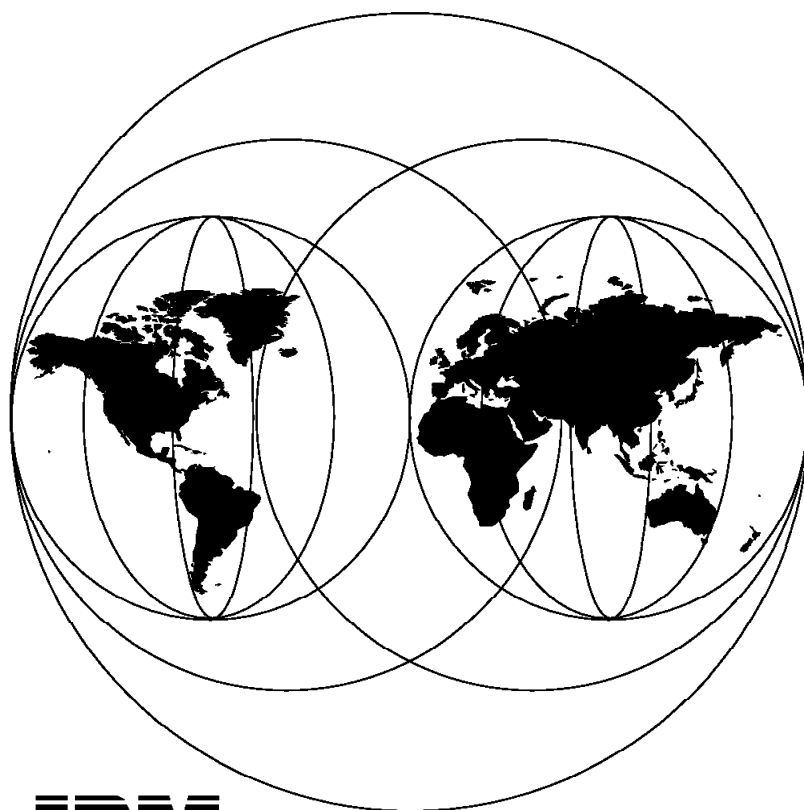


CICS and VSAM Record Level Sharing: Implementation Guide

April 1997



IBM

**International Technical Support Organization
San Jose Center**



International Technical Support Organization

SG24-4766-00

**CICS and VSAM Record Level Sharing:
Implementation Guide**

April 1997

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 169.

First Edition (April 1997)

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

Warning

This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. It is recommended that, when the product becomes generally available, you destroy all copies of this version of the book that you have in your possession.

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

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Preface	ix
How This Redbook Is Organized	ix
The Team That Wrote This Redbook	x
Comments Welcome	xi
Chapter 1. Introduction to CICS Transaction Server for OS/390	1
1.1 What Is CICS Transaction Server for OS/390, Version 1 Release 1?	1
1.2 The CICS Log Manager	2
1.2.1 Why Perform Logging	2
1.2.2 Log Streams and Log Structures	3
1.2.3 Managing the MVS Log Streams	3
1.2.4 Benefits of the CICS Log Manager	4
1.3 The CICS Recovery Manager	4
1.3.1 UOW Control	4
1.3.2 UOW Shunting	6
1.3.3 Benefits of the CICS Recovery Manager	6
1.4 The CICS Enqueue Domain	6
1.4.1 Resource Owners	7
1.4.2 Enqueue Requests	7
1.4.3 Supported Locks	7
1.4.4 CICS TS Enqueue Pools	7
1.5 VSAM Record Level Sharing	8
1.5.1 The SMSVSAM Server Address Space	8
1.5.2 Logging through CICS	8
1.5.3 Access Mode	9
1.5.4 Recovery Attributes	9
1.5.5 Sharing Control Data Set	9
1.5.6 Locks	9
1.5.7 Read Integrity	9
1.5.8 Restrictions for VSAM Record Level Sharing	10
1.5.9 Restrictions for Concurrent Access	10
1.6 Temporary Storage Data Sharing	11
1.6.1 Temporary Storage Server DFHXQMN	12
1.6.2 Temporary Storage Pool	12
1.6.3 CICS to Temporary Storage Server Connection	13
1.6.4 Managing Temporary Storage Pools	13
1.6.5 Application Considerations	13
1.6.6 Benefits	13
1.7 Resource Definition Online for Transient Data	14
1.7.1 RDO Support	14
1.7.2 New Options for Transient Data	14
1.7.3 Transient Data Queue Management	14
1.7.4 Benefits	15
1.8 New EXEC CICS CREATE Command	15
1.8.1 COBOL Example of EXEC CICS CREATE	15
1.8.2 Miscellaneous	16
1.8.3 Benefits	16
1.9 Database Management	17
1.9.1 IMS Database Support	17
1.9.2 DATABASE 2 Support	18
1.9.3 Summary	18

1.10	VTAM Generic Resource	18
1.11	Monitoring and Statistics	19
1.11.1	Performance-Class Data	19
1.11.2	Statistics	19
1.11.3	Statistics Utility Program	20
1.11.4	Statistics Sample Program	20
1.11.5	Summary	20
1.12	Miscellaneous Changes	20
1.12.1	Autoinstall for Virtual Terminals	20
1.12.2	Shippable Terminals	20
1.12.3	START Affinity Parameter	21
Chapter 2. Introduction to System-Managed Storage		23
2.1	The DFSMS Environment	23
2.2	Letting the System Manage Your Data	23
2.2.1	Benefits of SMS	24
2.3	SMS Classes and Groups	25
2.3.1	Using SMS Classes and Groups	25
2.4	The Storage Hierarchy	27
2.5	SMS and VSAM Record-Level Sharing	28
Chapter 3. Prerequisite Products		29
3.1	Hardware Prerequisites	29
3.2	MVS/ESA Operating System	29
3.3	DFP Environment	30
3.4	Database Products	30
3.4.1	IMS/ESA (5665-408)	30
3.4.2	DB2 (5665-DB2)	30
3.5	IBM MQSeries for MVS/ESA	30
3.6	IBM Telecommunications Access Methods	30
3.7	External Security Manager	31
3.8	Forward Recovery Software	31
3.9	Programming Languages	31
3.9.1	Execution-Time Support	31
3.9.2	Storage Protection—OS/VS COBOL Programs	32
3.10	Batch Support	32
3.10.1	PL/I	33
3.10.2	COBOL	34
Chapter 4. CICS Migration		35
4.1	Migration Environment	35
4.2	Prerequisites	36
4.3	Migrating to CICS TS	37
4.4	Implement Logging for CICS System Logs	39
4.5	Migrate Your CICSplex	40
4.5.1	Common Tasks	40
4.5.2	CICS Region Migration	41
4.5.3	Migrating the File-Owning Region	43
4.5.4	Migrating the Application-Owning Regions	45
4.5.5	Migrating the Terminal-Owning Region	45
4.5.6	Migrating the Queue-Owning Region	45
4.5.7	Final Steps	47
4.6	Implementing RLS	48
4.6.1	Preparing for RLS	48
4.6.2	Verify RLS Operation	49

4.6.3 Changing Access Mode	51
4.7 Mixed Environment Discussion	51
4.8 Miscellaneous Information	52
4.8.1 New and Changed Exits	52
4.8.2 New and Changed SIT Parameters	54
4.8.3 New ABEND Codes	55
Chapter 5. CICS Logging and Journaling	57
5.1 Overview of Logging and Journaling	57
5.2 Setting Up System Logs	58
5.2.1 Defining a Dummy System Log to CICS	58
5.2.2 Defining the MVS System Logger Components	59
5.2.3 Defining the CICS Log Manager Components	70
5.3 Setting Up General Logs	74
5.3.1 Defining User Journals and Autojournals	74
5.3.2 Defining the Forward Recovery Logs	74
5.3.3 Creating General Logs	74
5.3.4 Defining the Log of Logs	74
5.4 Management of CICS Logs and Journals	75
5.4.1 Availability of the MVS System Logger	75
5.4.2 CICS Logging Operations	75
5.4.3 Reading Log Stream Offline	76
5.4.4 Management of DASD Log Data Sets	78
5.4.5 Management of Staging Data Sets	78
5.5 Monitoring Log Streams	79
5.5.1 Online Statistics	79
5.5.2 MVS Log Streams Statistics	81
Chapter 6. Implementing RLS	89
6.1 A Brief Summary of VSAM Record Level Sharing	89
6.1.1 Summary of Prerequisites	90
6.2 Implementing VSAM Record Level Sharing	90
6.2.1 Selecting Data Sets for RLS Access Mode	90
6.2.2 Review of DFSMS 1.3 Related Tasks	91
6.2.3 Enable RLS Access Mode	91
6.2.4 Read Integrity	95
6.2.5 Plan for Fallback	95
6.2.6 Sharing Data Sets between CICS and Batch	95
6.3 RLS Failure Scenarios	98
Chapter 7. Coupling Facility Preparation	101
7.1 Coupling Facility Data Sets and Policies	101
7.1.1 Coupling Facility Resource Manager	101
7.1.2 MVS System Logger	103
7.2 Naming Convention	104
7.2.1 Log Streams	104
7.2.2 TS Pools	105
7.3 Log Stream Sizing	105
7.3.1 DFHLOG Log Streams	106
7.3.2 DFHSHUNT Log Streams	108
7.3.3 Log Streams of the Type Block Write Forced Periodically	109
7.3.4 Log Streams of the Type Block Write Not Forced	111
7.4 Temporary Storage Pool Sizing	111
7.5 Activating Definitions	111
7.6 MVS Commands	111

Chapter 8. Temporary Storage Data Sharing Servers	113
8.1 Setting Up the Server	113
8.1.1 Defining the AXM Subsystem	113
8.1.2 Defining the Temporary Storage Pool Structure	114
8.1.3 Defining Temporary Storage Server Regions	114
8.1.4 Connecting to CICS Regions	121
8.1.5 Defining Security Access	121
8.2 Monitoring	123
8.2.1 CEMT Transaction	123
8.2.2 Temporary Storage Server Statistics	123
8.2.3 Programs DFHSTUP and DFH0STAT	125
8.3 Customization	126
Chapter 9. CICS Utilities	127
9.1 Monitoring Log Streams	128
9.2 Transaction Affinities Utility	128
9.3 New Transactions	128
9.4 Shutdown Assist Program	129
9.5 Recovery Manager Utility Program	131
9.6 Batch Access to RLS Data Sets	134
9.6.1 Sample Programs to Allow the Switch from RLS to Non-RLS	134
9.6.2 Installation and Customization	136
9.7 Trace Utility Print Program (DFHTU510)	137
9.7.1 Trace Format	137
9.8 New Enqueue Function	143
9.8.1 Enqueue Pools	143
9.9 Resolving Enqueue Deadlocks	145
9.9.1 Example 1 of Resolving Enqueue Deadlocks	145
9.9.2 Example 2 of Resolving Enqueue Deadlocks	145
9.9.3 Example 3 of Resolving Enqueue Deadlocks	146
9.9.4 Example 4 of Resolving Enqueue Deadlocks	147
9.9.5 Example 5 of Resolving Enqueue Deadlocks	148
Chapter 10. Modifying SMS for RLS	149
10.1 Sharing Control Data Set (SHCDS)	149
10.2 Coupling Facility Structures	151
10.3 SMS Configuration Changes	153
10.4 SMS Storage Class Changes	154
10.5 Parameter Library (Parmlib) Changes	156
10.6 Resource Access Control Facility (RACF)	157
10.7 SMSVSAM	157
10.8 New Storage Management Commands	158
10.8.1 Access Method Service (AMS) Commands	163
10.9 Fallback Procedure	163
10.10 New Catalog Information	163
10.11 Maintenance and Messages	164
10.12 Step by Step Guide	165
Appendix A. System Measurement Facility and RLS	167
Appendix B. Special Notices	169
Appendix C. Related Publications	171
C.1 International Technical Support Organization Publications	171
C.2 Redbooks on CD-ROMs	171

C.3 Other Publications	171
How to Get ITSO Redbooks	173
How IBM Employees Can Get ITSO Redbooks	173
How Customers Can Get ITSO Redbooks	174
IBM Redbook Order Form	175
Glossary	177
List of Abbreviations	181
Index	183

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 implement VSAM RLS. You can also find detailed planning, installation and recovery information in the product manuals and in the following redbooks which are companion volumes to this book:

- *CICS and VSAM Record Level Sharing: Planning Guide (SG24-4765)*
- *CICS and VSAM Record Level Sharing: Recovery Considerations (SG24-4768)*
Not available until the second quarter of 1997.

How This Redbook Is Organized

The redbook is organized as follows:

- Chapter 1, "Introduction to CICS Transaction Server for OS/390"
In this chapter, we give a short overview of CICS Transaction Server for OS/390, Version 1 Release 1.
- Chapter 2, "Introduction to System-Managed Storage"
In this chapter, we introduce system-managed storage. System-managed storage is the IBM automated approach to managing auxiliary storage. It uses software programs to manage data security, placement, migration, backup, recall, recovery, and deletion to ensure that current data is available when needed, and obsolete data is removed from storage.
- Chapter 3, "Prerequisite Products"
This chapter gives information about the hardware and software environments required to use CICS Transaction Server for OS/390, Version 1 Release 1.
- Chapter 4, "CICS Migration"
In this chapter, we describe the migration steps we performed to implement CICS TS with VSAM RLS. We also discuss miscellaneous information we found useful during migration.
- Chapter 5, "CICS Logging and Journaling"
This chapter describes the CICS TS logging and journaling facilities.
- Chapter 6, "Implementing RLS"
In this chapter, we provide the information about the tasks you must perform in order to access your VSAM data sets (or some of them) in RLS mode.
- Chapter 7, "Coupling Facility Preparation"
In this chapter, we describe definitions of resources and policies to set up coupling facility to use RLS.
- Chapter 8, "Temporary Storage Data Sharing Servers"

This chapter describes the new temporary storage data sharing feature of CICS TS.

- Chapter 9, “CICS Utilities”

This chapter describes the new or changed utilities supplied by CICS.

- Chapter 10, “Modifying SMS for RLS”

In this chapter, we describe the tasks the Storage Administrator should perform to enable and maintain VSAM RLS. We assume that you have already installed DFSMS 1.3 and are familiar with the SMS constructs and classes. We do not provide a guide for implementing SMS.

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. He 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 preparing and holding workshops and meetings for CICS customers throughout Germany. He 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 *A Comparison of 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 On-Line Transaction Processing field. His areas of expertise include MVS, CICS, CICSplex SM, MQ Series, and application architectures.

Byron Jones is a Storage Analyst with 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.

Toru Yamazaki is an Advisory Storage Specialist at the International Technical Support Organization, San Jose Center. He substantially edited and completed this redbook after the end of Hugh's assignment.

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

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Use the electronic evaluation form found on the Redbooks Home Pages at the following URLs:

For Internet users	http://www.redbooks.ibm.com
For IBM Intranet users	http://w3.itso.ibm.com/redbooks

- Send us a note at the following address:

redbook@vnet.ibm.com

Chapter 1. Introduction to CICS Transaction Server for OS/390

In this chapter we give a short overview of the Customer Information Control System's (CICS) Transaction Server for OS/390, covering:

- Description of the CICS Transaction Server for OS/390, Version 1 Release 1
- The CICS log manager
- The CICS recovery manager
- The CICS enqueue domain
- VSAM record-level sharing
- Temporary Storage Data Sharing
- Resource definition online for Transient Data
- The new EXEC CICS CREATE command
- Database management changes
- VTAM generic resource changes
- Monitoring and statistics
- Miscellaneous changes

1.1 What Is CICS Transaction Server for OS/390, Version 1 Release 1?

CICS Transaction Server for OS/390, Version 1 Release 1 (CICS TS) provides an easy-to-use S/390 client/server package. It includes a CICS server, a CICS client, and the CICS management functions 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 available only as part of CICS TS. This base element includes features and products available with prior CICS versions:

- CICS Web Interface
- Open Network Computing (ONC), Remote Procedure Call (RPC)
- IBM CICS Transaction Affinities Utility

We refer to this base element as CICS, or CICS TS when we need to differentiate it from earlier releases of CICS.

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)

1.2 The CICS Log Manager

The CICS log manager is a new CICS domain, introduced in CICS TS. It replaces the journal control management function of previous CICS releases, and is responsible for all logging done by CICS. The CICS log manager works with the MVS system logger, which is introduced in MVS/ESA 5.2, to provide a focal point for all system log, forward recovery log, and user journal output within a single MVS sysplex. It provides flexible mapping of CICS journals to MVS log streams, which you can use, for example, to merge all the forward recovery logs for a given VSAM data set from many CICS systems on several MVS images. The MVS system logger also enables faster CICS restart, dual logging, and flexible log and journal archiving.

Figure 1 shows an overview of the CICS log manager.

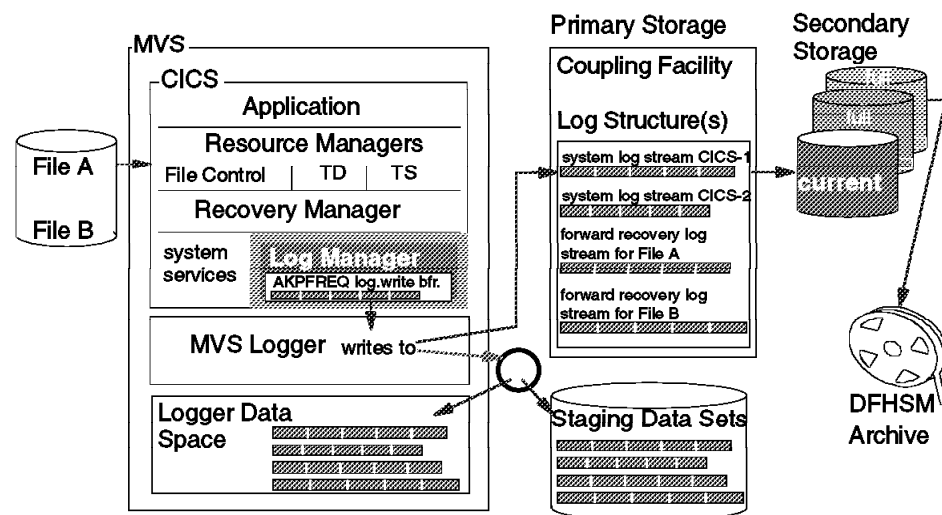


Figure 1. Overview of CICS Logging

1.2.1 Why Perform Logging

Log records held on the system log allow CICS to back out recoverable resources that have been updated by a transaction that fails at some time during its life. If the transaction fails, CICS is able to restore the resources to the state they were in before the transaction started. This is known as dynamic backout. In earlier releases of CICS, the information required for dynamic backout was stored in CICS storage, as well as on the CICS system log. In CICS TS, there is no in-storage dynamic log. All log records required for backout are written only to the CICS system log.

If CICS fails, all transactions that are in-flight at the time CICS fails must be backed out. The data needed for backout must be written to a medium that is independent of the failure of CICS; otherwise the data needed for backout would be lost if CICS fails. CICS TS achieves this by using the MVS system logger. In earlier releases of CICS, log records are written to DASD by CICS journal control.

The system also needs to capture log records to allow forward recovery of recoverable resources in the event of a failure. For example, if the DASD

volume holding a VSAM data set fails, the data set is recovered by restoring the latest backup copy of the data set to a new DASD volume, and then applying the forward recovery log records to the data set. In earlier releases of CICS, the forward recovery log records could be written either to the system log or to a user log. In CICS TS, the system log is used only for information relating to backout. All forward recovery information is written to forward recovery logs.

1.2.2 Log Streams and Log Structures

The CICS log manager does not write the log records itself. It uses services provided by the MVS system logger to write records to:

- The CICS system log, which is also used for dynamic transaction backout.
- Forward recovery logs
- Autojournals
- User journals

Forward recovery logs, autojournals, and user journals are collectively known as general logs. The MVS system logger is a component of OS/390 or MVS/ESA 5.2, which provides a programming interface to access records on a log stream. A log stream is a sequence of data blocks, with each log stream identified by its own log stream identifier—the log stream name (LSN). The CICS system log, forward recovery logs, autojournals, and user journals map to specific MVS log streams.

The MVS system logger initially writes the log streams to a coupling facility list structure, which you must define during CICS installation. List structures are used for many different purposes. We use the term *log structure* to refer to a coupling facility list structure that holds MVS system logger log streams. The log structure is an area of coupling facility storage reserved for the log streams.

Figure 1 on page 2 shows that log records are written to either to an MVS data space or to DASD, in addition to being written to the coupling facility. This ensures that log records are available if the coupling facility fails for any reason. This topic is discussed further in 5.2.2.7, “Defining DASD Staging Data Sets” on page 67.

Using the MVS system logger means that the CICS journal data sets of earlier releases and the journal control table (JCT) are redundant. Journal data sets are replaced by MVS log streams, and the JCT macros defining journaling are replaced by JOURNALMODEL resource definitions (defined using the CICS resource definition transaction, CEDA) in the CICS system definition file (CSD).

1.2.3 Managing the MVS Log Streams

Figure 1 on page 2 shows that the MVS system logger initially writes the log records to coupling facility log structures. These log structures are known as primary storage. To prevent the coupling-facility log structures from filling with log streams, the MVS system logger automatically spills the older log records into secondary storage, which consists of data sets managed by the Storage Management Subsystem (SMS). Each log stream, identified by its LSN, is written to its own log stream data sets. The MVS system logger dynamically allocates a

new log stream data set should the active one become full, up to a limit of 168 data sets¹ per defined LSN.

Eventually, older records are migrated from secondary to tertiary storage (DASD data sets or tape volumes), as specified in your hierarchical storage manager policy.

1.2.4 Benefits of the CICS Log Manager

The CICS log manager greatly improves the management of CICS system logs, forward recovery logs, auto journals, and user journals. It simplifies recovery procedures and operations, so the danger of operational errors is minimized. The CICS log manager, with the MVS system logger, improves management of system log data by avoiding log wraparound and automatically deleting obsolete log data for completed units of work (UOWs). The CICS log manager can retrieve data directly as well as sequentially, and most system log data is immediately available from the coupling facility. CICS restart processing is thus faster as a result of faster access to a reduced volume of log data.

In a large parallel sysplex, it would be impossible for each CICS region to write its own forward recovery log records for a file that is shared among these regions, and then merge these logs should recovery be necessary. With the new logging mechanism, you define one log stream for a recoverable file, and the MVS system logger merges log records from multiple CICS regions while writing the log stream. All records are written in the correct time-stamp sequence. You can choose to have only one log stream for a set of recoverable files that form a logical entity and therefore always need to be recovered together.

1.3 The CICS Recovery Manager

The CICS recovery manager is a new CICS domain. Its purpose is to ensure the integrity and consistency of recoverable resources, such as files and databases, both within a single system and distributed over interconnected systems in a network. In particular, it resolves the “in-doubt” problem that can inhibit application designs in which multiple CICS systems participate in updates of recoverable resources within a single UOW.

1.3.1 UOW Control

The CICS recovery manager maintains, for each UOW in a system, a record of the changes of state that occur during its lifetime. Some examples of events that cause state changes are:

- Creation of the UOW
- Premature termination of the UOW because of transaction failure
- Receipt of a synch point request
- Entry into the in-doubt period during two-phase commit processing
- Synch point rollback
- Normal termination of the UOW.

The identity of a UOW and its state are owned by the CICS recovery manager and are recorded in storage and on the system log. The system log records are

¹ OS/390 Release 3 removes the limit of 168 data sets for each log stream.

used by the CICS recovery manager during emergency restart to reconstruct the state of the UOW that were in progress at the time of the earlier system failure.

