SG24-4765-00

CICS and VSAM Record Level Sharing: Planning Guide

September 1996

IBM
International Technical Support Organization
San Jose Center

International Technical Support Organization

SG24-4765-00

CICS and VSAM Record Level Sharing: Planning Guide

September 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special Notices” on page 95.

First Edition (September 1996)

This edition applies to Version 1, Release 1 of CICS Transaction Server for OS/390, Program Number 5655-147.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. 471 Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Contents

Preface

You should read this redbook if you are planning to implement VSAM record-level sharing (RLS) with CICS Transaction Server for OS/390 Version 1 Release 1 and DFSMS/MVS Version 1 Release 3 or are evaluating whether you should use VSAM RLS.

You should read this redbook early in the planning process, well before you install CICS TS and start to implement RLS. The book focuses on the things you have to do to to prepare for CICS TS, DFSMS/MVS 1.3, and VSAM RLS. It does not give detailed information about installing CICS TS or DFSMS/MVS 1.3 or implementing VSAM RLS. You can find detailed implementation information in the product manuals and in the following redbooks, not available until 1Q97, which are companion volumes to this book:

- *CICS and VSAM Record Level Sharing: Implementation Guide* (SG24-4766)
- *CICS and VSAM Record Level Sharing: Recovery Considerations* (SG24-4768)

How This Redbook Is Organized

The redbook is organized as follows:

- Chapter 1, "Planning Is the Key"

  In this chapter we give a brief introduction to some of the tasks you must perform to implement CICS TS, DFSMS/MVS 1.3, and VSAM RLS.

- Chapter 2, "VSAM Record-Level Sharing"

  This chapter gives an overview of the benefits of VSAM RLS, the products that you will need to implement it, and additional benefits that accrue from using those products.

- Chapter 3, "Prerequisite Hardware and Software"

  You must implement specific prerequisite hardware and software products to access VSAM files in RLS mode. In this chapter we list the *minimum* levels of support required.

- Chapter 4, "Migrating to VSAM Record Level Sharing"

  In this chapter we highlight the major migration steps you must take to implement VSAM RLS.

- Chapter 5, "The Coupling Facility"

  In this chapter we introduce some of the coupling facility functions and provide information to help you decide which type of coupling facility, how many coupling facilities, and how much storage you need to implement VSAM RLS.

- Chapter 6, "MVS/ESA Version 5 Release 2"

  In this chapter we introduce the functions that are relevant to VSAM RLS, most importantly the MVS system logger. We also discuss other functions that are important for operating a VSAM RLS environment in a Parallel Sysplex.

- Chapter 7, "DFSMS/MVS Version 1 Release 3"

In this chapter we describe the tasks that the storage administrator must carry out with the MVS systems programmer and the CICS systems programmer to prepare to use VSAM RLS. We make the assumption that you are already running a Parallel Sysplex environment and SMS is active.

- Chapter 8, “CICS Transaction Server for OS/390 Version 1 Release 1”

  In this chapter we highlight the planning items for the migration to CICS Transaction Server for OS/390 Version 1 Release 1, with emphasis on the implementation of VSAM RLS. We assume that you have planned for all of the prerequisite items, such as installing the Parallel Sysplex with the necessary base software.

- Chapter 9, “Application Programming Considerations”

  In this chapter we deal with planning considerations for application programs where there are significant changes.

- Chapter 10, “Operations”

  In this chapter we deal with operational considerations, specifically changes to:

  – CICS startup
  – CICS shutdown
  – Batch processing of online VSAM RLS data sets
  – Operating the MVS system logger
  – Temporary storage queue server region

- Chapter 11, “Planning Checklist for VSAM Record-Level Sharing”

  To fully implement VSAM RLS, you have to consider the items outlined in the planning checklist in this chapter.

- Appendix A, “Classroom Education”

  In this appendix we describe a course that is available to help you successfully implement CICS and VSAM RLS.

The Team That Wrote This Redbook

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

Hugh Smith, an Advisory Information Technology Consultant at the International Technical Support Organization, San Jose Center, is a CICS specialist with 10 years of experience working with customers. He writes extensively and teaches IBM classes worldwide on all areas of CICS. Hugh has presented several papers at technical conferences, including the CICS Technical Conference in the United States. Before joining the ITSO four years ago, Hugh worked closely with CICS development at IBM UK. Hugh has a Master′s Degree in Natural Sciences from Cambridge University, England.

Karl-Heinz Marquardt is an Advisory Information Technology Consultant for Transaction Systems, working as second level field support for IBM Germany. He has more than 25 years of experience with IBM, most recently working with CICS customers throughout Germany. He prepares and holds workshops and briefings for customers throughout Germany and is one of the authors of *Designing a Distributed CICS Application* (GG24-4361). His areas of expertise include MVS, CICSPlex SM, CICS, and client/server architecture.

Luigi Ingrosso is an Information Technology Coordinator in IBM Italy. He has nearly 30 years of experience working with IBM customers in Italy and is one of the authors of *CICS/ESA 3.1 Migration - Planning and Implementation* (GG24-3485) and *S/390 Configurations - Parallel and Traditional* (SG24-4514). His areas of expertise include MVS, teleprocessing, and CICS.

Rayno van Zyl is a Senior Information Technology Specialist in South Africa. He has 11 years of experience in the OLTP field. His areas of expertise include MVS, CICS, CICSPlex SM, MQ Series, and application architectures.

Byron Jones is a Storage Analyst in IBM UK. He has worked for IBM for 14 years and has 9 years of experience in the storage products field. He holds a degree in geography from the University of Swansea. His areas of expertise include DFSMS, DFDSS, DFHSM, and ABARS.

Maddelena Tosoni is an Information Technology Specialist in MVS and DFSMS in the IBM Southern Europe/Middle East/Africa (SEMEA) Availability Center of Rome, Italy. She has seven years of experience in the MVS and storage products field. She holds a degree in Information Technology from the G. Armellini Institute in Rome, Italy. Her areas of expertise include MVS and related products such as DFSMS, DFDSS, DFHSM, DFSORT, RACF, SMP/E, and JES2.

We would like to extend our thanks to:

Bob Haimowitz
International Technical Support Organization, Poughkeepsie Center

Nigel Williams
Installation Support Center, IBM Hursley UK

Bob′s help in preparing and supporting our Parallel Sysplex was invaluable. Nigel provided his experience from development as well as many insights we would not have gained otherwise. We could not have completed this project without them.

Thanks also to the following people for the time they spent reviewing this book, and for their suggestions for improvement:

Jim Coolbaugh
Application Control Services, Raleigh

Catherine Moxey
CICS Development, IBM UK Laboratories

Tom Russell
International Technical Support Organization, Poughkeepsie Center

Henrik Thorsen
International Technical Support Organization, Poughkeepsie Center

Bob Yelavich
Dallas Systems Center

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Planning Is the Key

VSAM record-level sharing (RLS) is an exciting new function that brings significant benefits to users of CICS for MVS/ESA, especially those who want to fully exploit a Parallel Sysplex environment.[1] You must ensure that several important components are in place before you can realize these benefits. Some of these components may be new to you, some you may already have, some you may need to use in new ways. Whichever is the case, the Boy Scout motto, "Be prepared," summarizes the task ahead of you. This book helps you be prepared.

1.1 Education

The first step is to understand the new environment. Depending on your job, you may have to understand some or all of the following:

- What is RLS?
- What is a coupling facility, and why do I need one?
- Why do I need MVS/ESA Version 5 Release 2, and what does it give me?
- What is new in DFSMS/MVS Version 1 Release 3?
- What is new in CICS Transaction Server for OS/390 Version 1 Release 1 (CICS TS)?

1.1.1 What Is Record-Level Sharing?

Chapter 2, "VSAM Record-Level Sharing" on page 3 gives you a very brief overview of RLS. For more details, read the information in the *CICS Release Guide*, which contains an excellent overview of the RLS functions.

1.1.2 What Is a Coupling Facility?

Chapter 5, "The Coupling Facility" on page 19 tells you what a coupling facility is, which software components use it, and how to gain an initial understanding of the coupling facility storage size you need.

1.1.3 Why Do I Need MVS/ESA Version 5 Release 2?

Chapter 6, "MVS/ESA Version 5 Release 2" on page 37 introduces the major functions of MVS/ESA 5.2 and explains why it is a prerequisite product for CICS TS and VSAM RLS.

1.1.4 What Is New in DFSMS/MVS Version 1 Release 3?

Chapter 7, "DFSMS/MVS Version 1 Release 3" on page 45 describes the changes in DFSMS/MVS 1.3 to support VSAM RLS and discusses the tasks that the storage administrator must carry out with the MVS and CICS system programmers to prepare to use VSAM RLS.

---

[1] A sysplex is a collection of MVS system images that are coupled together through certain hardware and software products to process work. A coupling facility is needed to create a Parallel Sysplex, which is required to provide the services for data sharing. See *System/390 MVS Sysplex Overview: Introducing Data Sharing and Parallelism in a Sysplex.*

1.1.5 What Is New in CICS Transaction Server for OS/390 Version 1 Release 1?

Chapter 8, "CICS Transaction Server for OS/390 Version 1 Release 1" on page 57 gives an overview of the new functions in CICS TS. The *CICS Release Guide* gives a more detailed description of the new functions of CICS TS.

1.2 Planning

When you have completed your education, you can build your plan. The bulk of this book is dedicated to assisting you in that endeavor. It helps you with the questions you are likely to have before you actually install CICS TS and DFSMS/MVS 1.3.

You should allow ample time for planning before you implement VSAM RLS. Chapter 4, "Migrating to VSAM Record Level Sharing" on page 13 lists the prerequisite products that you must have in place before you can use VSAM RLS. Chapter 4, "Migrating to VSAM Record Level Sharing" on page 13 discusses the migration paths you can take for implementing the prerequisite products and reviews the considerations for migrating the CICS regions in your CICSplex to CICS TS.

Remember that your planning must include a fallback plan!

1.3 Implementation

This book is not a detailed implementation guide. Use the *CICS and VSAM Record Level Sharing: Implementation Guide* for detailed implementation guidance when you start to install CICS TS and DFSMS/MVS 1.3 in preparation for VSAM RLS.

Chapter 2. VSAM Record-Level Sharing

In this chapter we present an overview of the benefits of VSAM RLS, the products you need to implement VSAM RLS, and additional benefits that come from using those products.

2.1 Sharing VSAM Data

VSAM data sets often have to be shared among several different applications in an MVS system image or among applications on several different MVS system images. For example, transactions running in different CICS regions may have to access the same VSAM data set at the same time, or a CICS transaction may have to access a VSAM data set at the same time that a batch job is using the data set. The requirements for sharing can vary. Sometimes applications only have to read the data set. Sometimes an application has to update the data set while other applications are reading it. The most complex case is when all applications have to update the data set, and all require complete data integrity.

2.1.1 VSAM Sharing: Before Record-Level Sharing

VSAM in a non-RLS environment provides only limited support for the sharing of data sets. It does not provide the functions that are required to enable multiple users to update a shared data set with complete integrity.

CICS users have been able to share VSAM data sets with integrity by using function shipping. With function shipping, one CICS region accesses the VSAM data set on behalf of other CICS regions. Requests to access the data set are shipped from the region where the transaction is running to the region that has access to the file. Function shipping provides a solution for the CICS user, but it does have limitations: The processor cost of implementing function shipping can be high, and function shipping does not address the problems of sharing data sets between CICS regions and batch jobs.

Database management systems such as IBM DATABASE2 (DB2) and IBM Information Management System (IMS) Database Manager resolve the problem of sharing data with integrity among multiple users. Now, with DFSMS/MVS 1.3 and CICS TS, VSAM has many of the functions that database management systems provide, including support for the Parallel Sysplex environment.

2.1.2 VSAM Sharing: With Record-Level Sharing

RLS is a new VSAM function provided by DFSMS/MVS 1.3 and exploited by CICS TS. VSAM data sets are opened in RLS mode, which allows them to be shared, with full update capability, among many applications running in many CICS regions. As part of VSAM RLS, DFSMS/MVS 1.3 supports a new data sharing subsystem, SMSVSAM, which runs in its own address space. SMSVSAM provides the VSAM RLS support required by CICS application-owning regions (AORs) and batch jobs within each MVS system image in a Parallel Sysplex environment. The SMSVSAM subsystem, which is generally initialized automatically during an MVS initial program load (IPL), uses the coupling facility for its cache structures and lock structures. It also supports a common buffer pool for each MVS system image.

2.2 VSAM Sharing: The Record-Level Sharing Environment

VSAM RLS exploits the Parallel Sysplex environment. RLS can be used on a single MVS system image or across many MVS system images. Figure 1 shows how CICS AORs and batch jobs located on different MVS system images can share access to VSAM data sets. Access is through the SMSVSAM subsystem.

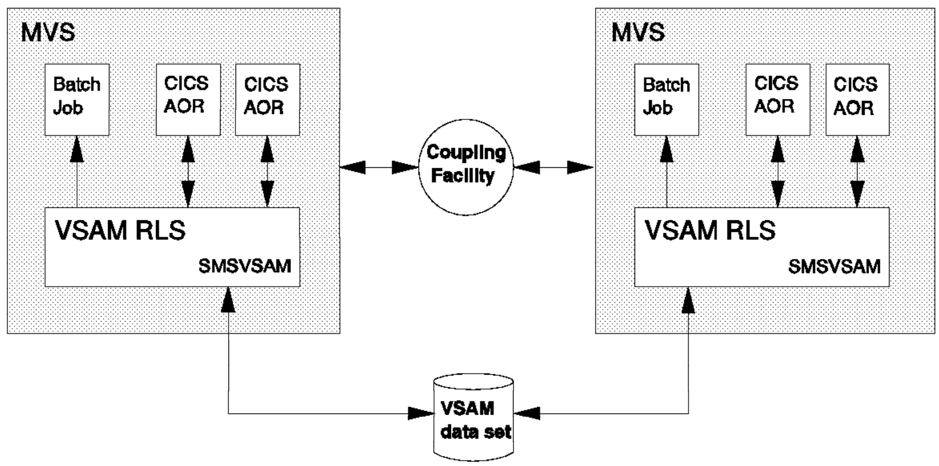

Figure 1. The Target Environment

Applications running in CICS AORs can share access to VSAM data sets opened in RLS mode with full update capability. Batch jobs have only read access to recoverable VSAM data sets opened in RLS mode, as shown in Figure 1. Batch jobs can update nonrecoverable VSAM data sets opened in RLS mode.

Note: Because the CICS recoverable files function provides transactional recovery for applications, **VSAM RLS is expected to be used primarily by CICS applications**.

2.3 Benefits of VSAM Record-Level Sharing

VSAM RLS improves price for performance, improves the availability of systems using VSAM data sets, enhances the integrity of systems using VSAM data sets, extends the processing capacity of your system, and provides flexible ways to configure your environment and balance workloads without requiring changes to your existing applications.

2.3.1 Price for Performance

VSAM RLS provides a data solution that is powerful enough to handle complex business requirements but can be run on one or more smaller, more competitively priced System/390 (S/390) microprocessors.

2.3.2 Availability

Users increasingly require access to data held in VSAM data sets every hour of the day, every day of the year. In a non-RLS system, all CICS application access to a specific VSAM data set is typically through a single file-owning region (FOR). The FOR thus becomes a single point of failure. RLS enables sharing of VSAM data among many applications running in many CICS regions, possibly on several MVS system images, all within a single sysplex. RLS improves the availability of VSAM data sets during both planned and unplanned outages. In a VSAM RLS system, CICS application access to VSAM is through the SMSVSAM address space, and there is one SMSVSAM address space in each MVS system image. Thus if one of the MVS systems or subsystems is not available, applications can access their data from another MVS system or CICS region.

VSAM RLS also improves data integrity and availability through the use of common locking and buffering.

RLS Improves Availability

RLS provides improvements for availability.

2.3.3 Integrity

VSAM RLS and CICS TS improve data integrity in several ways.

2.3.3.1 No Dirty Reads

VSAM RLS extends the scope of locking to provide support for read integrity. Some applications are intolerant of so-called dirty reads.[2] VSAM RLS ensures that an application can read a record from a file and, if required, be assured that it receives the latest committed version of the data.

2.3.3.2 Correct Data Set Information

VSAM data sets accessed in non-RLS mode have no way of propagating hi-used relative byte address (HURBA) and hi-allocated relative byte address (HARBA) information to other systems. Therefore a batch program cannot read all records in the data set if a CICS region on any other system is updating the data set. (The HURBA and HARBA are only updated when CICS closes the non-RLS mode data set.) VSAM RLS removes this restriction.

2.3.3.3 RLS Locks Held If CICS Fails

Locks for RLS mode VSAM data sets are held by the SMSVSAM subsystem, not by CICS. Thus the locks obtained by in-flight and in-doubt CICS tasks can be held if an AORs fails, preventing other applications from accessing uncommitted data until CICS is restarted or the locks are released by operator command.

Locks for non-RLS mode VSAM data sets are held by CICS. This is true for CICS TS and all earlier releases of CICS. In CICS TS the locks for non-RLS mode data sets (and other resources, such as transient data queues) are held by a new CICS domain, the enqueue domain. If CICS fails, the locks are unavailable. Another application could open the non-RLS mode VSAM data set before CICS is restarted and access the uncommitted records.

---

[2] A read request that does not involve any locking mechanism, and which may obtain invalid data—that is, data that has been updated, but is not yet committed, by another task. This could also apply to data that is about to be updated, and which will be invalid by the time the reading task has completed.

2.3.3.4 Other Improvements

A new component of CICS TS, the recovery manager domain, rebuilds the enqueue domain locks when CICS TS is restarted. As with earlier releases of CICS, CICS TS backs out uncommitted data for an in-flight unit of work (UOW). Unlike earlier releases of CICS, CICS TS keeps locks for in-doubt UOWs until either it contacts the coordinator for the UOW and synchronization can resume, or a predefined in-doubt wait period expires and a decision is taken to commit or back out based on the INDOUBT option of the TRANSACTION definition. This arbitrary decision by one partner is usually referred to as a *heuristic decision*. Releases earlier than CICS TS always make a heuristic decision during restart processing or when the transaction goes in doubt. The decision is controlled by the WAIT option of the transaction definition.

The CICS interface with SMSVSAM is through a control access control block (ACB), and CICS registers with this ACB to open the connection. RLS mode VSAM data sets are thus treated similarly to local resources for recovery purposes. CICS applications can update VSAM resources and any other subsystems that support the two-phase commit protocol (for example, DB2, DBCTL, MQ Series, and other CICS TS regions) in the same UOW. The recovery manager domain ensures that all resources involved in a UOW have the same state.

RLS Improves Data Integrity

RLS offers improvements for data integrity by preventing dirty reads, providing correct data set information, supporting retained locks, and ensuring that all resources in a UOW have the same state.

2.3.4 Extends Capacity

You can use a CICS FOR to allow shared access to a VSAM data set from multiple AORs. As the volume of transactions using the FOR to access VSAM data sets increases, it is possible that the central processor (CP) requirement of the FOR can exceed the capacity of a single processor in a central processor complex (CPC), because most of the work in the FOR takes place under its quasi-reentrant task control block (TCB). This CP constraint can cause performance problems. Without VSAM RLS the options for dealing with CP constraint in the FOR are limited. Methods that have been considered in the past include:

- Spreading the data among more than one FOR. If you consider this option, be aware of the possible risk under specific circumstances of data inconsistencies due to the heuristic decisions made for in-doubt UOWs. The recovery manager domain of CICS TS removes this exposure.
- Moving applications and copies of the data to another MVS system. This method requires separate copies of the data, which causes considerable maintenance overhead.
- Moving the whole system to a larger processor. This method can be expensive.

Parallel Sysplex technology and VSAM RLS enable you to add another VSAM RLS subsystem to another CPC and access the common (shared) data sets. An FOR, and thus function shipping, is not required.

RLS Extends Capacity

VSAM RLS extends capacity.

2.3.5 Flexible Configurations and Workload Balancing

The configuration of the system environment is much more flexible with the availability of VSAM RLS. All AORs in the Parallel Sysplex can access the same VSAM data sets through their local VSAM RLS subsystem.

The whole CICS workload can be spread over all available CICS regions in a sysplex, independent of VSAM data location. Systems staff planning and execution time is saved if there are workload or system design changes because the VSAM data will be available from each system.

RLS Allows for Flexible Configurations and Workload Balancing

VSAM RLS provides flexibility for configuration and workload balancing.

2.4 Products to Support Key RLS Functions

VSAM RLS requires certain prerequisite products to be in place. These products have to support key functions of VSAM RLS, and thus they rely on other prerequisite products (whether or not you actually use VSAM RLS).

The key VSAM RLS functions are described below.

2.4.1 Central Logging

VSAM data can be accessed by multiple users (for example, CICS regions on different MVS images). The forward recovery logs recording changes to the data must reflect all of the changes. We therefore require a mechanism for merging log records from multiple sources.

The **MVS system logger** of MVS/ESA 5.2 provides support for central logging in this environment by holding logs centrally in one or more coupling facilities for all CICS regions in a sysplex. The MVS system logger merges log records from multiple CICS TS regions into one log as required. CICS TS exploits this new facility.

No logging for batch programs

Note that, as with earlier releases of CICS and DFSMS, CICS logs the updates to RLS mode VSAM data sets. Because logging is not performed for batch jobs, the batch jobs are not allowed to update recoverable VSAM data sets that have been opened in RLS mode.

IBM has made the following statement of direction on future enhancements to CICS TS:

> IBM intends to improve the cost of computing for single system customers by enhancements to the new CICS Logging and Journal management facilities announced September 10, 1996, by providing support for a new type of log stream. This log stream is single system in scope, that is, only one system in a sysplex can have active connections

to the log stream. This type of log stream will enable the benefits of the System Logger to be available to single OS/390 systems without requiring additional hardware.

IBM may change its product plans in the future for business or technical reasons.

2.4.2 Central Locking

In order for all subsystems accessing VSAM data to have a common understanding of which records are available to be read or updated, locking must be provided from a central place. DFSMS/MVS 1.3 provides central locking management. Locks are held in structures within coupling facilities.

2.4.3 Central Unit-of-Work Control

All parties must understand UOW control, so that they do not work with uncommitted data. The recovery manager together with the locking facilities of DFSMS/MVS 1.3 provide the UOW control required and allow CICS TS to demonstrate the properties of atomicity, consistency, isolation, and durability (ACID).

2.4.4 Retained Locks

Locks must be retained if a subsystem fails and leaves updated records in an uncommitted state. If a CICS region fails during sync point processing, an application program does not want another user to access the data before it knows whether it has to commit or back out (as in the case of DB2 and DBCTL data sharing). Retained locks ensure that data integrity is maintained until the failed CICS region has completed its recovery processing.

DFSMS/MVS 1.3 provides support for retained locks.

2.4.5 Summary

VSAM RLS requires:

- MVS/ESA 5.2—for the MVS system logger
- DFSMS/MVS 1.3—for locking (including retained locks)
- CICS TS—for UOW control
- A coupling facility—for the MVS system logger and the locking structures

Chapter 3. Prerequisite Hardware and Software

You must implement specific prerequisite hardware and software products to access VSAM files in RLS mode. In this chapter we list the *minimum* levels of support required.

3.1 Hardware Requirements

You must have an S/390 Parallel Sysplex with at least one coupling facility to implement VSAM RLS. The coupling facility provides shared storage and shared storage management functions for the Parallel Sysplex. A coupling facility is required for VSAM RLS because:

- VSAM uses a lock structure (IGWLOCK00) in the coupling facility to allow data sharing among multiple users with data integrity.
- VSAM uses cache structures within the coupling facility to enable data to be stored in the coupling facility buffers.
- The MVS/ESA 5.2 system logger writes log data into list structures in the coupling facility.

You can implement your coupling facility in several ways, as discussed in 5.1, “What Type of Coupling Facility?” on page 19. Whichever way you implement it, we strongly recommend that you implement at least two coupling facilities. The reasons for this are discussed further in 5.2, “How Many Coupling Facilities?” on page 21. We do not recommend a single coupling facility configuration. Refer to Chapter 5, “The Coupling Facility” on page 19 for a detailed description of setting up your coupling facility.

3.2 Software Requirements

The following software is required for VSAM RLS:

- **MVS/ESA 5.2**

  MVS/ESA 5.2 is the minimum level of operating system needed to provide the common logging support required by the CICS TS log manager. If you are not already using MVS/ESA 5.2, you may find it easier to implement OS/390 instead. See Chapter 6, “MVS/ESA Version 5 Release 2” on page 37 for more detailed information.

- **DFSMS/MVS 1.3**

  VSAM data sets opened in RLS mode must be SMS managed, as must other data sets such as the offload data sets for the MVS system logger. DFSMS/MVS 1.3 provides new storage classes to support these requirements. DFSMS/MVS 1.3 also provides the common locking facility needed for RLS mode access.

  Note: Within a sysplex, an SMS configuration can be shared between systems running DFSMS/MVS 1.3 and earlier levels of DFSMS/MVS; however, toleration PTFs must be applied to all pre-DFSMS/MVS 1.3 systems so that they do not conflict with RLS access.

  See Chapter 7, “DFSMS/MVS Version 1 Release 3” on page 45 for more detailed information.

- **CICS TS**

  To exploit the functions that VSAM RLS provides, you need CICS TS, which provides:

  – RLS support, enabling CICS systems to share the VSAM data sets directly rather function shipping to an FOR

  – Application programming interface (API) extensions for RLS

  – The log manager

  – The recovery manager

  See Chapter 8, "CICS Transaction Server for OS/390 Version 1 Release 1" on page 57 for more detailed information.

- **Global resource serialization - or an equivalent function**

  Global resource serialization (GRS) is required to ensure cross-system serialization of VSAM resources and other DFSMS/MVS control structures altered by VSAM RLS. Non-RLS open requests for a data set are not allowed if the data set has been opened in RLS mode or RLS transaction recovery for the data set is pending.

- **Appropriate levels of COBOL, PL/I, C/370, and FORTRAN**

  COBOL, PL/I, C/370, and FORTRAN run-time libraries can be updated to support VSAM RLS for batch applications. This does not automatically imply that your application is VSAM RLS tolerant. You must determine whether your applications run correctly in an RLS environment before accessing VSAM data sets in RLS mode. See 7.1.1, "Who Can Use VSAM RLS?" on page 45 for more information.

  With allocations based on data set definition (DD) statements in your JCL, you can use the RLS parameter to specify RLS access and read integrity options. You do not have to recompile your applications to access the data set in RLS mode. For FORTRAN dynamic allocation, the language supports the specification of RLS access through the FILEINF command. In this case, a recompile of the program is necessary.

