPACE
IDC11995I RECOMMENDATION: ISSUE IDCAMS EXAMINE AND VERIFY RECOVERY TO ATTEMPT TO
REPAIR THE PRECEDING ERROR
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 4
VERIFY DATASET(SYSPLEX.VSMEVR03.KSDS01) RECOVER
IDC11997I THE PRECEDING IDC11728I ERROR HAS BEEN ADDRESSED.
IDC11994I RECOMMENDATION: RERUN IDCAMS EXAMINE.
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

Consider the following points regarding Example 4-21:

- The **IDC11994I** message is shown after **IDC11996I** or **IDC11997I** to recommend users to rerun **IDACAMS EXAMINE** to verify whether the preceding error messages that were found by IDCAMS EXAMINE are now fixed.
- The IDC11995I message indicates that at least one error in the data set was found that can be addressed by issuing **IDCAMS EXAMINE** and **IDCAMS VERIFY RECOVER** in the same job step.
- The IDC11997I message indicates that attempt an attempt was made to repair the preceding IDC117xxl errors.
- The IDC11998I message indicates that **IDCAMS VERIFY RECOVER** does not attempt to repair the preceding messages because of concurrent access on the data set.
- The **IDC11999I** message indicates that **IDCAMS VERIFY RECOVER** attempted to repair the preceding IDC117xxI message but failed.

4.7.2 PARM(TEST) command

The **PARM(TEST)** command is used to display diagnostic tracing. Debugging is enhanced at z/OS V2R2 to include 31-bit support and allocate memory and process work above the line. The use of above the line memory is a constraint relief and reduces the chances of abends that are caused by lack of memory.

4.7.3 PRINT and REPRO commands

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

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

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

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

Example 4-22 shows how this new keyword is used.

Example 4-22 CIMODE example

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

4.7.4 Modify CATALOG commands

Catalog address space (CAS) can be operated for display and actions. In the following table, we grouped the commands into types of actions to outline which commands could be useful for you in these scenarios. We split the commands into the following groups:

- Operator commands
- Information about CAS and CAS activity
- Fixing temporary catalog problems
- Temporary modifications to ICF catalogs
- Catalog recovery
- RLS-related catalog commands
- Catalog parmlib commands
- Enhanced catalog sharing commands

The commands related to these main actions are listed according to this grouping. Due to this grouping, some commands might be repeated, because they can appear in more than one context.

The commands that you might use for a certain task are listed in Table 4-1.

Table 4-1 ICF catalog MODIFY commands grouped by type

Type of command	Command	Description
Operator command	F CATALOG,CLOSE	Close an ICF catalog.
	F CATALOG,SYS%	Apply maintenance from one system.
	F CATALOG,UNALLOCATE	Deallocate a catalog from CAS.
	F CATALOG,VCLOSE	Close a VSAM volume data set (VVDS).
	F CATALOG,VUNALLOCATE	Deallocate a VVDS from CAS.
Information about CAS and CAS activity	F CATALOG,ALLOCATED	Display allocated ICF catalogs.
	F CATALOG,CONTENTION	Display the catalog contention setting.
	F CATALOG,ENTRY	Display catalog entry points.
	F CATALOG,LIST	Lists currently active CAS service tasks, their related job names, elapsed times, and unique identifications.
	F CATALOG,LISTJ(jobname),DETAIL	List a job in queue in a catalog request.
	F CATALOG,ECSHR(STATUS)	Display Enhanced Catalog Sharing (ECS).
	F CATALOG,REPORT	Create a catalog report.
	F CATALOG,REPORT,PERFORMANCE	Create a performance report.
	MODIFY,CATALOG,REPORT,CATSTATS	Create a statistics report.
	MODIFY,CATALOG,REPORT,CATSTATX	Create a statistics report (extended).
	F CATALOG,TAKEDUMP	Have a catalog take an SVC memory dump.
Fixing temporary catalog problems	F CATALOG,ABEND	Terminate a catalog task.
	F CATALOG,CLOSE	Close a catalog.
	F CATALOG,END	End a catalog task.
	F CATALOG,RESTART	Restart CAS.
	F CATALOG,UNALLOCATE	Deallocate a catalog.
	F CATALOG,VCLOSE	Close a VVDS.
	F CATALOG,VUNALLOCATE	Deallocate a VVDS.

Type of command	Command	Description
Temporary modifications to ICF catalog	F CATALOG,ALIASLEVEL	Change an aliaslevel.
	F CATALOG,ALLOCATE	Open a catalog.
	F CATALOG,CATMAX	Change the maximum number of allowed catalogs.
	F CATALOG,CLOSE	Close a catalog.
	F CATALOG,{ISC|NOISC}	Set the usage of In Storage Cache (ISC).
	F CATALOG,{SYS%ON|SYS%OFF	Apply maintenance from one system.
	F CATALOG,TASKMAX(nnn)	Change the maximum allowed tasks.
	F CATALOG,UNALLOCATE	Deallocate a catalog.
	F CATALOG,{VLF|NOVLF}	Add or remove a catalog to or from the virtual lookaside facility (VLF).
	F CATALOG,ENABLE(DELFORCEWNG)	Enables the display of message IDC1997I and IDC1998I when a **DELETE VVDS RECOVERY** or **DELETE USERCATALOG FORCE** is performed.
	F CATALOG,DISABLE(DELFORCEWNG)	Disables the warning message IDC1997I or IDC1998I when attempting to use the **DELETE VVDS RECOVERY** or **DELETE USERCATALOG FORCE** command.
	F CATALOG,DISABLE(DSNCHECK)	Perform a syntax check on added names.
Catalog recovery	F CATALOG,RECOVER,LOCK(ucat name)	Lock a catalog for recovery.
	F CATALOG,RECOVER,UNLOCK(ucat name)	Unlock a command after recovery.
	F CATALOG,RECOVER,SUSPEND(ucat name)	Suspend a catalog for recovery.
	F CATALOG,RECOVER,RESUME(ucat name)	Resume a catalog after a suspend.
RLS-related catalog commands	D SMS,CFLS	Display the coupling facility (CF) lock structure.
	D SMS,CFLF(IGWLOCK00)	Display the IGWLOCK00 structure.
	F CATALOG,RLSENABLE(ucatname)	Enable the ICF catalog for RLS usage.
	F CATALOG,RLSQUIESCE(ucatname)	Quiesce an RLS-enabled ICF catalog.
	F CATALOG,RLSQUIESCE,SYSTEM	Quiesce a whole system from RLS.
	F CATALOG,RLSENABLE,SYSTEM	Enable a whole system for RLS.