Each local resource manager can write UOW-related log records to the local system log, which the CICS recovery manager may subsequently be required to present to the resource manager again during recovery from failure.

To enable the CICS recovery manager to deliver log records to each resource manager in the appropriate way, it needs to capture suitable information when the log records are created. All logging by resource managers to the system log is therefore always performed using interfaces provided by the CICS recovery manager and never directly to the CICS log manager. Figure 2 shows the clients of the CICS recovery manager and how it supports the resource managers.

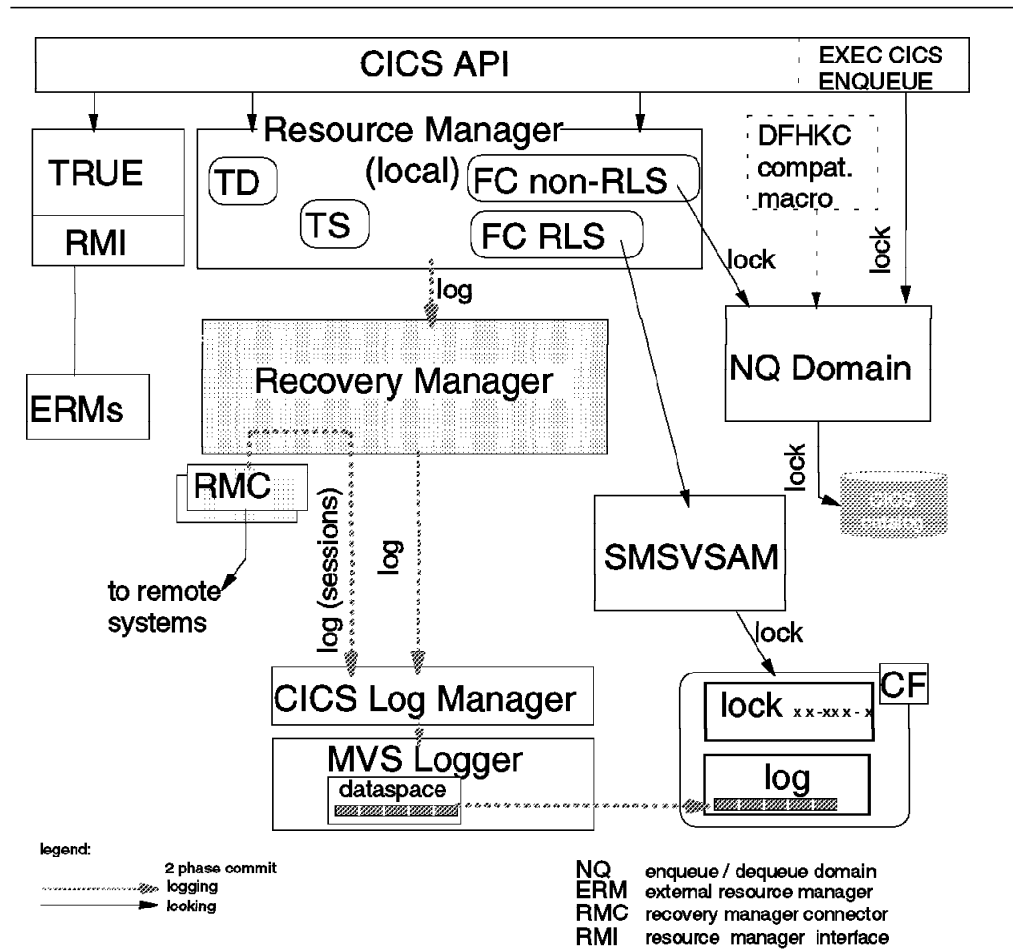


Figure 2. CICS Recovery Manager Clients.

1.3.2 UOW Shunting

The CICS recovery manager can temporarily suspend (shunt) completion, and later resume (unshunt) completion, of UOWs that cannot immediately complete commit or back-out processing because the required resources are unavailable, because of system, communication, or media failure.

When the CICS recovery manager shunts a UOW, it:

- Moves all relevant log information from the active system log stream (DFHLOG) to the secondary system log stream (DFHSHUNT).
- Because it is undesirable for transaction resources to be held up for too long, CICS attempts to release resources that are not required for recovery, such as terminals, sessions, and task storage.
- Transforms locks currently held on recoverable resources from active status to retained status.

After the cause of the failure has been removed, the UOW is unshunted again to complete its commit or back-out processing. This is normally an automatic process. For the rare case where CICS is unable to resolve a failed UOW automatically, the CICS recovery manager provides an interface to CEMT to allow you to issue INQUIRE or SET commands for shunted UOWs. This enables you to track down the problem and take manual actions to unshunt the UOW.

1.3.3 Benefits of the CICS Recovery Manager

The benefits you can gain with the new CICS recovery manager are:

- Full support for the preservation of data integrity.
- Exploitation of the two-phase-commit protocol to ensure that in-doubt failures are resolved correctly.
- Less interruption for VSAM errors, as data consistency is restored online. Transactions can access all but the directly affected records, allowing the option of continuing to use the file if the damage is localized.
- Performance improvement for emergency restart because of parallel backout while restarting.
- Operator and systems programming interface (SPI) to enable tracking and repair of shunted UOWs.
- Improved shutdown mechanism, which eliminates hanging transactions and minimizes the chance of creating retained locks.
- Improved error diagnosis.

1.4 The CICS Enqueue Domain

The new enqueue (NQ) domain provides locking services on behalf of the local clients (or resource owners) of CICS recovery manager.

1.4.1 Resource Owners

The local resource owners who use the services of the NQ domain are:

- File control: for non-RLS VSAM files
- Transient data
- Temporary storage
- The NQ domain itself is a client of CICS recovery manager for the synch pointing process.

1.4.2 Enqueue Requests

EXEC CICS ENQUEUE requests are routed directly to the NQ domain. The former DFHKC ENQ service is now available only as a macro compatibility layer; the macro routes directly to NQ domain.

1.4.3 Supported Locks

The NQ domain supports all the lock types required for full data integrity support and read integrity support, such as exclusive, active, retained, and shared locks. Note that the NQ domain itself does not provide any logging of enqueues. Therefore, the respective resource owner must do the logging (or rather request it through the CICS recovery manager), to be able to correctly reacquire an enqueue after a failed CICS is restarted.

1.4.4 CICS TS Enqueue Pools

ENQ pools are created by the users of the NQ domain during CICS initialization. Each enqueue request is allocated in a specific pool with its own hash table and characteristics. The pool name identifies the type of enqueue a task is waiting for. If, for example, you do a CEMT INQUIRE TASK, you can easily identify the type of enqueue request the task is waiting for. Enqueues may have the same name in different pools. Figure 3 shows the enqueue pools and indicates their purpose.

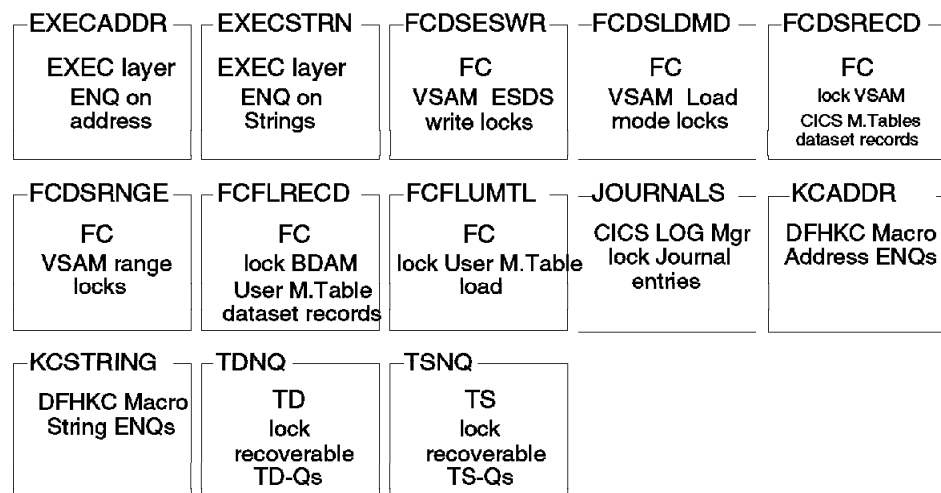


Figure 3. Overview of Enqueue Pools.

1.5 VSAM Record Level Sharing

VSAM RLS is a new function introduced with DFSMS 1.3 and exploited by CICS TS. It allows sharing of VSAM data sets among multiple applications running in multiple CICS TS regions that reside in any MVS/ESA 5.2 in a parallel sysplex. VSAM RLS is used for all VSAM files controlled by CICS file control (with some restrictions), including the CICS system definition file (CSD).

Figure 4 gives an overview of the components needed for VSAM RLS.

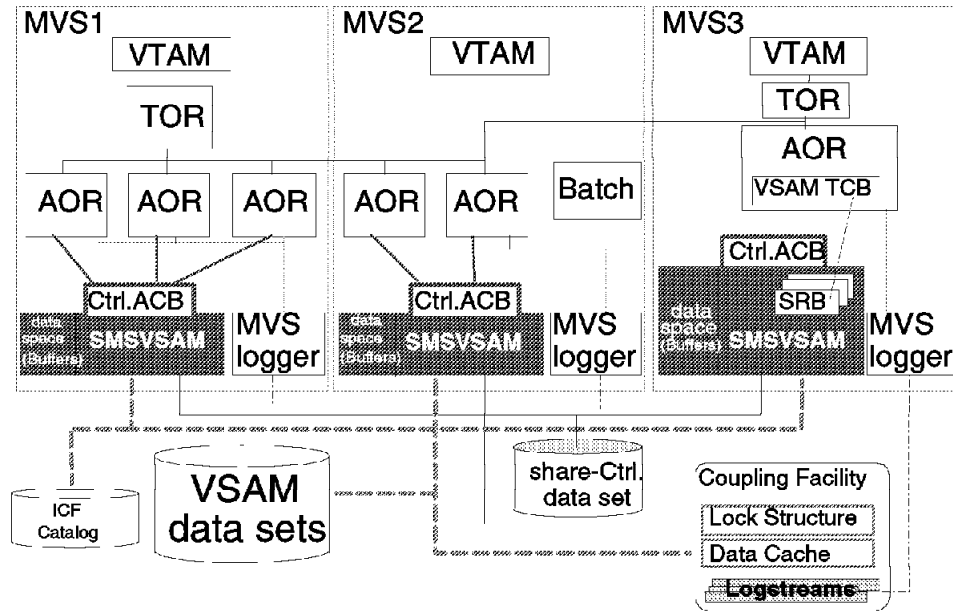


Figure 4. The Components of VSAM Record Level Sharing

1.5.1 The SMSVSAM Server Address Space

To support VSAM RLS, DFSMS 1.3 introduces a new MVS subsystem, the Storage Management Subsystem (SMS) VSAM server address space. Once installed, SMSVSAM can be automatically initialized whenever you perform an initial program load (IPL) of MVS. SMSVSAM implements locking through coupling facility lock structures. Locking for RLS mode access is performed at a record level, rather than at the control interval level as in non-RLS mode. SMSVSAM also maintains a cache in the coupling facility to hold shared data. Additionally, the SMSVSAM server allocates a local data space that holds the VSAM buffers and control blocks. The strings required for VSAM data access are dynamically allocated so that you do not have to define strings as for non-RLS files.

1.5.2 Logging through CICS

Any CICS TS region that needs access to shared VSAM files is registered automatically with the SMSVSAM control ACB at CICS initialization. To activate RLS for a CICS region you must set RLS=YES in the system initialization table (SIT). To access a file in RLS mode, you must define RLSACCESS(YES) in the file definition in CICS. SMSVSAM does not support logging. Logging and journaling remain the responsibility of CICS.

1.5.3 Access Mode

You specify the access mode when you open the file (or in the case of CICS, CICS specifies the access mode based on the RLSACCESS parameter of the file definition). The access mode is an RLS, or a non-RLS access mode such as LSR or NSR.

1.5.4 Recovery Attributes

For RLS data sets, the recovery attributes must be stored in the ICF catalog, so that all systems using the data set share a common understanding of the recovery attributes. You must provide the necessary parameters when the data set is defined. The parameters define attributes such as “forward recovery wanted” and log stream data set names. These parameters must be defined for an RLS mode file.

CICS uses the recovery information contained in CICS file definitions only if there is no recovery information in the ICF catalog; this applies solely to non-RLS mode files. You should consider keeping all recovery information in the catalog to ease maintenance, regardless of whether the VSAM data set is accessed in RLS or non-RLS mode.

1.5.5 Sharing Control Data Set

Sharing control is a key VSAM RLS component that maintains data integrity in the event that the lock structure is lost. You should always run with two active and one spare sharing control data sets. The spare data sets are used when an error occurs on an active sharing control data set. Sharing control data sets should be placed for maximum availability. Up to five active and five spare sharing control data sets are allowed.

Refer to the *DFSMSdfp Storage Administration Reference* for further information.

1.5.6 Locks

SMSVSAM maintains two different types of record-level locks in its coupling-facility lock structure: exclusive locks, and shared locks. Exclusive locks are used for any kind of update request, while shared locks are used to support read integrity. The lock includes the name of the corresponding lock owner—that is, the CICS application ID and the unit of work ID. CICS initiates lock release using the SMSVSAM interface.

1.5.7 Read Integrity

To support read integrity, CICS provides three new options. You can specify these options either on CICS application programming interface (API) commands or in the file definitions:

- UNCOMMITTED, which gives no read integrity. Programs work the same way as with earlier releases of CICS.
- CONSISTENT, which allows only committed data to be read. This causes a shared lock to be established on the record, and the lock is held for the duration of the read request.
- REPEATABLE, which is similar to CONSISTENT. The shared lock, however, is held until synch point, so no updates are possible on this record for the duration of the reading UOW. This ensures that if the program issuing the original lock rereads the record, it will retrieve the same data.

Your application programs may receive new response codes when they try to access a record. They will, for example, receive a RECORDBUSY when an active lock is held on the record and NOSUSPEND has been specified, and a LOCKED when SMSVSAM holds a retained lock. See the *CICS Application Programmers Reference* for more information.

1.5.8 Restrictions for VSAM Record Level Sharing

RLS cannot be used for

- Transient data intrapartition data sets
- Temporary storage
- Local or global catalog data sets
- Key sequenced data sets (KSDSs) cannot be accessed using relative byte address (RBA)
- Key range data sets are not supported.
- Temporary data sets are not supported.
- An alternate index cannot be directly opened and accessed; path mode is allowed.
- An RLS data set cannot be used as the source of a CICS-maintained data table, but it can be used for a user-maintained data table.
- RLS data sets must be SMS managed.
- IMBED option is not supported. However, 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 simply copy it using REPRO to create a new cluster that does not specify IMBED. IMBED today is a largely obsolete performance enhancement, which is mostly invalidated by modern caching disk controllers.

The REPLICATE option is also worth considering. Although REPLICATE is supported by RLS, it does not provide any performance benefit and can be omitted to save DASD space.

1.5.9 Restrictions for Concurrent Access

There are some restrictions you have to consider when sharing VSAM data sets in your parallel sysplex.

For recoverable data sets

- Multiple CICS TS regions in a parallel sysplex can update the data set concurrently.
- Multiple batch jobs can read the data set while CICS is updating it.
- No batch job can open a recoverable data set for output using RLS access.

Quiescing a data set

Before you can update a recoverable data set from a batch job you must quiesce the data set to stop all RLS activity. The quiesced state is stored in the ICF catalog. After a quiesce has completed, all CICS files associated with the data set are CLOSED. A quiesced data set can be opened in non-RLS mode only if no retained locks are presented. Once in non-RLS mode, the data set can be opened again in RLS mode only after it is UNQUIESCED. Quiesce and unquiesce operations are performed using the CEMT command SET DSNAME(*dsxy*) QUIESCED|UNQUIESCED, or by using the equivalent SPI command within a program.

Nonrecoverable data sets

Multiple CICS TS regions and batch jobs can update a nonrecoverable data set concurrently. However, users need to ensure that their batch applications can cooperate with CICS, changing things at the same time.

VSAM share options

Any VSAM share options specified for a data set are ignored if the data set is opened in RLS mode. However, if you specify a disposition of DISP=OLD in the file DD statement of any job, sharing is inhibited, and the CICS region or batch job that opens the data set has exclusive use of the data set.

Deadlocks

An additional function of SMSVSAM is to detect deadlocks. If a deadlock occurs, the deadlock timeout mechanism of SMSVSAM returns a deadlock error to the programs involved in the deadlock, and provides diagnostic information. Deadlock timeout is based on the DTIMOUT parameter in the CICS transaction resource definition, or on the new SIT parameter FTIMEOUT (DTIMOUT takes precedence). Deadlocks are more likely to occur in a sharing environment—for example, when several transactions executing in different CICS regions concurrently access the same record.

Benefits of RLS

RLS has these benefits:

- Shared concurrent access to VSAM files by many CICS regions.
- Improved availability of VSAM data sets.
- Exploitation of the parallel sysplex capabilities.
- Improved data integrity
- Improved system management
- New and improved commands

Additional VSAM extensions not directly related to VSAM RLS are these:

- Support for variable relative record data sets (VRRDS)
- Removal of the limitation of 4 GB for KSDS data sets accessed in non-RLS mode
- Increase in the number of VSAM data sets that can be allocated by CICS to approximately 10,000.

1.6 Temporary Storage Data Sharing

In this section, we provide an overview of temporary storage data sharing introduced with CICS TS. Temporary storage code has been entirely restructured in a new CICS domain, the temporary storage (TS) domain.

The new TS domain interfaces with the CICS recovery manager and the NQ domain to provide full integrity for recoverable TS queues. Restart times can be improved for regions that handle large amounts of TS data, since the CICS recovery manager removes the need to read the entire recoverable queue during restart. The new NQ domain serves the TS locking requests for recoverable TS queues.

To enable sharing of TS queues in a parallel sysplex environment, a new server subsystem has been introduced; the TS server DFHXQMN. Exploitation of a parallel sysplex environment can be limited if you cannot share your TS queues among multiple CICS regions, because you might have to tolerate affinities to a queue-owning region (QOR) in your dynamic transaction routing program. With the new TS server you can access a shared TS queue from any

application-owning region (AOR) in the parallel sysplex. The shared TS queues reside in a designated TS pool in the coupling facility. CICS internal queues are not supported; they cannot be shared. See Figure 5 on page 12 for an overview of temporary storage data sharing in a parallel sysplex.

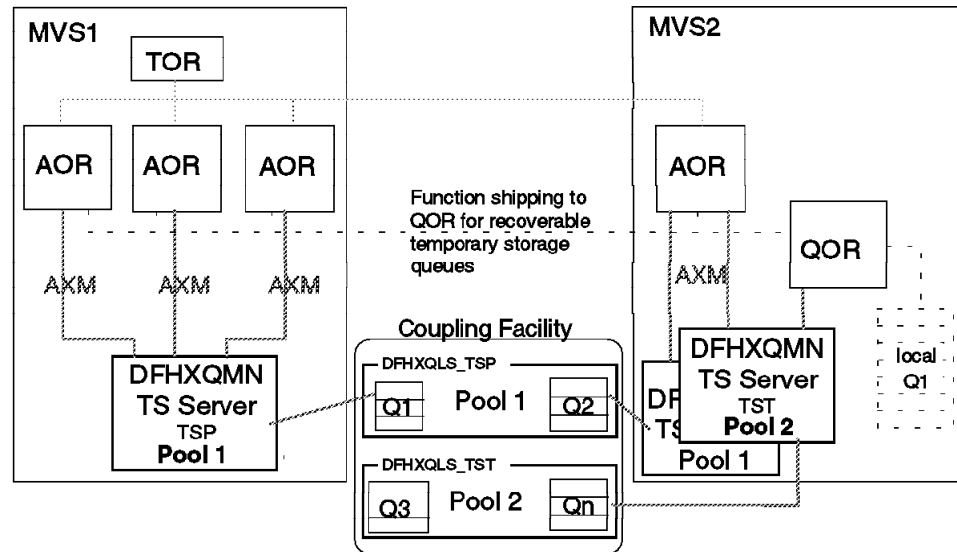


Figure 5. Temporary-Storage Data Sharing Overview

1.6.1 Temporary Storage Server DFHXQMN

The TS server DFHXQMN is a new MVS subsystem that serves all AORs in an MVS image. You need one TS server per MVS image for a given TS pool. The TS server avoids the installation of a CICS QOR, which has limitations in terms of availability (it is a single point of failure) and performance (function shipping overhead). The AORs interface with the TS server using authorized cross memory services (AXM). In CICS, the TS server is identified by the systemID (SYSID) specified in the TS table (TST). The TS server accesses the shared TS queues that reside in a designated TS pool in the coupling facility. You may have more than one TS server running in an MVS image, where each server manages one TS pool in the coupling facility. The server knows which pool to manage by a parameter you specify in the startup JCL for the server (giving the pool name). You can use MVS modify commands to change some of the parameters for the TS server dynamically.

1.6.2 Temporary Storage Pool

The TS pool resides in a coupling facility list structure that must be defined in your CFRM policy (using the IXCMIAPU utility). The structure name for a TS pool must always begin with DFHXQLS_. If the pool structure size you define is not big enough, you can get NOSPACE conditions in your applications. The TS pool does not have a spill mechanism such as the MVS system logger and the maximum structure size cannot be changed dynamically. You should plan your pool sizes carefully, according to the requirements in your installation.

The structure is initialized by the first TS server that starts up. Remember that the queues in the TS pool are not recoverable. However, if a CICS system that

uses a shared queue fails, the shared queue still exists in the coupling facility after CICS restart. This might have implications for your applications.

1.6.3 CICS to Temporary Storage Server Connection

When a CICS TS region initializes, it checks whether there are shared queues defined in the TST. If so, it builds a vector that contains the details for the connection to the TS server. With the first TS request that references a shared queue, CICS checks if the respective TS server is active and then sends a connect request. Should the server or the connection fail, then CICS tries to reconnect with each new TS request that is issued. If the server is available but cannot currently handle a TS request (server is busy), CICS queues the request and waits for an ECB to be posted by the TS server.

1.6.4 Managing Temporary Storage Pools

A number of display commands are available to get information about the TS server and the usage of the coupling facility structure. The CEMT command INQUIRE TSQUEUE has been extended with new attributes. A utility is provided that you can use to unload or reload the contents of a TS pool to DASD. You may need to specify a new pool name when reloading a TS pool.

It is possible to have a shared TS queue and a local TS queue with identical queue names. You differentiate between them using the SYSID option, as you do for identically named queues in different regions with earlier releases of CICS.

1.6.5 Application Considerations

The queue update commands (for example, WRITEQ TS) can now receive a locked condition, for example when the queue belongs to an in-doubt UOW.

If your application requires a recoverable queue, then the solution is to implement a QOR in order to be able to share such a queue. TS data sharing does not support recoverable queues.

New causes for a SYSID error (SYSIDERR) exist for shared TS queues, such as:

- CICS to TS server connection is interrupted.
- CICS is not authorized by RACF to use the TS server.
- TS server is not started or has abended.

The NOSPACE condition is raised if the coupling facility pool structure has reached its maximum limit, as defined in CFRM policy.

1.6.6 Benefits

The new TS domain, together with the new TS server, can offer you the following benefits:

- TS data sharing enables you to exploit dynamic transaction routing without having to tolerate affinities due to a temporary storage queue.
- Availability of shared TS queues to your applications is improved.
- Performance is better than that of a QOR.
- Queue management is improved.