  Both COBOL and FORTRAN can create self-contained load modules. At a minimum, applications using these languages must relink in order to use RLS access.

  You must also provide environment run-time libraries for batch applications that will use VSAM RLS data access. See 9.4, "Programming Language Considerations" on page 81 for more detailed information.

In addition to the software listed above, you should implement software for security checking and for controlling forward recovery of your VSAM data sets. You can use either IBM products or products from other vendors. If you implement the IBM products, you must use these levels of software:

- **Resource Access Control Facility Version 2 Release 1 (RACF 2.1)**

  RACF 2.1 provides new function for authorization and protection of data and system resources:

  – A new class name, SUBSYSNM, authorizes a subsystem (such as CICS TS) to open a VSAM ACB and use the new VSAM RLS functions.

  – A new RACF FACILITY class profile, STGADMIN.IGWSHCDS.REPAIR, controls access to the AMS SHCDS command functions, which you can use to list outstanding SMSVSAM recovery requirements and control that recovery.

If you do not use RACF to secure your installation, you must ensure that your security product supports the new class name and the new FACILITY class profile. We do not discuss RACF 2.1 further in this book.

- **CICS VSAM Recovery Version 2 Release 3 (CICSVR 2.3)**

  Your forward recovery utility must interact with SMSVSAM to accomplish its function. It must also understand the new CICS logging structure introduced with CICS TS. Therefore you need CICSVR 2.3, or a functionally equivalent product, for forward recovery of VSAM data sets.

3.3 Non-IBM Software Requirements

Check with the vendors that their products exploit VSAM RLS and are compatible with CICS TS and DFSMS/MVS 1.3.

Chapter 4. Migrating to VSAM Record Level Sharing

In this chapter we highlight the major migration steps you must take to implement VSAM RLS. We do not provide a detailed migration guide. Our main purpose is to show you how, from a specific starting point, you can get to the software levels needed to use VSAM RLS.

4.1 Software Migration

Chapter 3, "Prerequisite Hardware and Software" on page 9 lists the products required to implement VSAM RLS . Figure 2 shows you some possible migration paths for reaching the levels you need for VSAM RLS. Pick the starting point that matches your current level of operating system software. The underlined products show the level of software we recommend for that level of operating system.

Follow the flow from each starting level. Those are the steps we strongly recommend you go through to implement MVS/ESA 5.2, CICS TS, and DFSMS/MVS 1.3.

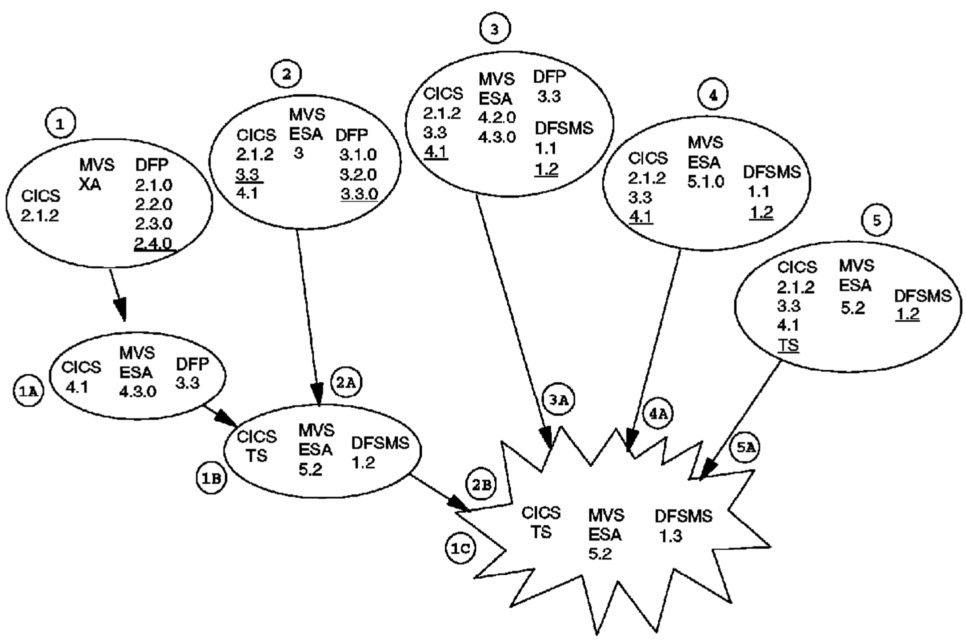

Figure 2. Suggested Migration Paths

Migration steps :

1. To go from point **1** to point **1A**, migrate MVS and DFP first and then CICS.

   To go from **1A** to **1B**, migrate MVS and DFSMS first and then CICS.

   To go from **1B** to **1C**, change only the DFSMS release.

2. To go from point **2** to **2A**, migrate MVS and DFSMS first and then CICS.

   To go from **2A** to **2B**, change only the DFSMS release .

3. To go from point **3** to **3A**, migrate MVS and DFSMS first and then CICS.

4. To go from point **4** to **4A**, migrate MVS and DFSMS first and then CICS.
5. To go from point **5** to **5A**, change only the DFSMS release .

We recommend that you go through the above steps, because they represent the different phases you should exploit for a soft migration. (With a soft migration, the gaps and differences between the starting level and the target level are not very deep and numerous.)

Because the version of CICS you install is MVS dependent, you have to migrate MVS and DFP or DFSMS first. You cannot install CICS TS, for example, if you do not have MVS/ESA 5.2.

DFP or DFSMS is the data management portion of the MVS operating system and is deeply intertwined with MVS usability. Therefore, when you install a new MVS, you should also install the latest version of DFP or DFSMS that goes with it.

4.2 Hardware Migration

If you do not already have a Parallel Sysplex, you have to implement one to support VSAM RLS. Whatever your current hardware configuration, you have several basic hardware migration routes to Parallel Sysplex. Some, or all, of these will be feasible in your installation. Chapter 3 of *Parallel Sysplex: Migration Paths* lists the basic options and includes an example of each option. Chapter 8 of the same book gives more details on how to select appropriate coupling facilities and processors. The *MVS/ESA SP Version 5 Sysplex Migration Guide 5.1.0 Version 5.2* contains detailed information about MVS/ESA 5.2 and the hardware components that are required to build a Parallel Sysplex.

We do not cover hardware migration further in this book, as it is well documented in the two books referenced above.

4.3 CICS Region Migration

You do not have to change your FOR for SMSVSAM and migrate all of your CICS regions to CICS TS regions, all at the same time. You can migrate progressively, especially during the test and development phases before cutover to production.

4.3.1 Migration Path 1

One migration order is to migrate your FOR to CICS TS first. For example, if you currently have a number of CICS/ESA 4.1 regions that access their VSAM files through a CICS/ESA 4.1 FOR, you could do the following:

1. Begin by migrating the FOR to a CICS TS region.
2. Leave the AORs at the CICS/ESA 4.1 level, continuing to function ship file control requests to the new FOR. Initially, the new FOR can continue using VSAM files in non-RLS mode.
3. When you are satisfied that the CICS TS FOR is functioning correctly, redefine the files as RLSACCESS(YES). The AORs continue to function ship their file requests, but the FOR actually uses SMSVSAM to access the data sets.

4. You can now progressively migrate the AORs to CICS TS, changing the remote file definitions to local file definitions and the RLSACCESS(NO) attribute to RLSACCESS(YES).

5. If you have applications that cannot use RLS mode (for example, if you have applications running in a CICS/ESA 3.3 AOR, or if you must support function shipping requests from outside your Parallel Sysplex), retain the CICS TS FOR. Otherwise you can dispense with the FOR.

Figure 3 illustrates this gradual migration process.

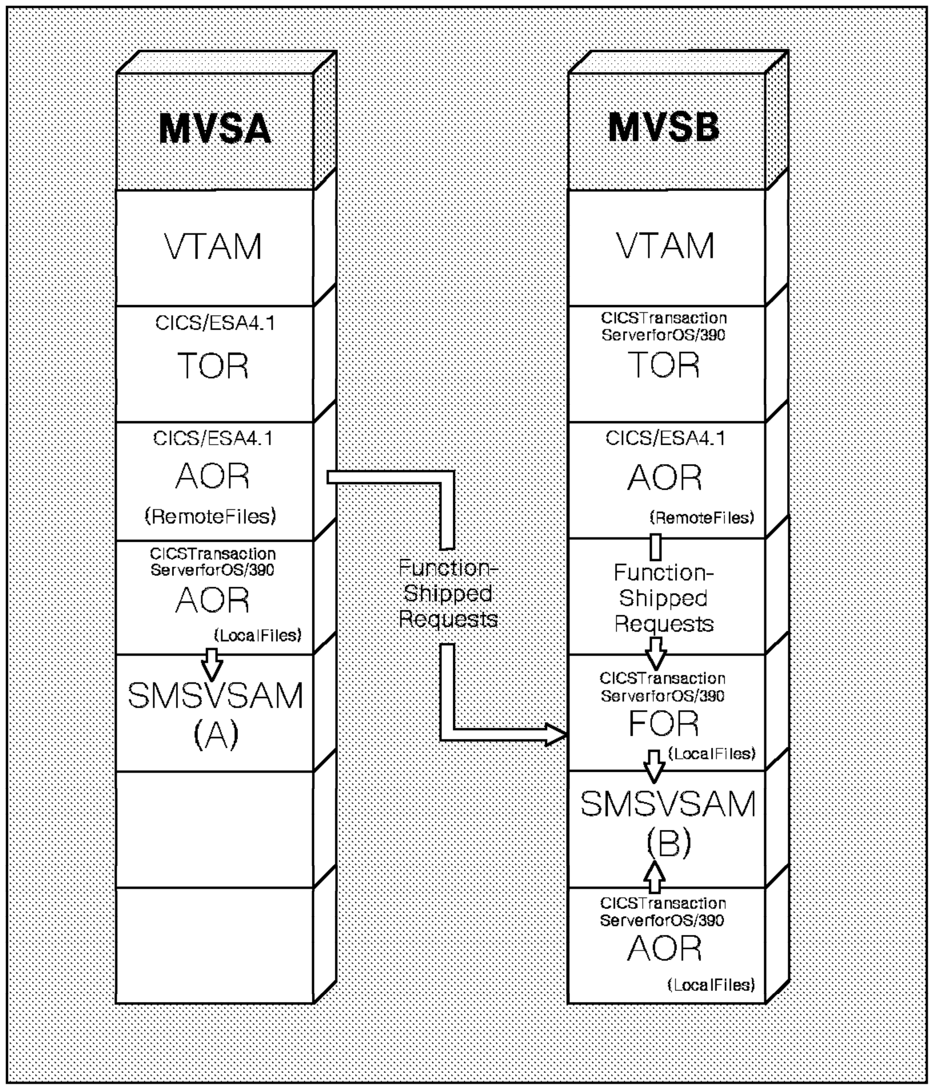

Figure 3. Migration Scenario: Function Shipping and RLS

The diagram shows the point in the migration process when two of the AORs remain at the CICS/ESA 4.1 level and two are migrated to CICS TS. The VSAM files in the CICS/ESA 4.1 regions are defined as remote, and file requests continue to be function shipped to the FOR. The AORs running under CICS TS access files directly in RLS mode through the services of SMSVSAM.

4.3.1.1 Advantages of Migration Path 1

There is a performance advantage to migrating the FOR before migrating any AORs connected to the FOR by multiregion operation (MRO). If you migrate the AORs first, and continue using the MRO-connected FOR at the old release level, you could incur significant additional overhead. This overhead results from the extra logging that can occur in CICS TS AORs function shipping to CICS/ESA 4.1 FORs, depending on the nature of the UOWs. You avoid the risk of increased overhead if you migrate the FOR either first or at the same time as the AORs.

4.3.1.2 Disadvantages of Migration Path 1

Migration path 1 requires you at some stage to both function ship requests from an AOR to an FOR and use RLS mode to access the VSAM data set from the FOR. This process imposes an additional pathlength, and you therefore may prefer the path discussed in 4.3.2, "Migration Path 2."

Migrating the FOR is the most challenging part of moving to CICS TS; it requires the greatest amount of work in planning for and defining log streams to the MVS logger. Therefore, you may prefer the path discussed in 4.3.3, "Migration Path 3" on page 17.

4.3.2 Migration Path 2

An alternative migration order is to change steps 4 and 5 in 4.3.1, "Migration Path 1" on page 14. The migration path then becomes:

1. Migrate the FOR to a CICS TS region.
2. Leave the AORS at the CICS/ESA 4.1 level, continuing to function ship file control requests to the new FOR. The new CICS TS FOR continues using VSAM files in non-RLS mode.
3. When you are satisfied that the CICS TS FOR is functioning correctly, you can progressively migrate the AORs to CICS TS. The CICS TS AORs still function ship their VSAM file requests to the CICS TS FOR.
4. Redefine the files as RLSACCESS(YES) in the FOR, and change the remote file definitions to local file definitions with RLSACCESS(YES) in the AORs.
5. If possible, dispense with the CICS TS FOR.

4.3.2.1 Advantages of Migration Path 2

Migration path 2 has the same logging performance advantage as migration path 1. It offers further advantages in that file requests use either function shipping *or* RLS mode, not both, to access the VSAM data set from the AOR.

4.3.2.2 Disadvantages of Migration Path 2

Migration path 2 still requires you to migrate the FOR and all associated log streams before you have gained significant operational experience with CICS TS. Therefore, you may prefer the path discussed in 4.3.3, "Migration Path 3" on page 17.

Migration path 2 requires you to migrate directly from non-RLS mode to RLS mode with update from multiple AORs. Therefore, you may prefer the path discussed in 4.3.1, "Migration Path 1" on page 14, or you may prefer to continue function shipping to an FOR and accessing files in RLS mode from there until you have gained experience in RLS mode operation. The latter option has the performance disadvantage of requiring both function shipping and RLS mode access for any file request.

4.3.3 Migration Path 3

Migration path 3 depends on migrating at least one AOR before you migrate your FOR to CICS TS:

1. Migrate one AOR to CICS TS.
2. Fully test the operation of your application programs in this environment, and become familiar with the operation of the new logging functions. You can also test the operation of the recovery manager domain, using the new CIND transaction.
3. Once you are satisfied with your testing of the CICS TS AOR, follow the path described in 4.3.1, "Migration Path 1" on page 14 or 4.3.2, "Migration Path 2" on page 16 to migrate the FOR and the remaining AORs.

4.3.3.1 Advantages of Migration Path 3

Migration path 3 enables you to become familiar with CICS TS in an operational environment *before* you make the major changes to the logging subsystem required to support access to VSAM files (in either RLS or non-RLS mode).

4.3.3.2 Disadvantages of Migration Path 3

Migration path 3 can have the additional logging overhead discussed in migration path 1. Migrating only one AOR minimizes the effect of this overhead, while enabling you to gain operational experience with CICS TS.

4.4 Additional Migration Planning Documentation

See the following publications for additional information about migration:

DFSMS/MVS Version 1 Release 3 Planning for Installation
DFSMS/MVS V1R2.0 Planning for Installation
DFSMS/MVS Version 1 Release 1 Planning for Installation
MVS/DFP Version 3.3 Planning Guide
MVS/ESA Planning: Installation and Migration for MVS/ESA System Product Version 4
MVS/ESA Planning: Installation and Migration with JES2 MVS/ESA System Product Version 5
CICS Release Guide
CICS Migration Guide
CICS/ESA 4.1 Release Guide
CICS/ESA 4.1 Migration Guide
CICS/ESA 3.3 Release Guide

Chapter 5. The Coupling Facility

A coupling facility provides locking, caching, and list services between coupling-capable S/390 processors running MVS/ESA Version 5 or later. Coupling facility links are used to connect the coupling facility to the coupling-capable processors. In this chapter we introduce some of the coupling facility functions and provide information to help you decide which type of coupling facility you need to implement VSAM RLS, how many coupling facilities you need, and how much storage you need to implement VSAM RLS.

The coupling facility control code (CFCC) provides the coupling facility functions. The CFCC can run in a processor resource/systems manager (PR/SM) logical partition (LP) of an IBM Enterprise System/9000 (ES/9000) 9021 711-based processor, an IBM S/390 Parallel Enterprise Server 9672 (IBM 9672), or an IBM S/390 Coupling Facility 9674 (IBM 9674).

You can use an Integrated Coupling Migration Facility (ICMF) for migration and for testing software that requires a coupling facility. ICMF provides the CFCC functions without coupling facility links and is limited to MVS/ESA systems running in an LP on a single physical system. ICMF is available on the 9021 711-based processors, 9121 511-based processors, and IBM 9672s.

The coupling facility enables software on different systems in the Parallel Sysplex to share data with the assurance that the data will not be corrupted and will be consistent among all sharing users. To share data, systems must have connectivity to the coupling facility through coupling facility links.

Note: You must have at least CF level 2 microcode installed before using VSAM RLS.

5.1 What Type of Coupling Facility?

Your coupling facility can be one of the types discussed below.

5.1.1 Dedicated Coupling Facility

The dedicated coupling facility is a CPC that runs only the CFCC. The CFCC runs in a single LP with dedicated processor resources. There are two kinds of dedicated coupling facilities: the stand-alone IBM 9674, which can run only the CFCC; and IBM 9672 CPCs, which give you the option of dedicating the CPC to run only the CFCC when you define the coupling facility LP.

5.1.2 Coupling Facility LP Defined with Other LPs on a Processor

You can define the coupling facility LP to run on a CPC in the S/390 microprocessor cluster, a 9021 711-based model, or a 9121 511-based model with other LPs defined to the CPC. The CPC in the S/390 microprocessor cluster and the 9021 711-based models enable you to use coupling facility links to connect the coupling facility LP to MVS images on the same processor or on other processors that are capable of connecting to the coupling facility. On the 9121 511-based models, you must use ICMF on the processor to define an LP to run the coupling facility, and you do not install coupling facility links.

5.1.3 Coupling Facility LP and the ICMF

For testing coupling facility function, you can define coupling facility LPs with other LPs on a processor and use ICMF to simulate coupling facility links. With ICMF you do not use coupling facility links to connect the coupling facility, LPs, and MVS images, and you cannot connect the coupling facility LP to MVS images on other processors. ICMF is not generally recommended in a production Parallel Sysplex environment for CICS TS or VSAM RLS. The function is available on the following processors:

- 9021 711-based models
- 9121 511-based models
- IBM 9672 processors

5.1.4 Volatile or Nonvolatile Coupling Facility

A volatile coupling facility is one where the contents of memory are lost if the power supply is interrupted.

You can make a dedicated coupling facility nonvolatile by adding an uninterruptible power supply (UPS) to the coupling facility. The UPS supplies power to the coupling facility during power failures.

As an alternative to a UPS, you can use either battery backup or a local uninterruptible power supply. Battery backup uses internal batteries to provide power during an outage. The batteries will last for a few minutes if the coupling facility continues to run. You can use the batteries to supply power for *power save state*, in which case only the memory is kept active by the internal batteries during a power outage. In power save state the internal batteries last up to 80 minutes.

IBM also supplies a Local Uninterruptible Power System (Local UPS) model for IBM 9672s and IBM 9674s, to provide enhanced power protection. The IBM 9910 Model B89 provides full power for between 5 minutes and 1 hour. In power save state the 9910 provides power for up to 1.5 hours, or for up to 12 hours for a coupling facility that uses the optional battery feature.

5.1.5 Recommended Coupling Facility

Use a dedicated coupling facility for your production VSAM RLS environment. We recommend an IBM 9674, although any dedicated coupling facility is acceptable. An IBM 9674 provides a coupling facility that is external to all processors in the Parallel Sysplex, thus affording the highest availability. If any of the processors fails, the information held in the coupling facility is still available to the other processors, and the rest of the Parallel Sysplex can continue to run. If the coupling facility were an LP in one of the processors, failure of that processor would stop the whole Parallel Sysplex, because both copies of information for that processor (within the MVS system and in the coupling facility) would be lost. We also recommend that your coupling facility be nonvolatile.

For more information about this subject, see the section entitled “Coupling Facility Selection” in Chapter 8, “ Hardware,” of *Parallel Sysplex: Migration Paths*.

5.2 How Many Coupling Facilities?

You will have to decide how many coupling facilities you need in your Parallel Sysplex; one, two, or more.

For maximum availability, we recommend that you set up at least two coupling facilities with global connectivity. As with any component, the coupling facility could fail. You also have to plan for maintenance activities, which could require removing a coupling facility from the Parallel Sysplex. For maximum availability, you should therefore configure at least two coupling facilities.

A subsystem builds a structure in one coupling facility. If that coupling facility should fail, the subsystem is notified. Most subsystems attempt to rebuild the structure in another coupling facility. A subsystem running on any MVS system in the Parallel Sysplex that attempts a rebuild will rebuild the structure from information kept in processor storage. Thus, to achieve high availability, backup coupling facility capability must be available. The backup coupling facility must have processor and storage resources immediately available to it. This can be achieved by providing a standby coupling facility or spare capacity in other coupling facilities. We recommend that the second coupling facility be an IBM S/390 Coupling Facility 9674 or another dedicated coupling facility.

A coupling facility is a storage device, and you must ensure that all data has been successfully moved elsewhere before IPLing it. Unless you can rebuild into another coupling facility, it is vital that subsystem activities are shut down and the MVS logger has completed the offload to DASD.

For more details about the coupling facility, refer to *MVS/ESA Setting Up a Sysplex* and *System/390 MVS Sysplex Hardware and Software Migration*.

5.3 What Is in the Coupling Facility?

You will have to decide how much storage your coupling facility requires. To help with this decision, we discuss how storage in the coupling facility is used. In 5.5, “Sizing Requirements for Record-Level Sharing” on page 29 we take you through some exercises that will help you determine the CICS TS, DFSMS/MVS 1.3, and VSAM RLS storage requirements.

The size of the coupling facility is defined in terms of the total amount of storage to be used. Total storage includes both the control areas required by the coupling facility control code and data areas used by the application. The size is affected by coupling facility allocation rules and the coupling facility allocation increment size, a function of the level of the coupling facility control code.

Storage in a coupling facility is divided into distinct objects called *structures*. These structures are defined by the coupling facility resource management (CFRM) policy. Authorized programs use the structures to implement data sharing and high-speed serialization. The structure types are *cache*, *list*, and *lock*, each providing a specific function to the application.

Note:

The CFRM policy allows for a maximum of 128 structures for all purposes.

For performance and recovery purposes, you can distribute structures on different coupling facilities. Having more than one coupling facility enables you to plan for recovery because structures can be rebuilt on another coupling facility if a structure or the coupling facility itself fails. Other factors to consider when you define structures include the storage reserved for dumping coupling facility data and rebuilding structures in the coupling facility.

The more MVS systems that participate in data sharing, the greater is your need for increased coupling facility capacity and coupling facility link connectivity in the Parallel Sysplex.

Table 1 on page 23 lists all of the IBM product functions that use the coupling facility to allocate their structures.

Function	Structure Type	Documentation
XCF signaling	List	*System/390 MVS Sysplex Hardware and Software Migration*
VTAM generic resources function	List	*VTAM V4R2 Network Implementation Guide*
JES2 checkpoint data set in a coupling facility	List	*MVS/ESA SP V5 JES2 Initialization and Tuning Guide*
IMS DBCTL data sharing with IRLM 2.1 **IRLM locking** **OSAM buffer invalidate** **VSAM buffer invalidate**	 Lock Cache Cache	*System/390 MVS Sysplex Application Migration*
RACF database in a large Parallel Sysplex environment	Cache	*RACF V2 System Programmer's Guide*
Automatic tape switching	List	*System/390 MVS Sysplex Hardware and Software Migration*
DB2 data sharing **IRLM locking** **Group buffer pools** **Shared communication area**	 Lock Cache List	*IBM DATABASE2 for MVS/ESA V4 Data Sharing: Planning and Installation*
MVS system logger	List	*MVS/ESA SP V5 Assembler Services Guide*
CICS log manager	List	This book
VSAM record-level sharing **Lock structure** **Cache structure**	 Lock Cache	This book

Table 1. Coupling Facility Structures for IBM Products

IBM also provides some tools that help in determining an initial environment, such as the CP90 S/390 Parallel Sysplex Quick-Sizer in CPSTOOLS. The Quick-Sizer is a PC-based productivity tool for assessing the overall hardware requirements when moving applications from one or more System/370 (S/370) or System/390 (S/390) processor environments to a Parallel Sysplex. If you are not an IBM employee, please ask your IBM representative about CP90. For more information about sizing the IMS and DB2 requirements for the coupling facility, see *Parallel Sysplex Capacity Planning*.

The International Technical Support Organization is planning to publish the *OS/390 Parallel Sysplex Configuration Cookbook* early in 1997. You should refer to that book for consolidated information about coupling facility structures.

5.3.1 Lock Structure for Record-Level Sharing

To use VSAM RLS, you must define a single coupling facility lock structure, which is used to enforce the protocol restrictions for VSAM RLS data sets and maintain the record-level locks. Ensure that the coupling facility lock structure has universal connectivity, so that it is accessible from all systems in the Parallel Sysplex that support VSAM RLS. For high availability environments, we recommend that you use a nonvolatile coupling facility for the lock structure. If you maintain the lock structure in a volatile coupling facility, a power outage could cause a failure and loss of information in the coupling facility lock structure. Should that happen, all outstanding recovery (CICS restart and backout) for any affected data sets must be completed before new sharing work is allowed for those data sets. The coupling facility lock structure is named IGWLOCK00. See 5.5.1, “Sizing the Lock Structure for Record-Level Sharing” on page 29 for information about estimating the size of IGWLOCK00.

For more details about the coupling facility structures, see the *DFSMS/MVS Version 1 Release 3 DFSMSdfp Storage Administration Reference*.

5.3.2 Cache Structures

Coupling facility cache structures provide a level of storage between local memory and DASD cache. They are also used as a system buffer pool for VSAM RLS data when that data is modified on other systems. Each coupling facility cache structure is contained in a single coupling facility. You might have multiple coupling facilities and multiple coupling facility cache structures.

5.3.3 CICS TS Logger Structures

In CICS TS logging is performed with the MVS system logger. Both the system logger and CICS have extensive recovery procedures to prevent any loss of logged data. If data loss occurs, the procedures minimize the impact on the integrity of the CICS system.

CICS uses several log streams, which differ in terms of the size of the average block written and the length of time the data should be retained in the coupling facility structure to optimize CICS performance. You should define a number of different structures to CFRM and the MVS system logger for CICS system logs.