Type of command	Command	Description
Catalog parmlib command	F CATALOG,ALIASLEVEL(n)	Display the number of alias levels.
	F CATALOG,CONTENTION,(ALLOCLOCK,nn,n,a,)	Display or modify the contention setting on ALLOCLCK.
	F CATALOG,ECSHR(AUTOADD)	Activate ECS on ECS-eligible catalogs.
	F CATALOG,CATMAX(1024)	Display the maximum number of open catalogs.
	F CATALOG,DISABLE(DELFORCEWNG),ENABLE	Turn warning message IDC1997I on or off.
	F CATALOG,DELRECOVWNG(ENABLE | DISABLE)	Turn warning message IDC1999I on or off.
	F CATALOG,DSNCHECK(ENABLE | DISABLE)	Perform a syntax check on added names.
	F CATALOG,DUMPON	Set dynamic DUMP on CAS.
	F CATALOG,EXTENDALIAS(ENABLE | DISABLE)	Enable or disable Extended Alias.
	F CATALOG,NOTIFYEXENT(nn)	Notify if the ICFCAT maximum number of allowed extents was exceeded.
	F CATALOG,SYMREC(ENABLE | DISABLE)	Enable SYMREC creation.
	F CATALOG,CONTENTION,(SYSIGGV2,nn,a)	Display or modify the SYSIGGV2 contention setting.
Catalog parmlib command (continued)	F CATALOG,CONTENTION(SYSZTIOT,nn,a)	Display or modify the SYSZTIOT contention setting.
	F CATALOG,CONTENTION(SYSZVVDS,nn,a)	Display or modify the SYSZVVDS contention setting.
	F CATALOG,TASKMAX(nnn)	Set the maximum number of user service tasks in CAS.
	F CATALOG,TASKMIN(nnn)	Set the minimum number of user service tasks in CAS.
	F CATALOG,UPDTFAIL(ENABLE | DISABLE)	Activate the IEC390I catalog update error.
	F CATALOG,VVDSSPACE,(prim,sec)	Change the VVDS allocation default.
	F CATALOG,VVRCHECK(ENABLE | DISABLE)	Enable or disable a VVR check.
Enhanced Catalog Sharing commands	F CATALOG,ECSHR(DISCONNECT)	Disconnect ECS from a structure.
	F CATALOG,ECSHR(CONNECT)	Connect ECS to a structure.
	F CATALOG,ECSHR,STATUS(ucatname)	Display ECS status.
	F CATALOG,ECSHR,REMOVE(catname)	Remove a catalog from ECS.

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

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

The only requirement for activating this facility is to define a new **OPERCMDS** resource profile, as shown in Example 4-23.

Example 4-23 New OPERCMDS resource profile

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

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

4.7.5 TSO LISTCAT command

The **LISTCAT** command is used to list catalog entries. A new parameter **PREFIX/NOPREFIX** was introduced to **LISTCAT** at z/OS V2R2 that allows the user to add the TSO user ID when **LISTCAT** is run, as shown in Example 4-24.

Example 4-24 New LISTCAT parameter

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

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

5

Record-level sharing support for ICF catalogs

This chapter describes the Virtual Storage Access Method (VSAM) record-level sharing (RLS) for integrated catalog facility (ICF) catalogs, and consists of the following areas:

- RLS for catalog overview, including requirements, support for catalogs, possible states, enhanced features, and new parameters.
- Considerations for implementing RLS, including preparing to migrate, sizing, migrating, fallback, and maintenance.
- Basic tests results, including benchmarks and documentation.

RLS access for VSAM is widely used today for IBM z/OS Data Facility Storage Management Subsystem (DFSMS) Transactional VSAM Services (DFSMStvs) and non-transactional applications. RLS provides multi systems sharing at a logical record level across a sysplex by using the coupling facility (CF). VSAM RLS uses a CF-based Lock Manager and CF-based Cache Manager in the implementation of record-level sharing. The intelligence of VSAM RLS is in an address space that is named SMSVSAM. The function requires data sets that use RLS control to be SMS-managed.

RLS is one of many buffering modes to VSAM and is set at OPEN time. The following types of VSAM buffering modes are available:

- Non shared resources (NSR)
- Local shared resources (LSR)
- Global shared resources (GSR)

These buffering modes can be referred to as *non-RLS access* to distinguish RLS from other modes.

This chapter includes the following topics:

5.1 RLS for shared catalogs overview

DFSMS V2.R1 introduces Record Level Share (RLS) usage for shared ICF catalogs. The purpose of the enhancement is to improve performance and availability on ICF catalogs.

Consider the following points:

- ICF catalogs must be shared.
- ICF catalogs are VSAM data sets.
- VSAM features an effective way of implementing data sharing through the use of the Coupling Facility, which is named RLS.

Therefore, ICF catalogs are RLS.

The use of RLS for ICF shared catalogs provides the following performance advantages:

- Locking at a logical record level, improving performance by not having to serialize on the SYSIGGV2 bcs`name`; that is, a better granularity.
- No need to keep a self-describing VVR round robin register (in the catalog or in the coupling facility) to keep the ICF catalog read integrity by having in the buffer pools always with the most current contents.
- Moving to 64-bit VSAM buffering (above the bar), when the use of RLS reduces the input/output (I/O) activity.

When the ICF shared catalog uses RLS buffering mode, a SMSVSAM task holds SYSIGGV2 bcsname SHARE while a catalog is opened for RLS access. This configuration ensures catalog data integrity from programs relying on SYSIGGV2 to serialize the catalog.

RLS is also beneficial to non-shared catalogs and shared catalogs. This benefit is realized because concurrent catalog requests might run on the same system and encounter update contention in the same fashion as when sharing a catalog cross system.

The improvements that are described here are also expected to solve performance bottlenecks and reduce the need for customers to split ICF catalogs to meet availability requirements.

5.1.1 RLS requirements

RLS use for ICF shared catalogs requires the base RLS setup, which includes an IBM Parallel Sysplex. In addition, the SMSVSAM address space must be active. As always required with VSAM RLS data sets, you must associate the data set to a SMS storage class that is assigned with a Cacheset and optional a Lockset identifying cache and lock structures in the CF.

The SMSVSAM started task at the SMSVSAM address space must be up and running. The SMSVSAM address space automatically starts at initial program load (IPL) if you specify the `RLSINIT (YES)` keyword in the IGDSMSxx member of `SYS1.PARMLIB`. If SMSVSAM is not started, you can start the address space after IPL by issuing the **`V SMS,SMSVSAM,ACTIVE`** command from the operator console.

To define the buffering mode as RLS, the following requirements must be met:

- Use the DD job control language (JCL) **`RLS`** parameter, or by specifying **`MACRF=RLS`** in the access method control block (ACB).

- Have their attributes changed as related to *unrecoverable* and *recoverable* options. Alter at DEFINE CLUSTER IDCAMS, the attribute to `LOG(NONE)` if unrecoverable, or to `LOG(UNDO)` or `LOG(ALL)` if data set is recoverable.
- Add a DFSMS storage class for the created for RLS data set to have appropriate cache set and CF cache assigned.

Sharing control data set

You must define the sharing control data sets (SHCDSs), which are key for RLS recovery because they hold vital information for it. For example, SMSVSAM restart depends on the SHCDSs being available.

The following information is part of the contents of the SHCDSs:

- The name of the CF lock structure in use
- A list of subsystems (transaction managers) and their status
- A list of open data sets using the CF
- A list of data sets with unbound locks
- A list of data sets in permit non-RLS state

SHCDS data sets are VSAM linear data sets. Define and activate two of these data sets for normal use, and one for spare purposes.

For information about how to set up the initial RLS environment, see *z/OS DFSMSdfp Storage Administration*, SC23-6860.

5.1.2 Support for ICF catalogs using RLS

New and existing parameters are needed to define new, or to alter existing, ICF catalogs to support RLS access mode. The `LOG` attribute must be defined. However, in this implementation, only unrecoverable catalogs are supported, so the attribute is `LOG(NONE)`. You also need parameters to set the type of initial access (non-RLS or RLS) at the first access of the catalog. Catalogs can be defined or altered in one of four RLS states:

- Eligible
- RLS quiesced
- RLS enabled
- RLS mode

For more information about these states, see 5.1.3, “ICF catalog possible RLS states”.