1.7 Resource Definition Online for Transient Data

In this section, we provide an overview of changes associated with resource definition online (RDO) for transient data.

With CICS TS you can define your transient data queues online using the CEDA transaction. The destination control table (DCT) is no longer needed. For migration purposes, the DCT is still supported; there are, however, some new parameters, which you can only define using RDO. DFHCSDUP has been extended to support migration of your DCTs to RDO.

1.7.1 RDO Support

CEDA is extended to support define, alter, copy, delete, display, install, expand, move, rename, user define, and view operations for transient data definitions. All transient data queues types are supported: extrapartition, intrapartition, indirect, and remote queues. Recovery for installed transient data definitions is provided at warm start and emergency restart, using the CICS system log and global catalog.

1.7.2 New Options for Transient Data

RDO for transient data provides new options, available only for RDO-defined transient data queues. The new options are:

DSNAME
SYSOUTCLASS
BLOCKFORMAT
PRINTCONTROL
DISPOSITION
WAIT
WAITACTION

Refer to the *CICS Resource Definition Guide* for more information on these new options.

1.7.3 Transient Data Queue Management

Intrapartition queues:

- Can be installed only if the DFHINTRA data set is successfully opened during startup.
- Must be disabled and empty and must not have any indirection queues before they can be altered, reinstalled, or deleted.
- Cannot be disabled as long as there are retained locks for an in-doubt UOW.
- Those belonging to CICS (internal queues) cannot be discarded. The transient data queues used by CICS itself are defined in the IBM-supplied group DFHCTG.

Extrapartition queues must be closed before any alter, reinstall, or delete operation; otherwise they have the same restrictions as do intrapartition queues.

Indirect and remote queues do not have special restrictions since they cannot be used by a local region.

1.7.4 Benefits

RDO for transient data improves availability of your CICS regions since you do not have to recycle CICS to activate changes for your transient data queues. System management for transient data queues is simplified and you no longer have to predefine your extrapartition transient data queue data sets in the startup JCL.

1.8 New EXEC CICS CREATE Command

You can use this new API command to dynamically create resource definitions in a CICS TS region. The resource definitions that you can create using the EXEC CICS CREATE command are:

- CONNECTION
- FILE
- JOURNALMODEL
- LSRPOOL
- MAPSET
- PARTITIONSET
- PARTNER
- PROFILE
- PROGRAM
- SESSIONS
- TDQUEUE
- TERMINAL
- TRANCLASS
- TRANSACTION
- TYPETERM

On a warm start or emergency restart, your definitions are recovered from the global catalog and the CICS system log. After a cold start, your dynamic definitions have been discarded.

1.8.1 COBOL Example of EXEC CICS CREATE

Figure 6 on page 16 shows a COBOL sample of a program using the EXEC CICS CREATE command.

```

DATA DIVISION.
WORKING-STORAGE SECTION.

01 PNAME          PICTURE X(8).
01 PATTRLEN      PICTURE S9(4) COMPUTATIONAL.
01 PATTR.
   05 LANG1      PICTURE X(9) VALUE ' LANGUAGE(' .
   05 LANGUAGE   PICTURE X(9).
   05 CBRL       PICTURE X(2) VALUE ') ' .
   05 RESI       PICTURE X(9) VALUE ' RESIDENT(' .
   05 RESIDENT   PICTURE X(3).
   05 CBRR       PICTURE X(2) VALUE ') ' .

   ...
PROCEDURE DIVISION.
   ...
*****
*   PREPARE ATTRIBUTES FOR PROGRAM DEFINITION   *
*****
MOVE 'MYPROG' TO PNAME
MOVE 'COBOL' TO PATTR.LANGUAGE
MOVE ..... TO .....
* COMPLETE MOVES TO FILL STRUCTURE *
   ...
EXEC CICS CREATE PROGRAM (PNAME)
                      ATTRIBUTES (PATTR)
                      ATTRLEN (PATTRLEN)

END-EXEC.
EXEC CICS RETURN END-EXEC.
GOBACK.

```

Figure 6. COBOL Example of EXEC CICS CREATE

Some resource definitions need several commands, for example connections and sessions. You may have to use the COMPLETE command to end a sequence of commands, for example:

```

EXEC CICS CREATE CONNECTION(...)
EXEC CICS CREATE SESSIONS( )
EXEC CICS CREATE CONNECTION COMPLETE

```

1.8.2 Miscellaneous

All CREATE commands support RESP, RESP2, NOEDF, and NOHANDLE. Command execution can begin in the third CICS initialization phase.

The use of EXEC CICS CREATE requires ALTER access permission for the resource class in your security definitions.

1.8.3 Benefits

Resource definitions can be provided dynamically from a CICS program and without reference to the CSD. This means you can provide autoinstall for resources just before they are needed in an application, and only then. The CSD can sometimes be a bottleneck, especially in a parallel sysplex, and sharing the CICS system definitions is not always a practical solution for everybody.

1.9 Database Management

In this section, we look at the improvements implemented in CICS TS for the support of Information Management System (IMS) databases and IBM DATABASE 2 (DB2) databases.

1.9.1 IMS Database Support

You must use the DBCTL interface to access your IMS databases from a CICS TS region. The local Data Language 1 (DL/I) interface is no longer supported. You can, if necessary, ship DL/I function requests from a CICS TS region to a data-owning region (DOR). The DOR can access its DL/I databases using DBCTL, or using the local DL/I interface if it is running CICS/ESA 4.1, or an earlier release of CICS.

1.9.1.1 DBCTL Extensions

DBCTL and its task-related user exit support the in-doubt WAIT parameter. This means DBCTL can hold locks in case a unit of work fails. If CICS shunts a task, the task's DBCTL thread is freed to avoid unnecessary use of resources. This can cause database resource adapter (DRA) snap dumps to occur, depending on the IMS release. The following program temporary fixes (PTFs) avoid this dumping:

- For IMS/ESA 4.1, use PN62412 (PTF UN68325).
- For IMS/ESA 5.1, use PN62480 (PTF UN68717).
- For IMS 3.1, no PTF is required.

1.9.1.2 Miscellaneous:

DBCTL introduces changes in several areas. For example, DBCTL provides extended scheduling for program specification blocks. With extended scheduling, a PSB is scheduled even if some of the corresponding databases are not currently available. Applications should check the return code from a schedule request to ensure that all databases are available. Application programs will detect that the database is unavailable when they try to access it.

The operations interface of DBCTL is very different for CICS users. You cannot use the CEMT SET and INQUIRE DLIDATABASE commands to control and inquire about DL/I databases in a DBCTL environment.

If you are not already using DBCTL from CICS, you should prepare a plan for migrating from local DL/I to DBCTL as part of your plan for implementing CICS TS.

1.9.1.3 Benefits of DBCTL

Using DBCTL has a number of benefits, including these:

- You can make image copies of a database while it is online using the DFSUICP0 utility.
- DBCTL uses the IMS logging system and utilities. This provides advantages such as dual logging.
- Implementing DBCTL can provide virtual storage constraint relief for CICS, because no IMS code resides in CICS storage except the DRA.
- Data sharing capability is improved, including the sharing of databases among batch and online applications.
- CICS applications can access data entry databases (DEDBs).
- DBCTL provides a release-independent interface.

1.9.2 DATABASE 2 Support

In this section we look at some highlights of the interaction between CICS and DB2.

1.9.2.1 INDOUBTWAIT

The DB2 attachment supports the in-doubt WAIT parameter. Locks for a shunted UOW are held until the situation in doubt is resolved. Threads are terminated to release unused resources.

1.9.2.2 New Resource Control Table Options

CICS TS introduces a new parameter for the TYPE=INIT macro of the DB2 resource control table. The new parameter is STANDBY. In addition, the STRTWT parameter has a new option, AUTO. For further information, refer to Section 11.2, "CICS and DB2," in the *CICS Release Guide*.

1.9.3 Summary

- Local DL/I is no longer supported.
- CICS is now being shipped completely generated.
- Database attachments use an in-doubt protocol.

1.10 VTAM Generic Resource

In this section, we highlight the improvements implemented for users of the VTAM option called *VTAM generic resource*.

VTAM generic resource was introduced with CICS/ESA 4.1; the CICS/ESA 4.1 implementation; however, has limitations. For a logical unit type 6.2 (LU6.2) session logon, the generic resource name must be used. Such sessions then have a permanent affinity to one specific member of the VTAM generic resource group. LU6.2 sessions among CICS regions that are members of a VTAM generic resource group within a sysplex are not possible, since all have the same generic resource name. This also inhibits intersysplex communication among VTAM generic resource members in different sysplexes.

To overcome these problems, some installations implemented a hub CICS region, with all its drawbacks in a parallel sysplex environment (single point of failure being one of them).

With the enhancements implemented in CICS TS, you no longer need to implement a hub region, provided all your TORs are running CICS TS. Thus the single point of failure and the performance bottleneck of a hub system can be avoided. You can now have LU6.2 connections within a sysplex and also set up intersysplex communication using the generic resource name.

The CEMT and SPI commands to inquire about and specify VTAM and connection resources have been extended to include VTAM generic resource information. Refer to the *CICS Intersystem Communication Guide* for more information.

1.11 Monitoring and Statistics

In this section, we provide some information about changes for monitoring and statistics in CICS TS.

1.11.1 Performance-Class Data

A number of new performance-class data items have been added to reflect the new functions of CICS TS, and also to improve the contents and efficiency of monitoring. The new performance-class data items include:

- Information on the transaction type (user, system, or mirror)
- Information on MVS workload manager interactions
- Recovery manager information, including in-doubt wait and in-doubt resolution
- Recovery manager UOW ID of the current transaction
- RLS I/O wait time: the transaction wait time for RLS file I/O
- Synch-point elapsed time: entire duration of the synch-point process (commit or rollback) including:
 - Local CICS system time
 - Remote CICS system times
 - Time for synch-pointing with resource managers
 - Transaction dispatch and suspend times during synch-pointing
- CICS log manager write counts to the MVS system logger for a given transaction
- CPU time consumed by RLS requests

Several of the existing performance class data have been changed to better support a CICSplex environment and to gain more exact monitoring information. Refer to the *CICS Customization Guide* for more information on the new and changed performance classes.

1.11.2 Statistics

New and changed statistical data is available with CICS TS, such as:

- New global statistics for CICS recovery manager, including the number of synch points.
- Detailed information about shunted UOWs and forced in-doubt actions.
- New global statistics for the enqueue domain, showing the total number of enqueues, number of waits for enqueues including wait time, and more.
- New resource statistics for defined log streams, such as number of log stream writes, peak wait times, number of times the buffer was full, and more.

Much of the statistics data has been changed: for example the connection resource statistics, the dispatcher global statistics, the file resource statistics, temporary storage statistics and many more.

1.11.3 Statistics Utility Program

The statistics utility program (DFHSTUP) has been enhanced. There are new reports for the new and changed domains.

1.11.4 Statistics Sample Program

The statistics sample program (DFH0STAT) has been changed and enhanced. You can now select new reports for most of the resources

1.11.5 Summary

The new and changed performance class data and statistics enable you to better analyze transactions and system activities in a CICSplex and parallel sysplex, in order to take proactive actions if necessary.

1.12 Miscellaneous Changes

To round off the picture, we highlight in this section some other items of interest coming with CICS TS.

1.12.1 Autoinstall for Virtual Terminals

The CICS common client allows you to communicate from a client application (running on almost any workstation) with any CICS server. For example, you can have an application with a nice graphical user interface on your PS/2 running under OS/2 which communicates with a CICS TS server that runs transactions on behalf of the client. A common set of APIs enable the client code to initiate such a request most efficiently. The APIs directly invoke the code in the CICS common client that executes also in the workstation.

One of the methods to communicate using the common client APIs is called the *external presentation interface* (EPI). When using EPI, the CICS server sees an ordinary terminal. EPI uses virtual terminals when logging on to CICS, and the virtual terminal ID (TERMID) must either be specified in the client program or you can let the CICS client support generate one for you dynamically. While automatic generation is the most convenient way to do this, it causes problems with CICS releases earlier than CICS TS because duplicate TERMIDs are generated from different clients. In CICS TS, the autoinstall program is invoked for each install of a virtual terminal. If a duplicate TERMID is detected, an alternative TERMID is generated.

1.12.2 Shippable Terminals

To avoid conflicts for shipped terminals, the TERMID has to be unique in CICS releases earlier than CICS TS.

In CICS TS, the autoinstall program is called for shipped terminals and connections and you can now assign an alias TERMID or connection attribute to avoid name clashes. See the sample exit programs DFHZATDX and DFHZATDY in the CICS TS SDFHSAMP library.

1.12.3 START Affinity Parameter

A new SIT parameter FSSTAFF=YES|NO (function ship start affinity) causes a shipped START function request to return to the TOR that initiated it. This is not always the case in CICS releases earlier than CICS TS if you had VTAM generic resource and thus had more than one TOR connected to your AORs.

There are a number of additional smaller enhancements and changes we decided not to describe in this book. See the *CICS Release Guide* for a complete description.

Chapter 2. Introduction to System-Managed Storage

System-managed storage is the IBM automated approach to managing auxiliary storage. It uses software programs to manage data security, placement, migration, backup, recall, recovery, and deletion to ensure that current data is available when needed, and obsolete data is removed from storage.

System-managed storage is tailored to your needs. You define the requirements for performance, security, and availability, along with storage management policies used to automatically manage the direct access, tape, and optical devices that use any MVS/ESA operating systems.

In this chapter, we introduce system-managed storage in these topics:

- The DFSMS environment
- Letting the system manage your data
- SMS classes and groups
- The storage hierarchy
- SMS and VSAM record-level sharing.

2.1 The DFSMS Environment

The combination of system-managed storage and related hardware and software products is called the DFSMS environment. To implement the DFSMS environment and to take advantage of all the functions available with MVS/ESA, you need to install a specific set of software products:

- **DFSMSdfp**, provides the storage, program, data, and device management functions of MVS/ESA. The storage management subsystem (SMS) component of DFSMSdfp is fundamental in providing these functions.
- **DFSMSdss**, copies and moves data for MVS/ESA.
- **DFSMShsm**, provides the backup, recovery, migration, recall, and space management functions in the DFSMS environment.
- **DFSMSrmm**, provides the management functions for removable media, including tape cartridges and reels, and optical volumes.
- **DFSORT**, sorts, merges and copies data.
- **RACF** (or an equivalent product) controls access to data and other resources in MVS/ESA.

The DFSMS/MVS product combines DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm into one package.

2.2 Letting the System Manage Your Data

The SMS consists of a logical set of policies. With SMS, you can define performance goals and data availability requirements, create model data definitions for typical data sets, and automate data backup. Based on your installation policy, SMS can automatically assign those services and data definition attributes to data sets when they are created. IBM storage

management-related products determine data placement, manage data backup, control space usage, and provide data security.

The goals of SMS are to:

- Improve the use of the storage media – for example, by reducing out-of-space abends and providing a way to set a free-space requirement.
- Reduce the labor involved in storage management by centralizing control, automating tasks, and providing interactive controls for storage administrators.
- Reduce the user's need to be concerned with the physical details of performance, space, and device management. Users can focus on using, instead of managing data.

2.2.1 Benefits of SMS

These are some examples of the benefits of SMS:

- **Simplified data allocation.** SMS enables you to simplify your data allocations. For example, without using SMS, you would have to specify the unit and volume on which the system should allocate your data sets. You would also have to calculate the amount of space required for the data set in terms of tracks or cylinders. This means you have to know the track size of the device which will contain the data set.

With SMS, you can let the system select the specific unit and volume for the allocation. You can also specify size requirements in terms of megabytes or kilobytes. This means you do not need to know anything about the physical characteristics of the devices in your installation.

- **Improved allocation control.** SMS enables you to set a requirement for free space across a set of direct access storage device (DASD) volumes. You can then provide adequate free space to avoid out-of-space abends. The system automatically places data on a volume containing adequate free space.
- **Improved I/O performance management.** SMS enables you to improve DASD I/O performance across your installation and at the same time reduces the need for manual tuning by defining performance goals for each class of data. You can use cache statistics recorded in system management facilities (SMF) records to help evaluate performance. You can also improve sequential performance by using extended sequential data sets. The DFSMS environment makes the most effective use of the caching abilities of the IBM 3990 Model 3 and Model 6 Storage Controls, as well as other new models.
- **Automated DASD space management.** SMS enables you to automatically reclaim space which is allocated to old and unused data sets or objects. You can define policies that determine how long an unused data set or object will be allowed to reside on primary storage (storage devices used for your active data). You can have the system remove obsolete data by migrating the data to other DASD, tape, or optical volumes, or you can have the system delete the data. You can also release allocated but unused space which is assigned to new and active data sets.
- **Improved data availability management.** SMS enables you to meet different backup requirements for data residing on the same DASD volume. Thus, you do not have to treat all data on a single volume the same way.

You can use DFSMSHsm to automatically back up CICS and DATABASE 2 (DB2) databases, partitioned data sets extended, and physical sequential, partitioned, virtual storage access method (VSAM), hierarchical file system (HFS), and direct access data sets. You can also back up other types of data and use concurrent copy to maintain access to critical data sets while they are being backed up. The processes concurrent copy, along with backup-while-open, have an added advantage: they avoid the invalidation of a backup of a CICS VSAM KSDS due to a control area or control interval split.

- **Simplified conversion of data to different device types.** SMS enables you to move data to new volumes without requiring updates to job control language (JCL). Because users in a DFSMS environment do not need to specify the unit and volume which contains their data, it does not matter to them if their data resides on a specific volume or device type. This allows you to easily replace old devices with new ones.
- **Other benefits.** SMS also provides automated tape space management, and automated optical space management.

2.3 SMS Classes and Groups

On systems that do not use DFSMS, storage management consists mostly of manual operations performed on individual data sets, and manual and automated operations performed on volumes. With SMS, you can automate storage management for individual data sets and objects, and for DASD, optical, and tape volumes. You use SMS classes and groups to define the goals and requirements that the system should meet for a data set or object.

2.3.1 Using SMS Classes and Groups

Automatic class selection (ACS) routines assign classes to data, based on requirements and attributes, and select the target storage groups.

The classes are:

Data Class	Data definition parameters
Storage Class	Performance and device availability requirements
Management Class	Data availability, space, and retention requirements
Storage Group	List of candidate allocation volumes

Figure 7 on page 26 shows how SMS classes and storage groups are assigned to data sets.

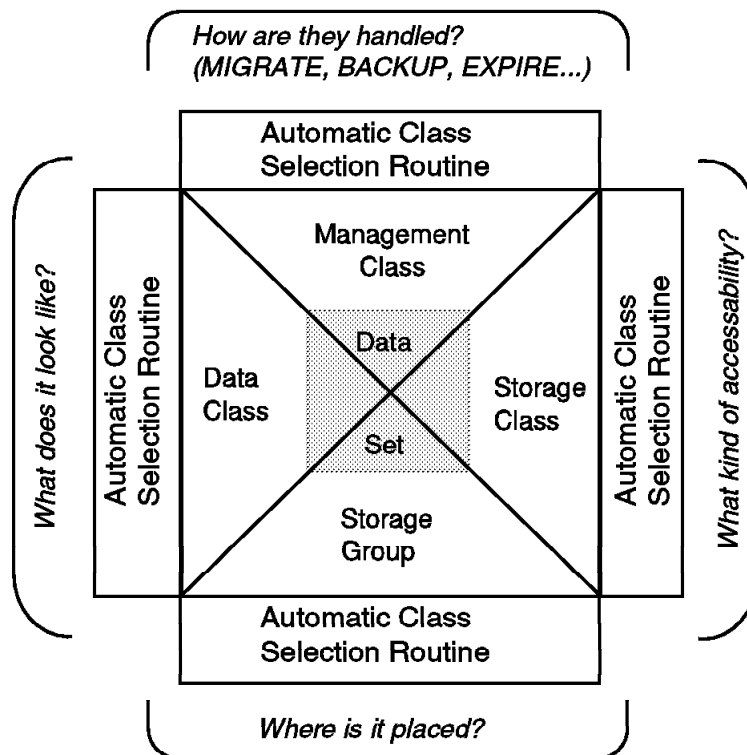


Figure 7. SMS Classes and Groups That Are Assigned to Data

You can use ACS routines to automatically determine the target storage group and assign data classes, storage classes, and management classes to SMS-managed data sets and objects. Data classes can be assigned to non SMS-managed data sets as well. Automatic class selection provides centralized control over data set allocation on SMS-managed volumes.

Note: If SMS is activated, all new data set allocations are subject to automatic class selection.

Your storage administrator writes an ACS routine for each of the three types of classes plus the storage groups. Your SMS configuration is composed of:

- Data class, storage class, management class, and storage group definitions
- ACS routines, to assign the classes and groups.
- The base configuration, which contains default information such as default unit type, default device geometry, and default management class. It also identifies the systems in the installation for which the subsystem manages storage.

The storage administrator stores the information on this configuration in a source control data set (SCDS).

Each time a new data set is allocated, SMS runs the ACS routines in the following order:

1. The data class routine is run to determine whether a data class should be assigned to the data set.

2. The storage class routine is run and, if a storage class is assigned, the data set is put under SMS control.

Note: If the storage class is set to *null*, then the data set will not be SMS-managed.

3. The management class routine is run to assign a management class.
4. The storage group routine is run to determine candidate storage groups for the SMS-managed data set.

The selection of specific classes and groups is based on information from JCL or other allocation parameters, such as data set name, data set size, or job name.

2.4 The Storage Hierarchy

The storage hierarchy consists of levels of storage devices with each level characterized by different access speeds, costs per byte, and storage capacities. Figure 8 shows the different levels of the hierarchy in a pre-DFSMS 1.3 environment:

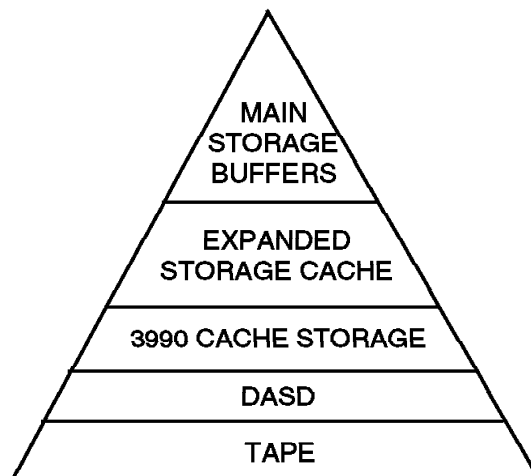


Figure 8. Storage Hierarchy before DFSMS 1.3

DFSMS 1.3 with the record-level sharing (RLS) function introduces a new level in the storage hierarchy. Figure 9 on page 28 shows you the new hierarchy.

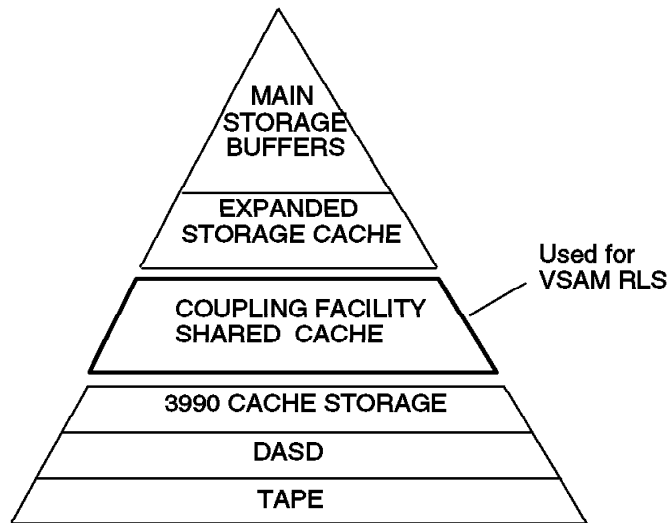


Figure 9. Storage Hierarchy with DFSMS 1.3

2.5 SMS and VSAM Record-Level Sharing

VSAM record-level sharing (RLS) extends the storage hierarchy to support a data sharing environment across multiple systems in a parallel sysplex. This support is designed primarily for VSAM data sets used by CICS applications.