For an overview of the operation of the MVS system logger, see 6.1, “The MVS System Logger” on page 37.

5.3.3.1 CICS System Log

There is one system log for each CICS region. The system log may be defined as DUMMY if recovery action is not required for the region. A DUMMY system log prevents warm and emergency restarts.

The CICS system log is a single, logical stream made up of two separately defined log streams:

- The primary log stream, called *DFHLOG* by convention, contains information about normal (short-lived) in-flight UOWs.
- The secondary log stream, called *DFHSHUNT* by convention, contains information about long-lived UOWs, that is, UOWs that could not complete because of backout failures, are in-doubt, or belong to long-running tasks. The secondary log stream is transparent to user applications. The system initialization table (SIT) parameter AKPFREQ effectively defines whether or

not a transaction is considered long running. A transaction that does not sync point (either implicitly or explicitly) within two AKPFREQ cycles is considered long running, and the corresponding log data is moved to the DFHSHUNT log.

The system log is intended only for recovery purposes, such as dynamic transaction backout or emergency restart. The CICS log manager reads the system log directly during dynamic transaction backout. It does not use the in-storage dynamic log mechanism that earlier releases of CICS used.

5.3.3.2 Log of Logs

CICS writes the log of logs (DFHLGLOG) to provide information to forward recovery programs such as CICS VSAM Recovery (CICSVR). The log of logs is a user journal containing copies of the tieup records written to forward recovery logs. Thus it provides a summary of the recoverable VSAM data sets that CICS has used, when it used them, and to which log stream the forward recovery log records were written.

5.3.3.3 Forward Recovery Log Streams

Forward recovery logs are used only by CICS file control. CICS does the logging for RLS and non-RLS data sets. You do not have to define a log stream for each data set; log data for multiple data sets can be written to the same log stream. There is a tradeoff between transaction performance, fast recovery, and having a large number of log streams to manage.

Redundant data must be deleted from log streams periodically so that the logger inventory entry for the stream does not fill up. For a forward recovery log, the deletion process will be related to the data backup frequency. You can keep four generations of backup. When you take the next backup, as well as when you delete the oldest backup generation, you should also delete records on the forward recovery log older than the oldest backup, because they will no longer be needed during forward recovery.

5.3.3.4 User Log Streams

User log streams have similar characteristics to forward recovery log streams. They are used for application purposes. For example, an EXEC CICS WRITE JOURNAL command would specify a user log stream as the target for the command. The log of logs is a special user journal, used by CICS itself.

User log streams have similar characteristics to forward recovery log streams. They are used for application purposes. For example, an EXEC CICS WRITE JOURNAL command would specify a user log stream as the target for the command.

5.3.3.5 Autojournal Log Streams

With automatic journaling, CICS automatically writes records to an autojournal log stream when records are read from or written to files or when there are input or output messages for terminals accessed through VTAM. You specify that you want autojournaling for VSAM files by using the autojournaling options on the file resource definition in the CSD. For BDAM files, you specify the options on a file entry in the file control table. You specify that you want terminal control autojournaling on the JOURNAL option of the profile resource definition referenced by your transaction definitions. You could use these messages to create audit trails.

5.3.4 CICS TS Log Structure Names and Characteristics

We recommend a naming convention such as LOG_purpose_nnn, where *purpose* identifies the usage type, and *nnn* is the number of structures to be allocated to a *purpose*.

You might therefore create the following structures:

LOG_DFHLOG_001
: CICS systems logs. This structure should be quite large. If CICS can keep all of the data required for backout within the structure, the need for data to spill to DASD data sets is reduced. The average buffer size tends to be small.

LOG_DFHSHUNT_001
: CICS secondary system logs. This structure can be quite small. Data is written to it infrequently; when data is written to it, large buffer sizes are used.

LOG_USERJRNL_001
: User journals where block writes are never forced (using EXEC CICS WAIT or the WAIT option of WRITE JOURNALNAME). For these log streams, the average and maximum buffer sizes will be the same.

LOG_GENERAL_001
: Forward recovery logs and user journals where blocks are forced to the log periodically.

Others
: LOG_OPERLOG_001, LOG_LOGREC_001, and so on, for non-CICS uses of the MVS system logger.

5.3.5 CICS TS Log Structure Allocation

When deciding on the number of structures to allocate, consider the following:

- The CFRM policy allows for a maximum of 128 structures for all purposes.
- Smaller structures can be allocated, rebuilt, and recovered faster than larger structures.
- It might be desirable to keep test CICS systems and other systems not in regular use in separate structures so that their use of the structure does not interfere with the space tuning for production CICS systems.
- Share structures between MVS images. If an MVS image or logger address space fails and the surviving MVS image is using the same log stream structures (not necessarily the same log streams), the surviving MVS image will be notified of the failure and can start immediate log stream recovery for the log streams used by the failing MVS. Otherwise, recovery will be delayed until the next time a system connects to a log stream in the affected structures or the failing system's logger address space restarts.
- Because of the manner in which the MVS system logger allocates space within the structure, try not to place log streams with dissimilar characteristics in the same structure, especially log streams with very different values for INITSIZE and AVGBUFSIZE. See 5.5.4, "CICS TS Logger Structure Sizings" on page 31 for an explanation of how to estimate values for INITSIZE and AVGBUFSIZE.

  To specify the number of log streams that use the resources of a single coupling facility structure, use the LOGSNUM parameter on the IXCMIAPU service. Each log stream is allocated a proportion of the structure space according to the number of currently connected log streams (up to the limit specified in LOGSNUM). For example, you can define a structure with a LOGSNUM value of, say, 30 log streams. If only 10 log streams are connected, each log stream can use one-tenth of the space in the structure. As other log streams are connected and disconnected, the MVS system logger adjusts the proportion of space to be used by each log stream.

If DFHLOG log streams for regions with different characteristics were to be allocated to the same structure, you would either waste space in the structure or find that one log stream is frequently spilling to DASD. For example, cloned AORs have the same logging characteristics and should be defined in the same structure. A terminal-owning region (TOR), however, is likely to require less structure space and have a smaller AVGBUFSIZE; it may be more appropriate, therefore, to define TORs in their own structure.

Estimate how many log streams will be connected to a structure and use that number when specifying the LOGSNUM parameter to DFHLSCU for the purpose of estimating the INITSIZE structure. You can use a higher value for LOGSNUM when defining the structure to the MVS system logger so that you have the flexibility to connect more log streams to the structure in the future (each LOGSNUM consumes approximately 2000 bytes of structure storage). A LOGSNUM value in the 10 - 20 range is optimum in many environments.

Note: It is not possible to change AVGBUFSIZE, MAXBUFSIZE and LOGSNUM after a structure has been defined, without deleting and redefining the log streams that are currently defined within the structure.

- Use the correct average buffer size. The average buffer size defined for a structure must be reasonably close to the actual buffer size of the log streams using the structure. Otherwise the structure can run out of usable space long before the structure is actually full.

- Set MAXBUFSIZE to slightly less than 64 KB, for example, 64000. With this value CICS can write out the maximum size user record (62 KB), and the coupling facility storage can be allocated in 256-byte elements. If you use the default value for MAXBUFSIZE, coupling facility storage will be allocated in 512-byte elements, which is more wasteful. There is no advantage to setting MAXBUFSIZE to less than 64000. Such a value would limit the maximum record size that can be written to the log but would not improve storage utilization.

  Some users have used the system log for long-lived data, including user journaling, and for audit trails. Therefore, to provide a migration path, the SYSLOG=KEEP option has been provided.

- Redundant data must be deleted from log streams periodically, so that the logger inventory entry for the stream does not fill up. The MVS system logger can keep track of up to 168 log stream data sets. If the log stream data sets are sized to contain about one day′s worth of data, and allowing for less activity on weekends and holidays, you can keep about six months′ worth of data on a log stream.

  If you want to keep data for longer than that, you would have to run a periodic (perhaps monthly) job to archive the log stream data to another medium. If your retention requirements are lower, you can just periodically run a job to delete the redundant data.

- CICS does not delete records from the log stream. It is the user′s responsibility to periodically delete redundant data.

- An eight-character journal name offers a lot of flexibility. We do not recommend that you create vast numbers of journals (for example, by using the terminal name or userid as part of a program-generated name).

5.3.6 CICS TS Temporary Storage Data Sharing

Nonrecoverable temporary storage queues can reside in queue pools in a coupling facility. These nonrecoverable queues may be accessed from different CICS regions. When you use temporary storage data sharing, you have to replace main or auxiliary storage for your temporary storage queues with one or more temporary storage pools, where the scope and function of each temporary storage pool is similar to a queue-owning region (QOR).

5.4 Other System Requirements

In this section we discuss the data set and DASD requirements for the coupling facility associated with CICS TS, DFSMS/MVS 1.3, and VSAM RLS.

5.4.1 Couple Data Sets

You define the coupling facility structures in the CFRM policy. To store policy-related information you have to define couple data sets.

In a Parallel Sysplex environment, you need several types of couple data sets:

- A sysplex couple data set, which contains information about the MVS systems, the cross-system coupling facility (XCF) groups and members running in the sysplex, and general status information
- The CFRM couple data set, which contains the CFRM policy that enables you to define how MVS is to manage your coupling facility resources.
- The sysplex failure management (SFM) couple data set, which contains the SFM policy that enables you to define how system failures, signaling connectivity failures, and PR/SM reconfiguration actions are to be managed.
- The workload management (WLM) couple data set, which contains the WLM policy that enables you to define service goals for workloads.
- The Automatic Restart Manager (ARM) couple data set, which contains the policy that defines how MVS is to manage restarts for specific batch jobs and started tasks registered as elements of ARM.
- The system logger (LOGR) couple data set, which contains the policy that enables you to define log stream or structure definitions.

For additional information about couple data sets, refer to *MVS/ESA Setting Up a Sysplex*.

5.4.2 Staging Data Sets

MVS usually keeps a second copy of the data written to log structures in the coupling facility in a data space, for use when rebuilding a coupling facility log in the event of an error. This is satisfactory as long as the coupling facility is failure-independent (in a separate CPC and nonvolatile) from MVS.

Where the coupling facility is in the same CPC or uses volatile storage, the MVS system logger supports staging data sets for copies of log stream data that would otherwise be vulnerable to failures that impact both the coupling facility and the MVS system images.

The size of each staging data set should be at least as big as the log stream's share of the coupling facility structure but rounding the average block size up to 4 K instead of 256 K.

5.5 Sizing Requirements for Record-Level Sharing

In this section we provide the information you need to make an initial sizing of your coupling facility storage requirements for DFSMS/MVS 1.3, CICS TS, and VSAM RLS.

5.5.1 Sizing the Lock Structure for Record-Level Sharing

You define the coupling facility lock structure, IGWLOCK00, in the CFRM policy, using the IXCMIAPU utility. To estimate the size of IGWLOCK00 (in MB), use the following formula:

```
10 MB * number_of_systems * lock_entry_size
```

where:

number_of_systems
is the number of systems in the Parallel Sysplex

lock_entry_size
is the size of each lock entry. This value depends on the MAXSYSTEM value that is specified to the IXCL1DSU couple data set format utility.

Use the information in Table 2 to determine the actual lock entry size for the different MAXSYSTEM setting values.

Table 2. Effect of MAXSYSTEM Value on Lock Table Entry Size

MAXSYSTEM Value	Lock Entry Size
7 or less	2 bytes
> = 8 and < 24	4 bytes
> = 24 and < 56	8 bytes

Table 3 shows some sample lock allocation estimates.

Table 3 (Page 1 of 2). Coupling Facility Lock Structure Sizing Examples

MAXSYSTEM Value	Number of Systems	Total Lock Structure Size (MB)
< = 7	2	40
	4	80
< = 23	8	320

Table 3 (Page 2 of 2). Coupling Facility Lock Structure Sizing Examples		
MAXSYSTEM Value	**Number of Systems**	**Total Lock Structure Size (MB)**
< default > = 8	2	80
	4	160
	8	320
32	2	160
	4	320
	8	320

Use these estimates as rough initial values. Other factors influence the size of the lock structure, such as false contentions and retained locks. For more detailed information about the sizing of the lock structure, refer to *DFSMS/MVS 1.3 DFSMSdfp Storage Administration Reference*.

For more information about the MAXSYSTEM parameter, refer to *MVS/ESA Setting Up a Sysplex*.

5.5.2 Sizing Cache Structures

To help you achieve the best possible performance with VSAM RLS buffering, the sum total of all of the coupling facility cache structure sizes you define (the coupling facility cache) should ideally be the sum total of the VSAM local shared resource (LSR) buffer pool sizes used in non-RLS mode. VSAM LSR buffer pool size is the sum of LSR pool size and, if used, the corresponding hiperspace pool size. You can run VSAM RLS with less coupling facility cache storage than this, but the coupling facility cache must be large enough for the coupling facility cache directories to contain an entry for each of the VSAM RLS local buffers across all instances of the RLS server. Otherwise the VSAM RLS local buffers become falsely invalid and must be refreshed. To minimize false invalidation, the minimum coupling facility cache structure size should be one-tenth the sum total of the local buffer pool sizes.

Table 4 shows the sum total of the local VSAM RLS buffer pool size before migrating to VSAM RLS for a sample CICS FOR configuration.

Table 4. Sample CICS File-Owning Region Configuration			
	Local Shared Resource Pool Size (MB)	**Hiperspace Pool Size (MB)**	**Sum Total (MB)**
FOR_1	20	30	50
FOR_2	40	No pool	40
FOR_3	30	50	80
			170

When migrating the CICS FOR configuration to VSAM RLS, the coupling facility cache you define should be at least 170 MB. In this way, cross-invalidated local RLS buffers can be refreshed from the coupling facility cache structures.

Performance should improve when the coupling facility cache is larger than the sum of the local VSAM LRS buffer pool sizes. When the coupling facility cache is smaller, performance depends on the dynamics of the data references among the systems involved. In some cases, you might want to consider increasing the size of very small (2 MB to 10 MB) coupling facility caches.

For additional details about CICS TS structures, see the *CICS Migration Guide*.

5.5.3 CICS TS Temporary Storage Data Sharing

Each temporary storage pool is defined, using MVS cross-system extended services (XES), as a keyed list structure in a coupling facility. Thus you must define CFRM policy statements. Using the CFRM policy definition utility, IXCMIAPU, you specify the size of the list structures required and their placement within a coupling facility. For an example of this utility, see member IXCCFRMP in the SYS1.SAMPLIB library, in *MVS/ESA Setting Up a Sysplex*. You can estimate the size of the structure you need according to the amount of data you are likely to use. At initialization this structure size is sufficient for your current needs, and the maximum number of lists you define should reflect your greatest possible needs, as this number cannot be changed. The name of the list structure for a temporary storage data sharing pool is created by appending the temporary storage pool name to the DFHXQLS_ prefix: DFHXQLS_poolname.

For additional details about CICS TS temporary storage pools, see the *CICS System Definition Guide*.

The size of a coupling facility structure, specified in 1 K blocks, includes the total amount of storage required by both the application and the coupling facility itself for structure control information.

Structure sizes are determined differently for each software product that uses the coupling facility. For details on sizing the structures, see *System/390 MVS Sysplex Hardware and Software Migration*.

5.5.4 CICS TS Logger Structure Sizings

We recommend that you use the CICS-supplied utility, DFHLSCU, to help you calculate the amount of coupling facility space you need and the average buffer size of your log streams. For details on how to use DFHLSCU, see the *CICS Operations and Utility Guide*.

If it is inappropriate for you to use DFHLSCU to help you size your coupling facility structures and log streams (perhaps you have no existing journal records to use as input to DFHLSCU, or you are capacity planning for new applications), the sections that follow will help you calculate your space requirements. All values arrived at in the formulae are for *nnn* KB. Also review the information under “Coupling Facility Sizing Considerations” in the *CICS Installation Guide*.

DFHLSCU gives values for the following attributes:

AVGBUFSIZE The average buffer size of your log stream. You supply this attribute in your DEFINE STRUCTURE jobs. To a great extent, AVGBUFSIZE controls the efficient use of space within the structure and can prevent undue DASD offloading.

INITSIZE The initial amount of space to be allocated for the structure in the coupling facility. You supply this attribute in the CFRM policy.

SIZE The maximum size of the log stream structure. You supply this attribute in the CFRM policy.

STG_SIZE The size of the staging data set required by the log stream. You supply this attribute in your DEFINE LOGSTREAM jobs.

The formulae given in the *CICS Installation Guide* will help you calculate values for AVGBUFSIZE, INITSIZE, and SIZE. In this section we discuss only the formulae for AVGBUFSIZE and INITSIZE. You can use SIZE to estimate the storage requirements in the coupling facility for log structures. You have to estimate AVGBUFSIZE in order to estimate INITSIZE.

5.5.4.1 Structure Size for DFHLOG

The system log coupling facility structures are much larger than those used for other log streams and have the greatest potential impact on system performance. Therefore you have to size the structures accurately so that you do not waste coupling facility space by overallocating or get poor performance from underallocating. When planning structure sizes for DFHLOG, ensure that you allow enough storage to avoid the log stream spilling to DASD. Generally, the volume of data that CICS keeps in the primary system log at any one time covers between two and three activity key points.

AVGBUFSIZE for DFHLOG: You can calculate AVGBUFSIZE for DFHLOG from the weighted average of the data logged by the most frequently executed transactions in the system, using the formula shown in Figure 4.

```
AVGBUFSIZE = (bytespersec / writespersec) + 48
```

Figure 4. Formula to Calculate Average Buffer Size for Primary System Log

Calculate bytespersec as:

```
bytespersec = (N1 * D1) + (N2 * D2) + ... (Nn * Dn)
```

Calculate writespersec as:

```
writespersec = lesser of (25 or bytespersec)
```

When you calculate bytespersec:

N1, N2, ... Nn are the transaction frequencies (transactions per second) of the most frequently executed transactions.

D1, D2, ... Dn are the bytes of data logged by each transaction.

INITSIZE for DFHLOG: To calculate INITSIZE use the formula shown in Figure 5 on page 33.

```
      (LOGSNUM *(2500 + (no. entries + 5) * (AVGBUFSIZE * 1.1289 + 195)))
310 + --------------------------------------------------------------------
                                   1024
```

Figure 5. Formula to Calculate Primary System Log Structure INITSIZE

LOGSNUM is the maximum number of log streams being written to the same structure. This formula assumes that the regions sharing the same structure for their log streams have similar characteristics. See 5.3.5, "CICS TS Log Structure Allocation" on page 26 for further discussion on this point.

You can calculate the value of the number of entries (no. entries) as follows:

```
                 (akpintvl + trandur) * writespersec
  no. entries =  -----------------------------------
                                0.9
```

akpintvl is the interval between activity keypoints. It varies with workload. You can calculate it as follows:

```
                            AKPFREQ
      akpintvl = ---------------------------------
                 (N1 * R1) + (N2 * R2) + (Nn * Rn)
```

When you calculate akpintvl:

N1, N2, ... Nn are the transaction frequencies (transactions per second) of the most frequently executed transactions.

R1, R2, ... Rn are the number of log records written by each transaction.

trandur is the execution time (between sync points) of the longest-running transaction that runs as part of the normal workload. If trandur is longer than the akpintvl value, you can either:

- Increase the value of AKPFREQ, so increasing the value of akpintvl (as long as this does not result in an unacceptably large coupling facility structure size).
- Change the application logic to cause more frequent sync points.

  or

- Calculate a structure size based on a shorter transaction duration and accept that DASD offloading occurs when the long-running transaction is used.

Round the final result of the INITSIZE formula up to the next multiple of 256. The formula for SIZE gives a result that is approximately 50% greater than the INITSIZE value.

Setting AKPFREQ: The activity keypoint frequency (AKPFREQ) is a SIT parameter. You must review the value you specify for AKPFREQ in CICS TS. Do not assume that the value you specified in earlier releases is appropriate for CICS TS.

AKPFREQ defines the number of blocks written to the system log data set after which CICS takes an activity keypoint. During the activity keypoint, CICS records the status of all running (or in-flight) transactions on the system log. During emergency restart, CICS reads the system log to find the activity keypoint and then looks for the beginning of all transactions that were in flight at the time of the failure, recording the before image of the updated data. The emergency restart then backs out any data modified or inserted by the in-flight transactions.

For releases earlier than CICS TS, too low an AKPFREQ value causes a lot of activity keypoints to be written to the system log. This degrades CICS performance. Too high a value increases emergency restart time because CICS needs more time to look for the activity keypoint. For releases before CICS TS, the optimal value is to write an activity keypoint every 15 to 20 minutes.

With CICS TS the system log is located in the coupling facility. Both the overhead to write an activity keypoint and the emergency restart time are minimized. CICS TS maintains the records of the system log structure in the coupling facility for approximately two activity keypoints. AKPFREQ becomes very important in defining the size of the system log structure in the coupling facility. In CICS TS AKPFREQ defines the number of *records* (not blocks) written to the system log data set after which CICS takes an activity keypoint. The default value is 4000 records. We suggest that AKPFREQ require an activity keypoint every 30 to 60 seconds, so you should start with a value approximately (1/30 * average number of log records per block in your current system) the size of your current specification. A larger value increases the coupling facility storage requirement without corresponding performance improvements. A lower value starts to degrade performance.

5.5.4.2 Structure Size for DFHSHUNT

It is very difficult to predict the frequency and block size for records written to DFHSHUNT. Generally, the DFHSHUNT log stream need be only a fraction of the size of the DFHLOG log stream. Use the formula shown in Figure 6 to calculate coupling facility space for DFHSHUNT.

```
INITSIZE = (150 * LOGSNUM) + 310
```

Figure 6. Formula to Calculate Secondary System Log Structure INITSIZE

5.5.4.3 Structure Size for Forward Recovery Logs

You can merge the forward recovery logs written by many CICS regions onto the same log stream. You can also use the same log stream for forward recovery data for multiple data sets.

AVGBUFSIZE for Forward Recovery Logs: First, calculate the AVGBUFSIZE using the formula shown in Figure 7.

```
AVGBUFSIZE = (bytespersec / writespersec) + 36
```

Figure 7. Formula to Calculate Average Buffer Size for Forward Recovery Logs

Calculate bytespersec as:

```
bytespersec = (N1 * Wr1 * (D1 + rechdr) +.. (Nn * Wrn * (Dn + rechdr)))
```

Calculate writespersec as:

```
writespersec = lesser of 25 or (N1 + ... + Nn)
```

rechdr is the record header length of each record. Forward recovery logs should only contain file control records. These can be data set name records, which consist of a 204-byte record header, and no further data, in which case rechdr is

204. Otherwise the records are WRITE ADD, WRITE ADD COMPLETE, or WRITE ADD DELETE records. In this case, rechdr is 84 and is followed by the record key, and the record data (including its key).

When you calculate bytespersec and writespersec:

N1, N2, ... Nn are the number of transactions per second writing to each data set.

Wr1, Wr2, ... Wrn are the number of write requests per transaction.

D1, D2, ... Dn are the average record lengths for each data set.

INITSIZE for Forward Recovery Logs: You can calculate INITSIZE, using the same formula as for the primary system log (see Figure 5 on page 33). When using this formula for forward recovery logs, the number of entries (no. entries) is calculated differently:

```
no. entries = writespersec * 12.5
```

writespersec has the same value as for calculating the AVGBUFSIZE for forward recovery logs.

5.5.4.4 Structure Size for User Journals and Autojournals

Structure sizes for user journals and autojournals are calculated in a similar way to those for forward recovery logs.

AVGBUFSIZE for User Journals and Autojournals: First, calculate the AVGBUFSIZE, using the formula shown in Figure 7 on page 34. Calculate bytespersec as:

```
bytespersec = (N1 * Wr1 * (D1 + rechdr) +.. (Nn * Wrn * (Dn + rechdr)))
```

Calculate writespersec as:

```
writespersec = lesser of 25 or ((N1 * Wa1) + ... + (Nn * Wan))
```

rechdr is the record header length of each record. Autojournal records are issued from file control. The records can be data set name records, which consist of a 204-byte record header, and no further data, in which case rechdr is 204. Otherwise the records are READ ONLY, READ UPDATE, WRITE UPDATE, WRITE ADD, or WRITE ADD COMPLETE records. In this case, rechdr is 84 and is followed by the file control record itself. User journal records consist of a 68-byte record header, followed by the user prefix, and the user data.

When you calculate bytespersec and writespersec:

N1, N2, ... Nn are the number of transactions per second writing to each data set.

Wa1, Wa2, ... Wan are the number of wait requests per transaction.

Wr1, Wr2, ... Wrn are the number of write requests per transaction.

D1, D2, ... Dn are the average record lengths for each data set.

INITSIZE for User Journals and Autojournals: You can calculate INITSIZE, using the same formula as for the primary system log (see Figure 5 on page 33). When using this formula for user journals and autojournals, the number of entries (no. entries) is calculated differently:

```
no. entries = writespersec * 12.5
```

writespersec has the same value as for calculating the AVGBUFSIZE for user journals and autojournals.

5.5.5 Staging Data Sets

Define each staging data set to be at least the same size as the log stream's share of the coupling facility, but round the average block size up to 4 K.

For example, the staging data set size corresponding to the basic coupling facility space requirement for each CICS system log stream (DFHLOG) can be calculated by the following formula:

```
                                          avgbufsize (rounded up to 4 K)
staging data set size = no. entries *  ----------------------------
                                                    4096
```

5.5.6 Log Stream Data Sets

Log stream data set sizing can be done at a gross level by estimating the total daily output to the log stream from all of the CICS regions and using that as the data set size. You can adjust this size as needed, based on experience, to ensure that periodic jobs are run to delete the log tail before the 168 data set limit is reached. For additional details about CICS TS structure sizing, see the *CICS Migration Guide*, *CICS Installation Guide*, and *CICS System Definition Guide*.

Chapter 6. MVS/ESA Version 5 Release 2

MVS/ESA Version 5 Release 2 (MVS/ESA 5.2) continues to build on the S/390 Parallel Sysplex by bringing value to the single-system as well as the multisystem environment. In this chapter we introduce the functions that are relevant to VSAM RLS, most importantly the MVS system logger. We also discuss other functions that are important for operating a VSAM RLS environment in a Parallel Sysplex. The MVS system logger can be exploited by both system and subsystem components. The ARM and dynamic reconfiguration of the coupling facility improve availability and resource management.