New operator commands and define parameters have also been provided to support the transition of ICF catalogs between RLS and non-RLS access. These commands work sysplex wide, even when submitted from only one logical partition (LPAR), and have the catalogs perform a serialized close across the sysplex. The status ends up being `RLS enabled` or `RLS quiesced`, ready for being accessed in `RLS mode`.

Catalog requests issued during transition do not fail. However, they are suspended while the **`QUIESCE`** or **`ENABLE`** process happens. Regard these commands as *migration* or *fallback* tools. These commands are not intended for daily usage. When an ICF catalog has been migrated to RLS mode, it should remain in RLS mode. The only exception is if you encounter an issue that forces you to revert to non-RLS mode.

5.1.3 ICF catalog possible RLS states

Note: Do not to enable catalogs for RLS that are accessed early in the IPL before the SMSVSAM and Catalog address spaces initializes.

In an environment that uses RLS, the ICF catalogs can be in one of the following states:

- Eligible

  The catalog is defined or altered to have the `LOG` attribute. This definition is mandatory to access the catalog in RLS mode (as for all other VSAM RLS data sets). The **`OPEN`** in this state still occurs as non-RLS.

- RLS quiesced

  The `RLS quiesced` indicator is set to `YES` in the catalog's VSAM volume record (VVR) entry. **`OPEN`** is only possible for non-RLS access.

  Important: The ability to move to RLS-accessed ICF catalogs from non-RLS-accessed catalogs is supported for user catalogs (UCATs) only. It is not supported for master catalog (MCAT).

- RLS enabled

  RLS enable indicator is set to `YES` in the catalog's VVR entry. As RLS enabled, the catalog can be opened in RLS mode only.

- RLS mode

  A catalog in RLS mode can be defined as a catalog that is open in RLS mode, or one that was last closed in RLS mode. The `RLS-IN-USE` indicator can also be set in the catalog's VVR.

The catalog can be taken out of RLS mode by command. For pre-DFSMS V2. R1 systems, this process requires that SMSVSAM be up and running, or that you use the IDCAMS **`SHCDS CFRESETDS(`*`user_cat_name`*`)`** command to reset the `RLS-IN-USE` indicator. In this case, it requires that all instances of SMSVSAM be shut down.

5.1.4 Comparison with other enhanced catalog features

RLS for catalog can coexist with other advanced catalog features. In this section, we compare the following enhanced catalog features with the RLS for catalog feature:

- Enhanced Catalog Sharing

  Enhanced Catalog Sharing (ECS) places the ICF catalogs self-describing VVR in the CF (SYSIGGCAC_ECS) to reduce I/O. ECS is a way of sharing an ICF catalog without the use of the coupling facility (RLS).

- Catalog Data Space Cache

  Catalog Data Space Cache (CDSC) uses virtual lookaside facility (VLF), and is intended for shared catalogs. Virtual storage is possible within 1 MB - 2 GB, divided into 4 KB blocks. Cached records are stored in a data space. This caching is designed for UCATs.

- In Storage Cache

  In Storage Cache (ISC) has a maximum of 3041 basic catalog structure (BCS) records. Storage is obtained in the catalog address space (CAS). It is intended for catalogs that are not changed frequently.

Comparing RLS with ECS

The enhanced catalog features can be replaced by RLS. For the MCAT, it still makes sense to use ECS because RLS mode is not supported for MCAT.

Catalogs that use RLS do not need to access the VVR in cache for ensuring its currency. For catalogs in RLS mode, the catalog records are placed in RLS local buffer pools or CF cache structures.

The **STORAGECLASS** cacheset defines which cache structure to use. The **DATACLAS** controls the buffer pool (64 bit) and caching options through the **RLSOVETHEBAR** and **RLSCFCACHE** settings. Locking happens at a record level to serialize access to individual catalog records rather than enqueuing on the SYSIGGV2 `bcsname` resource (entire catalog).

For other tuning parameters, such as `STRNO`, `BUFND`, and `BUFNI` on the **DEFINE USERCATALOG** command, these settings are ignored in an RLS environment. RLS obtains buffers dynamically as they are needed.

5.1.5 New parameters on the DEFINE USERCATALOG command

To support the transition to RLS, new parameters were added to the **DEFINE USERCATALOG** command. The parameters define the availability level for readiness to use RLS mode on the ICF catalogs.

Table 5-1 on page 92 is an overview of the parameters that were added in DFSMS V2.R1. For **SUSPEND** and **RESUME**, the catalog is in the `SUSPEND` state or the `RESUME` state. The **SUSPEND** and **RESUME** parameters can also be used to non-RLS catalogs.

This issue is similar for **RLSQUIESCE** and **RLSENABLE**, the two parameters that are directly related to RLS catalogs. The catalog status is `QUIESCED` or `ENABLED`.

Table 5-1 RLS added parameters

New DEFINE user catalog parameters	Description	Default
SUSPEND	Requests are suspended until a **RESUME** has been issued. The **SUSPEND** parameter was mutually exclusive with the **LOCK** parameter but since OA45623 this is no longer the case. See the OA45623 description for further information.	RESUME
RESUME	The request for the catalog is executed immediately. The command releases a **SUSPEND** state.	RESUME
RLSQUEISCE	The catalog is accessed in non-RLS mode following the define.	RLSQUIESCE
RLSENABLE	The catalog will be accessed in RLS mode after the allocation.	RLSQUIESCE
RECONNECT	Request the new catalog to use existing ALIAS information.	N/A
LOG(NONE)	Catalog is eligible for access in RLS mode.	Value appears as NULL

If the catalog is in **LOCKED** or **SUSPENDED** mode, an authorized user with READ access to the IBM Resource Access Control Facility (IBM RACF) STGADMIN profile can still access and repair a locked catalog. At the same time, other operations against the catalog are failed or queued.

If the IGG.CATLOCK profile is not defined, or if the RACF FACILITY class is not active, you cannot **LOCK**, **UNLOCK**, **SUSPEND**, or **RESUME** a catalog.

Example 5-1 shows how to define a new ICF catalog, using the new **RLSENABLE** parameter.

Example 5-1 Example of defining an ICF catalog using RECONNECT and RLSENABLE

```
DEFINE USERCATALOG -
(NAME(UCAT.RLSTST.BNC) ICFCATALOG -
VOLUME(MHL1A0) TRK(50 50) -
STORCLAS(SCRLS) -
DATACLAS(WELCHRLS) -
LOG(NONE) RECONNECT RLSENABLE -
FREESPACE(20 20) -
NOIMBED NOREPLICATE) -
DATA (CISZ(32768))
```

LOG(NONE) is used to prepare for RLS, and **RLSENABLE** enables **OPEN** in RLS mode.

The following new parameters support RLS:

- RLSQUIESCE: The catalog is accessed in non-RLS mode following the define.
- RLSENABLE: The catalog is accessed in RLS mode after the allocation.

5.2 Considerations for implementing RLS ICF catalogs

The following sections describe RLS ICF catalogs implementation considerations, including preparation, sizing, migrating, fallback, and maintenance.