VSAM RLS is a dataset access mode that allows multiple address spaces, CICS application-owning regions on multiple MVS systems, and batch jobs to access data at the same time. With VSAM RLS, multiple CICS systems can directly access a shared VSAM data set, eliminating the need for function shipping between AORs and file-owning regions (FORs). VSAM RLS provides data sharing with data integrity by using the lock structure in the coupling facility.

VSAM RLS uses the cache structures within the coupling facility to enable data to be stored in the coupling facility buffers. The coupling facility caches serve as a multisystem shared global pool.

Note: It is the storage class which determines if a data set can use the coupling facility cache structure. Therefore the data set must be SMS-managed to access the coupling facility cache structure.

CICS provides the logging, commit, and rollback functions for VSAM recoverable files; VSAM provides record-level serialization and cross-system caching. CICS, not VSAM, provides the recoverable files function. Whether a data set is recoverable or not determines the level of sharing allowed between CICS and batch applications. For additional information about DFSMS, refer to *DFSMS/MVS V1 R3 General Information*.

Chapter 3. Prerequisite Products

This chapter gives information about the hardware and software environments required to use CICS Transaction Server for OS/390, Version 1 Release 1. It addresses the following topics:

- 3.1, “Hardware Prerequisites”
- 3.2, “MVS/ESA Operating System”
- 3.3, “DFP Environment” on page 30
- 3.4, “Database Products” on page 30
- 3.6, “IBM Telecommunications Access Methods” on page 30
- 3.7, “External Security Manager” on page 31
- 3.8, “Forward Recovery Software” on page 31
- 3.9, “Programming Languages” on page 31

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

- *CICS Transaction Server for OS/390 Release Guide*
- *CICS Transaction Server for OS/390 Migration Guide*

3.1 Hardware Prerequisites

CICS TS runs on an Enterprise Systems Architecture/370 (S/370) or an Enterprise Systems Architecture/390 (S/390) processor that meets the requirements of the host operating system, CICS, the access methods and your application programs.

The CICS log manager requires a coupling facility. If you have multiple MVS images in your sysplex, or if you want to use RLS even within a single MVS image, you may need more than one coupling facility. Your coupling facility can be an Integrated Coupling Migration Facility (ICMF) or one of the following:

- An IBM S/390 Coupling Facility 9674 (IBM 9674)
- A logical partition on an IBM S/390 Parallel Enterprise Server 9672 (IBM 9672)
- A logical partition on an ES/9000 711-based processor

3.2 MVS/ESA Operating System

The CICS log manager uses the MVS system logger services, supplied in MVS/ESA SP 5.2 JES2 (5655-068) or JES3 (5655-069). This is the minimum level of operating system required for CICS. You can (and should) use later releases of software, such as a release of OS/390.

You need SMP/E (5668-949) Release 8 to install CICS.

3.3 DFP Environment

The minimum release level of Data Facility Product (DFP) is MVS/DFP Version 3 Release 3 (5655-XA3). DFSMS/MVS Version 1 Release 2 and DFSMS/MVS Version 1 Release 3 (5695-DF1) are also supported.

Installation staff planning to implement the VSAM Record Level Sharing (RLS) function provided with DFSMS/MVS 1.3 should first review the Preventive Service Planning(PSP) information for upgrade MVSDFSMS130 subset HDZ11C0/9628 and DATASHARING Contact your IBM marketing representative for details on obtaining a copy of the PSP.

3.4 Database Products

CICS TS supports the following IBM database managers: IMS/ESA (5665-408) and DB2 (5665-DB2).

3.4.1 IMS/ESA (5665-408)

CICS TS does not support the local DL/I interface to IMS/ESA Database Manager (IMS/DM). You must use the DBCTL interface, which is available in IMS/DM Version 3, Release 1 or later. If you want to exploit IMS database sharing among many CICS regions on several MVS images within a parallel sysplex configuration, you require IMS/DM Version 5, Release 1 or later.

3.4.2 DB2 (5665-DB2)

The minimum release level supported is DB2 Version 2, Release 3. If you want to exploit DB2 database sharing among many CICS regions on several MVS images within a parallel sysplex configuration, you require DB2 Version 4, Release 1 or later.

3.5 IBM MQSeries for MVS/ESA

The minimum release level supported is MQSeries for MVS/ESA Version 1, Release 1, Modification 3 (5695-137).

Support of in-doubt wait handling requires MQSeries for MVS/ESA Version 1, Release 1, Modification 4.

3.6 IBM Telecommunications Access Methods

CICS TS will work with VTAM Version 3, Release 4, Modification 1 (5685-085) or later. However if you wish to have a single network name for multiple terminal-owning CICS systems (TORs) on several MVS images within your parallel sysplex, you require VTAM Version 4, Release 2 (5695-117) or later.

If you use TCP/IP attached systems or workstations to communicate with CICS using the CICS basic TCP/IP sockets feature of TCP/IP for MVS, you need TCP/IP for MVS Version 3 (5655-HAL).

3.7 External Security Manager

If you use Resource Access Control Facility (RACF) as your external security manager for CICS, you need RACF Version 2, Release 1 (5695-039) plus PTF UW05554 for APAR OW02759, to provide the required support for the new LOGSTRM general resource class, and PTF OW15975, to enable general resource class profile names for CICS journals to be up to 17 characters long.

3.8 Forward Recovery Software

If you use CICS VSAM Recovery (CICSVR) for forward recovery of VSAM files (including RLS files), you need CICSVR Version 2, Release 3 (5695-010).

3.9 Programming Languages

CICS supports the following programming languages and environments:

- High Level Assembler/MVS (5696-234)
- VS/COBOL II (5668-958 and 5688-023). No support is provided for NORES programs (CICS or batch). Requires PTF for APAR PN43097 (Abend0c4 caused by IGZECIC returning to CICS with an incorrect mode).

These PTSs are as follows:

UN48282 for FMID JCL1331

UN48283 for FMID JCL1341

UN48284 for FMID JCL1403

- C/370 Version 1, Release 2 or later (5688-040)
- IBM C/C++ for MVS/ESA Version 3, Release 1 or later (5655-121)
- CSP Version 3 or later (5688-813)
- OS PL/I Optimizing Compiler Version 2, Release 1 or later (5668-910)
- OS PL/I Optimizing Compiler Version 1, Release 5.1 or later (5734-PL1)

CICS also supports IBM SAA AD/Cycle Language Environment/370 Version 1, Release 1 and Release 2 run-time environment (5688-198) with the following 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)

PTF for APAR PN79363 has to be installed to use LE/370 with CICS. The symptom of the problem is abend0C4 in CEEZLOD offset X' B0A' with REG10 containing an EBCDIC data string "MSGT."

3.9.1 Execution-Time Support

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

CICS TS retains execution-time support for application programs led by the following unsupported COBOL compilers:

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

Execution-time support is withdrawn for applications compiled by the old OS/VS COBOL compilers 360S-CB-545 and 5734-CB1.

3.9.2 Storage Protection—OS/VS COBOL Programs

If you migrate OS/VS COBOL application programs from a pre-CICS/ESA 3.3 environment, be aware of possible storage-protection exceptions.

Restricted OS/VS COBOL language statements that result in a call to MVS GETMAIN services, but which worked on earlier releases, might not work when CICS storage protection is active. For example, if a CICS application program written in OS/VS COBOL is defined with EXECCKEY(USER) and it issues a restricted COBOL verb that results in an MVS GETMAIN, it abends with an 0C4 abend. In such cases, it is not the application program itself that appears to cause the 0C4, but the OS/VS COBOL routines that execute statements such as INSPECT.

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, 1997

When announcing the extended support period, the announcement letter (393-087) stated:

“The OS/VS COBOL Product Support (Limited Offering) will provide support for OS/VS COBOL only when it is used with program products, program offerings, and operating systems released prior to June 30, 1994. The OS/VS COBOL Product Support (Limited Offering) is intended for customers who need continued access to the IBM Support Center.”

The extended support was specifically intended to provide:

“Continued access to the Customer Support Center, including usage of ETR functions, for customers migrating to current COBOL products.”

It was never intended to allow the customers additional time to continue using OS/VS COBOL without any intention of migrating to a current COBOL compiler.

3.10 Batch Support

The following considerations apply for all languages:

Runtime library RLS support for COBOL, PL/I, and FORTRAN is provided by Language Environment for MVS and VM R1.5. In general, HLL languages and VSAM using RLS access maintain compatible execution with VSAM NSR where the batch application has sole usage of the dataset. However, in a VSAM RLS multiple updater environment, source changes to existing batch applications may be needed to handle new VSAM RLS (locking) errors and also to 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 has exclusive control of particular records only at a given time, while other record updates, deletes, and additions may occur concurrently? This may require some analysis of the batch application.

These programs that may require change in a batch environment:

- Use of dynamic allocation (SVC99)
- Need to check for new return/error codes. For example, many COBOL programs that check file status values after each I/O request has completed do not check the VSAM feedback return code as well. The means for doing this is documented in the *IBM COBOL for MVS and VM Programming Guide*. 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.

Language Environment for MVS and VM R1.5 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 and 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, no recompilation is required for the VSAM RLS feature. However, there is no RLS support for VS COBOL II NORES programs. Also, OS/VS COBOL programs need to migrate to one of the COBOL compilers listed above.

Additionally, APAR PN80628 for COBOL and APAR PN77375 for PLI need to be applied to LE R1.5.

3.10.1 PL/I

PL/I batch support for VSAM RLS is provided in Language Environment for MVS and VM Release 5. This allows PL/I batch noncommit 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 Language Environment will execute with VSAM RLS without re-link-edit or recompile. In addition, if you are migrating to Language Environment from OS PL/I, Language Environment PL/I provides support for 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, recompile, or both. Review the *PL/I MVS and VM Migration Guide*, SC26-3118, which give detailed requirements about recompile and re-link-edit in different products. You may also find helpful

the Language Environment migration checklist in Chapter 1 of the *Language Environment Version 1, Release 4, Run-Time Migration Guide*, SC26-8232.

PL/I APAR PN77375 has a detailed description of the implications of RLS access on the PL/I I/O language statements. This includes programming concerns related to record locking and access limitations, as well as new error codes and messages.

3.10.2 COBOL

COBOL support for VSAM RLS is provided in Language Environment for MVS and VM Release 5. This allows COBOL batch noncommit 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 have been compiled using one of the compilers supported by Language Environment. VS COBOL II NORES cannot access VSAM data sets in RLS mode, and OS/VS COBOL programs are not supported by Language Environment.

In addition to the reasons already cited for making changes to the program source, here are some other migration considerations:

- AIXBLD runtime option is not supported. VSAM in RLS mode does not support opening of empty alternate indexes and COBOL is unable to issue BLDINDEX to overcome this while in RLS mode. The user must create the cluster and alternate index in non-RLS mode prior to using the same data set in RLS mode. OPEN EXTEND is the normal means of adding records to a VSAM data set in RLS mode when an alternate index is used.
- SIMVRD run-time option is not supported with VSAM in RLS mode. It is IBM's intention to drop this run-time option in a future release.
- Many new OPEN messages were introduced with DFSMS/MVS V1, R3.0 which COBOL cannot incorporate into a set of file status values. Most of the new OPEN errors are equivalent to COBOL file status 93 (SK93) which indicates "resource not available." Consider adding checks into your program to avoid unnecessary I/O requests if the OPEN has failed. All the new OPEN messages will be available via message IEC161I with documentation from VSAM.

Chapter 4. CICS Migration

In this chapter, we describe the migration steps we performed to implement CICS TS with VSAM RLS. We also discuss miscellaneous information we found useful during migration.

4.1 Migration Environment

Figure 10 shows our target environment.

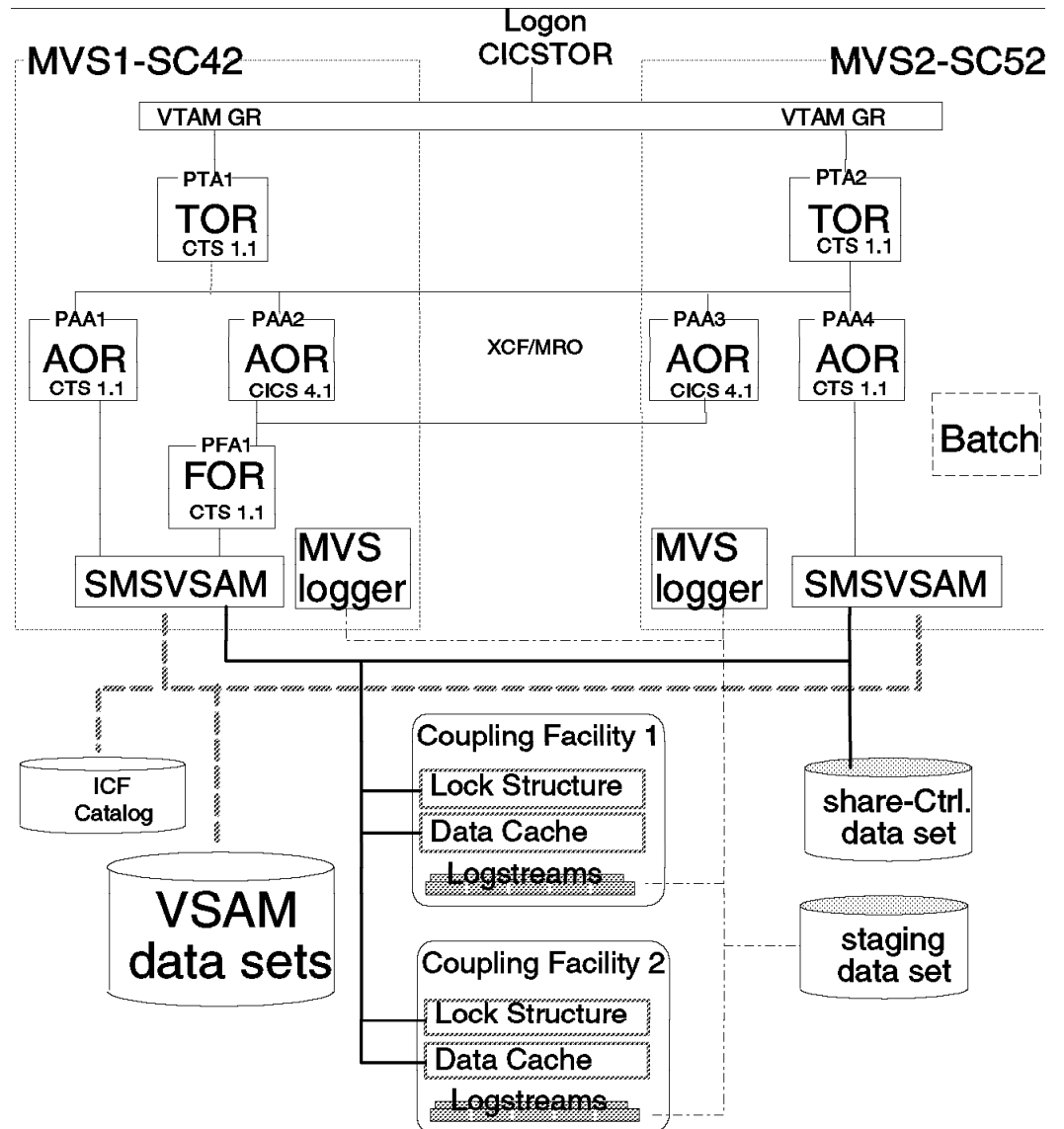


Figure 10. The Migration Environment

Initially all regions ran CICS/ESA 4.1. Our CICSplex runs in a parallel sysplex of two MVS images (SC42 and SC52), with two coupling facilities. We have a CICS terminal-owning region and two application-owning regions in each MVS image. The TORs have system IDs of PTA1 and PTA2. For all CICS regions, the

application ID is the system ID prefixed by SCSC (so the application ID for PTA1 is SCSCPTA1). Figure 10 shows only the system IDs of the regions. The two terminal-owning regions also share a generic application ID of CICSTOR.

Our target environment has one AOR running CICS/ESA 4.1 and one running CICS TS in each MVS image. This allows us to investigate a mixed-level environment.

We do not wish to access all data sets in RLS mode at this stage, so we retain our file-owning region (PFA1) in MVS SC42 to allow sharing of the non-RLS-mode data sets. Since the CICS/ESA 4.1 application-owning regions (PAA2 and PAA3) cannot access VSAM data sets in RLS mode, PFA1 is running CICS TS to allow the applications in PAA2 and PAA3 to access RLS mode data sets. The CICS TS application-owning regions (PAA1 and PAA4) do not have connections to PFA1. We use CICSplex System Manager workload separation to ensure that no transactions requiring access to non-RLS-mode data sets are routed to PAA1 or PAA4.

We also implement DFSMS 1.3 and start an SMSVSAM server address space in each MVS image.

4.2 Prerequisites

In this section we review the preliminary tasks for CICS TS installation. See the *CICS Installation Guide* for detailed installation steps. See Chapter 3, “Prerequisite Products” on page 29 for a list of prerequisite products.

1. CICS/ESA Version 3 and later releases do not support CICS internal security mechanisms. If you are migrating from CICS/MVS 2.1.2 (or an earlier unsupported release of CICS), you must implement an external security manager (ESM), such as RACF, if you are not already using an ESM.
2. Migrate from earlier CICS releases to CICS/ESA 4.1. CICS/ESA 4.1 is a sound base to prepare your applications for the exploitation of the parallel sysplex environment at your pace.
3. If you are using IMS databases and access these from CICS, you must migrate to DBCTL if you want to use these databases directly from a CICS TS region. DBCTL also allows you to share the IMS DBs among any AOR in the sysplex. You can, if necessary, function ship requests from a CICS TS region to a data-owning region (DOR) running CICS/ESA 4.1 (or an earlier release of CICS), and use the local DL/I interface in the DOR. We do not recommend this as a long-term solution, however.
4. Optionally, implement dynamic transaction routing (DTR). If you do not have a CICSplex environment already, you should begin to implement a CICSplex with dynamic transaction routing, based on your existing CICS regions. As a preliminary action for DTR we recommend you use the IBM CICS Transaction Affinities Utility (CAU), program number 5696-582. The CAU can detect and report the majority of affinities you may have in your applications.

Note: The presence of affinities does not mean you cannot implement a CICSplex. However, their presence limits your exploitation of the advantages of DTR and automatic workload balancing.

If you have not yet installed CICSplex System Manager (CICSplex SM), examine the benefits it will provide. CICSplex SM not only provides a ready-to-use DTR program, but also a single point of control over all of your

CICS regions, and you can define to CICSplex SM which affinities it must tolerate.

5. Install sysplex hardware installation

CICS TS requires the MVS system logger and therefore you have to set up a parallel sysplex with at least one coupling facility.

6. Install and test MVS/ESA 5.2 or later.

7. Set up a parallel sysplex. Aside from the software installation, one of the tasks required here is to implement a standard naming convention. See *System/390 MVS Sysplex Application Migration* for a sample naming convention in a parallel sysplex.

When you design your naming convention, include your CICSplex and the names of the CICS resources, such as data set names, queue names, connection and session definitions, and so on. Including all of these names allows you to work with wild characters and symbolic names when referencing resources and systems, and therefore facilitates cloning of systems as well as operation and maintenance.

8. Implement DFSMS 1.3. You can do this either before or after implementing CICS TS. You must implement both DFSMS 1.3 and CICS TS before implementing VSAM RLS. See Chapter 2, "Introduction to System-Managed Storage" on page 23 for more information.

See our discussion about a mixed environment in 4.7, "Mixed Environment Discussion" on page 51, in case you have to keep running an "old" CICS region. Generally such dependencies should not stop you from implementing a parallel sysplex. Start with the implementation as early as possible (now!).

4.3 Migrating to CICS TS

When we migrated our CICS/ESA 4.1 regions to CICS TS, we first set up a testing region. This region was initially implemented with no logging support. This allowed us to gain some experience with CICS TS before moving to a stage that required familiarity with the MVS system logger and the coupling facility structures required to support the MVS system logger. For a complete description of migration steps please refer to the *CICS Installation Guide* and the *CICS Migration Guide*. Items we considered more important are discussed in more detail.

1. Create the CICS data sets needed for a CICS region.

You do not have to create any journal data sets. Instead, you must provide JOURNALMODEL resource definitions so CICS can autoinstall the required log streams. Read more about logging and journaling in 4.4, "Implement Logging for CICS System Logs" on page 39.

When defining the local catalog data set (LCD) and the global catalog data set (GCD) you should carefully estimate their size. The catalog data sets are more critical in CICS TS than in earlier releases. The introduction of the CICS recovery manager means that the catalog data sets hold essential recovery information. You should rather overestimate than underestimate the catalog data set size; otherwise CICS TS will terminate when the catalog data set becomes full. See the *CICS System Definition Guide* for information on how to calculate the data set size for GCD and LCD. You can optimize

DASD space usage by allocating different sizes for each region type (FOR, TOR, AOR).

Initialize a newly created GCD using the new CICS recovery manager utility program DFHRMUTL, and set the SET_AUTO_START parameter to AUTOINIT. If you initialize the GCD with hexadecimal zeros as in earlier releases, you must set the START parameter in the SIT to INITIAL. This causes an INITIAL start of CICS. An INITIAL start is equivalent to the COLD start process in earlier releases of CICS. You can think of it as being a “supercold” or “icy-cold” start. Perform an INITIAL start of CICS the very first time you bring up a region, but not again, because all recovery information is lost after an INITIAL start. See the *CICS Migration Guide* and *CICS System Definition Guide* for more information.

2. Migrate the CICS system definition data set (CSD).

We recommend you prepare your CSD so that you can use the same CSD for all your CICS regions, including the regions using earlier releases of CICS. Each new CICS release changes the CICS-supplied groups of resource definitions that are included in the DFHLIST group list. In all cases, the old versions of the CICS resource definitions are retained in compatibility groups, which are needed to support earlier releases.

The CSD of CICS TS contains new resource types— for example, journal models and transient data queues. Sharing the CSD with prior releases could cause warning messages to appear. For example, a CICS/ESA 4.1 region using a CSD that has been upgraded to support CICS TS could produce the following message:

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

Because the CSD is a VSAM data set, it can be opened in RLS mode by CICS TS regions. However, the CSD cannot be opened in RLS mode by regions running CICS/ESA 4.1 or earlier releases of CICS. Only when all CICS regions sharing the CSD are on CICS TS can the CSD be opened in RLS mode.

Upgrade your CSD to support CICS TS using the DFHCSDUP utility with the COMPAT option. After upgrading your CSD to support CICS TS, you must include the appropriate DFHCOMPx compatibility groups in your startup group list if you want to share the CSD among regions running CICS TS and earlier releases of CICS. It is important that you install the compatibility groups in the correct order. Please refer to the *CICS Migration Guide* for detailed information, and to the *CICS Operations and Utility Guide* for information on the DFHCSDUP utility and the correct sequence of compatibility groups.