For more documentation on these subjects, see the following publications:

- *MVS/ESA Planning: Workload Management*
- *MVS/ESA SP Version 5 Sysplex Migration Guide 5.1.0 Version 5.2*
- *MVS/ESA Setting Up a Sysplex*
- *MVS/ESA Programming: Assembler Services Guide*
- *MVS/ESA Programming: Assembler Services Reference*
- *MVS/ESA Initialization and Tuning Reference Version 5.2*
- *CICS Release Guide*

6.1 The MVS System Logger

The MVS system logger is an integrated MVS function designed to support system and subsystem components in a Parallel Sysplex. It implements a set of services that enables applications to write, browse, and delete log data. The MVS system logger is designed to exploit coupling facility technology.

You can use MVS system logger services to merge data from multiple instances of an application, including from different systems across a Parallel Sysplex. For example, suppose you are concurrently running multiple CICS AORs on different MVS system images in a Parallel Sysplex, and each AOR can update a common data set. It is important for your installation to maintain a common log of all updates to the data set from across the Parallel Sysplex, so that, should the data set be damaged, it can be restored from the backup copy. The MVS system logger′s ability to merge the log data from these AORs into a single place in real time is a key supporting component for VSAM RLS. The MVS system logger significantly reduces the complexity of managing multiple logs (which would otherwise be required) and allows a single-system view of the log data in a Parallel Sysplex environment.

Log data is written to a log stream, which is simply a collection of data in log blocks residing in a coupling facility list structure, on DASD, or on both. Figure 8 on page 38 gives an overview of how the MVS system logger operates.

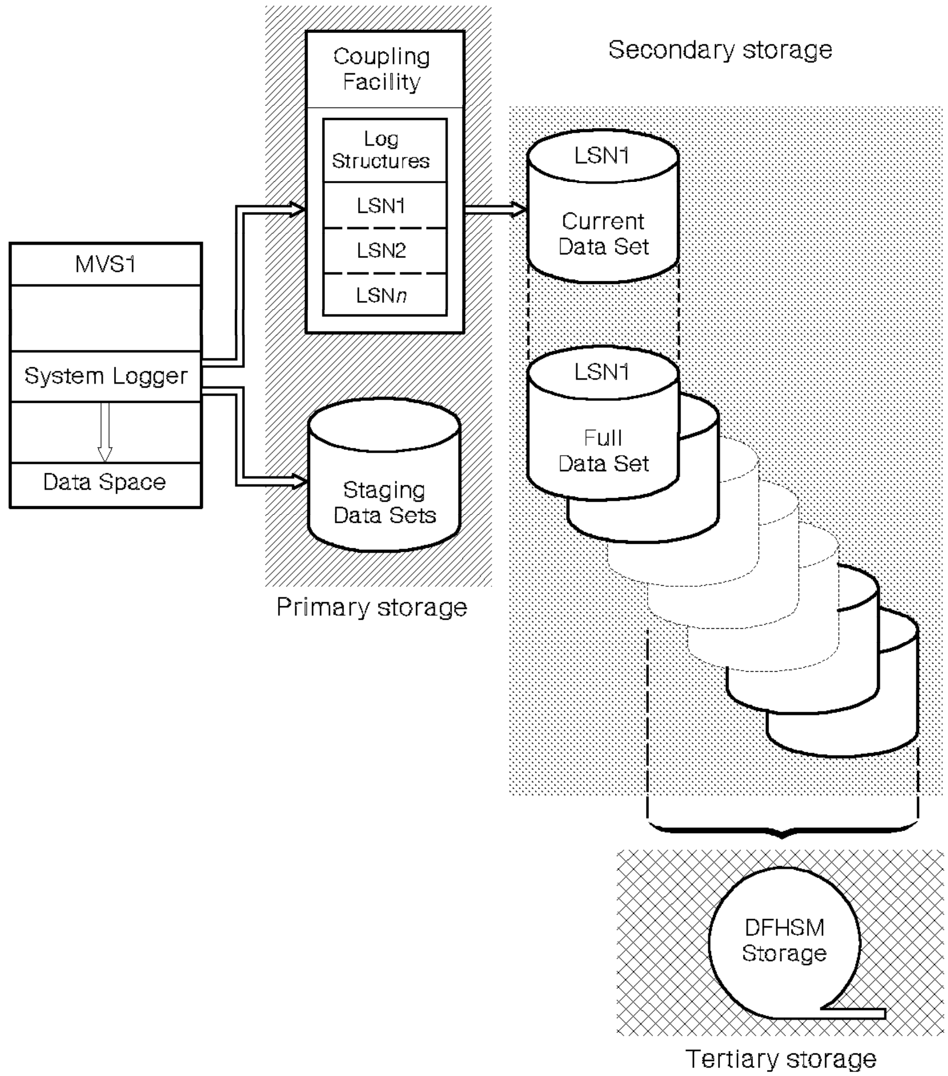

Figure 8. MVS System Logger Operation

When an application writes a log block to a log stream, the MVS system logger writes it first to a coupling facility list structure (primary storage). Each list structure can contain multiple log streams. Copies of the log blocks are written to a data space to protect against data loss if the coupling facility fails. Additional copies can be written to staging data sets if required, as explained in 5.4.2, "Staging Data Sets" on page 28. Over time the list structure fills. At a certain point (which you define), the MVS system logger moves the log block from the coupling facility structure to a log stream data set on DASD (secondary storage). The MVS system logger releases the storage in the list structure, so that the coupling facility space for the log stream can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent. All data in the coupling facility and in the log data sets on secondary storage is considered part of the log stream.

Redundant data must be removed from log streams periodically, so that the logger inventory entry for the stream does not fill up. The MVS system logger can keep track of up to 168 log stream data sets. Data can be deleted, or archived to another medium (tertiary storage).

6.1.1 Duplexing of Log Requests

To ensure maximum protection of log data, the MVS system logger supports the duplexing of log write requests to both the coupling facility and DASD. This duplexing capability provides failure isolation of committed log data, even when the coupling facility and a connected MVS are in the same failure domain (in the same CPC, or the coupling facility storage is volatile). The records are written temporarily to a staging data set associated with the log stream, until they are later offloaded asynchronously to permanent storage on the persistent log stream data sets. Offloading occurs when the HIGHOFFLOAD threshold of either the log stream or the staging data set has been reached.

You can specify whether duplexing to staging data sets is either conditional or unconditional, depending on whether or not the coupling facility is in an independent failure domain. If conditional, the MVS system logger chooses either the data space option or a staging data set; if unconditional, duplexing to a staging data set is forced. These options are illustrated in Table 5.

Table 5. Storage Used for Duplexing Log Stream Data Blocks

DUPLEX Option	DUPLEXMODE Option	Coupling Facility	Duplexed Data Storage
NO	Not applicable	Not applicable	Data space
YES	Conditional	Failure independent	Data space
YES	Conditional	Not failure independent	Staging data set
YES	Unconditional	Not applicable	Staging data set

6.1.2 CICS and the MVS System Logger

The CICS log manager improves the management of the CICS system log, forward recovery logs, autojournals, and user journals. Using services provided by the MVS system logger, the CICS log manager supports the CICS system log, forward recovery logs, autojournals, the log of logs, and user journals. The forward recovery logs, autojournals, user journals, and log of logs (a special user journal) are collectively called *general logs*.

The CICS log manager uses the services of the MVS system logger to enhance CICS logging in line with the needs of the Parallel Sysplex environment. In particular, it provides online merging of general log streams from different CICS regions, which may be on different MVS system images in a Parallel Sysplex.

The MVS system logger also improves media management and archiving and log data availability through direct and sequential access to log records.

The CICS log manager, with the MVS system logger, improves management of the system log and dynamic log data (all of which is written to the system log stream) by avoiding log wraparound and automatically deleting obsolete log data of completed UOWs.

All CICS logs (except for user journals defined as type SMF or DUMMY) are written to MVS system logger log streams. User journals of type SMF are written to the MVS SMF log data set. See 8.3.1, "The CICS Log Manager" on page 62 for more information about the CICS log manager.

6.1.3 Planning Items

Chapter 5, "The Coupling Facility" on page 19 describes how to size the structure that contains the CICS system log and Chapter 8, "CICS Transaction Server for OS/390 Version 1 Release 1" on page 57 describes how to define the system log for a CICS region.

6.2 The Automatic Restart Manager

ARM is a recovery function that provides improved availability for batch jobs and started tasks (STCs) by automatically restarting them after they unexpectedly terminate. In a Parallel Sysplex ARM provides additional benefit through its ability to restart registered clients on another MVS system image in the Parallel Sysplex if the MVS system image they were originally using fails.

Transaction managers, resource managers, and restartable subsystems can be restarted automatically. Any system requiring automatic restart must register with ARM. Systems affected by a failure are usually restarted on the same MVS system image, or on a different one if the MVS system image itself has failed. Thus the cost of computing, productivity, and availability are improved because shared resources and transactions in progress can be recovered and lost function and services restored.

The following points describe the main assumptions in the design of ARM:

- ARM provides job and STC recovery. Transaction or database recovery is the responsibility of the restarted applications.
- ARM does not start the applications initially (first IPL or subsequent IPLs after failures). This is the function of automation or production control products. Interface points are provided through exits, the event notification facility (ENF), and macros.
- The MVS system image or Parallel Sysplex must have sufficient spare capacity to guarantee a successful restart.
- A registered element (job or STC) of ARM that terminates unexpectedly is restarted on the same MVS system image.
- Elements that are on an MVS system image that fails are restarted on another MVS system image. You can group related elements and ensure that they are restarted on the same MVS system image.
- The intended exploiters of ARM function are the jobs and STCs of certain strategic transaction and resource managers. These include:
  - CICS/ESA
  - CICSPlex SM
  - DB2
  - IMS/ESA Transaction Manager (IMS TM)
  - IMS/ESA Database Manager (IMS DB)
  - ACF/VTAM

ARM minimizes failure impact by doing fast restarts without operator intervention. If programs are start-order dependent, ARM ensures that the order dependency is honored. The chance of work being restarted on systems lacking the necessary capacity is reduced as the best system is chosen according to statistics returned to ARM by the workload manager.

The main benefits of ARM are:

- Eliminates the need for operator-initiated restarts or restarts by other automatic packages, thereby:
  - Improving emergency restart times
  - Reducing errors
  - Reducing complexity
- Provides cross-system restart capability. ARM ensures that the workload is restarted on the MVS system image with spare capacity, by working with the MVS workload manager.
- Enables all elements within a restart group to be restarted in parallel. Restart levels are used to ensure the correct starting sequence of dependent or related subsystems. ARM support is part of the overall task of planning for and installing MVS Parallel Sysplex support. For more information about MVS automatic restart management, see *MVS/ESA Setting Up a Sysplex*.

Note: You cannot use ARM for CICS regions running with XRF. ARM is available only to non-XRF CICS regions.

MVS automatic restart management is available only to those MVS subsystem that register with ARM. CICS/ESA 4.1 and CICS TS regions register with ARM automatically as part of CICS system initialization. If a CICS region fails before it is registered for the first time with ARM, it will not be restarted. After the CICS region has registered, it is restarted by ARM according to a predefined policy for the workload.

You can find more about ARM processing for CICS in Chapter 7, “Automatic Restart Management,” of the *CICS Recovery and Restart Guide*.

Implementing ARM support for CICS generally involves the following steps:

1. Ensure that the MVS system images available for automatic restarts have access to the databases, logs, and program libraries required for your workload.

   You must ensure that you define the couple data sets required for ARM and that they are online and active before you start any CICS region for which you want ARM support. ARM registration fails if the couple data sets are not available at CICS startup. In this situation, CICS assumes that you do not want ARM support, and initialization continues.

2. Define ARM policies for your CICS regions.

6.3 Workload Management

To reduce the complexity of managing a Parallel Sysplex, MVS workload management provides dynamic Parallel Sysplex-wide management of system resources. MVS workload management is the combined cooperation of various subsystems (such as CICS, IMS, and VTAM) with the MVS/ESA Version 5 WLM component. The WLM manages Parallel Sysplex workload distribution, balances workloads, and distributes resources to competing workloads according to an installation-defined service policy. This type of workload management is different from the way workloads were managed in earlier MVS releases. The emphasis is on defining performance goals for work and having MVS and the subsystems adapt to meet the goals.

Workload management optimizes service by automatically routing transactions to the most appropriate CICS address space in the CICSplex, improving throughput, availability, and service levels. It also balances work between eligible CICS regions, provides workload separation, and handles affinities between transactions.

In a CICS environment, workload balancing is performed by the dynamic routing program. CICSPlex SM, which is part of CICS TS, provides a dynamic routing program, EYU9XLOP. You can specify that CICSPlex SM is to use either a goal mode algorithm or a queue mode algorithm for routing. If you specify that CICSPlex SM is to operate in goal mode, WLM provides the service class goal to CICSPlex SM, and CICSPlex SM uses the goal information to help decide where to route the transaction for processing to meet the goal. If you specify that CICSPlex SM is to operate in queue mode, CICSPlex SM does not use the service class goal but instead uses a *join shortest queue* algorithm.

WLM recognizes that the TOR and the AOR on one or more systems are involved in processing CICS transactions. Using RMF Monitor I reports, you can get information about CICS response times and any execution delays experienced by a service class for a single system or the Parallel Sysplex.

The new workload management concept addresses the need for distributing and balancing workloads that compete for resources, such as CPU and storage, by introducing a new concept in workload control: goal-oriented workload management.

Work is defined in terms of business service level goals, taking into account the response time needs of customer business applications, together with the relative importance of all work to be managed.

The new workload management idea is to define goals for work, using business terms, and to let the system decide how much work should run and how much resource to supply to meet the goals. For example, with a CICS environment in a Parallel Sysplex configuration, work is distributed among the different CICS AORs (on multiple MVS system images) depending on how the goals are achieved.

To ease implementation and provide fallback capability, the WLM operates in two modes:

Compatibility mode
: Uses the existing familiar system definitions. Reporting is enhanced to retrieve further information that can be useful in preparing the new WLM service definitions that are used in goal mode operation.

Goal mode
: Uses the new goal-oriented WLM service definitions

For more information about CICS and the WLM, see *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*.

WLM activation requires the following steps:

1. Identify and remove affinities.

The IBM CICS Transaction Affinities Utility MVS/ESA, program number 5696-582, has been included in CICS TS. You can use it to identify affinities and define them to CICSPlex SM.

2. Define policies to MVS.

   The MVS policies for the CICS transaction have to be defined.

3. Customize CICSPlex SM.

   The workload component of the CICSPlex SM product or a dynamic transaction routing exit with equivalent functions has to be customized to manage the dynamic routing of the CICS transactions to the CICS region that can guarantee the defined objectives.

6.4 Dynamic Reconfiguration of Coupling Facility Structures

Dynamic reconfiguration of coupling facility structures makes it possible to dynamically change the size of a coupling facility structure. Changes to a coupling facility structure may be required for such reasons as growth in the application data, variable use of the structure during different business periods, or growth in the number of processors or applications competing for coupling facility resources.

The size of a coupling facility structure includes the total amount of storage required by both the application and the coupling facility itself for structure control information and is specified in 1 K blocks. Determining the correct size requires an understanding of the application′s use of the structure and the characteristics of the workload using the structure. The application provides one or more formulae to enable you to estimate an appropriate size for a structure; use RMF reports to monitor the structure′s utilization.

The size of a structure is specified in a CFRM policy. The size that you request in the CFRM policy with the SIZE specification is the maximum structure size. In MVS/ESA 5.2 and OS/390, you can change the structure size with the ALTER function. ALTER enables you to vary the structure size between an initial size (specified by INITSIZE in the CRFM policy) and the maximum size.

Dynamic reconfiguration makes it easier to alter structure sizes. You must, however, size your structures as accurately as possible to avoid creating performance bottlenecks or wasting storage in the coupling facility.

Chapter 5, “The Coupling Facility” on page 19 describes how to define the coupling facility structures.

Chapter 7. DFSMS/MVS Version 1 Release 3

Planning for and installing VSAM RLS requires coordination with system hardware and software departments. In this chapter we describe the tasks that the storage administrator must carry out with the MVS systems programmer and the CICS systems programmer to prepare to use VSAM RLS. We make the assumption that you are already running a Parallel Sysplex environment and SMS is active. We describe the following:

- Applications and VSAM RLS
- Hardware and system connectivity requirements
- Preparing for VSAM RLS
- Activating VSAM RLS

See *DFSMS/MVS Version 1 Release 3 Planning for Installation*, *DFSMS/MVS V1 R3 DFSMSdfp Storage Administration Reference*, and *DFSMS/MVS 1.3 Using Data Sets* for a more detailed discussion of the tasks to prepare for VSAM RLS.

7.1 Applications and VSAM RLS

In this section we describe:

- Who can use VSAM RLS
- Processing restrictions
- Batch sharing and RLS
- Implications for data integrity

Note: Because the CICS recoverable files function provides transactional recovery for applications, **VSAM RLS is expected to be used primarily by CICS applications**.

7.1.1 Who Can Use VSAM RLS?

Applications using VSAM RLS benefit from increased data availability inherent in a shared environment that has read integrity and record-level locking (as opposed to control interval (CI) locking).

RLS is a mode of access to VSAM spheres; it is not an attribute of a sphere. RLS is an access option interpreted at open time. The option is selected by specifying either a new JCL parameter (RLS) or MACRF=RLS in the ACB. The MACRF=RLS option is mutually exclusive with MACRF=NSR (nonshared resources), LSR (local shared resources), and GSR (global shared resources) options. In other words, if a VSAM data set is open by any VSAM RLS application (CICS or batch application running in RLS mode), no other applications can gain access to the data set by using NSR, LSR, or GSR buffer management protocols.

NSR, LSR, and GSR protocols use the SHAREOPTIONS parameter values to determine the level of cross-region and cross-system sharing. These parameter values are irrelevant, and VSAM ignores them when accessing the data in RLS mode. With RLS, VSAM maintains a single, logical VSAM control block structure across all MVS images, whereas with NSR, LSR, and GSR, each system has its own independent control block structure, and it is an application responsibility to maintain cross-system integrity.

RLS mode access is supported for key sequenced data sets (KSDS), entry sequenced data sets (ESDS), relative record data sets (RRDS), and variable length relative record data sets (VRRDS). RLS mode access is supported for data sets accessed through VSAM alternate indexes. A direct open of an alternate index is not supported. RLS mode is not supported for CI access.

Your applications should fall into one of the following categories:

- VSAM RLS-tolerant. Applications in this category run correctly in a multiupdate environment when RLS is specified in the JCL or MACRF=RLS is specified on the ACB. The RLS JCL parameter allows batch read programs to use RLS without requiring a recompile of the program, provided they can use IBM Language Environment for MVS and VM, Version 1 Release 5, program number 5688-198 (see 9.4.2, "Batch Language Support" on page 82 for more information about language support). For batch update programs running against nonrecoverable VSAM RLS spheres, you may be able to modify the allocation from DISP=OLD to DISP=SHR.
- VSAM RLS-exploiting. The application recognizes when a VSAM data set can be shared at the record level and uses VSAM RLS functions to access the data. CICS applications are exploiting applications.
- VSAM RLS-intolerant. Applications in this category use facilities not supported by VSAM RLS, access VSAM internal data structures, or are incompatible with the functions of VSAM RLS. For example, the application might perform CI access against the data set.

Before changing an application that currently uses LSR or NSR access to use RLS access, consider the following:

- The program might not be coded to function in a multiupdate environment. For shared access, the application must wait for an exclusive record lock if another user has the record and is subject to deadlock or timeout return codes. The application might require modification to change allocations from DISP=OLD to DISP=SHR.
- The use of no read integrity (NRI) for read access gives the application the same locking rules with RLS as it has with NSR. The use of consistent read (CR) can result in the application waiting for the availability of a record and could result in a deadlock, timeout, or a retained lock condition with other applications.
- An explicit point is required to establish positioning at the beginning of a data set. The step is not required for single string applications using NSR.
- VSAM RLS ignores a UPAD exit. The program can be modified to use the new RLSWAIT exit.
- Control blocks other than the user ACB, RPL, and EXLST are not available to the application.
- VSAM open processing and record management can give a return code 16, which cannot happen with NSR.
- VSAM RLS also introduces new error codes and processing restrictions.

7.1.2 Processing Restrictions

Use of VSAM RLS introduces the following processing restrictions:

- VSAM RLS mode processing cannot be used for:
  - VSAM linear data sets
  - Key range data sets
  - Extended format KSDSs larger than 4 GB
  - Temporary data sets
  - VSAM volume data sets (VVDS)
  - Catalogs
  - The ISAM compatibility interface.
- A VSAM KSDS defined for RLS cannot have the index set embedded (with the IMBED option). If you want to use these data sets in RLS mode:
  - Redefine the data sets without the IMBED option.
  - Copy the old data sets to the new using the IDCAMS REPRO function.

  Reconsider the use of the REPLICATE option. Although REPLICATE is supported by RLS, it does not provide any performance benefit and can be omitted without penalty. Not using replication should save a little space on DASD.
- VSAM RLS does not allow addressed access (RPL OPTCD=ADR) to a VSAM KSDS or control interval access (RPL OPTCD=CNV) to any data set organization.
- Individual components of a VSAM cluster cannot be opened for VSAM RLS mode processing. A direct open of an alternate index is not supported.
- Data sets opened specifically for system use cannot take advantage of VSAM RLS.
- A checkpoint on a VSAM data set cannot be taken while a data set is open for RLS mode access in the address space.
- Opening a data set in RLS mode does not establish implicit positioning. An explicit POINT or GET NSP is required.
- The GETIX and PUTIX macros are not supported.
- VSAM RLS does not support Hiperbatch.
- Some parameters specified on the ACB are either ignored or not supported. These include buffer-related parameters BSTRNO, BUFND, BUFNI, BUFSP, and SHRPOOL, which are ignored. Data identifiers DDN and DSN are also ignored. DFR is ignored, and NDF is assumed for direct requests that do not specify NSP. CFX is ignored, and NFX is assumed. Control blocks in common (CBIC) and user buffering (UBF) are not supported. RMODE31 is assumed.

Note: VSAM RLS does not apply to IMS databases that are stored in VSAM data sets. IMS provides data sharing based on local shared resources (LSR) with the coupling facility cross-invalidate capability.

7.1.3 Batch Sharing with CICS

In addition to the data sharing across CICS applications, VSAM RLS enables **read**-with-integrity sharing by batch jobs. Batch jobs can share recoverable files while they are being modified by CICS applications. This sharing is possible because VSAM provides the record locking and buffer coherency functions across CICS and batch.

Note: Because VSAM RLS provides neither transactional recovery function nor logging of data set changes, it does not allow a batch job to open a recoverable data set for output.

7.1.4 Implications for Data Integrity

To help ensure data integrity in this expanded sharing environment, you need to address each of the following issues:

- Establish procedures for non-IBM products and user programs that can modify data on a volume without using a VSAM application program interface, DFSMSdss, or DFSMShsm. If these programs run at the time VSAM RLS data exists in coupling facility cache storage, they can compromise the integrity of the cached data.

  Note: Operations such as DISP=OLD serialization, VSAM enqueues, or VTOC serialization do not affect data in the coupling facility cache storage.

  Prior to running these programs, vary the volumes to be modified offline to each system in the Parallel Sysplex, or have them quiesced to the coupling facility by entering the command:

  ```
  VARY SMS,CFVOL(volid),QUIESCE
  ```

- ICKDSF REFORMAT is not supported for SMS-managed volumes, except to rebuild a volume label using the VTOCPTR option, without changing the VOLSER. Other uses of REFORMAT cause unpredictable results.
- Certain DFSMSdss commands require an operator to issue SMS coupling facility commands before the DFSMSdss operation.
- The use of volume reserves is inconsistent with the availability objectives of sysplex management. Consider converting your reserves to enqueues, especially on volumes where the sharing control data sets (SHCDSs) will reside. You can use GRS to set this up without changing source code for the applications that use the volumes.

7.2 Additional VSAM RLS Planning Documentation

See the following publications for additional information about VSAM RLS:

- *System/390 MVS Sysplex Hardware and Software Migration*
- *System/390 MVS Sysplex Application Migration*
- *DFSMS/MVS V1R3 DFSMSdfp Storage Administration Reference*
- *DFSMS/MVS V1R3 Using Data Sets*
- *DFSMS/MVS V1R3 Macro Instructions for Data Sets*
- *DFSMS/MVS V1R3 DFSMSdfp Advanced Services*
- *DFSMS/MVS V1R3 Access Method Services for ICF*
- *DFSMS/MVS V1R3 DFSMShsm Storage Administration Guide*

- *DFSMS/MVS V1R3 DFSMSdss Storage Administration Guide*
- *CICS Release Guide*
- *CICS Migration Guide*

7.3 Hardware and System Connectivity Requirements

You must have the appropriate hardware in place and global connectivity across the Parallel Sysplex.

7.3.1 Hardware

S/390 coupled systems provide the hardware technology for RLS. VSAM RLS exploits both the coupling facility locking and caching functions of this technology. This multisystem record sharing is optimized for the CICS OLTP environment, but the support also applies to other environments, such as batch, where shared access from multiple address spaces or multiple systems is required.

VSAM RLS applications cannot be run without the coupling facility and the correct level of MVS system software. You must have at least coupling facility level 2 microcode installed in your coupling facility before using VSAM RLS.

You must have at least one coupling facility connected to all systems capable of VSAM RLS within the Parallel Sysplex. The coupling facility should be large enough to contain both the caching and locking structures. It must also have enough surplus space to allow the structure to be modified or rebuilt.

For multiple coupling facilities, select one coupling facility with global connectivity to contain the master lock structure. You can attach additional coupling facilities to a subset of the systems, but system administrators and users must then ensure that their jobs run on a processor with the required connectivity.

Refer to Chapter 5, “The Coupling Facility” on page 19 for a detailed description of setting up your coupling facility.

7.3.2 Structure Global Connectivity

You must have same systems connectivity for coupling facility cache structures, the lock structure, and storage groups, so ensure that jobs running in the Parallel Sysplex have access to data in both the coupling facilities and storage groups. The lock structure must have global connectivity to all systems in the Parallel Sysplex.

Figure 9 on page 50 shows how connectivity among systems and storage groups matches connectivity among systems and coupling facilities.