5.2.1 Preparing to migrate to an RLS enabled ICF Catalog

When you have the base RLS environment in place, you can start migrating your first ICF catalog to use RLS mode. RLS includes the following basic requirements:

- SMSVSAM must be active.
- An RLS cache structure must be defined in the CF.
- An RLS Cache Set must be defined.
- An RLS Lock Set Name can be defined.
- The catalogs in scope must be SMS-managed.

Important: It is suggested that all systems in the sysplex be migrated to DFSMS V2.R1 before starting the migration. Shared catalogs cannot be accessed in RLS mode if they are opened on a pre-DFSMS V2.R1 system.

5.2.2 Sizing the RLS catalog environment for RLS buffers and CF cache

The following scenarios are presented for sizing the RLS buffers and CF cache (choose whichever is more appropriate to your environment):

- Scenario 1: Calculating the sizes and then allocating the memory.
- Scenario 2: In a memory constrained environment using the available memory with which you can work.

Scenario 1 for buffers

In this first scenario, we describe the use of a memory above the bar. Complete the following steps:

1. Place the catalogs in the RLS 64bit pool by specifying the **DATACLAS** option **RLSAboveTheBar(YES)**. Because RLS allows for 64 bit buffering, its possible that all, or most, of the catalog data can be stored in memory.
2. To determine the amount of catalog data, issue the following command:

   ```
   LISTCAT ENT('catalogname') CATALOG('catalogname') ALLOCATION
   ```

   for each eligible catalog and tally the data and index components that used RBAs.

   Example 5-2 shows a batch job to issue the command. You can repeat the LISTCAT command for each catalog you want to include.

   Example 5-2 JCL to perform the LISTCAT command

```
//KWRESO1A JOB ,MSGCLASS=H,CLASS=A,NOTIFY=KWRESO1
//LISTC    EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD *
  LISTC ENT(UCAT.BH6CAT) CATALOG(UCAT.BH6CAT) ALLOCATION
/*
//
```

Example 5-3 shows an extract from the report that was generated by the IDCAMS command to show you where to find the HI-U-RBA value. This field indicates how many bytes the catalog used within the allocated extent. Do not confuse this field with the HI-A-RBA field, which tells you how many bytes are allocated. You also must check the number of extents. If the catalog expanded to secondary extents, you must include the HI-U-RBA for each extent.

Example 5-3 Catalog listing showing HI-U-RBA and extent information

```
LISTC ENT(UCAT.BH6CAT) CATALOG(UCAT.BH6CAT) ALLOCATION
IDCAMS  SYSTEM SERVICES
                            LISTING FROM CATALOG -- UCAT.BH6CAT
CLUSTER

   DATA ------- UCAT.BH6CAT

     ALLOCATION
       SPACE-TYPE---------TRACK     HI-A-RBA---------5406720
       SPACE-PRI----------110       HI-U-RBA----------983040
       SPACE-SEC----------10
     VOLUME
       VOLSER-------------BH6CAT    PHYREC-SIZE---------4096
HI-A-RBA---------5406720     EXTENT-NUMBER----------1
       DEVTYPE------X'3010200F'     PHYRECS/TRK-----------12
HI-U-RBA----------983040     EXTENT-TYPE--------X'00'
       VOLFLAG------------PRIME     TRACKS/CA------------10
       EXTENTS:
       LOW-CCHH-----X'00120000'     LOW-RBA----------------0
TRACKS---------------110
       HIGH-CCHH----X'00190004'     HIGH-RBA---------5406719
   INDEX ------ UCAT.BH6CAT.CATINDEX

     ALLOCATION
       SPACE-TYPE---------TRACK     HI-A-RBA----------435200
       SPACE-PRI----------10        HI-U-RBA-----------7680
       SPACE-SEC----------1
     VOLUME
       VOLSER-------------BH6CAT    PHYREC-SIZE---------2560
HI-A-RBA----------435200     EXTENT-NUMBER----------1
       DEVTYPE------X'3010200F'     PHYRECS/TRK-----------17
HI-U-RBA-----------7680      EXTENT-TYPE--------X'00'
       VOLFLAG------------PRIME     TRACKS/CA------------1
       EXTENTS:
       LOW-CCHH-----X'00190005'     LOW-RBA----------------0
TRACKS---------------10
       HIGH-CCHH----X'0019000E'     HIGH-RBA----------435199
```

3. It is recommend that at least enough extra buffer pool space is added to accommodate the index components, and as much of the data components as is practical. What is practical is determined by the amount of CF caching available (see “Coupling Facility Structure Sizer tool” on page 95), and how much real memory is on each LPAR to back the buffer pool.

Note: Defining a RLS buffer pool that exceeds real memory results in paging the buffer pool and defeating the purpose of keeping data in memory.

Scenario 2 for buffers

To ensure sufficient RLS local buffering and CF cache size for your RLS-managed ICF catalogs, look at the buffer sizes in your current environment. Add up the buffer sizes based on the number of catalogs and the type of caching. Look into the following values:

- Current catalog data space cache (CDSC) that uses VLF. Capture the current `MAXVIRT` value (in 4 KB blocks). Sum up the total for the entire sysplex.
- Take the default for ISC buffers.

Calculate 3041 records per catalog. BCS records vary in size. An average size that is chosen can be 2000 bytes (total is then 608,200 bytes per catalog). Regard this size as a minimum and add more as needed to meet performance requirements. Again, calculate the total for the sysplex.

Consider the following sample scenario:

- You have a sysplex with four LPARs that share 50 ICF catalogs (10 ISC-managed and 40 in CDSC).
- Your COFVLFxx member specifies 16,384 for the `MAXVIRT` value (times 4 KB) for a total of 67,108,864 bytes for the VLF-managed catalogs. For four systems, this figure is 268,435,456 bytes. The 10 ICF catalogs that use ISC cache use (10 x 3041 x 2000) = 60,820,000 bytes. For four LPARs, this amount is 243,280,000 bytes. The requirement for buffering all catalogs then totals as listed in Table 5-2.

Table 5-2 Buffer requirement calculation

Component	Current usage in bytes	Current usage in MB
COFVLF	268,435,456	262,144
ISC	243,280,000	237,578
Total in bytes and MB	511,715,456	499,722

The current buffer need totals over 500 MB. A CF cache size of 500 MB is then allocated as a start and monitored as the conversion of the ICF catalogs to RLS access mode progresses.

RlsFixedPoolsize and RlsAboveTheBarMaxPoolSize parameters

To meet the requirement in the previous scenario regarding RLS local buffers, set **`RlsFixedPoolsize`** in IGDSMSxx to 500 MB. This parameter page fixes (pins) the specified amount of buffers in the RLS buffer pool. The parameter applies to all RLS data (not catalogs only).

Note: Caution must be used with this parameter because it page fixes buffers on a first come, first served basis for the requested CISIZE. Workloads with many different CISIZEs might not have access to the fixed buffers.

Coupling Facility Structure Sizer tool

Next, the size of the CF cache structures that are to be used by ICF catalog RLS must be defined. The Coupling Facility Structure Sizer Tool (CFSizer) is a web-based application that returns structure sizes that are based on the latest CFLEVEL for the IBM products that use the CF. CFSizer can be used to generate the exact settings and the JCL for the utility IXCMIAPU with the input to CFRM couple data set. This information is used during the Cache Structure creation process. The CFSizer is available at this website:

`http://www.ibm.com/systems/support/z/cfsizer`

To size a structure, complete the following steps:

1. Open the CFSizer web page. Select **VSAM RLS** from the product links on the left side navigation bar, as shown in Figure 5-1.