3. Define a test group list.

To quickly bring up a CICS TS region, we defined a test group list (testlist), where we added JOURNALMODEL definitions for the CICS system log that contained the type DUMMY. This enables you to bring up a test region without first having to perform all the definitions required for the MVS system logger. The following example shows the definition we used for a journal model. This definition is used as input to DFHCSDUP:

```

DEFINE GROUP(TEST) DESC('Dummy JMs CICS test')
    JOURNALMODEL(SLOG) JOURNALNAME(DFHLOG) TYPE(DUMMY)
        STREAMNAME( )
    JOURNALMODEL(SSHUNT) JOURNALNAME(DFHSHUNT) TYPE(DUMMY)
        STREAMNAME( )

```

This is only a test; you should not set the system log to DUMMY in a production system, because that inhibits CICS from doing any recovery and forces you to perform initial starts.

4. Create a SYSIN data set for SIT overrides.

See 4.8, “Miscellaneous Information” on page 52, for an overview of important new and changed SIT parameters.

5. Create the startup procedure and include it in the procedure library.

We place the SIT parameters which we want to vary when starting a specific region (as started task) in the PARM string of the EXEC DD statement.

Below is part of our procedure to start a CICS TS AOR.

```

//C51AOR  PROC START=' AUTO' ,
//          SYSIDNT=PAA1,    ** PAA1, PAA2, PAA3, etc
//          REG='32M' ,
//          OUTC='*'
//CICS510 EXEC PGM=DFHSIP,REGION=&REG,TIME=1440,
//          PARM=(' START=&START' ,
//              ' APPLID=SCSC&SYSIDNT' ,
//              ' GRPLIST=(DFHLIST,PAALIST,RLSLIST)' ,
//              ' MNSUBSYS=&SYSIDNT' ,
//              ' SYSIDNT=&SYSIDNT' ,
//              ' SYSIN')

```

6. Start the CICS TS region using START=INITIAL.

We can now use the following command to start an AOR using an INITIAL start:

```
S C51AOR,SYSIDNT=PAA1,START=INITIAL,JOBNAME=SCSCPAA1
```

The system should come up now, warning you that there is no system log.

4.4 Implement Logging for CICS System Logs

In this section, we briefly describe the tasks you have to perform in order to enable CICS to write to its system log. 1.2, “The CICS Log Manager” on page 2 gives an overview of the new logging mechanism introduced in CICS TS. You can find detailed descriptions and examples of sizing in Chapter 7, “Coupling Facility Preparation” on page 101, and of coupling facility and logger definitions in Chapter 5, “CICS Logging and Journaling” on page 57. We do not provide a detailed description in this section. For a description of migration tasks see the *CICS Migration Guide*

The tasks you must perform to enable the CICS log manager are:

1. Calculate the sizes of the coupling facility system log structures (based on log stream space requirements) and (optionally) staging data sets. The CICS-supplied utility DFHLSCU provides recommended values by analyzing your CICS/ESA 4.1 (or CICS/ESA 3.3) system logs.
2. Define two coupling facility structures, one for primary and secondary log streams (DFHLOG) and one for the “long-term” log records (DFHSHUNT).

You need the sizes you obtained from DFHLSCU before you make any definitions.

Use the IXCMIAPU utility to update the coupling facility resource management (CFRM) policy. Include the new coupling facility structures for DFHLOG and DFHSHUNT.

3. Use the IXCMIAPU utility to define and update the logger policy (LOGR). Include the definitions for the log streams of the CICS system log, DFHLOG and DFHSHUNT.
4. Create JOURNALMODEL resource definitions in CICS. Create one model for DFHLOG and another one for DFHSHUNT. Include the log stream name templates in the journal model. Note that the log type in the JOURNALMODEL definition must be "MVS" for the CICS system log.
5. Activate the new CFRM policy dynamically using the MVS command

```
SETXCF START,POLICY,POLNAME=polycyname,TYPE=CFRM
```

The first time that you start CICS with a new system log, you must specify START=INITIAL, or use the DFHRMUTL utility to set the SET-AUTO-START parameter to AUTOINIT.

4.5 Migrate Your CICSplex

In this section we outline what we consider the optimal way to upgrade a CICSplex to CICS TS regions. However, optimality very much depends on the environment and therefore may not be true for your environment. At this point in your migration, you have succeeded in bringing up one CICS TS region that uses the new MVS system logger for its system logs.

We suggest that you migrate your CICSplex step by step, but leave the CICSplex architecture as it is. That is, do not migrate to RLS yet and do not yet remove the FORs and QORs.

The very first thing to do is to install the CICS TS interregion program (DFHIRP) in a suitable LPA library and IPL MVS. The new DFHIRP is compatible with earlier CICS releases and can also communicate with earlier versions of DFHIRP in other MVS images. See the *CICS Installation Guide* for detailed instructions.

4.5.1 Common Tasks

These tasks are common for all region types (TOR, AOR or FOR):

1. Update the CSD and create a group list that includes only the resource groups you need for the specific region type (TOR, AOR, or FOR).
2. Review exit programs you are currently using and verify if they are still needed. Remove all the obsolete exits from your resource definitions. See the *CICS Migration Guide* and the *CICS Customization Guide* for details.
3. If you currently use your own shutdown procedure derived from DFH\$SDAP in earlier releases, you should review the new CICS-supplied shutdown transaction, CESD, and the sample shutdown program DFHCESD.
4. Review your transaction restart program. DFHREST has been changed to handle three new abend codes for deadlock detection. See 4.8, "Miscellaneous Information" on page 52.

5. If you have definitions for local transient data queues, migrate your destination control table (DCT) to the CSD. See 4.5.6, “Migrating the Queue-Ownning Region” on page 45 for more information.
6. If you have definitions for local temporary storage queues, recompile your TST using CICS TS macros.

4.5.2 CICS Region Migration

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

4.5.2.1 Migration Path 1

One migration sequence 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 AOR at the CICS/ESA 4.1 level, continuing to ship file control function 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 requests shipped from outside your parallel sysplex), retain the CICS TS FOR. Otherwise you can dispense with the FOR.

Figure 11 on page 42 illustrates the gradual migration process described.

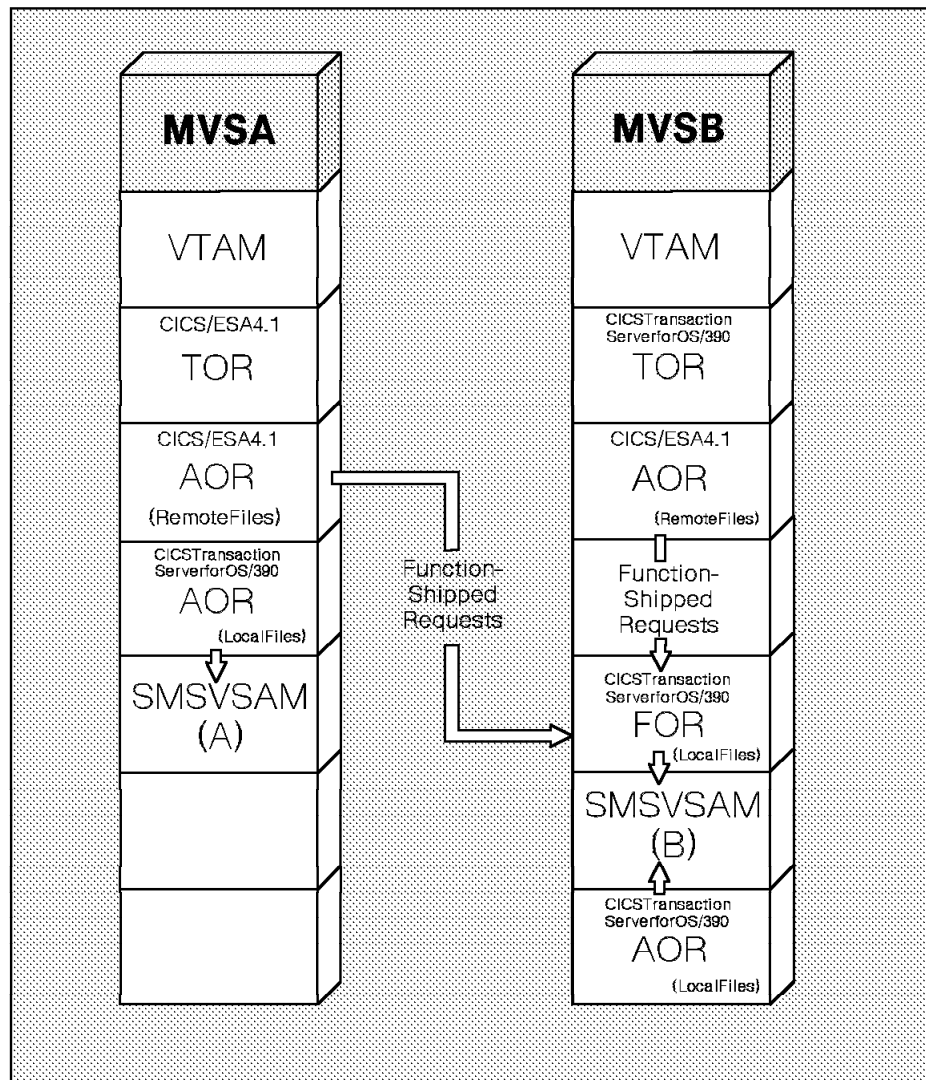


Figure 11. Migration Scenario: Function Request Shipping and RLS

Figure 11 shows the point in the migration process when two of the AORs remain at the CICS/ESA 4.1 level and two have migrated to CICS TS. The VSAM files in the CICS/ESA 4.1 regions are defined as remote, and file requests continue to be shipped to the FOR. The AORs running under CICS TS access files directly in RLS mode through the services of SMSVSAM.

Migration path 1 requires you, at some stage, to both ship function 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 path length, and you therefore may prefer the path discussed in 4.5.2.2, "Migration Path 2" on page 43.

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. Again, you may prefer the path discussed in 4.5.2.3, "Migration Path 3" on page 43.

4.5.2.2 Migration Path 2

An alternative migration sequence is to change Steps 3 and 4 in 4.5.2.1, “Migration Path 1” on page 41. 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 ship file control function 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 ship their VSAM file function 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.

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 4.5.2.3, “Migration Path 3.”

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 4.5.2.1, “Migration Path 1” on page 41, or you may prefer to continue function-request shipping to an FOR and accessing files in RLS mode from there until you have gained experience in RLS mode operation.

4.5.2.3 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 4.5.2.1, “Migration Path 1” on page 41, or 4.5.2.2, “Migration Path 2” to migrate the FOR and the remaining AORs.

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).

4.5.3 Migrating the File-Owning Region

Before you start, we recommend that you create image copies of your files, job control language, and definitions for the FOR and work out a fallback plan. See the *CICS Migration Guide* for a discussion about fallback planning. See Chapter 6, “Implementing RLS” on page 89 for more details on RLS.

The steps you have to perform to migrate a file-owning region to CICS TS are:

1. See the common steps described in 4.5.1, “Common Tasks” on page 40.
2. Decide on the number of log streams you intend to have. See 1.2, “The CICS Log Manager” on page 2 for an overview of the new logging mechanism and

Chapter 5, “CICS Logging and Journaling” on page 57 for more details about LOGR definitions.

3. Create a coupling facility structure for the forward-recovery log stream you need. Decide how many structures you need for your forward-recovery log streams based on the number of log streams and their sizes. For a discussion on sizing see Chapter 7, “Coupling Facility Preparation” on page 101.

We recommend you create additional coupling facility log structures for user journals and autojournals, separate from the forward-recovery log structure. This creates a more granular system, and makes it easier to tune coupling facility storage utilization.

4. Update the LOGR policy using the IXCMIAPU utility.
5. Create model journal resource definitions for those files requiring forward recovery, as well as for the autojournals and user journals you need, so CICS can create or associate a log stream with the file.

You can provide the forward recovery log stream name for a file in the ICF catalog; even if the file is not used in RLS mode. Review the CICS-supplied RDO group DFHLGMOD to see if it meets your requirements.

Note: Any log stream can be defined with a TYPE of SMF or DUMMY. This may be appropriate for a test environment. However, do not use a TYPE of DUMMY or SMF for CICS system logs or for forward-recovery logs in a production environment. See the *CICS Resource Definition Guide* for more information.

6. Update your file control resource definitions:

JID=SYSTEM should not be used in FILE definitions. It is obsolete. Review any FILE definitions that specify JID=1. JID=1 does not write to the system log, but to a user journal with the name DFHJ01. Review the applications that use files for which you have specified JID=SYSTEM, because it is very likely that the application wants to access the system log.

7. Review your global user exits (GLUEs) for file control:

CICS TS introduces new exits, makes some exits obsolete, and changes the function of others. The new exits are:

- XFCBFAIL
- XFCBOUT
- XFCBOVER
- XFCLDEL
- XFCQUIS
- XFCVSDS

The changed exits are:

- XFCREQ
- XFCREQC
- XFCSREQ
- XFCSREQC

The obsolete exits are:

- XRCINIT and XRCINPT

These general exits are now obsolete for most resources, including file control, although they are still used for user log records.

- XDBIN and XDBINIT

These general exits are now obsolete for all resources, including file control.

- XDBFERR and XRCOPER

These file-control-specific exits are now obsolete.

The function of many of the obsolete exits is now provided by new exits.

See 4.8.1, “New and Changed Exits” on page 52 for more information on these exits.

You should now be able to start and use the FOR. The files are not yet used in RLS mode.

4.5.4 Migrating the Application-Owning Regions

You may migrate your AORs to CICS TS one at a time. We assume here that your applications have been running successfully in CICS/ESA 4.1 regions. Do the following:

1. See the common steps described in 4.5.1, “Common Tasks” on page 40.
2. Review the program error program DFHPEP; it has changed because of the new CICS recovery manager.
3. The exits XRCINIT and XRCINPT have changed. Review them to decide if you still need them.

You should now be able to start and use the AOR, using function-request shipping to the CICS TS FOR. RLS access is not yet implemented.

4.5.5 Migrating the Terminal-Owning Region

Follow these steps:

1. See the common steps described in 4.5.1, “Common Tasks” on page 40.
2. Create a group list that includes the definitions for terminals you want your TORs to handle.
3. Review the new autoinstall program that comes with CICS TS and optionally adapt it to your local needs. See 4.8, “Miscellaneous Information” on page 52 for a short description.
4. If you have non-VTAM devices defined in your terminal control table, you must reassemble the terminal control table using the CICS TS macros.

You should be able to start and use the TOR now. Your dynamic transaction routing program should work unchanged.

4.5.6 Migrating the Queue-Owning Region

In this section we list the tasks to migrate a QOR

4.5.6.1 Transient Data Queues

1. We recommend that you migrate your DCT to the CSD, and use RDO to maintain your transient data queue definitions. To migrate the DCT, you have to change the TYPE=INITIAL macro of the DCT to look like this:

```
DFHDCT TYPE=(INITIAL,MIGRATE)
```

Once you have made this change, reassemble and link-edit the DCT. Then use the DFHCSDUP utility with the migrate function to migrate the DCT definitions to the CSD.

2. If you plan to continue using your existing DCT instead of migrating the entries to the CSD, you must remove all the CICS-required DCT entries from the source code and reassemble the table. The DCT entries are installed before CICS installs the start-up group lists and therefore prevent CICS from installing the correct level of definitions for the queues required by CICS. The CICS internal transient data queues are defined in the CSD group DFHDCTG.
3. Review the options that can be specified for a transient data queue relating to in-doubt actions (WAIT and WAITACTION). These options should conform to the respective options in your transaction definitions.

4.5.6.2 Temporary Storage Queues

You may now also migrate to shared temporary storage queues. For an overview of shared temporary storage, see 1.6, “Temporary Storage Data Sharing” on page 11. See Chapter 8, “Temporary Storage Data Sharing Servers” on page 113 for a description of how to implement shared temporary storage. The steps are these:

1. Decide which temporary storage queues can or should be shared. (You can share only nonrecoverable temporary storage queues.)
2. Make sure you have implemented a proper naming convention. Coding the exit XTSQRIN, you can manipulate the queue names, which may help to enforce your naming convention, in case your applications cannot be adapted immediately. See 4.8.1, “New and Changed Exits” on page 52, for an overview of new and changed temporary storage exits.
3. Install the temporary storage queue server.
4. Calculate the coupling facility structure size you require for the temporary storage pool.
5. Create the coupling facility structure definition and update your coupling facility policy.
6. Change the TST source to contain the new SHARED option and the SYSID of the temporary storage server, and reassemble the TST.

The addition of TS data sharing means there are now three types of TS queue:

Local queues—It is now possible to identify queues matching certain DATAIDs as local, as exceptions to following TYPE=REMOTE entries.

Remote queues—The meaning of the SYSIDNT parameter has changed. The SYSIDNT parameter is used to decide if this request is for a shared TS queue or a queue owned by a QOR.

Shared queues—The distinction between SHARED TS pools and queues owned by QORs is made with the new DFHTST TYPE=SHARED definition. This associates a SYSIDNT value with a shared TS pool rather than a QOR.

It is now possible to specify a DATAID which matches all queues using **DATAID=()**. This is also allowed on TYPE=RECOVERY, TYPE=SECURITY and TYPE=LOCAL.

7. Install the new assembled temporary storage table (TST) in all AORs. Make sure you tidy up the previous resource definitions for temporary storage queues, because CICS allows the same queue name to be used for shared as well as for local queues. The differentiation is made on the SYSID as it is today, with remote queues.

After recycling your AORs you should now be able to use the shared temporary storage queues residing in the coupling facility from all AORs. See the *CICS Release Guide* and the *CICS Application Programming Guide* for details.

4.5.7 Final Steps

In this section we list some steps needed to clean up and optimize your CICSplex:

1. Include your new regions in the CICSplex definitions of your CICSplex system manager, if applicable.
2. Review the new and changed parameters related to the CICS recovery manager:

- INDOUBT attributes on transaction definition

The INDOUBT attributes of the TRANSACTION resource definition are changed.

ACTION indicates the action to be taken if a CICS region fails or loses connectivity with its coordinator during the in-doubt period for this transaction. The action is dependent on the WAIT keyword. If WAIT specifies YES, then ACTION has no effect until the WAITTIME expires.

The INDOUBT keyword is redundant in CICS TS. but is retained for compatibility with previous releases.

The significance of the new in-doubt support to you largely depends on whether you currently suffer from in-doubt failures. For communication failures, you can check this by looking for instances of message DFHZN2101 issued by your CICS/ESA 4.1 and CICS/ESA 3.3 regions.

- XLNACTION option on connection definition

This new option specifies the action to be taken when a new log name is received from a partner system. You need to choose an appropriate value for your environment. The default value of KEEP will maintain compatibility with previous CICS releases.

- Changes to startup

An initial start has the same effect as a cold start in pre CICS TS regions.

In a cold start, 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 start to any remote partners to which it was connected during the previous run.

An initial start should be necessary only the first time you start a CICS region. You are advised not to cold start any CICS regions if possible. However, if you need to cold start for any reason, you need to be aware of the process for finding out when it is safe to cold start. This process is described in the *CICS Intercommunication Guide*.

- Changes to shutdown

There are changes to shutdown whereby you can specify a transaction to assist with the shutdown process. See 9.4 for more information.

3. For DB2: review the resource control table (RCT) and specify the parameters you require for handling in-doubt units of work. You have to reassemble the RCT using the DSNCRCT macro supplied in the CICS TS macro library. See *CICS Migration Guide* for more details.
4. Review your security requirements. We strongly recommend that you use an ESM to control access to your log stream data. If you migrate to RACF 2.1, you can use the new LOGSTRM resource class to protect log stream access. If you have an existing JCICSJCT or KCICSJCT RACF profile protecting resource DFHJ01, remember that this now protects only the user journal named DFHJ01, not the CICS system log.

4.6 Implementing RLS

In this section, we provide a list of steps you must perform to access VSAM files in RLS mode. See 1.5, “VSAM Record Level Sharing” on page 8, for an overview of VSAM RLS.

If you followed the steps we have outlined thus far in this chapter, your CICSplex should now be operating with all regions upgraded to CICS TS, but you still have an FOR. We suggest keeping this setup until after successful implementation and testing of RLS.

4.6.1 Preparing for RLS

CICS TS runs with MVS/DFP 3.3, DFSMS 1.2 or DFSMS 1.3. All of the migration steps discussed thus far can be carried out using any of these levels of software. DFSMS 1.3 is a prerequisite for VSAM RLS, so if you have not migrated to DFSMS 1.3 you must do so before you can implement VSAM RLS. First upgrade to DFSMS 1.3 without making any configuration changes, then follow the outline given in Chapter 10, “Modifying SMS for RLS” on page 149, to prepare your SMS system.

When you have implemented DFSMS 1.3, and made the changes needed to support VSAM RLS, you should have one VSAM server (SMSVSAM) running in each MVS image. During SMSVSAM installation, update the coupling facility policy with a lock structure for SMSVSAM and with a cache structure for VSAM data caching.

In CICS, you have to perform the following steps:

1. Permit CICS to open the SMSVSAM control ACB. With RACF, an example for this would be:

```
RDEFINE SUBSYSNM CAPPLID UACC(NONE) NOTIFY(userid)
PERMIT CAPPLID CLASS(SUBSYSNM) ID(reguid) ACCESS(READ)
```

CAPPLID is the CICS application ID (applid).

reguid is the CICS region user ID (userid).

When CICS tries to register with SMSVSAM, SMSVSAM calls RACF to check that the CICS region user ID has read-access authorization to a profile name in the SUBSYSNM class. The profile name must match the CICS applid.

2. Select the VSAM data sets you want to be shared among the AORs in your parallel sysplex.
3. Create image copies of the data sets you want to change to RLS mode.
4. Review your requirements for forward recovery. For your RLS data sets, the data set name of the forward recovery log stream must be specified in the ICF catalog. You can use the same forward recovery log stream for multiple data sets. For example, if you have a group of logically related data sets that always need mutual forward-recovery after a data set failure, you could send all forward-recovery records for this group to the same forward-recovery log stream.
5. Change the ICF catalog to contain the required recovery information for RLS, using an IDCAMS ALTER job. See Chapter 6, “Implementing RLS” on page 89, for details.
6. Update your file resource definitions in CICS using RDO to contain the RLSACCESS=YES attribute. Make sure you delete any remote system attributes present, because they are still valid in RLS mode. Unless your applications have a need for read integrity, let the READINTEG attribute default to UNCOMMITTED, because that is what you have today.

We suggest that you copy your existing file definitions to a new group for RLS files. You can easily add this new group to your AOR group list. Then delete the non-RLS files from this new group and the RLS files from the old group.

7. Review any file control exits you are currently using and check if you still require the logic they provide. Most of the exit points for file control have been changed. See 4.8.1, “New and Changed Exits” on page 52, for an overview and the *CICS Release Guide* for details.
8. Enable RLS mode in the SIT (RLS=YES), and stop CICS. You will probably want to perform a warm start and use CEDA to install your new file definitions into the running system, although if this is not a production region you could use a cold start and implement the new file definitions immediately.

The CICS region should now connect to SMSVSAM automatically and you should be able to access the defined files in RLS mode.

When migrating from an AOR/FOR to the AOR/RLS configuration, the logging activity from the FOR will be distributed across the AORs. This will impact both the average buffer size and the structure space requirements for each of the AORs. The structure space requirement of each AOR will be similar to the current space requirement of the FOR.