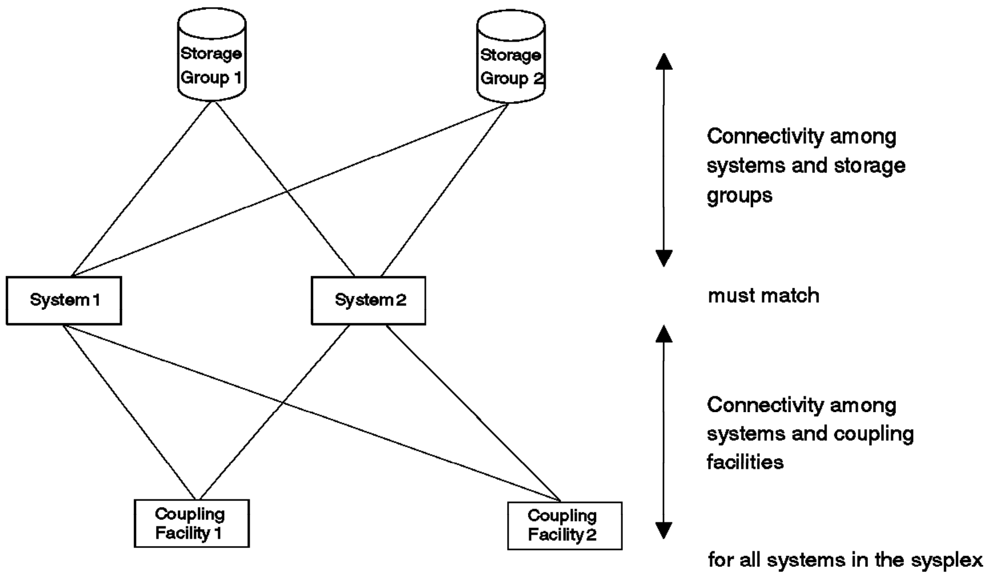

Figure 9. Global Connectivity in a Parallel Sysplex

Figure 10 illustrates the problems that can occur if you do not have global connectivity across the Parallel Sysplex.

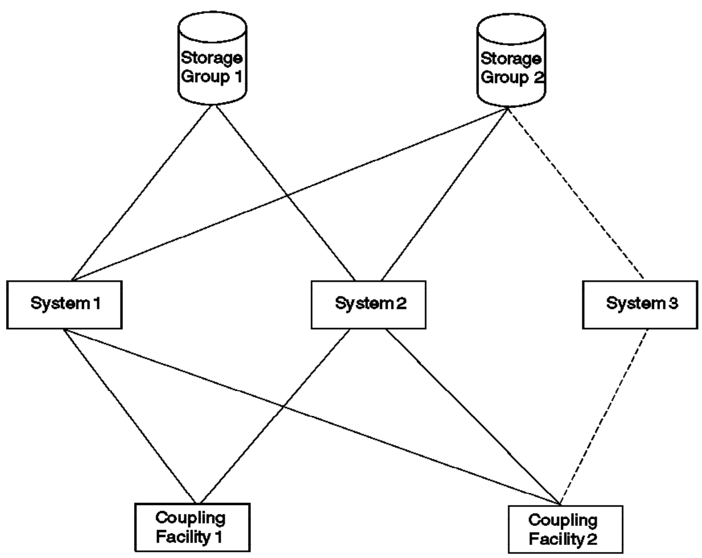

Figure 10. Insufficient Connectivity in a Parallel Sysplex

In Figure 10, System 3 is connected to both Storage Group 2 and Coupling Facility 2. This meets the minimum connectivity requirements but can still cause problems. Because Storage Group 2 data can be placed in Coupling Facility 1, jobs in System 3 could fail if they attempt to access data sets in Storage Group 2 that have been assigned to Coupling Facility 1. You can avoid such a scenario by ensuring that, for all systems in the Parallel Sysplex, connectivity among systems and storage groups matches connectivity among systems and coupling facilities.

7.4 Preparing for VSAM RLS

The following steps prepare your system for VSAM RLS:

1. Determine coexistence requirements
2. Define coupling facility cache and lock structures
3. Define coupling facility cache structures in the SMS base configuration
4. Define storage classes for VSAM RLS
5. Define the SHCDS
6. Modify SYS1.PARMLIB member IGDSMSxx
7. Plan for fallback
8. Protect VSAM record-level sharing functions

7.4.1 Determine Coexistence Requirements

You can share an SMS configuration between systems running DFSMS/MVS 1.3, DFSMS/MVS 1.2, DFSMS/MVS 1.1, or MVS/DFP 3.3 within a sysplex. You must apply toleration PTFs to all pre-DFSMS/MVS 1.3 systems so that they do not conflict with RLS access. If you convert your SMS configuration to SMS 32-name mode under DFSMS/MVS 1.3, however, toleration support is not provided. Therefore do not convert to SMS 32-name mode until all systems in the sysplex have been converted to DFSMS/MVS 1.3 and you are sure that you will not have to fall back to an earlier release of software on any of the systems.

You can only define cache sets and other RLS-related SMS functions through Interactive Storage Management Facility (ISMF) on a DFSMS/MVS 1.3 system. An SMS source control data set (SCDS) altered by ISMF on a DFSMS/MVS 1.3 system can be activated from any of the systems that share the configuration.

You can share a catalog between DFSMS/MVS 1.3 and earlier releases. LISTCAT output from DFSMS/MVS 1.3 systems, or down-level systems with toleration maintenance, indicates whether RLS information is present in the catalog for SMS-managed data sets.

For more information about coexistence issues, see “VSAM Record-Level Sharing” in Chapter 2 of *DFSMS/MVS Version 1 Release 3 Planning for Installation*.

7.4.2 Define Coupling Facility Cache and Lock Structures

Coupling facility cache structures must be defined to MVS/SP and in the SMS base configuration. The coupling facility lock structure is defined to MVS/SP with the required name of IGWLOCK00.

You must:

- Determine the coupling facility cache structure size.
- Determine the coupling facility lock structure size.
- Define the coupling facility cache structures.
- Define the coupling facility lock structure.

For more detailed information about defining the coupling facility structures, refer to Chapter 5, “The Coupling Facility” on page 19.

7.4.3 Define Coupling Facility Cache Structures in the SMS Base Configuration

In order for DFSMSdfp to use the coupling facility for VSAM RLS, after you define one or more coupling facility cache structures to MVS, you must also add them to the SMS base configuration.

To add coupling facility cache structures to the base configuration, you associate them with a cache set name. This cache set name is also specified in one or more storage class definitions. When a storage class associated with a data set contains a cache set name, the data set becomes eligible for VSAM RLS and can be placed in one of the coupling facility cache structures associated with the cache set. The system selects the best cache structure in which to place the data set.

See *CICS and VSAM Record Level Sharing: Implementation Guide* for more detailed information about defining cache structures to SMS.

7.4.4 Define Storage Classes for VSAM RLS

You must assign the cache set names defined in the base configuration to a storage class so that data sets associated with that storage class can be eligible for VSAM RLS and use coupling facility cache structures.

Note: In a JES3 environment, define cache set names only in those SMS storage classes that are used by data sets opened for VSAM RLS processing. When you define a cache set name in a storage class, any job accessing a data set associated with that storage class is scheduled on a VSAM RLS-capable system. If all storage classes have cache set names defined for them, all jobs accessing SMS-managed data sets are scheduled to VSAM-RLS-capable systems. This could cause a workload imbalance between those systems and down-level systems.

For information about changing SMS storage classes to use the coupling facility, refer to the *CICS and VSAM Record Level Sharing: Implementation Guide*.

7.4.5 Define Sharing Control Data Sets

Sharing control is a key element in maintaining data integrity in a shared environment. Because persistent record locks are maintained in the coupling facility, several new classes of failure could occur, such as a Parallel Sysplex, system, or SMSVSAM address space restart or a coupling facility lock structure failure. The SHCDS is designed to contain the information required for DFSMS/MVS to continue processing with a minimum of unavailable data and no corruption of data when failures occur.

The SHCDS acts as a log for sharing support. It is a logically partitioned linear data set that can be defined with secondary extents, although all extents for each data set must be on the same volume.

An SHCDS contains the following information:

- Name of the coupling facility lock structure in use
- System status for each system or failed system instance
- The time that the system failed
- List of subsystems and their status
- List of open data sets using the coupling facility

- List of data sets with unbound locks
- List of data sets in permit non-RLS state

Give careful consideration to the allocation and maintenance of your SHCDSs:

- Allocate SHCDSs so that the number of active and spare data sets ensures that the data is always duplexed.

  If a permanent I/O error occurs for an active SHCDS, or if an SHCDS becomes inaccessible from one or more systems, it is automatically replaced by one of the spare SHCDSs. When a system is forced to run with only one SHCDS, it issues a message requesting that you add another active SHCDS. If any system does not have access to an SHCDS, all opens for VSAM RLS processing are prevented on that system until an SHCDS becomes available.

  SMSVSAM will not start unless you have two active SHCDSs and at least one spare SHCDS.

- Place the SHCDSs on volumes with global connectivity. VSAM RLS processing is available only on those systems that currently have access to the active SHCDS.
- Ensure that the space allocation for active and spare SHCDSs is the same.
- Use the

  ```
  VARY SMS,SHCDS
  ```

  and

  ```
  DISPLAY SMS,SHCDS
  ```

  commands to maintain your SHCDSs.

Note: The information in SHCDSs is continuously updated. Therefore, backup and restore procedures for SHCDSs are ineffective.

7.4.6 Modify SYS1.PARMLIB IGDSMSxx Member

The IGDSMSxx member of SYS1.PARMLIB includes five new parameters to support the coupling facility and VSAM RLS processing:

- DEADLOCK_DETECTION, which specifies the interval for detecting deadlocks between systems
- SMF_TIME, which is used to align the interval-type SMF 42 records for DFSMS with the SMF_TIME interval
- CF_TIME, which is used to align creation of all CF-related SMF 42 subtypes
- RLSINIT, which is used to activate the SMSVSAM address space
- RLS_MAX_POOL_SIZE, which specifies the maximum size in megabytes of the SMSVSAM local buffer pool.

You can modify these parameters at any time during VSAM RLS processing.

See Chapter 14, “Administering VSAM Record-Level Sharing,” in the *DFSMS/MVS V1 R3 DFSMSdfp Storage Administration Reference* for information about changing these parameters.

7.4.7 Plan for Fallback

If you have to fall back from DFSMS/MVS 1.3 to an earlier release, you must delete RLS information from the catalog. Use the SHCDS CRFESET command on a DFSMS/MVS 1.3 system to delete RLS information from the catalog.

When returning to a previous level of DFSMS/MVS, CICS installations must revert to using CICS file control table (FCT) definitions even if the level of CICS is not changing. For additional CICS and CICSVR fallback considerations, refer to the *CICS Recovery and Restart Guide*.

For more information about fallback planning, see "VSAM Record-Level Sharing" in Chapter 2 of *DFSMS/MVS Version 1 Release 3 Planning for Installation*, and the *DFSMS/MVS 1.3 DFSMSdfp Storage Administration Reference*.

7.4.8 Protect VSAM Record-Level Sharing Functions

You have to establish the following authorizations to restrict access to certain VSAM RLS capabilities:

- Applications, such as CICS, using the VSAM RLS commit protocol interface, are capable of releasing locks that protect uncommitted RLS updates. You have to restrict the use of this interface through the RACF SUBSYSNM class. Use the SUBSYSNM class to authorize the CICS subsystems that are allowed to use the VSAM RLS commit protocols. Read access is required.
- The IDCAMS SHCDS command provides recovery functions that can also release locks protecting uncommitted RLS updates. Its functions result in the special handling of transaction backouts. Use the STGADMIN.IGWSHCDS.REPAIR FACILITY class to protect these SHCDS functions.
- To issue the V SMS,SMSVSAM,FALLBACK command to fall back from VSAM RLS processing, you must be authorized to the STGADMIN.VSAMRLS.FALLBACK FACILITY class.

  Refer to *DFSMS/MVS V1R3 Access Method Services for ICF* for further details.

7.5 Activate VSAM RLS

Once you have completed the hardware and software setup for RLS, you can IPL your MVS system to start SMSVSAM. SMSVSAM is a new system address space that is started automatically when you IPL an MVS system containing DFSMS/MVS 1.3 if you specify RLSINIT(YES) in IGDSMSxx member of SYS1.PARMLIB. The default is RLSINIT(NO).

SMSVSAM holds most of the new VSAM RLS code, as well as the media manager, channel programs, control blocks related to the locks held for records in the coupling facility, and SMF statistics for VSAM RLS activities.

VSAM RLS processing will be available after IPL as long as the following requirements are met:

- All systems are running as part of a Parallel Sysplex.
- SMSVSAM is active.
- At least one existing coupling facility cache structure is defined to MVS and to the SMS base configuration.

- At least two SHCDSs and one spare SHCDS have been activated. If a subset of the cache structures is not available, some data sets might not be accessible.
- The coupling facility lock structure, IGWLOCK00, is available.

Note: If VSAM RLS processing is not enabled, all attempts to open a data set where VSAM RLS is specified on the ACB or the JCL fail.

Chapter 8. CICS Transaction Server for OS/390 Version 1 Release 1

In this chapter we highlight the planning items for the migration to CICS TS, with emphasis on the implementation of VSAM RLS. We assume that you have planned for all of the prerequisite items, such as installing the Parallel Sysplex with the necessary base software, including CICS TS base installation. See Chapter 11, "Planning Checklist for VSAM Record-Level Sharing" on page 91 for a complete overview of migration tasks.

CICS TS provides an easy to use S/390 client/server package. It includes CICS server, CICS client, and CICS management function in a single solution package for a single price. CICS TS contains both exclusive and nonexclusive elements. The base element, the successor to CICS/ESA 4.1, is exclusive (in other words, it is available only as part of CICS TS). This base element includes features and products available with previous CICS versions:

- CICS Web Interface
- Open Network Computing Remote Procedure Call (ONC RPC)
- IBM CICS Transaction Affinities Utility MVS/ESA.

The nonexclusive elements of the product, also available as separate products, are:

- CICS Distributed Data Management (DDM)
- CICS Application Migration Aid Version 1 Release 1
- CICSPlex SM Version 1 Release 2
- CICS Clients Version 2
- Transaction Server for OS/2 Warp Version 4 (90-day evaluation copy)

8.1 CICSplex in a Parallel Sysplex

In this section we describe the major elements to be considered for setting up a CICSplex in a Parallel Sysplex environment. Implementing a CICSplex is not a prerequisite for CICS TS or VSAM RLS. However, a Parallel Sysplex is a prerequisite for CICS TS and VSAM RLS, and you should implement a CICSplex to gain the full benefits of your Parallel Sysplex.

A CICSplex consists of several CICS regions, usually connected in a configuration where each region performs a specific portion of the transaction workload. A CICSplex thus normally consists of:

- One or more TORs: The TORs handle all interactions with the connected terminals and control the associated sessions. Another major task of a TOR is to route the transactions to the available AORs using either static or dynamic routing.
- Multiple AORs: The AORs run the application programs to process a transaction they receive from a TOR.
- One or more FORs: The FOR allows you to share VSAM data sets among several AORs with integrity. The FOR owns the VSAM data sets and serializes access to the data. However, once you have fully converted to VSAM RLS, you no longer need an FOR.
- If you have to share other resources, such as temporary storage queues or transient data queues, you may need a QOR. You can use the new CICS TS

support for shared temporary queues instead of a QOR for nonrecoverable temporary storage queues.

You can find more information about CICSplexes n Chapter 1 of *Dynamic Transaction Routing in a CICSplex*.

8.1.1 Planning Items

In this section we give an overview of the tasks you should plan to set up a CICSplex in a Parallel Sysplex.

1. Decide on your naming convention.

   A sensible naming convention is required to support cloning of CICS regions as well as workload balancing and recovery management. Cloning means you have predefined several regions that all have the same characteristics and therefore share one set of startup parameters and resource definitions. Such a setup enables you to quickly start up an AOR should an increasing workload demand additional processing capacity. You can even automate this process by using the MVS workload manager.

   Chapter 3, "Planning Naming Conventions for CICS and Related Subsystems" of *OS/390 Parallel Sysplex Application Migration* recommends a naming convention that is based on the "C T G I" template, where:

   C identifies a logical collection of subsystems, for example a CICSplex, but it can also include non-CICS subsystems

   T identifies the type of a region, for example, *A* for AOR, *T* for TOR, *B* for DB2, and *C* for CMAS.

   G identifies a recovery group, for example, all subsystems that usually run on the same MVS image.

   I serves as an iteration identifier.

   Using the above example, a sysid of PTA1 identifies a TOR that belongs to CICSplex *P* (for example, production plex), and the TOR is usually started up on MVS-A. The chosen naming convention is used for subsystem identifiers, VTAM APPLIDs, and connection names.

2. Ensure that terminal IDs are unique within the CICSplex.

3. Start the CICS regions as started tasks and specify JOBNAME=*name*, where *name* is equal to the CICS APPLID.

4. Use the CICS APPLID to uniquely identify data sets owned by individual regions, for example, the global catalog.

   Use a generic identifier for data sets that are shared within the CICSplex, for example, the CICS system definition (CSD) file.

5. Consider implementing static routing

   If any of your CICS regions is being used as a combined TOR and AOR, consider splitting them into two regions. You can use static routing as an intermediate solution while planning the next step to dynamic routing.

6. Implement dynamic transaction routing (DTR)

   DTR was first introduced in CICS/ESA 3.3, and enhancements were made in CICS/ESA 4.1. You must implement DTR to exploit workload balancing efficiently. If your transactions have any form of affinity, it can constrain your use of DTR. For example, an intertransaction affinity exists if two transactions pass data to each other using techniques that force the transactions to run on the same CICS region and thus inhibit routing of the second transaction to a different AOR. You can use the IBM CICS Transaction Affinities Utility MVS/ESA to assist with identifying transaction affinities. The utility identifies affinities in a running CICS region and produces a report of transactions which use commands that may cause affinities. You can find more information about affinities in Chapter 4 of *Dynamic Transaction Routing in a CICSplex* or in Chapter 2 of the *CICS Application Programming Guide*.

You can write your own DTR program or use the program provided by CICSPlex SM. The DTR program must take account of AOR availability, transaction affinities, workload separation, and workload balancing. The CICSPlex SM DTR program includes:

- The ability to handle affinities
- The ability to provide definitions for workload separation, which allows you to route certain transactions to a subset of AORs according to various criteria, such as LU names.
- The ability to dynamically distribute transactions among the available AORs, taking into account criteria such as health and performance of the AORs. This process is known as workload balancing.

Note: It is not possible to dynamically route START requests. The use of START commands that are function shipped between AORs is, however, common. A better approach is to use EXEC CICS RETURN with the TRANSID and IMMEDIATE options, so that the next transaction can be routed dynamically to a target AOR.

For more information about DTR and workload balancing, see *Dynamic Transaction Routing in a CICSplex*, *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*, or Chapter 4 of *CICSPlex System Manager for MVS/ESA Concepts and Planning*.

8.2 Release Coexistence

During your implementation of CICS TS, you will probably have a period of coexistence, where you are running both CICS TS and one (or more) earlier levels of CICS. In this section we look at some possible configurations and discuss some of the considerations for running mixed levels of CICS regions.

You do not have to migrate all of your CICS regions to CICS TS in a “big bang” in order to implement VSAM RLS. Instead we recommend migrating progressively, especially during the test and development phases before cutover to production. We discuss possible paths for migrating your CICSplex to CICS TS and VSAM RLS in 4.3, “CICS Region Migration” on page 14.

8.2.1 CICS Interregion Program Coexistence

The CICS interregion program (DFHIRP) is responsible for communication among connected CICS regions. DFHIRP must be placed in the MVS link pack area (LPA), and the same copy of DFHIRP is used by all CICS regions in the same MVS system image. You *must* use the CICS TS version of DFHIRP during your period of coexistence. DFHIRP is designed to be downward compatible, so CICS regions using earlier levels of CICS can participate in communication without problems.

The CICS TS version of DFHIRP allows all CICS regions within a Parallel Sysplex to communicate with CICS regions in different MVS images within the Parallel Sysplex using XCF/MRO rather than VTAM, thus avoiding the VTAM overhead. This applies to CICS/ESA 3.3 and CICS/MVS 2.1.2 regions, as well as to CICS TS regions. You define the link between regions as an MRO link, and CICS automatically selects XCF/MRO for cross-MVS-boundary communication. This function is also available for CICS/ESA 4.1.

8.2.2 CICS SVCs

DFHSVC (the CICS type 3 SVC) must be placed in the MVS LPA. DFHCSVC is designed to be downward compatible, so CICS regions using earlier levels of CICS can use it without problems. You can, optionally, continue to use an older version of DFHCSVC for your older levels of CICS. If you want to do this, you must rename one of the versions of DFHCSVC (probably the new CICS TS version). You *must* use the CICS TS version of DFHCSVC for your CICS TS regions. Here is the list of planning items related to interregion communication. Please refer to the *CICS Installation Guide* for detailed information.

1. Copy the new CICS TS type 3 SVC routine (DFHCSVC) into the LPA.
2. Copy the CICS TS DFHIRP into the LPA.
3. Unless you are using the MVS modified link pack area (MLPA), you have to re-IPL MVS to activate the new copies of DFHCSVC and DFHIRP.
4. If you choose to have more than one SVC level active, you have to rename one of the SVCs before you copy the new SVC into the LPA.
5. After successful tests of MRO, implement MRO security. This requires defining the DFHAPPL *APPLID* profiles in the RACF FACILITY general resource class. See the *CICS RACF Security Guide*

8.2.3 Sharing the CSD

You can share your CSD between CICS TS and earlier releases of CICS. You must upgrade an existing CSD, or use a new CSD created for CICS TS.

The CSD is a VSAM data set and therefore is eligible for RLS mode access. If you plan to share a CICS TS CSD with earlier releases of CICS, you cannot define the CSD as RLS eligible in the CICS TS regions.

If you share your CSD between regions on different levels of CICS, you should ensure that *all* updates are made from CICS TS regions. You can do this by restricting earlier releases to read-only access. Sharing the DFHCSD with earlier releases could cause warning messages to appear. For example, a CICS/ESA 4.1 region using a DFHCSD that has been upgraded to a CICS TS level could produce the following message:

```
DFHAM4887 I SCSCPTA1 Unrecognized resource type found in the CSD file
            and has been ignored.
```

If you use the sample SIT (DFHSIT6$) provided with CICS TS, you have to define an MVS log stream as the forward recovery log for your CSD. DFHSIT6$ specifies CSDFRLOG=1, which indicates that your log stream must map onto journal name DFHJ01. If you want to be able to forward recover your CSD, you can use the default CSD recovery option of CSDRECOV=ALL in conjunction with the CSDFRLOG specification. If you specify CSDFRLOG=nn, the journal name used is DFHJnn. You will get an error on attempting to open the CSD if the log stream is not defined.

If you do not have to forward recover the CSD, override the default CSD recovery option by specifying CSDRECOV=NONE or CSDRECOV=BACKOUTONLY.

8.2.4 Macro-Level Applications

If you still have to run macro-level applications and want to share VSAM data sets used by those applications as well as others running in many AORs in your Parallel Sysplex, you need:

- One CICS/MVS 2.1.2 AOR to run the macro-level application. In terms of workload balancing this is an inhibiting dependency (affinity), because all transactions that use macro-level programs must be routed to this region.

  Note: Support for CICS/MVS 2.1.2 ends at the end of 1996. Make strong efforts to remove any dependency on macro-level code.

- One CICS TS region that acts as an FOR for the CICS/MVS 2.1.2 region.

  Note: If you access the file in RLS mode, your programs may receive the new LOCKED condition. Unless your program is designed to handle unspecified error conditions, it will probably abend under these circumstances. Another consideration is any additional overhead required for access between the FOR and the VSAM server (SMSVSAM).

If you choose to keep using non-RLS mode, either NSR or LSR, you still can get the benefit of an "already merged" forward recovery log when the FOR is a CICS TS region.

8.2.5 Local DL/I

CICS TS does not support the local DL/I interface to IMS databases. CICS/ESA 4.1 is the last release that supports local DL/I. If you want to access an IMS database from a CICS TS region, you must either function ship your request to a CICS region that supports the local DL/I interface or use the database control (DBCTL) interface.

We strongly recommend that you use DBCTL. Only function ship requests to a data-owning region (DOR) that owns the database and provides a local DL/I interface as a temporary measure.

For more information about DBCTL, see the *CICS DBCTL Guide*.

8.2.6 Shared Data Tables

As with any VSAM file accessed in non-RLS mode, you need an FOR that owns the source data set for a shared data table (SDT), if you want to share the SDT among several AORs. Remember that the SDT is shared only among CICS systems within the same MVS image. If you use SDTs to enhance the performance of an application, ensure that all AORs supporting that application are located on the same MVS image. The source data set can be accessed from CICS regions in other MVS images through function shipping.

The source data set for a user-maintained data table (UMT) can be accessed in RLS mode, whereas the source data set of a CICS-maintained data table (CMT) cannot. Therefore you can share the source data set of a UMT in a Parallel Sysplex just as you can any other RLS file. However, note that you are sharing the source data set, not the SDT in storage. For more information, see the *CICS Shared Data Tables Guide*.

8.3 Logging in CICS TS

In this section we provide a brief introduction of the new logging mechanism in CICS TS and the related tasks you have to perform when migrating to CICS TS.

8.3.1 The CICS Log Manager

The journal control function from earlier releases of CICS has been completely replaced in CICS TS by the new CICS domain called *CICS log manager*. This function now controls all logging within CICS, handing over the individual log records to the MVS system logger. The MVS system logger, a new feature available with MVS/ESA 5.2, does the actual write to log by writing log streams out to the respective coupling facility structure you defined.

Figure 11 shows an overview of the logging mechanism introduced with CICS TS and MVS/ESA 5.2.

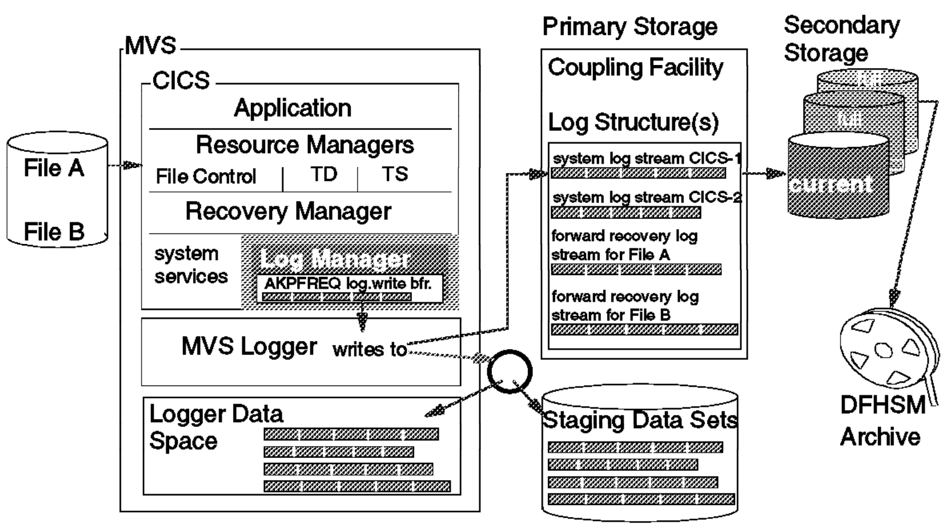

Figure 11. Overview of CICS Logging. The CICS log manager receives logging requests from the CICS resource managers, constructs a log stream, and passes it to the MVS system logger. The MVS system logger provides the physical write of log streams to the designated coupling facility structure. To avoid a single point of failure, the log stream is additionally written either to a logger data space or a staging data set, depending on the type of coupling facility you have.