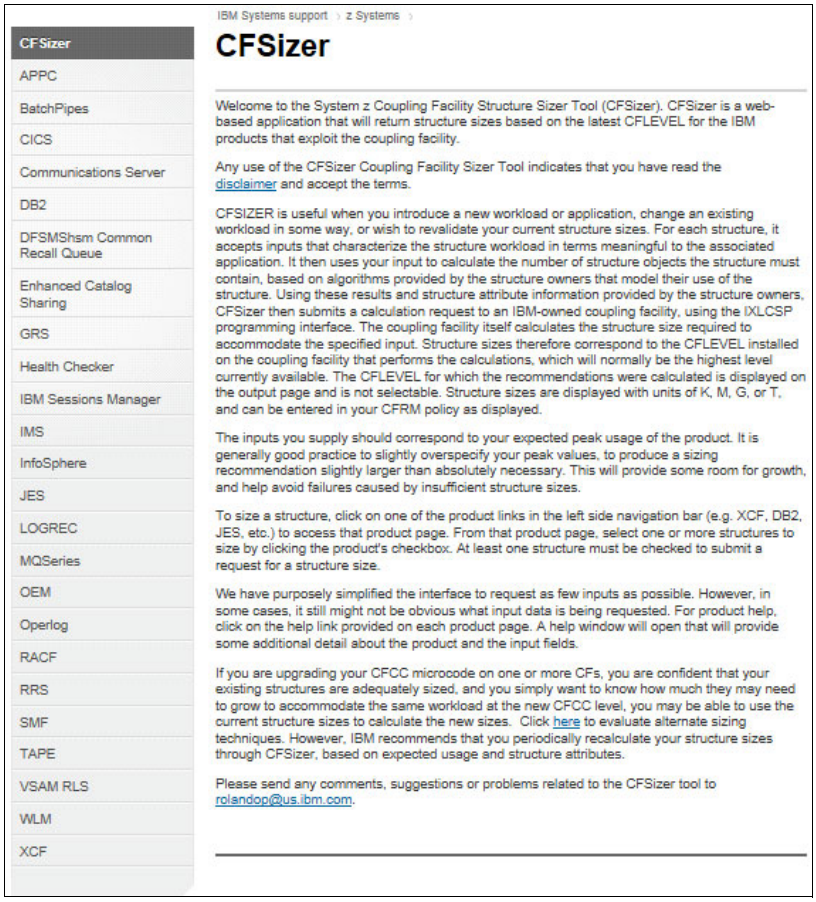
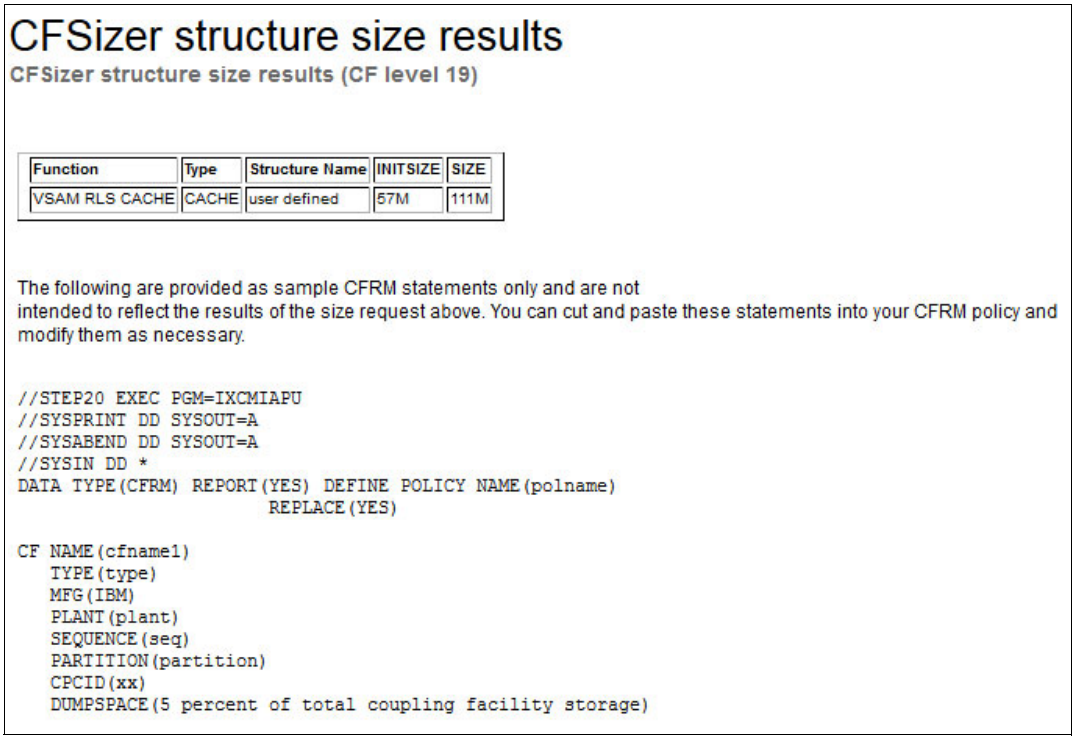


Figure 5-1 CFSizer Tool

2. In the VSAM RLS page, select one or more structures to size by selecting the cache structure that you want to size. At least one structure must be selected to submit a request for a structure size. Enter the required information and click **Submit** to continue. We selected the VSAM RLS CACHE structure and used the estimated value of 500 MB, as shown in Figure 5-2.

Figure 5-2 VSAM RLS CACHE structure sizing using CFSizer tool

3. The CFSizer presents you with sample coupling facility resource management (CFRM) statements that are based on the input that you provided. You can paste these statements into your CFRM policy and modify them as necessary. Calculated results that are based on your input are shown in Figure 5-3.



CFSizer structure size results

CFSizer structure size results (CF level 19)

Function	Type	Structure Name	INITSIZE	SIZE
VSAM RLS CACHE	CACHE	user defined	57M	111M

The following are provided as sample CFRM statements only and are not intended to reflect the results of the size request above. You can cut and paste these statements into your CFRM policy and modify them as necessary.

```
//STEP20 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSIN DD *
DATA TYPE(CFRM) REPORT(YES) DEFINE POLICY NAME(polname)
                       REPLACE(YES)

CF NAME(cfname1)
   TYPE(type)
   MFG(IBM)
   PLANT(plant)
   SEQUENCE(seq)
   PARTITION(partition)
   CPCID(xx)
   DUMPSPACE(5 percent of total coupling facility storage)
```

Figure 5-3 CF sizer results for allocating VSAM RLS cache structure for RLS

It is suggested that you create a dedicated cache structure for ICF catalogs. This structure is to ensure that if catalog buffers are flushed from the cache, it is based on the catalog activity and on other activity in this cache CF structure.