Refer to Chapter 6, “Implementing RLS” on page 89, for more details on RLS implementation.

4.6.2 Verify RLS Operation

To verify if RLS is active you may perform a quick test. Use CEMT to inquire about a file that should be open in RLS mode, as shown in Figure 12 on page 50:

```

CEMT I FILE(ACCUNTDDB)
STATUS: RESULTS - OVERTYPE TO MODIFY
          Fil(ACCUNTDDB) Vsa Ope Ena Rea          Sha Rls  NORMAL
                   Dsn( CICSDSW.VSAMU.ACCUNTDDB          )

```

Figure 12. Using CEMT to Inquire about an RLS Mode VSAM File

This will work only if you executed the IDCAMS ALTER job, so the catalog contains a valid value for LOG in the RLSDATA section. A null value (instead of Rls) on the CEMT panel means that the file is not accessed in RLS mode.

Now use IDCAMS to issue LISTCAT command:

```
LISTCAT ENTRY(' CICSDSW.VSAMU.ACCUNTDDB') ALL
```

This will verify the catalog entries. Your LISTCAT output should resemble that in Figure 13.

```

CLUSTER ----- CICSDSW.VSAMU.ACCUNTDDB
IN-CAT --- CATALOG.TOTICF1.VTOTCAT
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1996.172
  RELEASE-----2      EXPIRATION-----0000.000
SMSDATA
  STORAGECLASS ----CICSRLS      MANAGEMENTCLASS-CICSRLSM
  DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
  BWO STATUS-----00000000      BWO TIMESTAMP---000000 00:00:00.0
  BWO----- (NULL)
RLSDATA
  LOG -----ALL      RECOVERY REQUIRED --(NO)
  VSAM QUIESCED -----(NO)      RLS IN USE -----(YES)
  LOGSTREAMID-----CDSWV.FWRLOG.ACCUNTDDB

```

Figure 13. Sample Output from LISTCAT Operation for RLS Data Set

If you now quiesce the data set using CEMT (as shown in Figure 14), you cannot reopen it in RLS mode until you unquiesce it.

```

CEMT SET DSN(CICSDSW.VSAMU.ACCUNTDDB) Q
STATUS: RESULTS - OVERTYPE TO MODIFY
          Dsn(CICSDSW.VSAMU.ACCUNTDDB          ) Vsa      BEING QUIESCED
          Fil(0001) Val Bas Und Und          Ava      Unq

```

Figure 14. Using CEMT to Quiesce an RLS Mode VSAM File

Look at the file status in CICS; it is now closed and disabled, as shown in Figure 15 on page 51.


```

CEMT I FILE(ACCUNTDDB)
Fil(ACCUNTDDB) Vsa Clo Une Rea          Sha Rls
      Dsn( CICSDSW.VSAMU.ACCUNTDDB      )

```

Figure 15. Using CEMT to Inquire about Quiesced RLS Mode VSAM File

CICS shows the file status as closed. The VSAM catalog shows the file is in the quiesced state. Your LISTCAT output should be similar to that in Figure 16.

```

RLSDATA
LOG -----NONE    RECOVERY REQUIRED --(NO)
VSAM QUIESCED -----(YES)    RLS IN USE -----(YES)
LOGSTREAMID----- (NULL)
RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'

```

Figure 16. Sample Output from LISTCAT Operation for Quiesced Data Set

RLS IN USE -----(YES) will be changed to NO as soon as the data set is opened next, because it can now be opened only in non-RLS mode.

Use CEMT to unquiesce the data set again and then run a function test, still using the FOR, but accessing the files in RLS mode.

If this test is successful, you can update the file definitions in your AORs to RLS mode and delete the remote system attributes, so the RLS files no longer point to the remote FOR.

Your VSAM RLS environment should work now.

4.6.3 Changing Access Mode

If you need to access the data set in non-RLS mode – for a daily batch update for example – you must quiesce the data set before you run the batch job, and then unquiesce it afterward. CICS provides a set of sample programs for mode change to help you automate this process. See the *CICS Operations and Utility Guide* for a description.

Note: Update your procedures for recovery and restart before you implement VSAM RLS in a production environment. We do not discuss recovery and restart in this book. See the *CICS Recovery and Restart Guide* and *CICS and VSAM Record Level Sharing: Recovery Considerations* for information on recovery and restart.

4.7 Mixed Environment Discussion

In this section, we discuss the options you have if you must keep one or more back-level CICS regions connected in your parallel sysplex.

- If you still need local DL/I support, then you have not yet migrated to DBCTL. The answer is:
 - Work out a plan for migrating to DBCTL. If you take a close look at the benefits you get with DBCTL, you should come to the conclusion that the migration cost will pay off, especially in a parallel sysplex environment.

- Until you have migrated, you can keep a CICS/ESA 4.1 region as a data-owning region (DOR), which owns the DL/I databases.
- You can then ship the DL/I function requests from a CICS TS application-owning region to the data-owning region.
- If you need a recoverable temporary storage queue, and if a recoverable TS queue must be shared among several AORs, then you must set up a queue-owning region rather than using the new shared temporary storage support.
- You are advised to upgrade all systems in a sysplex to DFSMS/MVS 1.3. However, if you intend to have a sysplex with mixed levels of DFSMS, you should plan the coexistence of the systems carefully.

If you haven't converted all your MVS systems to DFSMS 1.3, then 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 1.3 systems so that they do not conflict with RLS access. However, if you convert your SMS configuration to SMS 32-name mode under DFSMS 1.3, toleration is not supported. Therefore, do not convert to SMS 32-name mode until all systems in the sysplex have been converted to DFSMS 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 define cache sets and other RLS-related SMS functions only through Interactive Storage Management Facility (ISMF) on a DFSMS 1.3 system. An SMS source control data set (SCDS) altered by ISMF on a DFSMS 1.3 system can be activated from any of the systems that share the configuration.

You can share a catalog between DFSMS 1.3 and earlier releases. LISTCAT output from DFSMS 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 1.3 Planning for Installation*.

4.8 Miscellaneous Information

In this section, we provide information we found useful during our own migration. The information also supplements the migration steps given in this chapter.

4.8.1 New and Changed Exits

This section lists the important global user exits and user-replaceable modules, which you should review if you are currently using exits in your installation. See the *CICS Customization Guide* for details of exit programs and the *CICS Release Guide* for a complete list of new, changed, and obsolete exits:

- XRCINIT and XRCINPT – These existing global user exits are invoked at warm and emergency restart for user back-out only. These exits are now invoked for user log records only, not for any of the other resources that they used to be for. These exit points are referenced by SIT parameter TBEXITS, name1 and name2.
- DFHZATDY – The terminal autoinstall program

The interface between DFHZATA and the terminal autoinstall URM has been extended. This may affect your autoinstall program. The CICS-supplied

DFHZATDY includes code to handle APPC generic resources and the deletion of APPC sessions and other changes.

- DFHREST – The transaction restart program

The CICS-supplied transaction restart program DFHREST has been modified to restart transactions that abended with the new deadlock abend codes:

- ADCD – DBCTL deadlock
- AFCF – Deadlock detected by CICS file control for non-RLS files.
- AFCW – Deadlock detected by VSAM for RLS files.

If you have your own restart program, update it to be able to handle these abends correctly.

- New and changed exits for file control – CICS provides four global user exits that you can use in connection with file control recovery operations. These are:
 - XFCBOUT – Invoked when CICS is about to back out a file update. XFCBOUT replaces XDBIN and XRCINPT for file control. The exit point is referenced by SIT parameter TBEXITS, name6.
 - XFCBFAIL – Invoked when an error occurs during back-out. XFCBFAIL replaces the back-out failure function of XDBFERR, XRCFCER, and XRCOPER. The exit point is referenced by SIT parameter TBEXITS, name3.
 - XFCLDEL – Invoked when backing out a write to a BDAM file or a VSAM ESDS data set. These file structures do not allow physical deletes, so the exit performs a logical delete to back out the write. XFCLDEL replaces the logical delete function of XDBFERR and XRCFCER. The exit point is referenced by SIT parameter TBEXITS, name4.
 - XFCBOVER – Invoked when CICS is about to skip unit-of-work back-out, because a batch program has overridden RLS retained lock protection and opened a data set for batch processing (which is possible as a last resort but not recommended for normal operation). The exit point is referenced by SIT parameter TBEXITS, name5.
- XFCSREQ and XFCSREQC have both been extended to support a cancel or close request. Cancel or close is issued by CICS when you cancel (through unquiescing) a previously issued quiesce request.
- XLGSTRM – A new exit in the CICS log manager domain. You can use XLGSTRM to modify the log stream name at the time CICS requests the MVS system logger to create a new log stream.
- New and changed exits for temporary storage include these:
 - XTSQRIN – Invoked before execution of a temporary storage request (for example WRITEQ TS). You can modify the temporary storage type parameter (UEP_TS_STORAGE_TYPE) but only if this request actually creates the queue. This new exit allows the exit program to change a queue from MAIN to AUXILIARY, or vice versa.
 - XTSQROUT – Invoked after execution of a temporary storage request. No parameters can be modified.
 - XTSPTIN – Invoked before the execution of a temporary storage interface request for a CICS internal queue (for example, for interval control or BMS queues). You can modify the temporary storage type parameter (UEP_TS_STORAGE_TYPE) but only if this request actually creates the queue.

- XTSPTOUT – Invoked after execution of a temporary storage interface request for a CICS internal queue. No parameters can be modified.

Temporary storage exits XTSREQ, XTSIN, and XTSOUT have been removed.

4.8.2 New and Changed SIT Parameters

See the *CICS System Definition Guide* for details.

- The following SIT parameters have been changed:
 - AILDELAY – Now applies to APPC connections as well as terminals.
 - AIRDELAY – Now applies to APPC connections as well as terminals.
 - AKPFREQ – Make sure the value is not set to 0. This parameter influences the offloading of log streams from the coupling facility and deletion of log data that is no longer needed. For a rough estimation, you can set an AKPFREQ-value of 4000 to be approximately equivalent to 30 seconds in a heavily loaded CICS system, meaning that an activity key point is taken every 30 seconds. You should tune up to the point, where most of your transactions can finish or write a synch point within one AKPFREQ period. That avoids writing system log records to the log stream data set, but only if you specify SYSLOG=NOKEEP in the SIT.
 - CSDFRLOG and CSDRECOV – If you specify CSDRECOV=NONE or BACKOUTONLY then CICS does not write forward-recovery log data for the CSD. If you specify CSDRECOV=ALL and CSDFRLOG=1 or “nn” CICS will write a forward-recovery log for your CSD. In this case, however, you must define a JOURNALMODEL so CICS can create a log stream where the last qualifier is set to DFHJ01 or “nn,” respectively. Journal DFHJ01 is not the CICS system log, as in was in earlier releases! If CSDRLS is YES, CSDFRLOG and CSDRECOV do not specify the recoverability attributes of the CSD. In that case, the recoverability must be defined in the ICF catalog.
 - DCT – NO is now the default. The DCT is no longer needed because you can use RDO to provide TD resource definitions.
 - PDIR – NO is now the default. CICS TS supports DL/I database access only through DBCTL or remote DL/I. Therefore, PDIR is valid only for remote DL/I access.
 - START – INITIAL has been added; COLD has changed its scope.
 - SPCTRxx and STNTRxx – You can specify new component codes for tracing to include the new CICS domains (LG, NQ, RI, and RM).
 - TBEXITS – Two exit names are added and the others now refer to different global user exit points. See also 4.8.1, “New and Changed Exits” on page 52.
- The following SIT parameters are new:
 - CSDINTEG – You can now specify read-integrity options for a CSD that is accessed in RLS mode.
 - CSDRLS – YES specifies that the CSD is to be accessed in RLS mode.
 - FTIMEOUT – Specifies the default timeout value for access requests to files opened in RLS mode. This parameter is used only if you do not specify DTIMOUT in a transaction definition. Both parameters are used by SMSVSAM to determine the timeout for a deadlock situation.
 - OFFSITE – YES specifies that CICS is to perform an OFFSITE restart, which is intended for use when carrying out remote site (or disaster) recovery.

- QUIESTIM – Specifies the timeout for the SET DATASET(xy) QUIESCE operation. A data set opened in RLS mode has to be quiesced before it can be opened in non-RLS mode.
- RLS – YES specifies that you want CICS to support VSAM RLS.
- SDTRAN – Specifies the name of your shutdown transaction. CESD is the new CICS-supplied default shutdown transaction.
- SYSLOG – Specifies whether you want CICS to delete obsolete log data, also known as log-tail deletion. IBM recommends setting this parameter to NOKEEP (which is default). If you set it to KEEP, the log-tail deletion is inhibited and all log data is written to the log stream data sets. If you specify KEEP, the maximum allowed number of log streams will quickly be reached and CICS will be stopped.
- TDINTRA – Specifies whether CICS should purge (EMPTY) or recover (NOEMPTY) your intrapartition TD queues on a warm start or emergency restart.
- UOWNETQL – Allows you to specify a qualifier for the network UOW-ID (NETUOWID). This parameter is used to qualify UOWs initiated in the local CICS region, if the networkID cannot be obtained from VTAM.

4.8.3 New ABEND Codes

The following is a list of new abend codes that may occur. The list is not comprehensive (see the *CICS Messages and Codes* for details):

- AEX4 – Occurs if a transaction receives the RECORDBUSY condition and does not handle this.
- AEX8 – Occurs if a transaction receives the LOCKED condition and does not handle this.
- AFCK – Occurs if a transaction issues a record management update against a data set, for which a sharp image copy is in progress initiated by DFSMS 1.3.
- AFCR – Occurs if a transaction issues a request against a file opened in RLS mode, but a catastrophic VSAM failure has occurred.
- AFCS – Occurs if a transaction issues a request against a file opened in RLS mode, but RLS has been disabled due to an earlier VSAM failure.
- AFCT – Occurs if a transaction issues a request against a file opened in RLS mode and a VSAM failure occurs.
- AFCU – Occurs if a transaction issues a request against a file opened in RLS mode, but VSAM detects lost locks due to an earlier VSAM failure.
- AFCW – Occurs if a transaction attempts to update a data set for which a copy is in progress initiated by DFSMSdss. This can occur on the following file API commands:
 - DELETE
 - READ UPDATE
 - READNEXT UPDATE
 - READPREV UPDATE
 - REWRITE
 - WRITE
- AFCV – Occurs for transactions that are waiting for a lock, but the deadlock timeout mechanism of SMSVSAM has been triggered. The value for

deadlock timeout is specified either in the DTIMOUT parameter of the transaction resource definition, or in the FTIMEOUT SIT parameter.

Chapter 5. CICS Logging and Journaling

This chapter describes the CICS TS logging and journaling facilities.

It addresses the following topics:

- 5.1, “Overview of Logging and Journaling”
- 5.2, “Setting Up System Logs”
- 5.3, “Setting Up General Logs”
- 5.4, “Management of CICS Logs and Journals”
- 5.5, “Monitoring Log Streams”

5.1 Overview of Logging and Journaling

CICS TS introduces a new domain, the CICS log manager. The CICS log manager replaces the journal control function of earlier releases. The CICS log manager improves the management of the CICS system log, forward-recovery logs, autojournals, and user journals.

The CICS log manager uses services provided by the new MVS system logger, which is an integrated MVS logging facility, designed to exploit coupling-facility technology. The ability to have real-time merging of logs significantly reduces the complexity of managing multiple logs.

The MVS system logger is a set of services that allows an application to write, browse, and delete log data. You can use MVS system logger services to merge data from multiple instances of an application, including merging data from different systems across a parallel sysplex.

For example, suppose you are concurrently running multiple instances of an application in a parallel sysplex, and each application instance can update a common data base. It is important for your installation to maintain a common log of all updates to the data base from across the parallel sysplex, so that if the data base should be damaged, it can be restored from the backup copy. You can merge the log data from applications across the parallel sysplex into a log stream, which is simply a collection of data in log blocks residing in a coupling facility list structure, on DASD, or both.

When an application writes a log block to a log stream, the MVS system logger writes it first to a coupling facility list structure. Over time, the MVS system logger moves the log block from the coupling facility list structure to DASD, so that the storage in the coupling facility occupied by the list structure can be used to hold new log blocks. The actual location of the log data in the log stream is transparent to the application.

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, which is also used for dynamic transaction back-out. (The CICS internal dynamic log of earlier releases does not exist in CICS.)
- Forward-recovery logs, autojournals, and user journals (general log)

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 images in a parallel sysplex.

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

The CICS log manager, with the MVS system logger, improves the management of 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. All write requests to a log device with a type of DUMMY receive a response of NORMAL, but the data is discarded.

Each CICS region has only one system log, although this can be defined as a dummy if not required. In CICS, the system log is implemented as two MVS system logger log streams, but together they form a single logical log stream. The CICS system log is intended for use only for recovery purposes – for example, during dynamic transaction back-out, or during emergency restart. It is not meant to be used for any other purpose.

5.2 Setting Up System Logs

Implementing logging and journaling in CICS TS requires setting up two system components: the CICS log manager and the MVS system logger. The CICS log manager is customized by the new CICS resource type JOURNALNAME. The MVS system logger is customized by the IXCMIAPU utility to define the log stream and the coupling facility structure that contains the log stream.

5.2.1 Defining a Dummy System Log to CICS

For test purposes and for CICS regions that do not require any logging functions, the system log can be defined as DUMMY. Figure 17 shows a sample DFHCSDUP job that can be used to define a dummy system log.

```
//DFHCSDUP JOB (999,P0K),'CICS510',CLASS=A,MSGCLASS=T,
//          NOTIFY=&SYSUID
//UPGRCD   EXEC PGM=DFHCSDUP,REGION=1M
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//DFHCSD  DD DSN=CICS.V5R1M0.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
ADD GROUP(TEST) LIST(TESTLIST)
DEFINE JOURNALMODEL(DFHLOG) JOURNALNAME(DFHLOG)
        TYPE(DUMMY)
        GROUP(TEST)
//
```

Figure 17. Sample DFHCSDUP Job to Define a Dummy System Log

The dummy system log definition can only be used with a INITIAL start of the CICS region, and the list created by the sample job, TESTLIST, must be

activated. If you use a dummy system log, the CICS log manager does not require a connection to the MVS system logger or the coupling facility list structure, and all the requests to update the system log are nullified.

CICS cannot perform a cold, warm or emergency restart if TYPE(DUMMY) is specified on the JOURNALMODEL definition.

5.2.2 Defining the MVS System Logger Components

The CICS system log is implemented as two MVS system-logger log streams. One stream is the primary system log stream, DFHLOG, which holds data for most normal (short-lived) in-flight units of work. The other stream is the secondary system log stream, DFHSHUNT, which holds information for UOWs that are not short-lived. Typically these are units of work that cannot complete because of back-out failures, or because they are designed as long-running tasks that issue infrequent synch points.

The two log streams are associated with structures managed by the coupling facility. We recommend that you specify different structures in the coupling facility when you define log streams for DFHLOG and DFHSHUNT.

5.2.2.1 Defining Coupling Facility Structures

Figure 18 on page 60 shows an IXCMIAPI job that can be used to define coupling facility structures, including the structures for the CICS system log.

```

//DEF CFRM JOB (999,POK), 'L06R', CLASS=A, REGION=4096K,
//          MSGCLASS=X, TIME=10, MSGLEVEL=(1,1), NOTIFY=&SYSUID
//IXCMIAPU EXEC PGM=IXCMIAPU
//*****
/*
/* DEFINE STRUCTURES FOR THE COUPLING FACILITY
/*
//*****
//STEPLIB DD DSN=SYS1.MIGLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(CFRM) REPORT(YES)

DEFINE POLICY NAME(CFRM11 )
*
* These define the CFs characteristics
*
CF NAME(CF01) DUMPSPACE(2048) PARTITION(1) CPCID(00)
TYPE(009672) MFG(IBM) PLANT(02) SEQUENCE(000000040104)

CF NAME(CF02) DUMPSPACE(2048) PARTITION(1) CPCID(01)
TYPE(009672) MFG(IBM) PLANT(02) SEQUENCE(000000040104)
*
* These define the structures for VSAM-RLS server
*
STRUCTURE NAME(CACHECICS) SIZE(70000)
INITSIZE(50000)
REBUILDPERCENT(75)
PREFLIST(CF02, CF01)

STRUCTURE NAME(CICS_CACHE) SIZE(70000)
INITSIZE(50000)
REBUILDPERCENT(75)
PREFLIST(CF02, CF01)
*
* These define the structures for CICS TS queues
*
STRUCTURE NAME(DFHXQLS_TSQSPQA1) SIZE(1000)
INITSIZE(500)
REBUILDPERCENT(10)
PREFLIST(CF02, CF01)
*
* These define the structures for FW-Recovery Logs
*
STRUCTURE NAME(LOG_CDSWU_001) SIZE(2320)
INITSIZE(1512)
REBUILDPERCENT(10)
PREFLIST(CF02, CF01)

STRUCTURE NAME(LOG_CDSWV_001) SIZE(2320)
INITSIZE(1512)
REBUILDPERCENT(10)
PREFLIST(CF02, CF01)
*
* These define the structures for CICS System Log
*
STRUCTURE NAME(LOG_DFHLOG_001) SIZE(24320)
INITSIZE(16128)
REBUILDPERCENT(10)
PREFLIST(CF02, CF01)

STRUCTURE NAME(LOG_DFHSHUNT_001) SIZE(2780)
INITSIZE(1812)
REBUILDPERCENT(10)
PREFLIST(CF02, CF01)
/*
//

```

Figure 18. Sample IXCMIAPU Job to Define System Log List Structures

The sample job contains only the structure definitions for CICS and VSAM RLS-related structures. You can find a complete description of the parameters

and the correct procedure to activate the coupling facility resource management (CFRM) policy in *MVS/ESA Setting Up a Sysplex*.

IXCMIAPU adds or changes the policy data in the administrative policies only. The utility does not change any information in the system copy of the active CFRM policy. To switch to the updated policy, you have to issue the following MVS command on the system console:

```
SETXCF START,POLICY,POLNAME=CFRM11,TYPE=CFRM
```

Chapter 7, “Coupling Facility Preparation” on page 101 describes how to size the coupling facility structures.

The maximum number of coupling facility structures that can be active for each policy is 128, and only one policy can be active at a time.

5.2.2.2 Defining Log Streams

You also use the IXCMIAPU utility to define log streams and the related structures, using the LOGR policies. The LOGR couple dataset must be activated and available before you add log stream and structure definitions to it.

The log stream can be defined as a model by specifying MODEL(YES). MVS system logger applications cannot connect to a model log stream. When defining a log stream model, make sure that no log streams are allocated yet. The log stream model can be referred to when you specify the LIKE parameter in a log stream definition or by applications (such as CICS) which use the MVS DEFINE command.

A log stream definition with the MODEL(NO) parameter, which is the default, can be used by applications to directly connect to the log stream.

The CICS log manager can either connect to a log stream defined as MODEL(NO) or create a log stream using characteristics taken from a log stream model. The models for the system logs are named &SYSNAME.DFHLOG.MODEL for the primary system log and &SYSNAME.DFHSHUNT.MODEL for the secondary system log. The log stream model can be defined once and referred to by all the CICS regions at connect time, decreasing the complexity of the resource definitions. A disadvantage of using the log stream model is that all log streams are mapped to the coupling facility structure defined in the model. Connection to a predefined log stream is faster for an INITIAL start because the CICS log manager does not have to manage the “not found” error when it tries to connect to its log stream using an IXGCONN request, and then recover issuing an MVS CREATE log stream request.