You have to make some decisions and provide a set of definitions for the new logging mechanism. These activities require some careful planning. Insufficient planning can lead to poor performance, or even CICS stalls, because of insufficient resources for logging. If you have to change some of the definitions, you can affect the availability of your CICS systems. Therefore try to avoid changes, especially in a production CICSplex.

8.3.1.1 Types of Log Records

The CICS log manager provides log records for:

- The CICS system log, which contains all information necessary to recover from transaction failures (transaction backout) and restart after a CICS failure (emergency restart).
- The forward recovery logs for recoverable data sets

- The log of logs
- User journals
- Autojournals

See 5.3.3, “CICS TS Logger Structures” on page 24 for more information about the different log types and structures.

8.3.1.2 Coupling Facility Structures

Log records are physically written to coupling facility structures by the MVS system logger (unless you specify SMF for user logs). Therefore, to support CICS logging, you have to provide structure definitions in a CFRM policy and in a logger (LOGR) policy. The structure definitions are stored in the coupling data set. The structure definitions are:

- Coupling facility log structure definitions in the CFRM policy

  This definition can be compared with a structure declaration. You specify the structure name, the initial and the maximum structure sizes and the recovery attributes associated with the structure. The sizing of the structures is critical and therefore needs careful planning. IBM provides a new utility, DFHLSCU, to help you with sizing your structures. *CICS and VSAM Record Level Sharing: Implementation Guide* gives detailed information about structure definition and sizing, including examples and recommendations.

- Coupling facility log structure definitions in the LOGR for:

  – CICS system logs DFHLOG and DFHSHUNT
  – Forward recovery logs for your recoverable data sets
  – User journals and autojournals

  The structure names must correspond to a structure name defined in the CFRM policy. The structure definition in the LOGR policy contains parameters for the number of log streams you allow to connect to the structure at a given time, the average size of log buffers you anticipate for the log streams written to this structure, and the maximum possible size of a log buffer.

Figure 12 on page 64 shows how log streams are populated in a given coupling facility structure and how the various parameters in the LOGR policy relate to the structure usage.

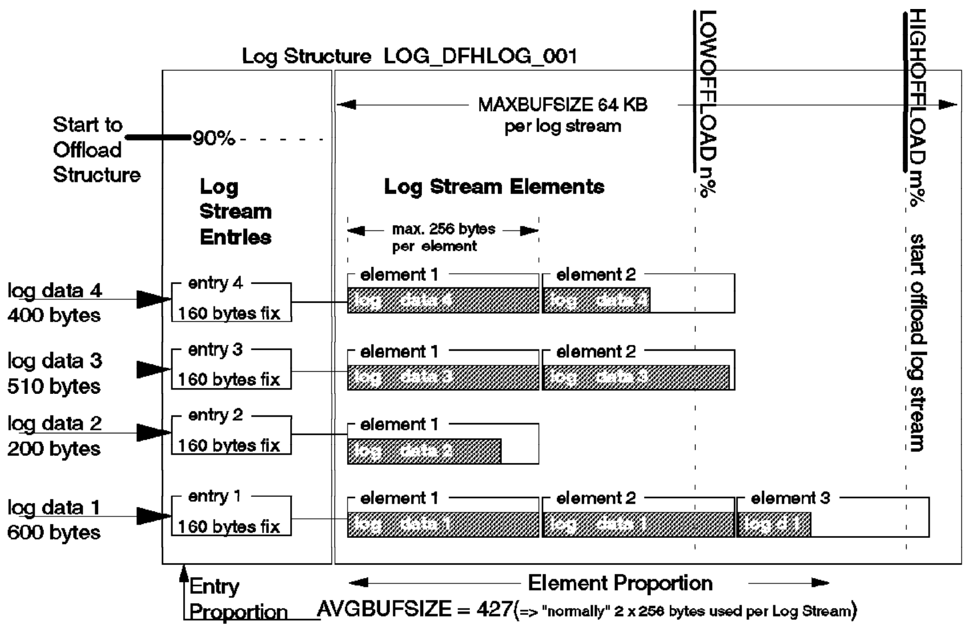

Figure 12. Log Stream Population in a Coupling Facility Structure

Figure 12 shows a coupling facility structure for the primary CICS system log stream named DFHLOG_001. Assume log data 1 through 4 are arriving from four CICS systems, PAA1, PAA2, PAA3 and PAA4, with different amounts of log data each. Each arriving log stream request causes a log stream entry to be created that contains control information. The log data is written into log stream elements, where each element is of a fixed size. The element size is dictated by the MAXBUFSIZE parameter specified in the structure definition. If MAXBUFSIZE is greater than 64 KB, the element size is 512 bytes; otherwise 256 bytes are allocated for each element. The available structure storage is proportionally divided into an element storage and an entry storage on the basis of the parameters you specify for average buffer size (AVGBUFSIZE), number of log streams (LOGSNUM), and the element size. The HIGHOFFLOAD percentage triggers the offloading of log streams to the respective log stream data sets. These are the static parameters that determine structure usage. During run time, structure usage is additionally influenced by the AKPFREQ value you specify and the load introduced in your CICS regions.

Theoretically you could define one single large structure in the coupling facility for all log streams created by all CICS regions in your Parallel Sysplex. This definition would result in poor performance, however, and insufficient use of coupling facility resources. Therefore we recommend that you define separate structures for each of the following log streams:

- The primary CICS system log, DFHLOG. Log data held in the CICS system log should normally be needed only for a relatively short time, that is, during the life of a transaction until it comes to a successful end or is backed out. Therefore the DFHLOG structure should be large enough to hold log streams for the active transactions of all AORs in your CICSplex. The MVS system logger automatically deletes log streams from the coupling facility structure that are no longer needed—the process is called *logtail deletion*

To optimize coupling facility capacity utilization, we further recommend having a separate DFHLOG structure for each type of CICS region, that is:

- One structure for your TORs
- One or more structures for the AORs. It might be efficient to define separate structures for AORs with different characteristics or if the number of AORs in your CICSplex is very high.
- One structure for the FOR (if applicable)
- One structure for the QOR (if applicable)

- One structure for the secondary CICS system log, DFHSHUNT. Depending on the nature of your transactions, the type of region, or the number of regions in your CICSplex, you may consider having more than one structure for the DFHSHUNT log streams as well, for example, if you have one AOR dedicated to long-running transactions or you do not migrate all of your data sets to VSAM RLS and keep an FOR.
- One or more separate structures for the general logs. Forward recovery logs, user journals, autojournals, and the log of logs (DFHLGLOG) are called *general logs*. DFHLGLOG can be viewed as a housekeeping log for the log stream data sets created by the MVS system logger. DFHLGLOG is used by forward recovery utilities such as CICSVR. Specify only one DFHLGLOG log stream for the entire sysplex.

  The number of structures you allocate for general logs depends on the number of log streams you anticipate to be written for the general logs.

If you have a volatile coupling facility, you must provide staging data sets. These data sets prevent the coupling facility from being a single point of failure for the log structures. Therefore the MVS system logger keeps a copy of the defined structures in the staging data sets.

8.3.1.3 Log Streams and Log Stream Data Sets

Log data is written to log streams that are stored in a coupling facility structure as described in 8.3.1.2, “Coupling Facility Structures” on page 63. You must provide the following log stream definitions in LOGR:

1. One separate log stream for the primary log, DFHLOG, of each CICS region in your Parallel Sysplex. The log stream name for DFHLOG must be unique across the Parallel Sysplex. Therefore we recommend including the MVS SYSID and the CICS region APPLID in the log stream name, for example: SC52.SCPAA1.DFHLOG for AOR PAA1 on MVS SC52.
2. One separate log stream for the secondary log, DFHSHUNT, of each CICS region in your Parallel Sysplex. The log stream name must be unique across the Parallel Sysplex, for example: SC52.SCPAA1.DFHSHUNT for AOR PAA1 on MVS SC52.
3. One or several forward recovery log streams for the forward recovery log data of your recoverable data sets. For more information see 8.3.1.4, “More on Log Streams and Log Stream Data Sets” on page 66.
4. One or several log streams for your user journals and autojournals. For more information see 8.3.1.4, “More on Log Streams and Log Stream Data Sets” on page 66.

A log stream definition contains parameters such as the name and size of a log stream, the structure to which you want the log stream to connect, and a “high watermark" (HIGHOFFLOAD) when you want the MVS system logger to start moving log streams to the log stream data set.

While you can share log streams for other log data, such as forward recovery logs (where sharing is even recommended), you must not share the same log stream for CICS system logs DFHLOG and DFHSHUNT.

8.3.1.4 More on Log Streams and Log Stream Data Sets

If the upper limit of a log structure is reached, the MVS system logger moves log streams out of the structure to the log stream data set you specify in the log stream definition. This process is triggered through the HIGHOFFLOAD parameter that you specify in the log stream definition for each log stream.

Some further considerations for log streams are:

- You do not have to define a separate log stream for each data set that needs a forward recovery log, or for each user journal or autojournal. The tradeoff is between transaction performance, fast recovery, and havi ng a large number of log streams to manage.

  If possible, all data sets used by one transaction should use the same log stream to reduce the number of logs to be written at sync point. A good starting point is to use the same forward recovery log ID that you use today.

- Group your data sets on log streams by the following criteria:
  - Similar security requirements
  - Similar backup frequency
  - Likelihood of having to restore the whole set at the same time

- Forward recovery log stream names should be related to the data sets, for example, all payroll data sets should write to the PAYROLL.FWDRECOV.PAYLOG log stream. The last qualifier of the stream name is used as the CICS resource name for dispatcher waits. Thus, if the name is self-explanatory, it is easier to interpret monitoring information and CICS traces.

- Do not mix data sets that have a high update frequency with data sets that have a low update frequency because more log data has to be read during the recovery of data sets that have a low update frequency.

- Do not log all data sets with a high update frequency to a single log stream because that can exceed the throughput capacity of the log stream.

- If you experience frequent structure full events (from the coupling facility), you have included too many data sets in a single log stream. The log stream cannot keep up with the amount of data arriving to be logged. Frequently monitor coupling facility usage, using the SMF 88 records, which you can extract by using the IXGRPT1 sample program in SYS1.SAMPLIB.

- Redundant data must be deleted from log stream data sets periodically so that the logger inventory entry, which is recorded for each log stream, does not reach the limit of 168 log streams per log stream data set name. Typically for a forward recovery log, the number of log stream data sets kept in the logger inventory relates to your data backup frequency (image copy). Example: Assume you always want to keep four generations of backup copies. After you create a new (fifth) generation, you delete the oldest (first generation). At the same time you should also delete the forward recovery logs that are older than the (first generation) image copy you just deleted.

You can delete log stream data sets by using the DFHJUP utility. Additional operational considerations regarding log stream data set deletion are provided in *CICS and VSAM Record Level Sharing: Recovery Considerations*. More information about logging is given in Chapter 6, “MVS/ESA Version 5 Release 2” on page 37.

8.3.2 Defining Log Streams: To Model or Not

All log streams must be defined to the MVS logger before they can be used. You can either define the log streams explicitly, using the IXCMIAPU utility, or you can let CICS define the log streams when they are first used. However, CICS cannot know how to specify all of the possible log stream definition attributes, so it specifies that the new log stream should be created like an existing model log stream. You are responsible for creating the model log stream definitions, using the IXCMIAPU utility, so that CICS definition attempts succeed.

It is worthwhile setting up models only if:

- Several log streams are to be defined.
- Each of the log streams has similar characteristics and should go into the same coupling facility structure.
- You do not know in advance the journal names that the CICS applications will use.

Otherwise it is probably less work to define the log streams manually, using IXCMIAPU.

You can also use the XLGSTRM user exit to set log stream attributes. It is called just before CICS defines a log stream to the MVS system logger.

8.3.3 Planning Items

In this section we list the tasks to be planned to enable the correct functioning of the CICS log manager. Details are documented in *CICS Release Guide* and *CICS Migration Guide*. More details on structure sizing are available in Chapter 5, “The Coupling Facility” on page 19, the *CICS Installation Guide*, and *CICS and VSAM Record Level Sharing: Implementation Guide*.

- Define the coupling facility log structures required for CICS logging
  This is a major task that has performance implications. It consists of the following steps:
  1. Design a naming convention for log structures and log streams.
  2. Determine the necessary log structure sizes and how many log structures you will need. Take your existing use of journals as a starting point. If you migrate from CICS/ESA 3.3 or CICS/ESA 4.1, the DFHLSCU utility (on the CICS TS library) can help you calculate the starting figures. DFHLSCU is described in the *CICS Operations and Utility Guide*. If you do not have access to DFHLSCU, or it is not appropriate for your system for any reason, use the formulae given in 5.5.4, “CICS TS Logger Structure Sizings” on page 31.
  3. Define log streams and/or log stream models to the MVS system logger.

     Decide on predefined log streams (using the IXCMIAPU utility) or dynamic creation by CICS. If you choose dynamic creation, you need JOURNALMODEL definitions in CICS (using RDO).

Note: You need two separate log streams for the system log of each CICS region in your Parallel Sysplex (DFHLOG and DFHSHUNT).

4. Create logical groups for your forward recovery logs, autojournals, and user journals to assign one log stream to each group, if possible.

5. Plan for log stream data set management.

   Delete redundant log streams from the MVS system logger inventory. The inventory limit is 168 data sets per log stream name.

6. Decide on log stream data set sizes. The more space you allocate for the log stream data sets, the longer it will take to reach the 168 limit. However, do not waste DASD space by specifying too much space for the primary CICS system log because ideally the primary log data should always be deleted from coupling facility storage before it gets spilled to the log stream data set.

7. Decide on the use and size of staging data sets.

8. Prepare for monitoring of the logger environment. The monitoring results might suggest tuning changes after implementation.

9. Review the AKPFREQ SIT parameter in each CICS region! The activity keypoint frequency influences your definitions for coupling facility log structure size and the dynamic use of the structure storage.

Note: Ensure that the AKPFREQ value is not 0 in any CICS region. A value of 0 would prevent the deletion of log data that is no longer needed, and therefore all log streams from such a region would be written to the log stream data set.

- Plan for the following changes, some of which might not be applicable to your environment (see the *CICS Release Guide*):
  - Changes to SIT and resource definition online (RDO) groups
  - Changes to JCL DD statements
  - Changes to API, systems programming interface (SPI), and exit programming interface (XPI)
  - Changes to global user exits and user-replaceable modules
  - Changes to utilities and CICS-supplied transactions
- Review CICS storage requirements. There is no longer a need for dynamic logging, but 64 KB are needed per log stream.
- Review security requirements.
- Review the use of journal APIs in your applications, for example, EXEC CICS WRITE JOURNALNUM...
- Review batch procedures and automation procedures related to pre-CICS TS journal control.

8.4 The CICS Recovery Manager

In this section we highlight the functions of the new recovery mechanism introduced in CICS TS and the related planning items. Details are documented in the *CICS Release Guide* and the *CICS Migration Guide*.

8.4.1 Overview

The CICS recovery manager is a new domain in CICS TS that is responsible for the coordination and synchronization of each UOW executed in CICS. The CICS recovery manager ensures integrity and consistency of recoverable resources in a single system as well as in a CICSplex.

One of the major functions of the new CICS recovery manager is the enhanced handling of errors during sync-point processing, for example, during the in-doubt window of two-phase-commit processing. If a UOW that has updated recoverable resources fails in flight, the CICS recovery manager automatically backs out the changes, just as in earlier releases. However, in the rare case when a UOW fails during sync-point processing (during the in-doubt window, backout, or commit), the CICS recovery manager *shunts* the UOW.

In shunting, CICS maintains information about the UOW on the system log and uses the information to resolve the UOW at a later time when the cause of the failure is resolved. Active locks, for the shunted UOW, are converted to retained locks so that others UOWs that try to access the recoverable resources are not forced to wait (instead they receive the LOCKED condition).

The commonest cause of shunted UOWs is likely to be in-doubt failures. When a UOW fails in doubt, you can control whether or not it gets shunted, using the new WAIT and WAITTIME attributes of the transaction resource definition. Through WAITTIME attribute, you can choose between a “wait forever” (until failure resolved) and a heuristic decision. In other words, you can specify to wait a certain time before a heuristic decision is made. If an in-doubt UOW is shunted, data integrity is preserved (as long as certain conditions are met), but resources are locked and unavailable to other transactions. If an in-doubt UOW is not shunted, then, just as in earlier CICS releases, data integrity may not be preserved.

Note: There are situations where it is not possible for the CICS recovery manager to shunt an in-doubt UOW (for example, if the UOW has multiple MRO links to back-level CICS regions).

8.4.2 Planning Items

Many of the planning items associated with the CICS recovery manager are related to the procedures you have implemented to restore data integrity following in-doubt and backout failures. Therefore a complete review of these procedures is necessary, including user exits and CICS-supplied utilities. See the *CICS Release Guide* and the *CICS Migration Guide* for details.

- Plan for education. You have to understand how the CICS recovery manager handles various error situations, and what to do if CICS cannot automatically resolve retained locks. You should review the information in *CICS and VSAM Record Level Sharing: Recovery Considerations*, which contains samples for handling UOW failures and how to resolve retained locks manually.
- Determine whether your resource managers support the new CICS recovery manager interface. Only if they do will you benefit from the new in-doubt support. IBM-supplied resource managers that support the CICS recovery manager interface to support in-doubt WAIT are:
  - CICS file control
  - DBCTL

- DB2
- CICS temporary storage domain
- CICS transient data program
- MQSeries for MVS/ESA

- We strongly recommend that you specify WAIT(YES), the default, for all transactions that update resources for which data integrity is important. It may be appropriate, in circumstances when you cannot afford for resources to be locked indefinitely, to specify a WAITTIME so that CICS backs out (or commits) resource updates, after the specified time, if the failure cannot be resolved by resynchronization with the coordinator. The tradeoff is between data integrity and resource availability.

- Review your procedures for reestablishing connections between CICS regions after a failure. The CICS recovery manager affects intercommunication between CICS regions with changes to the exchange lognames process. The effect of setting a NOTPENDING connection following an exchange lognames failure is more significant than in earlier releases. It is quite common for automated operators to be used to set NOTPENDING connections at regular intervals. In CICS TS this will force shunted, in-doubt UOWs to back out or commit. Use the new XLNACTION attribute of the CONNECTION resource definition to replace the function of any NOTPENDING automation that has been implemented. See *CICS and VSAM Record Level Sharing: Recovery Considerations* for more information.

- Review your procedures for stopping CICS. The CICS recovery manager introduces a new CICS shutdown process. See Chapter 10, "Operations" on page 85 for more information.

- The CICS recovery manager introduces a new utility called DFHRMUTL. See Chapter 10, "Operations" on page 85 for more information.

- Review the following changes to CICS-supplied transactions:

  - CEMT - has been extended to allow investigation of UOWs.

  - CETR - the CICS tracing transaction has been extended to support the new CICS domains. Review any automation product or operating procedures you have in production to include these new domains in tracing.

  - Use CIND to test recovery after a failure during the in-doubt window. CIND—the CICS in-doubt test transaction—is a new transaction that provides a way to cause a failure during the in-doubt window for testing purposes.

  - CESD - the CICS shutdown transaction is provided to shut down the system in a controlled way after a normal or immediate shutdown. You may want to add some routines that you are using today in DFH$SDAP to this transaction (program DFHCESD).

- Review the changes to the CICS API, XPI, and SPI, user exits, and utilities. Please refer to the *CICS Release Guide* for a detailed list of the changes.

8.5 VSAM Record-Level Sharing

In this section we provide a brief technical overview of VSAM RLS, discuss some restrictions associated with RLS processing, and review the items you have to plan to access files that are shared among several CICS TS regions using RLS.

8.5.1 Overview

VSAM RLS is a new function that comes with DFSMS/MVS 1.3. A new address space, SMSVSAM, acts as a server for all access to VSAM data sets in RLS mode. There is one SMSVSAM address space in each MVS system image. For more information, see Chapter 2, "VSAM Record-Level Sharing" on page 3.

CICS TS exploits RLS to provide:

- VSAM file sharing in RLS mode among *n* CICS transactions that run in *m* CICS regions in any MVS image in a Parallel Sysplex, with full update capability
- Shared read-only access to RLS files by a batch job
- Read integrity through the new VSAM locking mechanism
- Support of VRRDS

8.5.2 RLS Processing Restrictions

Keep in mind these restrictions associated with RLS:

- Only SMS-managed VSAM data sets can be opened in RLS mode.
- All data sets of a VSAM sphere can only be opened in either RLS or non-RLS mode at a given time.
- Concurrent batch update on RLS data sets is not possible, unless the data set is defined as nonrecoverable. See Chapter 10, "Operations" on page 85 for more information about how to switch modes if necessary.
- Recovery attributes for RLS data sets must be defined in the integrated catalog facility (ICF) catalog. For non-RLS data sets, recovery attributes can continue to be specified in the CICS file definitions. Note, however, that CICS takes the definitions from the ICF catalog if they are present, and the recovery attributes in the file definitions are ignored.
- The limitation of a 4 GB data set size has been removed for KSDSs *only* if they are accessed in non-RLS mode.
- In RLS mode, alternate indexes are supported only by path access.
- You cannot use the relative byte address (RBA) to access KSDSs in RLS mode.
- An RLS file cannot be:
  - A source file for a CICS maintained data table
  - A transient data intrapartition data set
  - A temporary storage data set
  - A local or global CICS catalog data set
  - A data set defined with the IMBED option

  Note: IMBED can be removed from the cluster definition without loss of function. If you want to use a cluster that specifies IMBED with RLS, you can use REPRO to create a new cluster that does not specify IMBED.

There are some new and changed messages related to RLS, as well as new abend codes. Therefore you have to review any automation product that relies on messages of type DFHFCxxxx. See the *CICS Messages and Codes* and the *CICS Migration Guide*. *CICS and VSAM Record Level Sharing: Implementation Guide* contains more details about VSAM RLS and provides samples for RLS setup.

8.5.3 Planning Items

The items you should plan for RLS implementation are listed below. For an overview of migration paths to VSAM RLS, see Chapter 4, "Migrating to VSAM Record Level Sharing" on page 13. Please refer to the *CICS Release Guide* and the *CICS Migration Guide* for details.

- Identify data sets eligible for RLS access in your environment and determine whether a data set must be shared or not.

  The main considerations for whether a data set has to be shared between CICS regions in RLS mode are:

  – Data sets that are currently accessed through an FOR are strong candidates for RLS.

  – Data sets that are accessed only by a single CICS region are best kept as non-RLS.

  – Data sets that are accessed only within a single MVS usage and are updated only by a single CICS region are more suitable for use as shared data tables (which provides better performance than RLS).

- Define the coupling facility structures needed for RLS—a cache structure to hold the VSAM records and a lock structure to store locks that are held on these records. See Chapter 5, "The Coupling Facility" on page 19 and Chapter 7, "DFSMS/MVS Version 1 Release 3" on page 45.

- Define the sharing control data sets (the minimum is three data sets). See Chapter 7, "DFSMS/MVS Version 1 Release 3" on page 45.

  Operators should be aware of the new SHCDS commands, such as:

  – LISTDS - list data set status

  – LISTSUBSYS - list status of a named CICS region

  – PERMITNONRLSUPDATE - override VSAM data integrity protection

- Define the SMS storage classes. See Chapter 7, "DFSMS/MVS Version 1 Release 3" on page 45.

- Define cache set names in the SMS base configuration. A cache set name must be defined in all storage classes used by RLS. See Chapter 7, "DFSMS/MVS Version 1 Release 3" on page 45.

- Review the IGDSMSxx parameters in SYS1.PARMLIB. See the *CICS Installation Guide* and the MVS library.

- Authorize each CICS region to open the SMSVSAM control ACB. See the *CICS RACF Security Guide* for more information.

- Review your batch update jobs if applicable. You need RLS quiesce/unquiesce procedures to switch from RLS mode to non-RLS mode. See Chapter 10, "Operations" on page 85.

- Review the DTIMEOUT parameter in your transaction definitions and the FTIMEOUT parameter in the SIT. These parameters influence SMSVSAM in deadlock timeout detection.
- Review read integrity requirements. Read integrity is recommended only for applications that cannot tolerate dirty reads. Before using it, review the application for increased risk of deadlock. See the *CICS Application Programming Guide* for information about the increased risk of deadlocks, particularly when defining read integrity on file definitions.
- Define the recovery attributes for RLS data sets (using IDCAMS) and possibly agree on standard placement of these attributes in the ICF catalog for all of your VSAM data sets. Modify the FC resource definitions for your VSAM files accordingly.
- Review global user exits (GLUEs). Review the backout user exits, XFCBFAIL and XFCLDEL, which replace XDBFERR, XRCFCER, and XRCOPER. You should also review the new user exit, XFCBOVER. If you access any VSAM data sets in non-RLS mode, you should also review the XFCBOUT user exit.
- Review recovery and restart procedures and adopt them to conform with the new logging mechanism (see Chapter 10, "Operations" on page 85).
- You must change any programs that have a dependency on the ACBTIOT field of the VSAM ACB. As a result of increasing the number of files that can be allocated to an address space, VSAM no longer sets the ACBTIOT field. If you have any programs that locate the VSAM ACB and use the value from ACBTIOT, the results are unpredictable. Make sure any vendor packages you use have been upgraded to reflect this change.
- If necessary, change application programs to correctly respond to new error conditions. See Chapter 9, "Application Programming Considerations" on page 79.

8.6 Databases

In this section we list some planning items related to databases used in CICS applications.

8.6.1 DB2 for MVS/ESA

The following DB2 releases are currently supported in CICS TS:

- DB2 Version 2 Release 3
- DB2 Version 3 Release 1
- DB2 for MVS/ESA Version 4 Release 1

You must use the CICS-DB2 attachment facility that is provided with CICS TS. It works with all releases of DB2 listed above.