5.2.3 Migrating to RLS enabled ICF catalog

After upgrading all of the systems in the sysplex to DFSMS V2.R1, you can start using RLS mode on ICF catalogs. This process is done by defining a new RLS ICF catalog for test purposes or by altering an existing ICF catalog to have the `LOG(NONE)` attribute. The `LOG(NONE)` attribute puts the catalog in RLS quiesced mode. By default, non-RLS catalogs have the value `NULL` in the **`LOG`** parameter. An example of a **`DEFINE`** is shown in Example 5-4 on page 98 (note the `LOG(NONE)` option) for defining a completely new RLS ICF catalog.

Keywords, such as `STRNO`, `BUFND`, and `BUFNI`, were removed because they are not used in an RLS environment.

Example 5-4 DEFINE USERCATALOG example of RLS ready ICF catalog

```
DEFINE USERCATALOG -
(NAME(UCAT.RLSTST) ICFCATALOG -
VOLUME(MHL1AO) TRK(5 1) -
STORCLAS(SCRLS) -
DATACLAS(WELCHRLS) -
LOG(NONE) -
FREESPACE(20 20) -
NOIMBED NOREPLICATE) -
DATA (CISZ(4096))
```

A **LISTCAT** following immediately after the **DEFINE** shows that the user catalog has the required LOG(NONE) attribute and is VSAM QUIESCED already, as shown in Example 5-5. No **RLS QUIESCE** command is needed at this point.

Example 5-5 Extract from LISTCAT of newly defined ICF catalog with LOG(NONE) attribute

```
LOG(NONE) RLSDATA
LOG ------------------NONE RECOVERY REQUIRED --(NO)
VSAM QUIESCED ------(YES) RLS IN USE ---------(NO)
```

To make the ICF catalog that we defined fully RLS-enabled, run the **F CATALOG,RLSENABLE** command. Example 5-6 shows SYSLOG messages issued in response to the command.

Example 5-6 F CATALOG RLSENABLE messages returned in SYSLOG

```
F CATALOG,RLSENABLE(UCAT.RLSTST)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I MODIFY CATALOG UCAT.RLSTST TO STATE RLSENABLE SUCCESSFUL
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

The **F CATALOG** command in Example 5-6 is directed toward a specific ICF catalog. If you want a system-wide scope to change all RLS enabled catalogs in one command across the sysplex, issue the **F CATALOG,RLSENABLE,SYSTEM** command. The same command syntax is valid for **RLSQUIESCE**.

A **LISTCAT** of the user catalog now shows the VSAM QUIESCED field changed from YES to NO. This means that the catalog is no longer quiesced, but enabled for RLS. RLS enablement is also indicated by RLS IN USE field, which has changed to YES, as shown in an extract from a **LISTCAT** of a user catalog in an RLS enabled state. See Example 5-7.

Example 5-7 Listcat example of ICF catalog in RLS enabled state.

```
RLSDATA
LOG ------------------NONE RECOVERY REQUIRED --(NO)
VSAM QUIESCED -------(NO) RLS IN USE --------(YES)
```

Another way of displaying the RLS readiness of an ICF catalog is to use the **F CATALOG,OPEN** command. This command lists all open catalogs, and displays settings for the open catalogs in this environment.

Example 5-8 shows a display of an RLS enabled ICF Catalog using the **F CATALOG,OPEN** command.

Example 5-8 F CATALOG,OPEN command output listing RLS state

```
-F CATALOG,OPEN
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC348I ALLOCATED CATALOGS
*CAS*************************************************************
* FLAGS -VOLSER-USER-CATALOG NAME
* YSU-R- SBXHS8 0001 UCAT.RLSTST
*****************************************************************
* Y/N-ALLOCATED TO CAS, S-SMS, V-VLF, I-ISC, C-CLOSED, D-DELETED,
* R-SHARED, A-ATL, E-ECS SHARED, K-LOCKED, U-RLS, W-SUSPENDED
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Note the U indicator for an RLS-enabled catalog.

If you have a lower-level system in your sysplex and a shared RLS candidate catalog is open in this system, you receive an error message when trying to enable this catalog.

Example 5-9 shows the error messages displayed when trying to RLS enable a catalog shared with a lower-level system.

Example 5-9 Error message trying to RLSENABLE an ICF catalog shared with lower-level system

```
F CATALOG,RLSENABLE(UCAT.RLSTST)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC353I SERVICE 'CLRC' RETURNED RC-236-030, QUIESOFF, UCAT.RLSTST
IEC353I MODIFY CATALOG UCAT.RLSTST TO STATE RLSENABLE UNSUCCESSFUL
IEC353I CATALOG ADDRESS SPACE MODIFY UNSUCCESSFUL
```

You receive a similar error message when trying to perform the RLS enable by using an IDCAMS **ALTER** command.

Example 5-10 shows error messages displayed with the IDCAMS **ALTER RLSENABLE** on a shared catalog.

Example 5-10 IDCAMS RLSENABLE command failure due to sharing with lower-level system

```
IDCAMS SYSTEM SERVICES TIME: 17:37:27
ALTER -
UCAT.RLSTST -
RLSENABLE
IDC3014I CATALOG ERROR
IDC3009I ** VSAM CATALOG RETURN CODE IS 236 - REASON CODE IS IGGOCLRC-30
IDC0532I **ENTRY UCAT.RLSTST NOT ALTERED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 8
```

5.2.4 Fallback from using RLS mode on ICF Catalogs

SMSVSAM recovery is essential for ICF catalogs that use RLS and for other users data sets. SMSVSAM is the server for all RLS processing, and RLS processing fails if SMSVSAM for some reason is not available. For more information about recovering SMSVSAM and the RLS environment, see *z/OS V2R1.0 DFSMSdfp Storage Administration*, SC23-6860.

For ICF catalogs that use RLS, the process to disable the use of RLS if SMSVSAM and the RLS environment encounter problems is straightforward.

Use the **F CATALOG,RLSQUIESCE(ucat name)** command or the **F CATALOG,RLSQUIESCE,SYSTEM** command, and catalog management reverts to normal catalog (non-RLS). Before enabling RLS use on ICF catalogs, it is also suggested that you have a fallback plan in place. New catalog commands support fallback, but earlier level systems need SMSVSAM active to switch catalogs out of RLS mode.

5.2.5 Catalog maintenance considerations in RLS environment

New keywords were introduced in support of RLS enablement for catalogs. In this section, we describe catalog maintenance in an RLS environment.

Backing up your RLS managed ICF catalogs

Backing up your RLS-managed user catalogs with IDCAMS (Export) can be done as in past releases. The following new IDCAMS keywords are available to manage RLS source or target catalogs:

- `RLSSOURCE(NO|YES|QUIESCE)`
- `RLSTARGET(NO|YES|QUIESCE)`

The use of these parameters results in the following influences on processing:

- `NO` indicates that the source and target data sets are opened by using NSR.
- `YES` indicates that the source and target data sets are opened by using RLS, and the data set includes consistent read integrity.
- `QUIESCE` indicates that the source and target data sets are opened by using RLS, and the data set is quiesced before processing any entries. Then, the access is quiesced during backup.

IDCAMS backup

Using IDCAMS for backing up a catalog, you can choose whether to access the catalog using RLS. When **QUIESCE** is specified, RLS access is QUIESCED during backup, and will be resumed when IDCAMS finishes backup. Using this parameter ensures integrity, but also suspends availability during backup. Example 5-11 shows a sample IDCAMS backup of an ICF catalog using **RLSSOURCE(QUIESCE)**.