Figure 19 on page 62 shows a sample job to define the LOGR policy. The sample defines explicit system log streams for CICS region SCSCPAA2, and log stream models for all other CICS regions initially started on MVS image SC52. The log streams are associated with the two system log structures we created in Figure 18 on page 60. The job shown in Figure 19 on page 62 also defines log related information for our two system log structures.

```

//DEF CFSTR JOB (999,P0K), 'CICS510', CLASS=A, MSGCLASS=T,
//          NOTIFY=&SYSUID
//IXCMIAPU EXEC PGM=IXCMIAPU
//*****
/*
/* DEFINE THE LOGSTREAMS ON THE LOGR DS
/*
//*****
//STEPLIB DD DSN=SYS1.MIGLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(LOGR) REPORT(YES)

/*-----*/
/* MAXBUFSIZE, AVGBUFSIZE AND STG_SIZE OBTAINED */
/* FROM THE DFHLSCU UTILITY */
/* LOGSNUM = 10 */
/*-----*/
DEFINE STRUCTURE
  NAME(LOG_DFHLOG_001)
  LOGSNUM(10)
  MAXBUFSIZE(64000)
  AVGBUFSIZE(256)
DEFINE LOGSTREAM
  NAME(SC52.DFHLOG.MODEL)
  STRUCTNAME(LOG_DFHLOG_001)
  STG_DUPLEX(YES)
  DUPLEXMODE(COND)
  STG_DATACLAS(NO_STG_DATACLAS)
  STG_MGMTCLAS(NO_STG_MGMTCLAS)
  STG_STORCLAS(NO_STG_STORCLAS)
  STG_SIZE(53014)
  LS_DATACLAS(NO_LS_DATACLAS)
  LS_MGMTCLAS(NO_LS_MGMTCLAS)
  LS_STORCLAS(NO_LS_STORCLAS)
  LS_SIZE(53014)
  HLQ(CICS)
  HIGHOFFLOAD(95)
  LOWOFFLOAD(60)
  MODEL(YES)
DEFINE LOGSTREAM
  NAME(SCSCPA2.TST.DFHLOG)
  LIKE(SC52.DFHLOG.MODEL)

DEFINE STRUCTURE
  NAME(LOG_DFHSUNT_001)
  LOGSNUM(10)
  MAXBUFSIZE(64000)
  AVGBUFSIZE(4096)
DEFINE LOGSTREAM
  NAME(SC52.DFHSUNT.MODEL)
  STRUCTNAME(LOG_DFHSUNT_001)
  STG_DUPLEX(YES)
  DUPLEXMODE(COND)
  STG_DATACLAS(NO_STG_DATACLAS)
  STG_MGMTCLAS(NO_STG_MGMTCLAS)
  STG_STORCLAS(NO_STG_STORCLAS)
  STG_SIZE(5000)
  LS_DATACLAS(NO_LS_DATACLAS)
  LS_MGMTCLAS(NO_LS_MGMTCLAS)
  LS_STORCLAS(NO_LS_STORCLAS)
  LS_SIZE(5000)
  HLQ(CICS)
  HIGHOFFLOAD(80)
  LOWOFFLOAD(0)
  MODEL(YES)
DEFINE LOGSTREAM
  NAME(SCSCPA2.TST.DFHSUNT)
  LIKE(SC52.DFHSUNT.MODEL)
/*

```

Figure 19. Sample IXCMIAPU Job to Define the LOGR Policy

The name of the explicitly defined log streams is fixed in the definition. The name of any log stream derived from the log stream model is determined when the log stream is created.

Note: The name of the explicitly defined log stream assumes that the VTAM application ID of the CICS region is unique inside the parallel sysplex configuration.

Methods for assigning appropriate values to MAXBUFSIZE, AVGBUFSIZE, and STG_SIZE are described in Chapter 7, “Coupling Facility Preparation” on page 101.

The LOGSNUM parameter on the DEFINE STRUCTURE statement defines the maximum number of log streams that can be mapped to the structure. This is one of the parameters that can optimize coupling facility utilization. Log streams can be assigned to coupling facility structures in a variety of ways: multiple log streams can share one structure, each log stream can have a unique structure, or related log streams can share the use of a structure. When you map more than one log stream to a single coupling facility structure, the MVS system logger divides the structure storage evenly among the log streams that have at least one connected MVS system logger application. For example, if an installation assigns three log streams to a single structure, but only one log stream has a connected MVS system logger application, then that one log stream can use the entire coupling facility structure. When an application connects to the second log stream, the MVS system logger dynamically divides the structure evenly between the two log streams. When an application connects to a third log stream, the MVS system logger allocates each log stream a third of the coupling facility space.

It is important that the log streams that share a coupling facility structure be similar in size and usage rates. Knowing size and rate can help you understand and plan how much coupling facility space you have for each log stream and predict when log stream data will be written to DASD as the coupling facility structures fill. The more connected log streams that share a coupling facility structure, the less facility space each has. In addition, when a structure is allocated for the MVS system logger, some of the structure space is taken up for control information for every log stream that could be connected to this structure. The higher the number for LOGSNUM, the more coupling-facility space is consumed.

In general, the value you define for LOGSNUM should be as small as possible. Try not to exceed 20 log streams per structure.

5.2.2.3 Defining Log Data Sets

When an MVS system logger application writes data to the log stream, the MVS system logger writes it first to the coupling facility list structure associated with the log stream. When the structure space allocated for the log stream reaches the installation-defined threshold, the MVS system logger moves (offloads) the log blocks from the coupling facility structure to VSAM linear data sets on DASD, so that the facility space for the log stream can be used to hold new log records.

A log stream can have data in multiple DASD log data sets. As a log stream fills log data sets on DASD, MVS system logger automatically allocates new ones for the log stream.

We recommend that you use Storage Management Subsystem (SMS) to manage DASD log data sets. The MVS system logger will tolerate the use of DFSMSHsm for another product to migrate data sets. However, a maximum number of 168 data sets can be allocated at any one time. This limit is lifted after you have installed OS/390 Release 3.

You can use the LS_SIZE option when defining the log stream, or Automatic Class Selection (ACS) routines, to define the size of the log data sets. Also, use the LS_DATACLAS and STG_DATACLAS options, or ACS, to ensure that logger offload data sets (and logger staging data sets) are allocated with share options (3,3).

5.2.2.4 Log Data Sets: Controlling Offloading

The HIGHOFFLOAD parameter on the log stream definition specifies the percent value you want to use as the high offload threshold for the coupling facility space allocated for this log stream. When the space is filled to the high offload threshold point or beyond, the MVS system logger begins its offload process. Ideally, when this occurs, CICS has flagged enough system log data for deletion (see 5.2.2.8, “Special Considerations for CICS System Logs” on page 68) such that it is not necessary for the system logger to actually offload data from the coupling facility. You should specify a HIGHOFFLOAD which avoids structure-full events; the recommended value is 95.

The LOWOFFLOAD parameter on the log stream definition specifies the percent value you want to use as the low offload threshold for the coupling-facility space allocated for this log stream. The low offload threshold is the point in the coupling facility, in percent value of space consumed, where the MVS system logger stops offloading log data to log stream DASD data sets. If data is to be offloaded, either by choice (SYSLOG=KEEP or because coupling facility space is severely limited) or by chance (because of fluctuating workload), then data should be offloaded in large chunks to prevent too frequent offloads (more frequently than once every 5 seconds). It is the difference between the HIGHOFFLOAD and LOWOFFLOAD values that determines how much data is offloaded at a time. The LOWOFFLOAD recommendation is to use the value estimated by DFHLSCU. Alternatively, you can use the formula provided in the *CICS Performance Guide* if you know the interval between activity keypoints and the execution time of your longest running UOW. If you do not want to use a formula, or you do not have an estimate from DFHLSCU, a LOWOFFLOAD value of 60 is a conservative estimate and should ensure that data is offloaded in reasonably sized chunks, but that data for inflight UOWs is never offloaded to DASD. For SYSLOG=KEEP, set LOWOFFLOAD to allow offloading to occur in large chunks (the recommendation is 5).

For primary system logs, the removal of log records by activity keypoint processing should make enough coupling facility storage available to avoid reaching the HIGHOFFLOAD point. There should be no need for any live data to be offloaded from a primary system coupling facility structure to DASD.

5.2.2.5 Log Data Sets: Additional Controls on Offloading

The HIGHOFFLOAD and LOWOFFLOAD parameters you use when you use the IXCMIAPI utility to define a log stream provide the most significant control on offloading. However, other log stream parameters can also affect offloading from the coupling facility list structure to DASD.

AVGBUFSIZE and MAXBUFSIZE Parameters: The log streams that share a coupling facility structure should have similar values for AVGBUFSIZE and MAXBUFSIZE. When a coupling facility structure is defined, it is divided into two areas, one of which holds list entries while the other holds list elements.

List elements are units of logged data and are either 256 bytes or 512 bytes long. List entries are index pointers to the list elements. There is one list entry per log record, and at least one element per record.

If you define MAXBUFSIZE with a value greater than 65276, data is written in 512-byte elements. If you define MAXBUFSIZE with a value less than or equal to 65276, data is written in 256-byte elements. The maximum value permitted for this parameter is 65532.

The proportion of the areas occupied by the list entries and the list elements is determined by a ratio calculated as follows:

AVGBUFSIZE
element size

The resulting ratio is $nn : 1$, where nn represents element storage, and '1' represents entry storage. The minimum ratio is 1:1. This ratio has performance significance for the following reasons:

- It may be inappropriate for different applications with different ratios to share the same coupling facility structure.
- If AVGBUFSIZE is not well tuned, you can incur more DASD offloads than you expect.
- If AVGBUFSIZE is not well tuned, you can waste list entry storage.

Element to Entry Ratio and Offloading: AVGBUFSIZE is set at the structure level and dictates the ratio for the whole structure. As a general rule, the greater the number of log streams per structure, the greater the chance that the element/entry ratio is inappropriate for certain applications using the log streams. If many applications write significantly different amounts of data to their log streams at significantly different intervals, some of those applications can experience unexpected DASD offloading, with the extra processing overhead that this incurs. The DASD offloading is unexpected because the log stream may not yet have reached its HIGHOFFLOAD threshold.

Each log record places an entry in the list entry area of the structure and the data is loaded as one or more elements in the list element area. If the list entry area exceeds 90% of its capacity, all log streams are offloaded to DASD. DASD offloading commences at this point, regardless the current utilization of the log stream, and continues until an amount of data equal to the difference between the HIGHOFFLOAD threshold and LOWOFFLOAD threshold has been offloaded.

For example, the list entry area may exceed 90% of its capacity while log stream A is only 50% utilized. Its HIGHOFFLOAD threshold is 80% and the LOWOFFLOAD threshold is 60%. Even though log stream A has not reached the HIGHOFFLOAD threshold, or even its LOWOFFLOAD threshold, data is offloaded until 20% of the log stream has been offloaded. This is the difference between 80% and 60%. After the offloading operation has completed, log stream A is at 30% utilization (50% minus 20%).

Thus, the log stream used by an application issuing very few journal requests may be offloaded to DASD because of frequent journal write requests by other applications using other log streams in the same structure.

If multiple log streams share the same structure, however, a situation where list entry storage reaches 90% utilization should occur only where all the log streams have a similar amount of logging activity. This situation is more likely to happen where AVGBUFSIZE is grossly overestimated.

OS/390 Release 3 introduces dynamic management of the entry-to-element ratio so that the setting of AVGBUFSIZE is less critical to the management of structure storage.

Figure 20 summarizes the various parameters that control offloading.

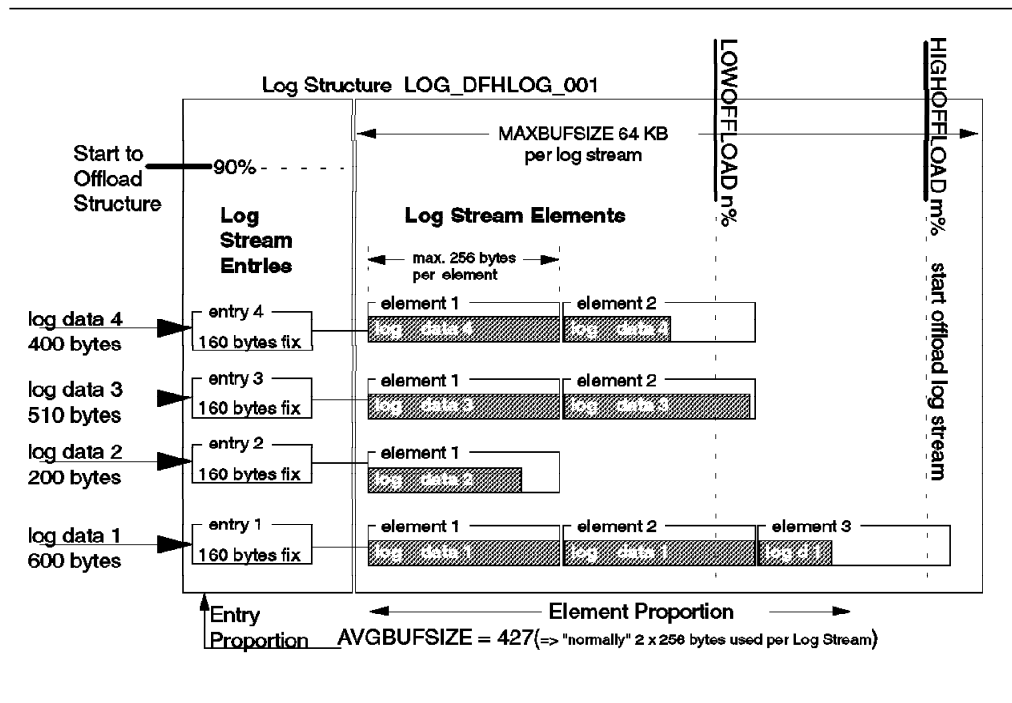


Figure 20. Log Stream Parameters Controlling Offloading

5.2.2.6 Frequency of DASD Offloading

System log DASD offloading can be caused by the following conditions:

- The coupling facility space allocated for a log stream reached the threshold defined by the HIGHOFFLOAD parameter, and an insufficient amount of space has been made available by activity keypoint processing.
- The staging data-set utilization reached the threshold defined by the HIGHOFFLOAD parameter (see 5.2.2.7, "Defining DASD Staging Data Sets" on page 67 for more information on staging data sets).
- The 90% limit of the coupling facility structure list entry is reached. In this case, all the log streams allocated to the structure are offloaded.

To monitor DASD offloading, review the fields "BYT DELETD FRM CF W/O DASD WRITE" and "BYT DELETD FRM CF AFTR DASD WRITE" in the SMF type 88

records for the coupling facility. These numbers should be high and low, respectively.

To minimize the frequency of DASD offloading, we recommend that you:

1. Check that the correct log streams are connected to the structure.
2. Check for long-running tasks (look for activity keypoint messages that are not associated with log-tail deletion messages).
3. Consider increasing the structure size.
4. Consider reducing AKPFREQ (this will only work if there are no long-running tasks).
5. If offloading is being triggered by the entry-full conditions (check the “NTRY-FULL” field of the SMF type 88 record) then AVGBUFSIZE could be set too high.
6. If offloading is being triggered by the staging data set threshold, then increase the size of the staging data sets.

Note: NTRY-FULL and STG-THLD are not reported in the IXGRPT1 sample program.

5.2.2.7 Defining DASD Staging Data Sets

MVS normally keeps a second copy of data written to 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 independent from MVS (in a separate central computer complex (CPC) and not volatile).

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

Every time an MVS system logger application writes a log block to a log stream coupling facility list structure, the MVS system logger automatically creates a duplicate copy of the data as insurance against data loss due to coupling facility or system failure. This duplicate copy is kept until the data is offloaded from the coupling facility structure to DASD log data sets. After the data is offloaded to DASD log data sets, the MVS system logger discards the duplicate copy of the log data. The MVS system logger uses VSAM linear data sets for its staging data sets, and if required allocates a staging data set for each log stream.

We recommend that the staging data sets reside on a device to which all systems in the parallel sysplex can connect through channel paths. This is important because other systems may need to access the backed-up log data in case of system or coupling facility failure.

The DASD staging data sets should be sized to contain all the data stored at the same time into the coupling facility space allocated to the log stream. The blocking of data into the staging data set (4096-byte blocks instead of 512-byte or 256-byte elements) is such that you have to allocate a staging data set size several times as large as the coupling facility structure size.

Use the following formula to help you tune the size of your staging data sets:

staging data set size = (NR × AVGBUFSIZE rounded up to next multiple of 4096)

where NR is the number of records to fill the log stream structure.

NR can be calculated as follows:

$$NR = \frac{\text{log stream structure size}}{(\text{AVGBUFSIZE rounded up to next element})}$$

Ensure that the coupling facility structure and staging data set can hold the same number of records.

The HIGHOFFLOAD threshold you define for the coupling facility list structure allocated to a log stream also applies to the staging data sets in use by that log stream. This means that when a staging data set reaches the high threshold, the MVS system logger immediately offloads data from the coupling facility to log data sets, even if the coupling facility usage for the log stream is below the HIGHOFFLOAD threshold.

The DFHLSCU utility can help you to estimate the size of staging data sets. Chapter 7, “Coupling Facility Preparation” on page 101, describes how to use this utility.

You control the choice between staging data sets and data spaces as follows:

- For system log and forward-recovery log streams, use STG_DUPLEX(YES) and DUPLEXMODE(COND). This ensures that the MVS system logger automatically uses staging data sets if it detects that the coupling facility is not failure-independent.
- Use STG_DUPLEX(YES) and DUPLEXMODE(UNCOND) if you only have only one coupling facility.

Figure 1 on page 2 shows the staging dataset flow.

5.2.2.8 Special Considerations for CICS System Logs

Ideally, all records for the primary system log should be kept in the coupling facility list structure and never offloaded to log data sets on DASD.

The new CICS log manager modifies the meaning of the SIT parameter AKPFREQ. AKPFREQ now defines the number of records written to the log stream of the system after which the CICS takes an activity keypoint. When CICS takes an activity keypoint, it writes the current status of all active tasks to the system log. During emergency restart, CICS uses back-out records from the system log to back out changes made by the transactions that were in flight at the time of the failure.

CICS TS introduces a new SIT parameter, SYSLOG, to control whether CICS deletes system log records at activity keypoint time. Specify KEEP if you want to retain the system log because you need an audit trail, or because you have user journal records on system log. The suggested and default value is NOKEEP. If you specify SYSLOG=KEEP, you must periodically delete data from the log stream before the limit of 168 data sets is reached.

System log records for all units of work start on the primary log stream. If you specify SYSLOG=KEEP, records are offloaded to log data sets as the coupling facility list structure fills. If you specify SYSLOG=NOKEEP, CICS uses a different mechanism to control the primary system log. When CICS takes an activity

keypoint, eligible records are removed from the primary log stream either by log tail deletion or by being moved to the secondary log stream.

A log tail is the oldest end of the log. At each activity keypoint, the CICS recovery manager asks the CICS log manager to delete the tail of the system log by establishing a point on the system log before which all older data blocks can be deleted. Thus, if the oldest live unit of work is in data block *x*, the CICS log manager asks the MVS system logger to delete all data older than *x* (*x*-1 and older).

Eligible long-running units of work, including shunted units of work, are moved to the secondary log during activity keypoint. A long-running unit of work that is eligible for shunting is one that exists for two complete activity keypoints without initiating any writes to the system log. After they have been moved, units of work remain on the secondary log until the unit of work is complete.

The secondary system log stream is purely a storage optimization mechanism, which prevents offloading of the primary system log.

For CICS releases earlier than CICS TS, coding a low AKPFREQ value causes many activity keypoints to be written to the system log, increasing the keypoint overhead. Coding a high value increases the emergency restart time because CICS needs more time to look for the activity keypoint. The optimized value is to write an activity keypoint every 15-20 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 keeps system log records inside the coupling facility for the duration of two activity keypoints. AKPFREQ becomes very important in defining the size of system log structure in the coupling facility. We recommend that you set AKPFREQ such that there is an activity keypoint every 30-60 seconds.

CICS writes a DFHJC5801 message to the CSMT transient data destination each time an activity keypoint is taken. Use CICS transaction statistics to see the frequency of the execution of the CSKP transaction.

Increasing AKPFREQ extends restart and XRF takeover times, and increases the amount of coupling facility space required for the primary system log. Decreasing AKPFREQ can reduce restart times. This is particularly important in high-availability systems such as those using XRF. Decreasing AKPFREQ tends to increase task wait time and processor cycles.

If data is offloaded from the primary system log, either by choice (because you have specified SYSLOG=KEEP or because coupling facility space is severely limited) or by chance (because of fluctuating workload), then data should be offloaded in large units to prevent an offload event occurring too frequently. Primary system log offloads should not occur more frequently than once every 5 seconds. The difference between the HIGHOFFLOAD and LOWOFFLOAD values determines how much data is offloaded at a time.

You can calculate a value for LOWOFFLOAD using the formula:

$$LOWOFFLOAD = \frac{trandur \times 90}{akpintvl + trandur} + 10$$

In this formula, `trandur` is the execution time between synch points of the longest-running transaction that runs as part of the normal workload, and `akpintl` is the average interval between activity keypoints. An estimate for `akpintl` is provided by `DFHLSCU`. For example, if `trandur` is 5 seconds and `akpintl` is 45 seconds then the formula gives a `LOWOFFLOAD` estimate of

$$\frac{(5 \times 90)}{(45 + 5)} + 10 \text{ or } 19.$$

If you do not want to use the formula, a `LOWOFFLOAD` value of 60 is a conservative estimate which should ensure that data is offloaded in reasonably sized amounts, but that data for in-flight UOWs is never offloaded to DASD.

If system log offloading is unavoidable, it is important to define a log data set size which avoids frequent allocation of new data sets. The recommendation is to initially make the logger data sets the same size as your CICS/ESA 4.1 system log data sets, and then monitor the SMF type 88 records for new data set allocations. You should aim for a frequency of less than one new data set allocated per hour.

5.2.3 Defining the CICS Log Manager Components

The CICS log manager domain replaces the journal control function of earlier CICS releases.

5.2.3.1 Defining JOURNALMODEL Resources

The `JOURNALMODEL` resource-type definition in the CSD allows autoinstall of CICS journals.

The CICS system log requires two `JOURNALMODEL` definitions, one with a `JOURNALNAME` parameter of `DFHLOG` and one with a `JOURNALNAME` parameter of `DFHSHUNT`. For both `JOURNALMODEL` definitions, the `TYPE` parameter must be `MVS` when you are defining the CICS system log for a production environment.

The `STREAMNAME` parameter associates the CICS definition with the log streams defined in the `LOGR` policies. The sample definitions shown in Figure 21 on page 71 associate the CICS system log with the log streams explicitly defined in the `LOGR` policy in Figure 19 on page 62.

```

//DFHCSDUP JOB (999,POK), 'CICSTS', CLASS=A,MSGCLASS=T,
//          NOTIFY=&SYSUID
//UPGRCSO EXEC PGM=DFHCSDUP,REGION=1M
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//DFHCSD  DD DSN=CICS.V5R1M0.DFHCSD,DISP=SHR
//CBDOU   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
DEFINE JOURNALMODEL(SYSLOG)
        GROUP(TEST)
        DESCRIPTION('DFHLOG LOGSTREAM EXPLICIT DEFINITION')
        JOURNALNAME(DFHLOG)
        TYPE(MVS)
        STREAMNAME(SCSCPAA2.TST.DFHLOG)
DEFINE JOURNALMODEL(SYSSHUNT)
        GROUP(TEST)
        DESCRIPTION('DFHSHUNT LOGSTREAM EXPLICIT DEFINITION')
        JOURNALNAME(DFHSHUNT)
        TYPE(MVS)
        STREAMNAME(SCSCPAA2.TST.DFHSHUNT)
//

```

Figure 21. Sample DFHCSDUP Job to Create Explicit CICS System Log Definitions

The job shown in Figure 21 creates explicit definitions for the primary and secondary system log for the test CICS region whose APPLID is SCSCPAA2. During CICS log manager initialization for this region, CICS issues an IXGCONN request to connect to log streams SCSCPAA2.TST.DFHLOG and SCSCPAA2.TST.DFHSHUNT.