If your DB2 system is at a different release, you cannot attach it to a CICS TS region. Thus you might have to keep a backlevel CICS region that supports your level of DB2. This has disadvantages in terms of workload balancing and continuous availability in a Parallel Sysplex. Therefore we recommend that you migrate your DB2 system to DB2 for MVS/ESA Version 4 Release 1 for full utilization of the Parallel Sysplex capabilities (data sharing, locking, logging, continuous availability, workload management). The migration is a major task that requires separate planning. See the DB2 for MVS/ESA Version 4 Release 1 library for more information.

As far as CICS is concerned, for DB2 you have to plan for the following:

- Install and activate the attachment facility of CICS TS.
- If you work with DB2 for MVS/ESA Version 4 Release 1, you have to apply APAR PN76625.
- Changes to the DSNCRCT macro in the resource control table (RCT)
- Changes in SPI commands
- Changes in problem determination (new tracing)
- The INITPARM SIT parameter must now be used to specify the DB2 RCT suffix, so you have to change your CICS startup procedures accordingly.

Please refer to the *CICS Release Guide* for more information.

8.6.2 IMS Database Support

In this section we list the planning items related to IMS databases used in CICS applications.

Because CICS/ESA 4.1 is the last release that supports local DL/I, you should plan to migrate to DBCTL. This is a major task that requires its own plan. See the *CICS DBCTL Guide* and the IMS library for more information. DBCTL gives you a number of benefits, one of which is sharing databases among any AORs in the Parallel Sysplex.

After successful migration to DBCTL (or if you already use DBCTL), for CICS you have to plan for:

- Unnecessary dump suppression

  If a UOW is shunted and DBCTL is participating in this UOW, DBCTL produces unnecessary database resource adapter (DRA) snap dumps. To avoid these dumps, apply an APAR according to the IMS version installed:

  - IMS/ESA Version 3 Release 1 - no APAR required. This is the minimum release level required to work with CICS TS.
  - IMS/ESA Version 4 Release 1 - APAR PN62412 (PTF UN68325)
  - IMS/ESA Version 5 Release 1 - APAR PN62480 (PTF UN68717)

- SIT changes

  The removal of local DL/I support makes almost all related SIT parameters obsolete. See the *CICS Release Guide* and the *CICS System Definition Guide* for more information.

- Batch jobs that currently use the CICS shared database interface to access local DL/I databases must be converted to IMS batch message processing (BMP) so they can directly access DL/I databases through DBCTL. The batch interface has been removed in CICS TS.

Should you decide not (yet) to migrate to DBCTL, you have to keep a DOR with a backlevel CICS (CICS/ESA 4.1, for example), to which you function ship all DL/I requests. Therefore you have to provide the necessary definitions for remote DL/I access in the requesting AORs. See the *CICS Release Guide* for more information.

8.6.3 Databases from Other Vendors

The basic question here is whether the specific database adapter supports the new CICS RMI to include in-doubt WAIT. If the answer is no, you cannot benefit from the in-doubt WAIT option.

You should negotiate with the database vendor as to which release of the database product is supported for CICS TS.

8.7 Other Functions in CICS TS

In this section we provide some highlights of the other functions available with CICS TS that are not related to VSAM RLS. The planning information we include here is only at an overview level.

8.7.1 VTAM Generic Resources

VTAM generic resources (GR) is a function of VTAM 4.2 exploited by CICS/ESA 4.1. The major benefit of VTAM GR is the ability to define several CICS regions (for example, TOR1, TOR2, TOR3) in a GR group and give it a generic name known to CICS users (for example, CICS). Thus users log on to CICS, but they actually are logged on to any of the three TORs in the group. A simple *join-shortest-queue* mechanism in VTAM provides some automatic workload balancing up front. In CICS TS some limitations with VTAM GR related to automatic transaction initiation (ATI) and advanced program-to-program communication (APPC) autoinstall have been removed. Please refer to *CICS Release Guide* for more information.

When migrating to VTAM GR from CICS/ESA 4.1 to CICS TS, consider the following:

- You can benefit from the VTAM GR enhancements only if all members of a GR group are at CICS TS.
- The autoinstall user replaceable module (URM) has to be adapted.
- AIRDELAY and AILDELAY SIT parameters now apply to APPC connections as well as terminals.

  AIRDELAY: delay period after an emergency restart, before autoinstalled terminals and APPC connections that are not in session are deleted.

  AILDELAY: delay period after sessions with CICS have ended, before the autoinstalled terminal or connection entry is deleted.
- You might have to code a batch program to end affinities owned by GR members that are out of service. See the *CICS Intercommunication Guide*.

8.7.2 CICS TS Shared Temporary Storage Queues

With CICS TS you can move your nonrecoverable temporary storage queues into the coupling facility and thus make them available to all CICS TS regions in the Parallel Sysplex. The temporary storage queues are accessed through a new temporary storage queue server subsystem, which must reside in each MVS image where you want CICS to access these shared queues. The nonrecoverable temporary storage queues can now reside in temporary storage pools, which you define in a coupling facility list structure. For planning, you roughly need one pool per "old" QOR. See Chapter 5, "The Coupling Facility" on page 19 for more information.

You need a sysplexwide naming convention for shared temporary storage queues to ensure that the queue names are unique. If you change any queue names, you may have to make corresponding changes to your application programs.

Note that shared temporary storage is supported only for nonrecoverable queues. So, if you have to share a recoverable temporary storage queue among several CICS regions, you still need a QOR that acts as a server where the AORs send their temporary storage requests by means of function shipping.

Note: Shared temporary storage adds to your coupling facility capacity needs.

CICS and VSAM Record Level Sharing: Implementation Guide contains a sample setup for shared temporary storage.

8.7.3 MVS Automatic Restart Manager

All planning items that relate to the MVS ARM are described in 6.2, "The Automatic Restart Manager" on page 40

8.7.4 EXEC CICS CREATE

The EXEC CICS CREATE command enables you to dynamically create CICS resource definitions out of a CICS program. You might want to consider using this new SPI for your own automation tools. For more information refer to the *CICS Release Guide* and the *CICS System Programming Reference*.

8.7.5 Resource Definition Online for Transient Data

CICS TS provides RDO support for transient data. CICS TS is the last release that supports the DFHDCT macro. Therefore you should plan to migrate your destination control table (DCT) to RDO. Additionally there are some new functions that are available only when transient data is defined using RDO. To aid migration you can use the new DFHDCT TYPE=(INITIAL,MIGRATE) macro.

Application programs have to be reviewed because of the new locked condition that can occur, for example, when a UOW is shunted.

The DD statements for extrapartition transient data data sets in your CICS startup JCL are no longer needed. You can define the data set name by using RDO.

8.7.6 Monitoring and Statistics

Numerous changes and additions have been made to CICS monitoring data and statistics in CICS TS. Both performance and exception class records have changed in this release to improve the use of this data for analysis as well as to reflect the new and changed CICS domains.

You should plan education on the use and analysis of monitoring data and statistics. You should also review and test the operation of any accounting and billing mechanisms that depend on CICS monitoring data and statistics.

Review and respecify your monitoring control table (MCT) to meet your needs. *CICS and VSAM Record Level Sharing: Implementation Guide* contains monitoring samples.

8.7.7 Fallback Planning

We strongly recommend working out a fallback plan for your production environment before migrating to CICS TS and VSAM RLS. Chapter 20, “Fallback Planning Considerations” in the *CICS Migration Guide* contains additional information on fallback planning.

Chapter 9. Application Programming Considerations

In this chapter we deal with planning considerations for application programs where there are significant changes.

The changes for batch programs consist of the processing restrictions documented in 7.1.2, “Processing Restrictions” on page 47, and the language restrictions documented in 9.4.2, “Batch Language Support” on page 82.

The rest of this chapter deals with considerations for CICS application programs. For full details, see the *CICS Application Programming Reference* manual.

9.1 Accessing VSAM Files from CICS

In this section we cover areas of enhancement to support VSAM RLS.

9.1.1 Enhanced Commands

The following CICS API commands have been enhanced to support VSAM RLS:

- DELETE
- READ
- READNEXT
- READPREV
- RESETBR
- REWRITE
- STARTBR
- UNLOCK
- WRITE.

The enhanced CICS supports:

- Update while in a browse
- Two forms of read integrity
- A conditional update option

See *CICS Application Programming Guide* and *CICS Application Programming Reference* for more information about the enhanced commands.

9.1.2 New Exception Condition

CICS TS adds a new exception condition, LOCKED, for applications accessing VSAM files (whether in RLS or non-RLS mode). In general, existing application programs that access files opened in RLS mode will run without modification. However, some programs could encounter the LOCKED condition. The new LOCKED condition occurs if a program attempts to access a record for which VSAM holds a retained lock. Requests for records protected by a retained lock are rejected immediately. The default CICS action for applications that do not handle the retained lock exception condition and do not specify HANDLE CONDITION ERROR is to abend the transaction.

You must check existing applications to ensure that they are correctly coded to deal with “unexpected conditions.” This should be of concern only in application programs that specify NOHANDLE, or imply NOHANDLE by means of the RESP and RESP2 options. Application programs that use NOHANDLE to deal with exception conditions in their own way must ensure that they are coded to handle unknown conditions, otherwise errors could occur. For example, if an application program uses an ELSE or WHEN OTHER clause to assume the last of a known set of conditions, unexpected results could follow from a retained lock condition.

9.1.3 RLS Alternate Indexes Restriction

It is also possible for a CICS application program to access a file that has been directly defined as an alternate index rather than a path. This results in index data rather than file data being returned to the application program. This operation is not supported for files opened in RLS mode.

9.2 Accessing Queues

CICS TS adds a new exception condition, LOCKED, to the following temporary storage and transient data commands:

- DELETEQ TS
- WRITEQ TS
- DELETEQ TD
- READQ TD
- WRITEQ TD

LOCKED means that the request cannot be performed because use of the queue has been restricted because a UOW has failed in doubt.

9.2.1 Recoverable Temporary Storage

WRITEQ TS and DELETEQ TS requests for recoverable queues may now return the LOCKED condition. This happens if the queue is owned by another UOW that has suffered an in-doubt failure. The WRITEG TS function, which writes data to a temporary storage queue, has changes in the AUXILIARY and MAIN options and the NOSPACE and SYSIDERR conditions. For details of how these changes affect the space requirements of the temporary storage data set, see the *CICS Application Programming Reference*.

9.2.2 Recoverable Transient Data

The LOCKED response is added to the EXEC CICS READQ TD, EXEC CICS WRITEQ TD, and EXEC CICS DELETEQ TD commands, with an EIBRESP of 100 and an EIBRESP2 of 0. The LOCKED response occurs when a request cannot be performed because access to the queue has been restricted owing to a UOW suffering an in-doubt failure. A LOCKED condition can occur on any request for a logically recoverable queue defined with the WAIT(YES) and WAITACTION(REJECT) attributes on the transient data queue definition.

9.3 Journals

The EXEC CICS WRITE JOURNALNUM command is replaced by the new EXEC CICS WRITE JOURNALNAME command. EXEC CICS WAIT JOURNALNUM continues to be supported for compatibility purposes.

If you have an application that currently writes data to the CICS system log (DFHJ01) using the EXEC CICS WRITE JOURNALNUM(01) command, this data is not written to the CICS TS system log (DFHLOG), but to a separate log stream. CICS uses a log stream called DFHJ01 as the journal name in this case. The same is true for any other journal write commands: CICS replaces the journal number *nn* with a journal name of DFHJ*nn*. This journal handling has been introduced in CICS TS to support your existing applications without change. If you need more than the 99 journals available in previous CICS releases, you have to change the EXEC CICS WRITE JOURNALNUM(nn) command to EXEC CICS WRITE JOURNALNAME(*jname*), where *jname* can be any eight-character journal name you choose. See the *CICS Application Programming Guide* for details.

Note: Instead of writing user journals and autojournals to MVS log streams, you can specify that they be written to the SMF data set.

JOURNALNAME introduces a corresponding change to the EXEC CICS QUERY SECURITY command, where JOURNALNAME is a new RESTYPE parameter.

9.4 Programming Language Considerations

All existing CICS programs, written in supported languages, can access RLS files without change. Batch RLS access has some language level restrictions.

9.4.1 CICS Language Support

All user application programs supported by CICS can access RLS files without change. Included are all programs assembled or compiled by a supported assembler or compiler and those programs compiled by an unsupported assembler or compiler for which CICS maintains execution time support.

CICS supports the following assembler, COBOL, PL/I, and C/370 compilers:

- High Level Assembler/MVS & VM & VSE Version 1.1 (5696-234)
- IBM PL/I for MVS & VM (5688-235)
- OS PL/I Optimizing Compiler Version 2 Release 1 (5668-910)
- OS PL/I Optimizing Compiler Version 1 Release 5.1 (5734-PL1) or later
- IBM COBOL for MVS & VM (5688-197)
- VS COBOL II (5668-958 and 5688-023)
  Requires PTF for APAR PN43097--see 9.4.1.1, “PTFs for APAR PN43097” on page 82 for details.
- IBM C/C++ for MVS/ESA (5655-121)
- C/370 (5688-040)

To assemble the SIT, you must use High Level Assembler/MVS & VM & VSE (ASMA90). The number of parameters defined in the DFHSIT macro exceeds the limit (240) supported by MVS Assembler H.

CICS also supports IBM SAA AD/Cycle Language Environment/370 (LE/370), program number 5688-198, with the following SAA AD/Cycle COBOL, C/370, and PL/I SAA AD/Cycle compilers:

- SAA AD/Cycle COBOL/370 (5688-197)
- SAA AD/Cycle C/370 (5688-216)
- SAA AD/Cycle PL/I (5688-235)

Note that batch support requires IBM Language Environment for MVS and VM, Version 1 Release 5 (LE 1.5). We therefore recommend that you use LE 1.5 in your CICS environment as well as in your batch environment.

9.4.1.1 PTFs for APAR PN43097

To prevent 0C4 abends caused by IGZECIC returning to CICS with an incorrect mode, you must apply the requisite PTFs for APAR PN43097. These are:

- PTF UN48282 for FMID JCL1331
- PTF UN48283 for FMID JCL1341
- PTF UN48284 for FMID JCL1403

9.4.1.2 Execution-Time Support for H Assembler

CICS retains translation and execution-time support for application programs assembled by the MVS Assembler H Version 2 (5668-962).

9.4.1.3 Execution-Time Support for Unsupported COBOL Compilers

CICS TS provides the same level of support for certain unsupported COBOL compilers as CICS/ESA 4.1 provided. It retains execution-time support for application programs compiled by the following unsupported COBOL compilers:

- Full American National Standard COBOL Version 4 (5734-CB2)
- OS/VS COBOL (5740-CB1)

We recommend that you migrate your old COBOL applications to a supported release of a COBOL compiler. Consider the following history of OS/VS COBOL:

- Withdrawal from marketing: December 18, 1992
- Discontinuance of support: June 30, 1994
- End of extended support: June 30, 1996

9.4.2 Batch Language Support

Run-time library RLS support for COBOL, PL/I, and FORTRAN is provided by LE 1.5, program number 5688-198. In general, high-level languages (HLLs) and VSAM using RLS access maintain compatible execution with VSAM NSR where the batch application has sole usage of the data set. However, in a VSAM RLS multiupdate environment, source changes to existing batch applications may be needed to handle new VSAM RLS (locking) errors and ensure that record locking meets the need of the batch application. For example, if a batch application previously had exclusive control over all its accessed records running in batch, can it now tolerate an RLS environment where it only has exclusive control of particular records at a time, while other record updates, deletes, and additions might be occurring concurrently? This may require some analysis of the batch application.

Programs that may require change in a batch environment are those that either use dynamic allocation (SVC99) or need to check for new return or error codes. For example, many COBOL programs that check file status values after each I/O request has completed do not also check the VSAM feedback return code. The technique for doing this is documented in the *IBM COBOL for MVS and VM Programming Guide*, SC26-4767. It can be used with COBOL/370 and VS COBOL II. We strongly recommend that all COBOL programs to be used with VSAM data sets in RLS mode use the second status area to obtain the VSAM feedback return code.

LE 1.5 supports applications generated with the following IBM compiler products:

- IBM PL/I for MVS and VM Release 1.1 (5688-235)
- IBM SAA AD/Cycle PL/I MVS & VM Release 1 (5688-235)
- IBM COBOL for MVS and VM Release 2 (5688-197)
- IBM SAA AD/Cycle COBOL/370 Release 1 (5688-197)
- IBM VS COBOL II Compiler and Library with Debug (5668-958)
- IBM VS COBOL II Compiler and Library (5688-023)
- IBM C/C++ for MVS/ESA Version 3 (5655-121)
- IBM SAA AD/Cycle C/370 Release 2 (5688-216)

For the supported compiler products, recompilation is not required for the VSAM RLS feature.

Note: There is no RLS support for VS COBOL II NORES programs.

Additionally, APAR PN80628 for COBOL and APAR PN77375 for PL/I has to be applied to LE 1.5.

9.4.2.1 COBOL

For batch access to RLS data sets, COBOL programs must use the LE 1.5 run-time modules. This allows COBOL batch non-commit-protocol applications to access VSAM files in RLS mode. These batch programs can access VSAM recoverable files in READ mode, and VSAM nonrecoverable files in either UPDATE or READ mode. Batch COBOL programs requiring RLS access must be compiled with one of the LE-supported compilers. There is no RLS support for VS COBOL II NORES programs. OS/VS COBOL programs have to migrate to one of the COBOL compilers listed above.

9.4.2.2 PL/I

PL/I batch support for VSAM RLS is provided in LE 1.5. This allows PL/I batch non-commit-protocol applications to open VSAM data sets in RLS mode. VSAM recoverable files can be opened for READ processing, and VSAM nonrecoverable files can be opened for either READ or UPDATE processing.

Existing PL/I batch applications that currently run under LE will execute with VSAM RLS without any need to re-link edit or recompile. In addition, if you are migrating to LE from OS PL/I, LE PL/I provides OS PL/I object and load module compatibility. However, special situations such as very old OS PL/I version 1 code and some Shared Library usage may require re-link edit, recompilation, or both. Review the *PL/I MVS and VM Version 1 Release 1 Migration Guide* for detailed requirements about recompilation, re-link edit, and product differences. You might also find the LE migration checklist in Chapter 1 of the *Language Environment for MVS and VM Release 5 Run-time Migration Guide* helpful.

PL/I APAR PN77375 has a detailed description of the implications of RLS access on the PL/I I/O language statements. Included are programming concerns related to record locking and access limitations as well as new error codes and messages. You should also review APAR PN85553 before accessing an RLS mode data set from a PL/I batch program.

9.4.2.3 FORTRAN

LE 1.5 supports a FORTRAN application program′s use of VSAM RLS. VSAM RLS is designed so that FORTRAN programs without dynamic allocation of VSAM data sets do not have to be modified to take advantage of this support.

The announcement letter for LE 1.5 details the support for different levels of FORTRAN compilers.

Chapter 10. Operations

In this chapter we deal with operational considerations, specifically changes to:

- CICS startup
- CICS shutdown
- Batch processing of online VSAM RLS data sets
- MVS system logger
- Temporary storage queue server region

10.1 CICS Startup

CICS TS introduces a new startup option, INITIAL, and changes the COLD start process. A new utility program, DFHRMUTL, extends the possible processing of an AUTO start.

10.1.1 Startup Options

CICS TS allows three types of startups: INITIAL, COLD, and AUTO.

10.1.1.1 INITIAL Starts

An INITIAL start is equivalent to the COLD start process in earlier releases of CICS. You can think of it as a “supercold” or “icy-cold” start. You can specify an INITIAL start by using the START=INITIAL parameter as a system initialization parameter or by specifying the AUTOINIT parameter to DFHRMUTL as an AUTO start override.

You would specify an INITIAL start when starting a new CICS region for the first time (with a new system log and new catalog data sets) or if a severe and unrecoverable error affects either the system log or one of the catalog data sets.

If the system log and catalog data sets are not damaged, and the problem is caused by unrecoverable data errors, you need only run DFHRMUTL to specify an INITIAL start. You do not have to redefine the data sets or log stream.

10.1.1.2 COLD Starts

The COLD start process has been changed in CICS TS: Some information from the previous run is preserved in both the global catalog and the system log. The purpose of the change is to allow a local CICS region to perform a COLD start with regard to all local resources while at the same time appearing to perform a warm or emergency restart to any remote partners to which it was connected during the previous run. Thus during a COLD start CICS preserves all information needed for resynchronization with its remote partners.

10.1.1.3 AUTO Starts

Another consequence of the changes to CICS startup is the effect on START=AUTO. If you have newly created catalog data sets, you must use DFHRMUTL to initialize the global catalog. If you initialize the global catalog with a low-values record, as in earlier releases, CICS abends during initialization if you specify START=AUTO.

In earlier releases, CICS determines the type of start to perform when you specify START=AUTO by reference to the control record in the global catalog. The state of this record, now the recovery manager′s control record, can be overridden by running DFHRMUTL, specifying the SET_AUTO_START parameter. Thus you do not have to modify your JCL to change the CICS START system initialization parameter. You can leave the system initialization parameter set as START=AUTO and use DFHRMUTL whenever you have to override the start type.

10.2 CICS Shutdown

The CICS TS shutdown process has been changed to provide a fast, controlled shutdown of CICS that retains the minimum number of locks for data integrity. Although a normal PERFORM SHUTDOWN is the safe and therefore recommended way to shut down CICS, in previous releases it proved unsatisfactory in many installations.

To help solve these problems, CICS TS provides a shutdown assist transaction.

10.2.1 Shutdown Assist Transaction

The purpose of the shutdown assist transaction is to eliminate, as far as possible, the need for immediate shutdowns of CICS. On an immediate shutdown in releases earlier than CICS TS, CICS does not allow running tasks to finish, and backout is not performed until emergency restart. In CICS TS, this could cause an unacceptable number of units of work to be shunted, and locks to be retained unnecessarily. We strongly recommend that you use a normal shutdown whenever possible.

CICS TS introduces a shutdown assist transaction, which can optionally be used with both normal and immediate shutdowns. The default shutdown assist transaction is CESD, which starts the CICS-supplied program, DFHCESD. DFHCESD attempts to purge and back out long-running tasks using increasingly stronger techniques. It ensures that as many tasks as possible commit or back out cleanly, enabling CICS to shut down in a controlled manner. You can replace the CICS-supplied transaction and CICS-supplied program with your own transaction and program.

If you currently use an immediate shutdown to terminate your CICS regions (perhaps because of long-running tasks), you must evaluate the potential impact of retained locks, and whether a normal shutdown with the shutdown assist transaction can eliminate the need for immediate shutdowns.

See the *CICS Operations and Utility Guide* for more information about the shutdown assist transaction.

10.2.2 MVS CANCEL Command

We strongly recommend that you do not terminate CICS by using the MVS CANCEL command, because it does not allow CICS to commit or back out in-flight UOWs. A consequence of canceling a CICS region is that active locks held by in-flight tasks at the time of the CANCEL are converted to retained locks. In an RLS environment, this not only prevents other CICS regions from accessing the locked data but also prevents batch jobs from opening the data sets affected by retained locks.

10.3 Batch Processing of Online VSAM RLS Data Sets

In this section we cover the batch processing of VSAM data sets that are accessed by CICS in RLS mode. Figure 13 on page 88 gives an overview of the options that you have for sharing a data set among CICS and batch jobs.

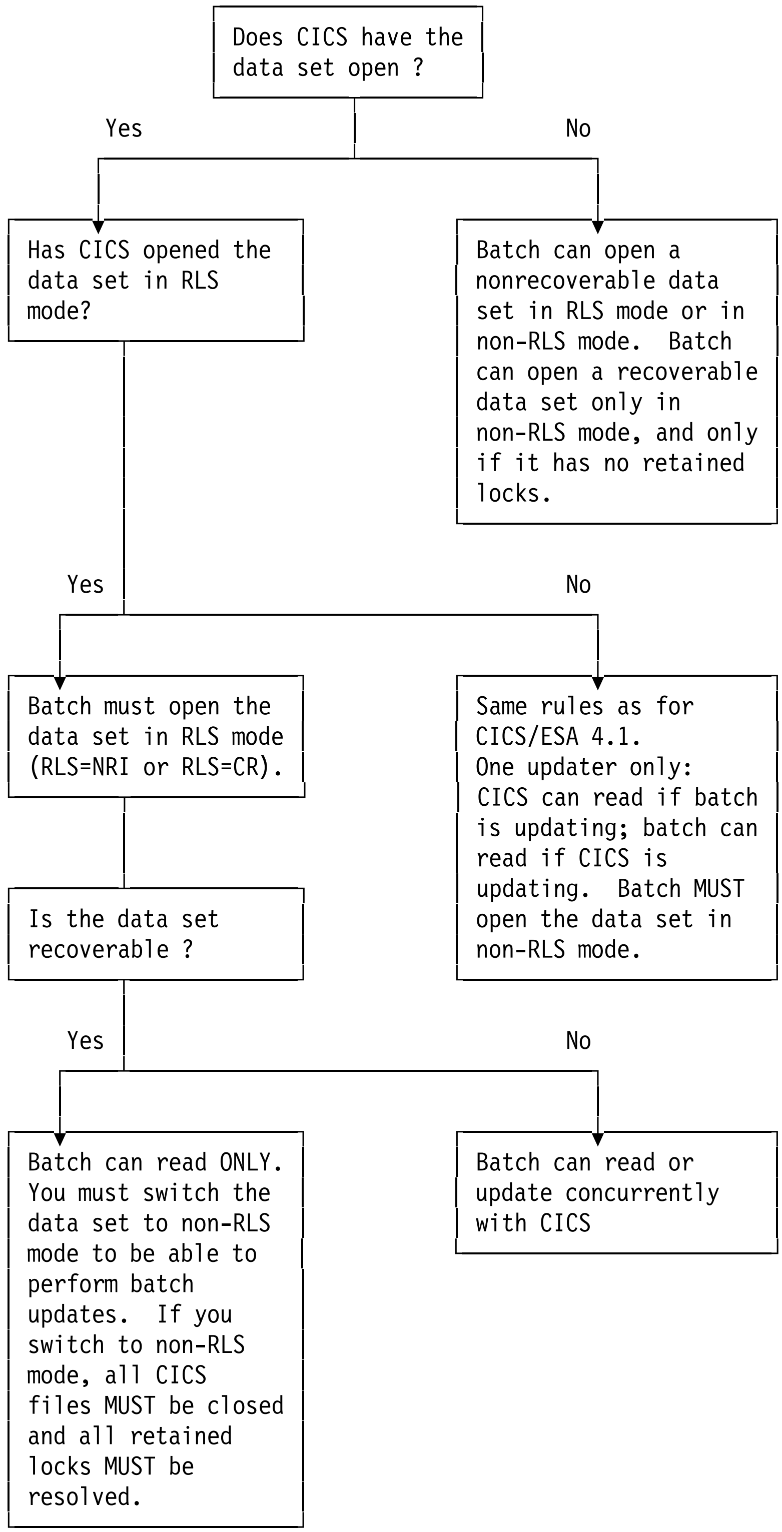

Figure 13. Sharing Data Sets Among CICS and Batch Jobs

For more information about the steps you must take to resolve retained locks in order to make a data set with retained locks available for use by a batch job, see *CICS and VSAM Record Level Sharing: Recovery Considerations* and the *CICS Recovery and Restart Guide*.