Example 5-11 IDCAMS backup of ICF catalog using the RLSSOURCE parameter

```
//CATOO4 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DD1 DD DISP=(,CATLG),
// DSN=MHLRES1.BACKUP1.RLSTST1,
// SPACE=(TRK,(2,1),RLSE)
//SYSOUT DD SYSOUT=*
//SYSIN DD *
EXPORT UCAT.RLSTST OUTFILE(DD1) TEMPORARY -
RLSSOURCE(QUIESCE)
```

DFSMS Data Set Services backup

When using DFSMS Data Set Services (DFSMSdss) for backing up RLS-managed catalogs, there is no specific keyword for QUIESCE. DFSMSdss performs the QUIESCE implicitly during backup, and enables RLS when the backup is finished. Also see an example of DFSMSdss backing up an ICF catalog in Example 5-12.

Example 5-12 Example of using DFSMSdss as the backup tool for ICF catalog

```
//MHLRES1D JOB ..................................
//STEPTOO6 EXEC PGM=ADRDSSU,REGION=4O96K
//SYSPRINT DD SYSOUT=*
//TAPE DD DISP=(NEW,CATLG),
// DSN=MHLRES1.DSS.BACK2.MHLRES1A,SPACE=(TRK,(2,1)),
// UNIT=339O,DCB=(BLKSIZE=27920)
//SYSIN DD *
DUMP DS(INC(UCAT.RLSTST )) -
OUTDDNAME (TAPE)
/*
```

DFSMS Hierarchical Storage Manager backup

DFSMS Hierarchical Storage Manager (DFSMShsm) also can back up catalogs. You must ensure that DFSMShsm has access to IGG.CATLOCK RACF profile to work with a locked catalog. Otherwise, DFSMShsm is a privileged user and can back up catalogs. DFSMShsm backs up aliases, and restores them when restoring the catalog.

Remember to **LOCK** or **SUSPEND** the catalog before performing the restore to control the integrity of that catalog.

5.2.6 Using REPRO MERGECAT on an RLS-managed catalog

Running the **REPRO MERGECAT** function in a shared environment from a non-RLS-enabled ICF catalog to an RLS-enabled ICF catalog requires the **QUIESCE** done on the target catalog before you are allowed to issue a **REPRO MERGECAT**.

Attempting to issue the **REPRO MERGECAT** without having the target catalog quiesced for RLS access results in the error message that is shown in Example 5-13.

Example 5-13 Example of REPRO mergecat from non RLS managed ICF catalog to RLS managed

```
REPRO -
INDATASET(UCAT.VSBOX01 ) -
OUTDATASET(UCAT.RLSTST.MHLRES1A ) -
LEVEL(MHLRES1A) -
MERGECAT
IDC3300I ERROR OPENING UCAT.RLSTST.MHLRES1A
IDC3351I ** VSAM OPEN RETURN CODE IS 168
IDC3003I FUNCTION TERMINATED. CONDITION CODE IS 12
```

Reporting on catalogs

Several reporting options are available to track your catalog efficiency, mainly connected with performance for non-RLS-managed and RLS-managed catalogs. In the following section, we review the reporting options.

Reporting on non-RLS-managed and RLS-managed catalogs is done through **F CATALOG** commands, or through standard z/OS tools, such as IBM z/OS Resource Measurement Facility (IBM RMF).

Using the **F CATALOG,REPORT,PERFORMANCE** command returns the I/O statistics of the catalog address space since the last IPL, or since the last reset of these statistics through the **F CATALOG,REPORT,PERFORMANCE(RESET)** command. An example of the output from this report is shown in Example 5-14. Note the `Statistics since` time stamp that shows the starting time for this summary report. This is not the entire report, which is extensive. It is just a snapshot from the beginning of the report.

Example 5-14 Extract of ICF catalog performance report

```
-F CATALOG,REPORT,PERFORMANCE
 IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
 IEC359I CATALOG PERFORMANCE REPORT
 *CAS************************************************
 *  Statistics since 11:05:51.79 on  11/01/2014       *
 *  -----CATALOG EVENT----   --COUNT--  ---AVERAGE---  *
 *  Entries to Catalog        598,187      1.627 MSEC  *
 *  BCS ENQ Shr Sys           889,257      0.106 MSEC  *
 *  BCS ENQ Excl Sys            8,538      0.365 MSEC  *
 *  BCS DEQ                   960,368      0.016 MSEC  *
 *  VVDS RESERVE CI           222,951      0.035 MSEC  *
 *  VVDS DEQ CI               222,951      0.020 MSEC  *
 *  VVDS RESERVE Shr            1,443K     0.048 MSEC  *
 *  VVDS RESERVE Excl          17,015      0.301 MSEC  *
 *  VVDS DEQ                    1,457K     0.036 MSEC  *
 *  SPHERE ENQ Excl Sys         5,312      0.096 MSEC  *
*   SPHERE DEQ                  5,312      0.014 MSEC  *
 *  CAXWA ENQ Shr              47,505      0.015 MSEC  *
 *  CAXWA DEQ                  47,505      0.002 MSEC  *
 *  VDSPM ENQ                 865,947      0.001 MSEC  *
 *  VDSPM DEQ                 865,947      0.001 MSEC  *
....continues ..
```

The REPORT PERFORMANCE command should be issued regularly to identify potential performance bottlenecks because your catalogs are a key component in I/O activity.

The most key information in this report that should be used for comparison purposes at different moments is the “Entries to Catalog” line. There, you can see the total amount of all catalog requests in this z/OS since the last reset operation (COUNT) and the total average response time (AVERAGE). All the other lines numerically describe all the partial functions that must be executed when ceasing an ICF catalog.

Cache data space and ISC buffering efficiency

The **F CATALOG,REPORT,CACHE** command reports on the cache data space and the ISC buffering efficiency. The hit rate is reported, and should be at least 20% to reach a reasonable value. The statistics must be measured over time to be valid. Measuring for an hour might not represent your normal catalog access. You can reset the statistics by issuing the **F CATALOG,CLOSE** command.

When the catalog is reopened after this rest, a new cache structure is built. Through this process, all counters are reset except for the purge values. A sample extract of the cache report is shown in Example 5-15. (All of the ICF catalogs in Example 5-15 are buffered at ISC.)

Example 5-15 Sample cache report based on F CATALOG,REPORT,CACHE command

```
-F CATALOG,REPORT,CACHE
 IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
 IEC359I CATALOG CACHE REPORT
 *CAS*****************************************************************
 *   HIT% -RECORDS- -SEARCHES --FOUND-- -DELETES- -SHR UPD- --PURGE-- *
 *                                                                     *
 * UCAT.ZOSR1A                                    (ISC)                *
 *    94%        1        17        16         0        0         0  *
 * UCAT.VTFM                                      (ISC)                *
 *    99%        6   335,705   335,699         0        0         0  *
 * UCAT.VSTEST1                                   (ISC)                *
 *    94%        1        17        16         0        0         0  *
 * UCAT.VSBOX11                                   (ISC)                *
 *    98%        3       220       217         0        0         0  *
.... continues ..
```

The **F CATALOG,REPORT,CATSTATS(catname)** gives you catalog accesses statistics, including BUFND, BUFNI, and STRNO information, about a specific user catalog, as specified in the **CATNAME** parameter. An extract of this report is shown in Example 5-16.