If a batch job requires only read access to a VSAM data set, it can run while the data set is open to CICS in RLS mode. Similarly the batch job can run if the data set is nonrecoverable, even if the batch job requires update access. However, if the batch job requires update access to a recoverable VSAM data set open in RLS mode, you must first switch the data set to non-RLS mode before the batch job can run.

10.3.1 Switching Data Sets from RLS Mode to Non-RLS Mode

In Figure 13 on page 88 we show that batch jobs cannot update recoverable VSAM data sets that are open in RLS mode. You must close the data set and then open it in non-RLS mode before the batch job can run. The recommended procedure for providing batch jobs with update access to recoverable VSAM data sets depends on whether CICS needs access to the data sets while the batch job is running.

10.3.1.1 No CICS Access to Data Set While Batch Is Running

The recommended procedure is:

1. Resolve any retained locks.
2. Quiesce the RLS data sets.
3. Optionally copy the data sets.
4. Run batch in non-RLS mode.
5. Copy the data sets.
6. Unquiesce the data sets.
7. Resume normal processing.

You must make a copy of the data set for recovery purposes after a batch run, regardless of whether you are switching from RLS to non-RLS access mode. You can optionally make a copy of the data set before the batch run. This is not required (you can forward recover the data set to the state it was in before the batch job ran) but speeds recovery if the batch job fails.

10.3.1.2 CICS Requires Access to Data Set While Batch Is Running

CICS can only read the data set while the batch job is updating it. The process is:

1. Resolve any retained locks.
2. Quiesce the RLS data sets.
3. Optionally copy the data sets.
4. Redefine the files as non-RLS and read-only in all relevant CICS systems. This requires reinstalling the file definitions.
5. Open the files for non-RLS, read-only processing in CICS.
6. Concurrently, run batch non-RLS.
7. When batch finishes, close the read-only, non-RLS files in CICS.
8. Reinstall the file definitions in CICS as RLS and update.
9. Unquiesce the data sets.
10. Open the files in CICS, if not using open at first reference.
11. Resume normal running.

10.3.2 Batch Read-Only Jobs

VSAM RLS enables read-with-integrity sharing between batch jobs and CICS applications. Batch jobs can share VSAM RLS recoverable files while they are being modified by CICS applications. This is possible because VSAM RLS provides the record locking and buffer coherency functions across the Parallel Sysplex.

10.4 MVS System Logger

All journals of previous releases of CICS are replaced by log streams. Redundant data must be deleted from log streams periodically so that the MVS system logger inventory entries do not exceed the limit of 168 log stream data sets per log stream. Typically, for a forward recovery log, deletion of old data is related to the data backup frequency.

10.5 Temporary Storage Queue Server Region

CICS temporary storage data sharing provides multiregion access to the nonrecoverable temporary storage queues for CICS TS.

Access to temporary storage queues by any CICS region within a Parallel Sysplex improves workload management of your CICS systems by removing affinities and simplifies the migration of existing CICS applications to the Parallel Sysplex environment.

As in previous releases of CICS, you can provide multisystem access for recoverable queues within the CICSplex by creating a QOR and function shipping temporary storage requests from the AOR to the QOR.

Although temporary storage data sharing queues are not recoverable, they are usually preserved across a CICS region restart or an MVS IPL, provided that the coupling facility is not stopped and does not fail.

CICS stores a set of temporary storage queues that you want to share in a temporary storage pool. Each temporary storage pool corresponds to a coupling facility list structure defined in the CFRM policy. You can create a single temporary storage pool or multiple temporary storage pools within the Parallel Sysplex, to suit your requirements. For example:

- You could create separate pools for specific purposes, such as a temporary storage pool for production, or a temporary storage pool for test and development.
- You could create more than one production pool, particularly if you have more than one coupling facility and you want to allocate temporary storage pool list structures to each coupling facility.

You must ensure that the temporary storage server region is activated before the CICS region needs it. A shared temporary storage pool consists of an XES list structure, which is accessed through a cross-memory queue server region. A shared temporary storage pool is started in an MVS image by starting up a queue server region for that pool as either a batch job or a started task. The batch job or started task invokes the queue server region program, DFHXQMN, which resides in an authorized program facility (APF) authorized library.

Chapter 11. Planning Checklist for VSAM Record-Level Sharing

To fully implement VSAM RLS, you must consider these planning items:

- Review your current use of VSAM data sets. There are some restrictions on accessing VSAM data sets in RLS mode. These are listed in 7.1.2, "Processing Restrictions" on page 47 and 8.5.2, "RLS Processing Restrictions" on page 71.
- Identify those data sets eligible for RLS access in your environment and decide which of them should be shared in RLS mode.
- Ensure that you have the correct levels of software to exploit VSAM RLS. See Chapter 3, "Prerequisite Hardware and Software" on page 9.
- Determine hardware, software, and availability requirements for the coupling facilities. See Chapter 5, "The Coupling Facility" on page 19.
- Evaluate applications to determine which can take advantage of VSAM RLS for their data.
- Map out the coupling facility cache structures; include number, sizes, and connectivity. See 7.3.2, "Structure Global Connectivity" on page 49 and 5.5.2, "Sizing Cache Structures" on page 30.
- Determine the size of the coupling facility lock structure. See 5.5.1, "Sizing the Lock Structure for Record-Level Sharing" on page 29.
- Map out the coupling facility list structures required to support the CICS log streams. See 5.3.5, "CICS TS Log Structure Allocation" on page 26.
- Size the coupling facility list structures for the CICS log streams. See 5.5.4, "CICS TS Logger Structure Sizings" on page 31.
- Plan the changes to MVS to define the coupling facility cache, lock, and list structures.
- Plan the changes to the SMS configuration; include base configuration definitions for coupling facility cache structures, new storage classes, storage group connectivity, and modifications to the ACS routines.
- Determine the size, number, and placement of SHCDs.
- Analyze the potential use of reserves on volumes where sharing control data sets will reside. Consider a conversion to enqueues.
- Plan new procedures to ensure data integrity with the coupling facility cache structure added to the storage hierarchy.
- Set up new operational procedures for coupling facility management.
- Add new parameters to SYS1.PARMLIB to control deadlock and SMF processing.
- Establish new authorization levels required for VSAM RLS access and support.
- Determine which VSAM data sets are eligible for RLS. Use IDCAMS DEFINE or ALTER to specify the LOG attribute for each eligible data set, declaring whether the data set is recoverable or nonrecoverable. If you specify LOG(ALL) to say that the data set is recoverable, use the LOGSTREAMID attribute to specify the forward recovery log stream for this data set. You

can also set the Backup While Open (BWO) attribute to TYPECICS if you have specified LOG(ALL).

- Establish application procedures for VSAM RLS; include changes to VSAM KSDS cluster definitions and procedures for batch jobs that update recoverable VSAM spheres.
- Resolve coexistence issues for DFSMS/MVS 1.3. See 7.4.1, "Determine Coexistence Requirements" on page 51 and *DFSMS/MVS Version 1 Release 3 Planning for Installation*.
- Have fallback and recovery procedures in place.

  Note: Fallback from VSAM RLS processing has CICS and SMS implications and changes:

  – If you are falling back to different CICS, CICSVR, or DFSMS/MVS release levels, see "Understanding the Product Environment for VSAM RLS" in the *DFSMSdfp Storage Administration Reference* and "Fallback Planning Considerations" in the *CICS Migration Guide*.

  – For falling back from RLS to non-RLS mode access, see the *DFSMSdfp Storage Administration Reference*.

Appendix A. Classroom Education

Chapter 2, “VSAM Record-Level Sharing” on page 3 states the importance of being prepared and the role education plays in a successful implementation of CICS TS, DFSMS/MVS 1.3, and VSAM RLS. In this appendix we describe the course that is available to help with your education.

A.1 CICS Transaction Server for OS/390 Planning and Transition

Course Code U3965

Duration 3 Days

A.1.1 Description

This course gives you a comprehensive look at the new features provided by CICS TS, such as CICS log manager, CICS recovery manager, VSAM RLS, RDO for transient data, temporary storage data sharing, and database management support.

To ease the migration from CICS/ESA 4.1, this course provides release planning information in the areas of operating system integration, resource definition changes, planning for recovery, and hardware and software planning.

A.1.2 Objectives

After completing this course, you should be able to develop a CICS TS migration plan that includes:

- System programming requirements
- Application programming extensions
- Parallel processing considerations
- Recovery considerations
- Hardware and software requirements

A.1.3 Prerequisites

Students should have one to two years of CICS experience in an MVS environment and some exposure to MVS in OS/390. Students should have completed the following courses or have equivalent knowledge:

- CICS/ESA Basic Tailoring (U3905)
- CICS/ESA Implementation of Intercommunication Facilities (U3913), if MRO or ISC is installed
- CICS DB2 Interface for System Programmers (U3924), if DB2 is attached to CICS

The following courses or equivalent knowledge will be helpful but are not required:

- S/390 Parallel Sysplex Planning (H3996)
- DFSMS/MVS Version 1.3 Update (SS120)

Appendix B. Special Notices

This publication is intended to help those who are responsible for planning for the implementation of a VSAM record-level sharing environment based on CICS Transaction Server for OS/390 Version 1 Release 1 and DFSMS/MVS Version 1 Release 3. The information in this publication is not intended as the specification of any programming interfaces that are provided by CICS Transaction Server for OS/390 Version 1 Release 1 or DFSMS/MVS Version 1 Release 3. See the PUBLICATIONS section of the IBM Programming Announcement for CICS Transaction Server for OS/390 Version 1 Release 1 and DFSMS/MVS Version 1 Release 3 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

ACF/VTAM	AD/Cycle
C/370	CICS
CICS/ESA	CICS/MVS
CICS/VSE	CICSPlex
COBOL/370	DATABASE 2
DB2	DFSMS
DFSMS/MVS	DFSMSdfp
DFSMSdss	DFSMShsm
Enterprise System/9000	ES/9000
Hiperbatch	Hiperspace
IBM	IMS
IMS/ESA	Language Environment
MQ	MQSeries
MVS	MVS/DFP
MVS/ESA	MVS/SP
MVS/XA	NetView
OS/2	OS/390
Parallel Sysplex	PR/SM
RACF	RMF
S/390	SAA
SP	System/390
VTAM	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

C++ is a trademark of American Telephone and Telegraph Company, Incorporated

Other trademarks are trademarks of their respective companies.

Appendix C. Related Publications

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

C.1 International Technical Support Organization Publications

For information about ordering these ITSO publications, see "How To Get ITSO Redbooks" on page 101.

- *MVS/ESA SP Version 5 Sysplex Migration Guide Version 5.1.0*, SG24-4581
- *Parallel Sysplex Capacity Planning*, SG24-4680
- *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*, GG24-4286
- *CICS and VSAM Record Level Sharing: Implementation Guide*, SG24-4766 (available January 1997)
- *CICS and VSAM Record Level Sharing: Recovery Considerations*, SG24-4768 (available January 1997)
- *OS/390 Parallel Sysplex Configuration Cookbook*, SG24-4706 (available February 1997)

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection (available soon)	SBOF-7250	SK2T-8042

C.3 Other Publications

These publications are also relevant as further information sources.

- *CICS DBCTL Guide*, SC33-1700
- *CICS Installation Guide*, GC33-1681
- *CICS Intercommunication Guide*, SC33-1695
- *CICS Messages and Codes*, GC33-1694
- *CICS Migration Guide*, GC33-1571
- *CICS Operations and Utility Guide*, SC33-1685
- *CICS RACF Security Guide*, SC33-1701

- *CICS Recovery and Restart Guide*, SC33-1698
- *CICS Release Guide*, GC33-1570
- *CICS Shared Data Tables Guide*, SC33-1702
- *CICS System Definition Guide*, SC33-1682
- *CICSPlex System Manager for MVS/ESA Concepts and Planning*, GC33-0786
- *IBM DATABASE2 for MVS/ESA V4 Data Sharing: Planning and Installation*, SC26-3269
- *DFSMS/MVS Version 1 Release 3 Access Method Services for ICF*, SC26-4906-02
- *DFSMS/MVS Version 1 Release 3 DFSMSdfp Advanced Services*, SC26-4921-02
- *DFSMS/MVS Version 1 Release 3 DFSMSdfp Storage Administration Reference*, SC26-4920-03
- *DFSMS/MVS Version 1 Release 3 DFSMSdss Storage Administration Guide*, SC26-4930
- *DFSMS/MVS Version 1 Release 3 DFSMShsm Storage Administration Guide*, SH21-1076
- *DFSMS/MVS Version 1 Release 3 Macro Instructions for Data Sets*, SC26-4913
- *DFSMS/MVS Version 1 Release 3 Using Data Sets*, SC26-4922-02
- *DFSMS/MVS Version 1 Release 3 Planning for Installation*, SC26-4919-03
- *DFSMS/MVS Version 1 Release 1 Planning for Installation*, SC26-4919
- *Dynamic Transaction Routing in a CICSPlex*, SC33-1012
- *IBM COBOL for MVS and VM Programming Guide*, SC26-4767
- *Language Environment for MVS and VM Release 5 Run-time Migration Guide*, SC26-8232
- *MVS/ESA Planning: Installation and Migration for MVS/ESA System Product Version 4*, GC28-1077-02
- *MVS/ESA Initialization and Tuning Reference*, SC28-1452
- *MVS/ESA Planning: Installation and Migration with JES2 MVS/ESA System Product Version 5*, GC28-1428
- *MVS/ESA Planning: Workload Management*, GC28-1493
- *MVS/ESA Programming: Assembler Services Guide*, GC28-1466
- *MVS/ESA Programming: Assembler Services Reference*, GC28-1474
- *MVS/ESA Setting Up a Sysplex*, GC28-1449
- *MVS/DFP Version 3.3 Planning Guide*, SC26-4561
- *PL/I MVS and VM Version 1 Release 1 Migration Guide*, SC26-3118
- *RACF V2 System Programmer's Guide*, SC23-3725-01
- *System/390 MVS Sysplex Application Migration*, GC28-1211
- *System/390 MVS Sysplex Hardware and Software Migration*, GC28-1210-01
- *System/390 MVS Sysplex Overview: Introducing Data Sharing and Parallelism in a Sysplex*, GC28-1208

- *VTAM V4R2 Network Implementation Guide*, SC31-6494

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **ITSO4USA category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Glossary

ACID properties. The term used to denote the properties of a transaction:

Atomicity A transaction′s changes to the state (of resources) are atomic: either all happen or none happens.

Consistency A transaction is a correct transformation of the state. The actions taken as a group do not violate any of the integrity constraints associated with the state.

Isolation Even though transactions execute concurrently, they appear to be serialized. In other words, it appears to each transaction that any other transaction executed either before it or after it.

Durability After a transaction completes successfully (commits), its changes to the state survive failures.

Note: In CICS, the ACID properties apply to a unit of work (UOW). See also *unit of work*.

alternate index. In systems with VSAM, a collection of index entries related to a given base cluster and organized by an alternate key, that is, a key other than the prime key of the associated base cluster data records. It gives an alternate directory for finding records in the data component of a base cluster.

base configuration. The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration is used to manage.

cache structure. A coupling facility structure that enables high-performance sharing of cached data by multisystem applications in a sysplex. Applications can use a cache structure to implement several different types of caching systems, including a store-through or a store-in cache.

central electronic complex (CEC). Synonym for central processor complex (CPC).

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load, and other machine operations.

central processor complex (CPC). A physical collection of hardware that includes main storage, one or more central processors, timers, and channels.

CICSplex. A group of connected CICS regions

cluster. See *VSAM cluster*.

coupling facility. A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

cross-system coupling facility (XCF). XCF is a component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.

data set. In DFSMS/MVS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In non-OpenEdition/MVS environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*.

DFSMS/MVS. An IBM licensed program that together with MVS/ESA SP makes up the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm.

DFSMSdfp. A DFSMS/MVS functional component that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS/MVS functional component used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS/MVS functional component used for backing up and recovering data and managing space on volumes in the storage hierarchy.

DFSMSrmm. A DFSMS/MVS functional component that manages removable media.

dirty read. A read request that does not involve any locking mechanism and may obtain invalid data—that is, data that has been updated but is not yet committed by another task. This could also apply to data that is about to be updated and will be invalid by the time the reading task has completed.

For example, if one CICS task rewrites an updated record, another CICS task that issues a read before the updating task has taken a sync point will receive the uncommitted record. This data could subsequently be backed out if the updating task fails, and the read-only task would not be aware that it had received invalid data.

See also *read integrity*.

file. A collection of information treated as a unit. In non-OpenEdition/MVS environments, the terms *data* *set* and *file* are generally equivalent and sometimes may be used interchangeably. See also *data set*.

global resource serialization. A function that provides an MVS serialization mechanism for resources (typically data sets) across multiple MVS images.

heuristic decision. A decision that enables a transaction manager to complete a failed in-doubt unit of work (UOW) that cannot wait for resynchronization after recovery from the failure.

Under the two-phase-commit protocol, the loss of the coordinator (or loss of connectivity) that occurs while a UOW is in doubt theoretically forces a participant in the UOW to wait forever for resynchronization. While a subordinate waits in doubt, resources remain locked and, in CICS/ESA, the failed UOW is shunted pending resolution.

Applying a heuristic decision provides an arbitrary solution for resolving a failed in-doubt UOW as an alternative to waiting for the return of the coordinator. In CICS, the heuristic decision can be made in advance by specifying in-doubt attributes on the transaction resource definition. These in-doubt attributes specify:

- Whether or not CICS is to wait for proper resolution or take heuristic action (defined by WAIT(YES) or WAIT(NO), respectively)
- The heuristic action that CICS is to take for the WAIT(NO) case (or is to take after the WAITTIME has expired, if a time other than zero is specified):
  - Back out all changes made by the UOW
  - Commit all changes made by the UOW

The heuristic decision can also be made by an operator when a failure occurs, and it can be communicated to CICS with an API or operator command interface (such as CEMT SET UOW).

in doubt. In CICS, the state at a particular point in a distributed UOW for which a two-phase-commit sync point is in progress. The distributed UOW is said to be in doubt when a subordinate recovery manager (or transaction manager):

- Has replied (voted) in response to a PREPARE request
- Has written a log record of its response to signify that it has entered the in-doubt state
- Does not yet know the decision of its coordinator (to commit or to back out).

The UOW remains in doubt until the coordinator issues either the commit or back-out request as a result of responses received from all UOW participants. If the UOW is in the in-doubt state, and a failure occurs that causes loss of connectivity between a subordinate and its coordinator, it remains in doubt until either:

- Recovery from the failure has taken place and synchronization can resume

  or

- The in-doubt waiting period is terminated by some built-in controls, and an arbitrary (heuristic) decision is then taken (to commit or back out).

Note: In theory, a UOW can remain in doubt forever if a UOW participant fails or loses connectivity with a coordinator and is never recovered (for example, if a system fails and is not restarted). In practice, the in-doubt period is limited by attributes defined in the transaction resource definition associated with the UOW. After expiry of the specified in-doubt wait period, the recovery manager commits or backs out the UOW, according to the UOW′s in-doubt attributes.

For cases where data integrity is of paramount importance, CICS supports “wait forever,” indicated by a WAITTIME of zero, in which case manual intervention is required to force a heuristic decision.

See also *two-phase commit* and *heuristic decision*.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS/MVS that gives users and storage administrators access to the storage management functions.

list structure. A coupling facility structure that enables multisystem applications in a sysplex to share information organized as a set of lists or queues. A list structure consists of a set of lists and an optional lock table, which can be used for serializing resources in the list structure. Each list consists of a queue of list entries.

lock structure. A coupling facility structure that enables applications in a sysplex to implement customized locking protocols for serialization of application-defined resources. The lock structure supports shared, exclusive, and application-defined lock states, as well as generalized contention management and recovery protocols.

logical unit of work (LUW). Old term used to describe a unit of work in earlier releases of CICS. The preferred term, adopted by CICS Transaction Server for OS/390 Version 1 Release 1, is *unit of work (UOW).* UOW is used as a keyword in a number of CICS interfaces in CICS TS.

See *unit of work*.

microprocessor. A processor implemented on one or a small number of chips.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

MVS system. An MVS image and its associated hardware, which collectively are often referred to simply as a system, or MVS system.

Parallel Sysplex. An MVS sysplex where all MVS system images are linked through a coupling facility.

read integrity. An attribute of a read request that ensures the integrity of the data passed to a program that issues a read-only request. CICS recognizes two forms of read integrity:

- Consistent
  A program is permitted to read only committed data—data that cannot be backed out after it has been passed to the program issuing the read request. Therefore, a consistent read request can succeed only when the data is free from all locks.
- Repeatable
  A program is permitted to issue multiple read-only requests, with repeatable read integrity, and be assured that none of the records passed can subsequently be changed until the end of the sequence of repeatable read requests. The sequence of repeatable read requests ends either when the transaction terminates, or when it takes a sync point, whichever is the earlier.

Contrast with *dirty read*.

record-level sharing. See *VSAM record-level sharing*.

sharing control data set (SHCDS). A VSAM linear data set that contains information DFSMS/MVS needs to ensure the integrity of the data sharing environment if the VSAM record-level sharing locks are lost.

source control data set (SCDS). A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated by using ISMF.

sphere. See *VSAM sphere*.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. See also *MVS system*, *Parallel Sysplex*.

two-phase commit. In CICS, the protocol observed when taking a sync point in a distributed UOW. At sync point, all updates to recoverable resources must be either committed or backed out. At this point, the coordinating recovery manager gives each subordinate participating in the UOW an opportunity to vote on whether its part of the UOW is in a consistent state and can be committed. If all participants vote "yes," the distributed UOW is committed. If any votes "no," all changes to the distributed UOW's resources are backed out.

This protocol is called the two-phase commit because there is a "voting" phase (the prepare phase), followed by the actual commit phase:

1. Prepare
   Coordinator invokes each UOW participant, asking each one if it is prepared to commit.
2. Commit
   If all UOW participants acknowledge that they are prepared to commit (vote "yes"), the coordinator issues the commit request.

   If only one UOW participant is not prepared to commit (votes "no"), the coordinator issues a back-out request to all.

unit of work (UOW). A sequence of processing actions (database changes, for example) that must be completed before any of the individual actions performed by a transaction can be regarded as committed. After changes are committed (by successful completion of the UOW and recording of the sync point on the system log), they become durable and are not backed out in the event of a subsequent failure of the task or system.

The beginning and end of the sequence may be marked by:

- Start and end of transaction, when there are no intervening sync points
- Start of task and a sync point
- A sync point and end of task
- Two sync points

Thus a UOW is completed when a transaction takes a sync point, which occurs either when a transaction issues an explicit sync-point request or when CICS takes an implicit sync point at the end of the transaction. In the absence of user sync points explicitly taken within the transaction, the entire transaction is one UOW.

In earlier releases of CICS, a UOW was referred to as a *logical unit of work (LUW)*.

VSAM cluster. A named structure consisting of a group of related components. For example, when the data is key sequenced, the cluster contains both the data and the index components.

VSAM record-level sharing (VSAM RLS). An extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. Record-level sharing uses the System/390 coupling facility to provide cross-system locking, local buffer invalidation, and cross-system data caching.

VSAM sphere. The collection of all component data sets associated with a given VSAM base data set—the base, index, alternate indexes, and alternate index paths.

List of Abbreviations

ACB	access control block
AOR	application-owning region
APF	authorized program facility
API	application programming interface
APPC	Advanced Program-to-Program Communication
ARM	Automatic Restart Manager
ATI	automatic transaction initiation
BMP	batch message processing
BWO	Backup While Open
CBIC	control blocks in common
CF	coupling facility
CFCC	coupling facility control code
CFRM	coupling facility resource management
CI	control interval
CICS	Customer Information Control System
CICSVR	CICS VSAM Recovery
CICS TS	Customer Information Control System Transaction Server
CMOS	complementary metal-oxide semiconductor
CMT	CICS-maintained data table
CP	central processor
CPC	central processor complex
CR	consistent read
CSD	CICS system definition
DBCTL	database control
DB2	DATABASE 2
DDM	Distributed Data Management
DOR	data-owning region
DRA	database resource adapter
DTR	dynamic transaction routing
ENF	event notification facility
ESDS	entry-sequenced data set
FCT	file control table
FOR	file-owning region
GRS	global resource serialization
GSR	global shared resources
HARBA	hi-allocated relative byte address
HLL	high-level language
HURBA	hi-used relative byte address
IBM	International Business Machines Corporation
ICF	integrated catalog facility
ICMF	Integrated Coupling Migration Facility
IMS	Information Management System
IPL	initial program load
ISMF	Interactive Storage Management Facility
ITSO	International Technical Support Organization
KSDS	key-sequenced data set
LP	logical partition
LPA	link pack area
LSR	local shared resources
MCT	monitoring control table
MLPA	modified link pack area
MRO	multiregion operation
MVS	multiple virtual storage
NSR	nonshared resources
NRI	no read integrity
ONC RPC	open network computing remote procedure call
PR/SM	processor resource/system manager
QOR	queue-owning region
RBA	relative byte address
RCT	resource control table
RDO	resource definition online
RLS	record level sharing
RMI	resource manager interface
RRDS	relative record data set
SCDS	source control data set
SDT	shared data table
SFM	sysplex failure management
SHCDS	sharing control data set
SIT	system initialization table
SPI	system programming interface
SMS	system managed storage
STC	started tasks
TCB	task control block
TOR	terminal-owning region
UBF	user buffering
UOW	unit of work

UPS uninterruptible power supply

URM user replaceable module

VRRDS variable length relative record data set

VSAM Virtual Storage Access Method

VVDS VSAM volume data sets

WLM workload management

XCF cross-system coupling facility

XES cross-system extended services

XPI exit programming interface

Index

T

U

V

X

Printed in U.S.A.

SG24-4765-00