Example 5-16 Example of report from F CATALOG,REPORT,CATSTATS command

```
-F CATALOG,REPORT,CATSTATS
 IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
 IEC359I CATALOG I/O STATS REPORT
 *CAS*****************************************************************
 *     ADDS  UPDATES     GETS   GETUPD  DELETES  BUFNI  BUFND STRNO *
 *                                                                   *
 * UCAT.ZOSR1A                                                       *
 *       0        0       18        0        0      10     16      8 *
 * UCAT.VTFMTAPE                                                     *
 *      57      181      752      181        0      10     16      8 *
 * UCAT.VTFM                                                         *
 *       0        0       23        0        0      10     16      8 *
 * UCAT.VSTEST1                                                      *
 *       0        0       18        0        0       4      4      2 *
... continues ..
```

The new **CATSTATX** parameter on the **F CATALOG,REPORT,CATSTATX(catname)** command reports the same information as **F CATALOG,CATSTATS**, but also includes CA-RECLAIMS and CA-REUSES information. This command can report on one or more catalogs with the use of asterisks (*) or wildcard in the **CATNAME** specification.

This report is reporting one ICF catalog at a time, as shown in Example 5-17.

Example 5-17 Example of F CATALOG,REPORT,CATSTATX report output

```
-F CATALOG,REPORT,CATSTATX(UCAT.RLSTST)
 IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
 IEC359I EXTENDED CATALOG STATS
 *CAS*****************************************************************
 *  CATALOG NAME     = UCAT.RLSTST                                   *
 *  INSERTS (ADDS)   =        0                                      *
 *  UPDATES          =        0                                      *
 *  RETRIEVES        =       18                                      *
 *  RETRIEVES FOR UP =        0                                      *
 *  ERASES (DELETES) =        0                                      *
 *  CA-RECLAIMS      =        0                                      *
 *  CA-REUSES        =        0                                      *
 *  BUFNI SETTING    =        4                                      *
 *  BUFND SETTING    =       10                                      *
 *  STRNO SETTING    =        2                                      *
 *  AVG ELAPSED TIME =     5.000 MSEC                               *
 *  AVG CPU TIME     =   842.000 USEC                               *
 *CAS*****************************************************************
```

The information about CA-RECLAIMS, AVG ELAPSED TIME, and AVG CPU TIME was added in z/OS V2.R1.

RMFMON III includes RLS activity reports. Choose the SYSPLEX option, and there are several reports on RLS. The RMFMON III Sysplex panel with these reports is shown in Example 5-18.

Example 5-18 Available VSAM RLS reports in RMF3

```
 1 SYSSUM   Sysplex performance summary                (SUM)
 2 SYSRTD   Response time distribution                 (RTD)
 3 SYSWKM   Work Manager delays                        (WKM)
 4 SYSENQ   Sysplex-wide Enqueue delays                 (ES)
 5 CFOVER   Coupling Facility overview                  (CO)
 6 CFSYS    Coupling Facility systems                   (CS)
 7 CFACT    Coupling Facility activity                  (CA)
 8 CACHSUM  Cache summary                              (CAS)
 9 CACHDET  Cache detail                               (CAD)
10 RLSSC    VSAM RLS activity by storage class         (RLS)
11 RLSDS    VSAM RLS activity by data set              (RLD)
12 RLSLRU   VSAM LRU overview                          (RLL)
```

RMF3 reports on RLS activity

Choose **RMF Sysplex Report**. Three RLS reports are available in this menu.

Coupling facility lock structure report

The **D SMS,CFLS** command displays information about the CF RLS lock structure. This information includes the lock rate, lock contention rate, false contention rate, average number of requests waiting for locks, the lock structure size, and primary structure information.

If the lock structure is in duplex mode, secondary structure information is also displayed.

Output from the **D SMS,CFLS** command is shown in Example 5-19.

Example 5-19 Coupling facility lock structure report.

```
D SMS,CFLS
IEE932I 853
IGW320I 07:42:24 Display SMS,CFLS(IGWLOCK00 )
PRIMARY STRUCTURE:IGWLOCK00 VERSION:CBE13E285CE22FD3 SIZE:14336K
RECORD TABLE ENTRIES:27365 USED:6
LOCK STRUCTURE MODE: SIMPLEX STATUS: ENABLE
System Interval LockRate ContRate FContRate WaitQLen
SC64 1 Minute 0.0 0.000 0.000 0.00
SC64 1 Hour 0.0 0.000 0.000 0.00
SC64 8 Hour 0.0 0.000 0.000 0.00
SC64 1 Day 0.0 0.000 0.000 0.00
(04) 1 Minute 0.0 0.000 0.000 0.00
(04) 1 Hour 0.0 1.923 0.000 0.00
(04) 8 Hour 0.0 5.048 0.000 0.00
(04) 1 Day 0.0 4.968 0.000 0.00
***************** LEGEND ******************
LockRate = number of lock requests per second
CONTRATE = % of lock requests globally managed
FCONTRATE = % of lock requests falsely globally managed
WaitQLen = Average number of requests waiting for locks
```

SMF reports

SMF is another source of information. You need a reporting tool that can post-process the SMF data. Choose the option that you have in your organization.

SMF type 42 subtypes 15 through 19 can be used for summary reporting on RLS use. The following subtypes are available:

- Subtype 15. VSAM RLS Storage Class Response Time Summary.
- Subtype 16. VSAM RLS Data Set Response Time Summary.
- Subtype 17. VSAM RLS Coupling Facility Lock Structure Usage.
- Subtype 18. VSAM RLS CF Cache Partition Usage.
- Subtype 19. VSAM RLS Local Buffer Manager least recently used (LRU) statistics summary.

Use the **`V SMS,MONDS(spherename),ON`** command to collect subtype 16 statistics. Additionally, data set collection for SMF type 42 must be turned on when using RMF. For catalogs and VSAM in general, all of the catalog-specific SMF records can be used:

- Type 60. VVR Updated.
- Type 61. ICF Define.
- Type 62. VSAM OPEN.
- Type 64. VSAM CLOSE (new for catalogs in z/OS 1.13 and later).
- Type 65. ICF Delete.
- Type 66. ICF Alter.

5.2.7 Migration considerations

SMSVSAM must be enabled on all systems in the sysplex by using RLS mode for catalogs. This requirement is valid for DFSMS V2.R1 systems and lower-level systems that are participating in the sysplex. It is also required that all systems in the sysplex are migrated to DFSMS V2.R1 before implementing RLS mode for shared ICF catalogs.

If you enable toleration support on the earlier systems, a candidate catalog to move to RLS is usable only in a system earlier than V2.R1 in RLS Quiesced mode. It is not usable when the catalog enters RLS Enabled mode.

Related publications

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

IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Note that some publications that are referenced in this list might be available in softcopy only:

- *IBM z/OS V2R2: Storage Management and Utilities,* SG24-8289
- *VSAM Demystified*, SG24-6105

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

ibm.com/redbooks

Other publications

The following publications are also relevant as further information sources:

- *z/OS DFSMS Managing Catalogs* SC23-6853
- *z/OS DFSMS Access Method Service Commands,* SC23 6846

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

A Practical Guide to ICF Catalogs

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8262-01

ISBN 0738442089

Printed in U.S.A.