This kind of definition requires an explicit JOURNALMODEL definition and an explicitly defined log stream in the coupling facility for each regions, but allows you to map the log streams to different coupling facility structures.

The STREAMNAME parameter can contain a symbolic qualifier, such as &APPLID (which is replaced by the CICS region application identifier). If you do not define any JOURNALMODELS, the CICS region tries to connect to a log stream named ®IONUSERID..&APPLID..DFHLOG. For a CICS-started task with an applid of SCSCPAA2, this resolves to STC.SCSCPAA2.DFHLOG. If the MVS system logger does not find such a log stream, it returns an error message. The CICS log manager then uses the default model (&SYSNAME..DFHLOG.MODEL) to dynamically create a log stream.

Figure 22 on page 72 shows how CICS maps the primary system log (DFHLOG) to a log stream name (LSN) during an INITIAL start. CICS uses the same process for the secondary system log (DFHSHUNT).

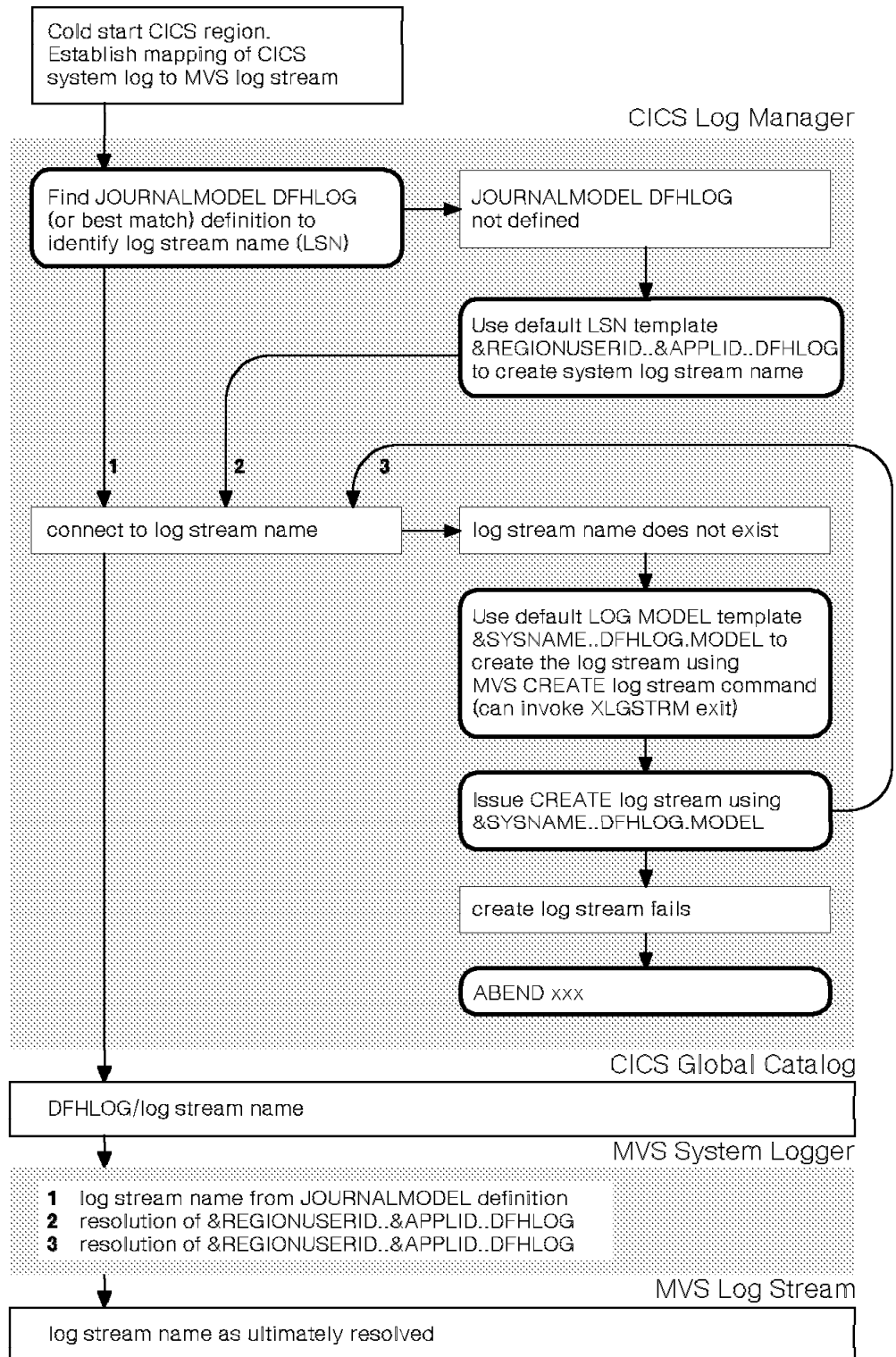


Figure 22. Mapping Logs to Log Streams

Figure 23 on page 73 shows the definition we used for the log stream initial allocation, using the model log stream.

```

//DFHCSDUP JOB (999,POK),'CICS510',CLASS=A,MSGCLASS=T,
//          NOTIFY=&SYSUID
//UPGRCSO EXEC PGM=DFHCSDUP,REGION=1M
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//DFHCSD  DD DSN=CICS.V5R1M0.DFHCSD,DISP=SHR
//CBDOUT  DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
DEFINE JOURNALMODEL(SYSLOG)
  GROUP(TEST)
  DESCRIPTION(Primary CICS system log)
  JOURNALNAME(DFHLOG) TYPE(MVS)
  TYPE(MVS)
  STREAMNAME(&APPLID..DFHLOG)
DEFINE JOURNALMODEL(SYSSHUNT)
  GROUP(TEST)
  DESCRIPTION(Secondary CICS system log, shunted tasks)
  JOURNALNAME(DFSSHUNT)
  TYPE(MVS)
  STREAMNAME(&APPLID..DFSSHUNT)

```

Figure 23. Sample Log Model Definitions for a CICS System Log

Using the JOURNALMODEL of the sample job, on the INITIAL start of the CICS region with an application identifier of SCSCPAA4, the CICS log manager tries to connect to a log stream named SCSCPAA4.DFHLOG. As this log stream does not exist, the MVS system logger returns an error message to CICS. The CICS log manager issues a CREATE log stream with the same name as before, and refers to a log model template named &SYSNAME.DFHLOG.MODEL. The MVS system logger creates the log stream, using the parameters of the log model template required and connects the CICS region to the coupling facility log stream.

Figure 24 shows the messages issued by CICS during the system log initialization for an INITIAL start.

```

+DFHLG0103I SCSCPAA4 System log (DFHLOG) initialization has started.
IXG231I IXGCONN REQUEST=CONNECT TO LOG STREAM SCSCPAA4.DFHLOG 652
COMPLETED FOR REQUESTOR SCSCPAA4. RETURN CODE: 00000008 REASON CODE:
0000080B DIAG1: 00000008 DIAG2: 0000F801 DIAG3: 05030004 DIAG4:
05020010
+DFHLG0104I SCSCPAA4 System log (DFHLOG) initialization has ended.
+DFHLG0103I SCSCPAA4 System log (DFSSHUNT) initialization has started.
IXG231I IXGCONN REQUEST=CONNECT TO LOG STREAM SCSCPAA4.DFSSHUNT 655
COMPLETED FOR REQUESTOR SCSCPAA4. RETURN CODE: 00000008 REASON CODE:
0000080B DIAG1: 00000008 DIAG2: 0000F801 DIAG3: 05030004 DIAG4:
05020010
+DFHLG0104I SCSCPAA4 System log (DFSSHUNT) initialization has ended.

```

Figure 24. Messages Issued during CICS INITIAL Start

This kind of model definition makes the system programmer's task easier, but associates all the system log streams of CICS regions running in a single MVS image with the same coupling facility structure.

5.3 Setting Up General Logs

The general logs are the user journal, autojournal, and forward-recovery logs. In this section we discuss considerations for setting up these logs.

5.3.1 Defining User Journals and Autojournals

User journals and autojournals can be defined as being DUMMY, MVS, or SMF. For a journal type of SMF, the journal records are written in SMF format to the MVS SMF log instead of to an MVS system logger log stream.

When type MVS is selected, the journal records are written to an MVS system logger log stream. The user journals and the autojournals map to explicitly defined log streams or to log streams that are automatically defined on first use of the journal, if a log stream model exists. By default, user journals and autojournals map to qualified log stream names of the form *region_userid.applid.journalname*, where *region_userid* and *applid* are taken from the CICS job. The *journalname* is the eight-character name of the user journal for user-defined names, or DFHJnn where nn is the journal number.

5.3.2 Defining the Forward Recovery Logs

Forward recovery logs are managed only by CICS file control. CICS obtains the log stream name of a VSAM forward-recovery log either from the ICF catalog or from the file definition within CICS. For files opened in RLS mode, the explicit log stream name is always obtained directly from the ICF catalog entry for the data set. For files opened in non-RLS mode, the log stream name is derived from the ICF catalog entry for the data set, or (if not defined in the ICF catalog) a journal model definition referenced by a forward-recovery journal name specified in the file resource definition. These names are of the form DFHJnn where nn is a number in the range 1-99 and is obtained from the forward-recovery log ID (FWDRECOVLOG) in the FILE resource definition.

5.3.3 Creating General Logs

If a general log referenced by CICS is not known to MVS, CICS requests the MVS system logger to create the log stream dynamically, using the attributes from a log stream model definition. By default, CICS constructs a qualified name for the MVS model definition by appending the name MODEL to all, or part, of the log stream name being created. CICS determines which parts of the log stream name to use for the model name as follows:

- If the log stream being created has a qualified name consisting of only two names (qualifier1.qualifier2), or has an unqualified name, CICS constructs the model name as qualifier1.MODEL, or name.MODEL
- If the log stream being created has a qualified name consisting of three or more names (qualifier1.qualifier2....qualifier_n) CICS constructs the model name as qualifier1.qualifier2.MODEL.

5.3.4 Defining the Log of Logs

You should define a log-of-logs log stream that is shared by all CICS regions in the parallel sysplex. This must be explicitly defined, because CICS does not support model log streams for dynamic creation of this log stream. You should also define the journal model resource definition that references the log-of-logs log stream. The CICS-supplied group, DFHLGMOD specifies a journal model for

the log of logs, called DFHLGLOG, and has a log stream name of region_userid.CICSVR.DFHLGLOG.

The log of logs contains records that are written each time a file is opened or closed. At file-open time, a tie-up record is written that identifies:

- The name of the file
- The name of the underlying VSAM data set
- The name of the forward-recovery log stream
- The name of the CICS region that performed the file open.

At file-close time, a tie-up record is written that identifies the name of the file and the name of the CICS region that performed the file close.

The log of logs assists forward-recovery utilities to maintain an index of log data sets.

5.4 Management of CICS Logs and Journals

After implementing CICS logging and journaling you should develop procedures and utilities to manage them.

5.4.1 Availability of the MVS System Logger

CICS itself checks for the availability of the MVS system logger. Every 10 seconds (or the interval specified in the ICV system initialization if this is greater than 10 seconds), CICS initiates a BROWSE READ operation on the system log. If this fails, CICS either abends or quiesces, depending on the returned MVS system logger reason code.

5.4.2 CICS Logging Operations

The CICS-supplied transaction CEMT has been modified to support the new CICS logging. See *CICS Supplied Transactions* for a detailed description of the command parameters.

The INQUIRE JMODEL command enables you to inquire about installed journal models to obtain corresponding log stream names:

```
CEMT I JM(*)
STATUS: RESULTS
Jmo(C51LGLOG) Jou(DFHLGLOG) Dum
Jmo(C51LOG ) Jou(DFHLOG ) Mvs
      Str(&APPLID..DFHLOG      )
Jmo(C51SHUNT) Jou(DFHSHUNT) Mvs
      Str(&APPLID..DFHSHUNT    )
```

The INQUIRE JOURNALNAME command enables you to inquire as to the status of the CICS system log and general logs:

```
CEMT I JO(*)
STATUS: RESULTS - OVERTYPE TO MODIFY
Jou(DFHLOG ) Mvs Ena      Str(SCSCPAA1.DFHLOG      )
Jou(DFHSHUNT) Mvs Ena      Str(SCSCPAA1.DFHSHUNT    )
```

The INQUIRE STREAMNAME command enables you to inquire as to the currently connected MVS log streams:

```

CEMT I STR(*)
STATUS: RESULTS
Str(SCSCPAA1.DFHLOG           ) Use(0000001) Ok Sys
Str(SCSCPAA1.DFHSHUNT        ) Use(0000001) Ok Sys

```

5.4.3 Reading Log Stream Offline

The LOGR subsystem allows you to read log stream output without having to rewrite your existing applications. For example, you may have existing applications that read log data in data set format and do not support the log stream format. The LOGR subsystem lets you access log stream data in data set format.

If your installation chooses to use the LOGR subsystem to read log stream output, you must activate it prior to using the log stream.

You can activate the LOGR subsystem in two ways. Either add a new SUBSYS statement such as

```
SUBSYS SUBNAME(LOGR) INITRTN(IXGSSINT)
```

in the IEFSSNxx parmlib member, or use the SETSSI command to add the subsystem dynamically:

```
SETSSI ADD,SUBNAME=LOGR,INITRTN=IXGSSINT
```

To use the subsystem interface to gain access to the log stream data, specify LOGR on the SUBSYS parameter on the log stream DD statement. Specifying the LOGR subsystem name on the SUBSYS parameter enables LOGR to intercept data set open and read requests at the subsystem interface (SSI) and convert them into log stream accesses.

Depending on the options specified on the SUBSYS parameter, general log stream records are presented either in a record format compatible with utility programs written for releases of CICS earlier than CICS TS, or in a format compatible with utility programs written for CICS TS or later.

The CICS log stream subsystem interface exit routine (DFHLG510) is provided for use with any journal utility program, including DFHJUP, that processes CICS journal data written as a general log stream (the exit does not handle CICS system-log log stream records).

The sample job shown in Figure 25 on page 77 prints the content of a journal log stream identified by the log stream name.

Note: DCB parameter BLKSIZE=32760 must be specified on the input data DD statement.

```

//LSTPRINT JOB (999,POK),' LOGR POLICY', CLASS=A,REGION=4096K,
//          MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//PRNTJNL EXEC PGM=DFHJUP
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*,DCB=RECFM=FBA
//SYSUT1 DD DSN=SCSCPA1.DFHJ01,
//          DCB=BLKSIZE=32760,
//          SUBSYS=(LOGR,DFHLG510,
//          'FROM=(1996/001,06:00),TO=(1996/200,23:59),LOCAL')
//SYSIN DD *
*-----*
* CONTROL STATEMENT : DEFAULTS *
* INPUT = SYSUT1 *
* OUTPUT = SYSPRINT *
* SELECTION QUALIFIERS : *
* 1. DEFAULT = ALL INPUT RECORDS *
*-----*
OPTION PRINT
END
/*

```

Figure 25. Sample Job to Print Journal Log Stream

The sample job shown in Figure 26 copies the content of a log stream identified by the DASD log data set name.

```

//LISTLGST JOB (999,POK),' CICS510', CLASS=A,MSGCLASS=T,
//          NOTIFY=&SYSUID
//PRNTJNL EXEC PGM=DFHJUP
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*,DCB=RECFM=FBA
//SYSUT1 DD DSN=SCSCPA2.DFHJ01.DA000001,
//          DCB=BLKSIZE=32760,
//          SUBSYS=(LOGR,DFHLG510,
//          'FROM=(1995/001,06:00),TO=(1995/007,23:59),LOCAL')
//SYSUT4 DD DSN=CICS.LISTLOG,DISP=(,CATLG),
//          UNIT=SYSDA,VOL=SER=TOTTS4,
//          SPACE=(CYL,(1,1))
//SYSIN DD *
*-----*
* CONTROL STATEMENT : DEFAULTS *
* INPUT = SYSUT1 *
* OUTPUT = SYSUT4 *
* SELECTION QUALIFIERS : *
* 1. DEFAULT = ALL INPUT RECORDS *
*-----*
OPTION COPY
END
/*

```

Figure 26. Sample Job to Copy Journal Log Stream

The sample job shown in Figure 27 on page 78 deletes the content of a log stream identified by the DASD log data set name.

```

//LSTDEL    JOB (999,POK), 'LOGR POLICY', CLASS=A, REGION=4096K,
//          MSGCLASS=X, TIME=10, MSGLEVEL=(1,1), NOTIFY=&SYSUID
//PRNTJNL   EXEC PGM=IEFBR14
//LOGSTR    DD DSN=SCSCPAA2.DFHJ01,
//          SUBSYS=(LOGR,DFHLG510,'TO=(1996/229,09:30)',DELETE)
/*

```

Figure 27. Sample Job to Delete Journal Log Stream

5.4.4 Management of DASD Log Data Sets

To optimize the system log functions, the log data should be held in coupling facility storage without offloading to DASD data sets. By contrast, data for general logs should instead be held in coupling facility storage for a comparatively short time and then offloaded to the DASD data sets.

If you log a lot of data, you may need to manage log data set storage consumption to ensure that you do not hit the directory entry limit of 168. Whenever the number of log data sets in a log stream's data set directory goes above 90% of the 168 allowed, the MVS system logger issues an action message, IXG257I. This message will be deleted when either the number of data sets for the log stream drops below 85% of the total allowed or the last connection to the log stream in the parallel sysplex disconnects. In the case of disconnects, the message will be reissued if an application reconnects to the log stream.

The IXG257I message cannot be detected by CICS, you should use your automation software, for example AOC to monitor the occurrences of this message.

MVS system logger will not delete a log data set until all data is marked for deletion or until the log stream associated with the data set is deleted from the LOGR policy. To manage DASD log data sets so that you do not hit the 168 directory extent limit, use one or more of the following methods to delete log data set from the log stream:

- Use the IXGDELET service to delete data from the log stream as soon as the data is no longer needed. This will help to manage log data set storage consumption by making sure log data sets are deleted as soon as possible.
- Use CICS/VR to archive or delete CICS log data sets for CICS log manager.
- Periodically, use an archiving procedure to remove data from the log stream. If you want to keep your data longer, that can be supported as long as it stays within the directory limit.

5.4.5 Management of Staging Data Sets

When a system's staging data set fills up, logging applications on that system will not be able to write to the log stream until log data can be offloaded from the coupling facility structure to DASD, which frees up space in the staging data set. Thus, when your staging data sets are too small, MVS system logger will perform coupling facility offloading more frequently than the HIGHOFFLOAD and LOWOFFLOAD thresholds defined for the log stream would otherwise require. This can degrade the performance of all log streams in that structure.

Use SMF record 88 to monitor staging data sets usage. Check field SMF88ETT to see whether the high threshold mark for a staging data set is being reached. If the high threshold mark is never reached, your staging data set may be larger than necessary. Check field SMF88ETF to see if you are getting one or more staging-data-set-full conditions. This indicates that your staging data set may be too small and should probably be enlarged.

5.5 Monitoring Log Streams

Chapter 6, "Tuning a Parallel Sysplex" in *MVS/ESA Setting Up a Sysplex*, describes the Resource Management Facility (RMF) reports which contain useful measurement data for analyzing the performance of the coupling facility. These reports are the *XCF Activity Report* and the *Coupling Facility Activity Report*.

CICS collect statistics on the data written to each journal and log stream, which can be used to analyze the activity of a single region.

5.5.1 Online Statistics

Figure 28 is an example of the report produced by the program DFH0STAT through the transaction STAT.

```

Applid SCSCPAA2  Sysid PAA2  Jobname SCSCPAA2  Date 06/20/1996  Time 16 :43:02  CICS for MVS/ESA Version 5 Release 1.0
32
-----
Journalnames
-----
Journal  Journal  Journal  Write  Bytes  Average  Buffer
Name     Status   Type     Logstream Name     Requests  Written  Bytes  Flashes
-----
DFHLOG   Enabled  MVS     SCSCPAA2.DFHLOG     0         0        0      0
DFHSHUNT Enabled  MVS     SCSCPAA2.DFHSHUNT  0         0        0      0
Applid SCSCPAA2  Sysid PAA2  Jobname SCSCPAA2  Date 06/20/1996  Time 16 :43:02  CICS for MVS/ESA Version 5 Release 1.0
33
-----
Logstreams
-----
Logstream Name  Use  Write  Bytes  Peak  Tot Force  Buff Full  Browse  Browse
                  Count Requests  Written  FWaiters  Waits  Waits  Starts  Reads
-----
SCSCPAA2.DFHLOG  1    11    5,680    1    11    0    0    0
SCSCPAA2.DFHSHUNT 1     0     0        0     0    0    0    0

```

Figure 28. Online Statistics

You can use the statistics utility program (DFHSTUP) to prepare and print the CICS statistics data. The data is recorded on the MVS system management facilities (SMF) SYS.MANx data sets. Figure 29 on page 80 shows a sample job that will print the statistics for several CICS applications. The data is presented in the same format as the statistics produced by the DFH0STAT program, but covers all the CICS regions selected.

```

//PRTSTACU JOB (999,P0K),'CICS.V5R1M0',CLASS=A,MSGCLASS=T,
//          NOTIFY=&SYSUID
//*****
//* STEP 1 - UNLOAD THE SMF DATA SET CONTAINING CICS DATA *
//*****
//SMFDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYS1.SC52.MAN1,DISP=SHR,AMP=('BUFSP=65536')
//      DD DSN=SYS1.SC52.MAN2,DISP=SHR,AMP=('BUFSP=65536')
//      DD DSN=SYS1.SC52.MAN3,DISP=SHR,AMP=('BUFSP=65536')
//*     DD DSN=SYS1.SC47.MAN1,DISP=SHR,AMP=('BUFSP=65536')
//*     DD DSN=SYS1.SC47.MAN2,DISP=SHR,AMP=('BUFSP=65536')
//*     DD DSN=SYS1.SC47.MAN3,DISP=SHR,AMP=('BUFSP=65536')
//OUTDD1 DD DSN=&&TEMP,DISP=(NEW,PASS),
//        SPACE=(CYL,(10,1)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
          INDD(INDD1,OPTIONS(DUMP))
          OUTDD(OUTDD1,TYPE(110))

/*
//*****
//* STEP 2 - DFHJUP SELECT RECORDS *
//*****
//RECSSEL EXEC PGM=DFHJUP
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//SYSUT1 DD DSN=&&TEMP,DISP=(OLD,PASS)
//SYSUT4 DD DSN=&&TEMP2,DISP=(NEW,PASS),
//        SPACE=(CYL,(5,2)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=6E,FLDLLEN=1,COND=M
OPTION COPY OFFSET=23,FLDTYP=X,VALUE=0002,FLDLLEN=2,COND=M
OPTION COPY OFFSET=47,FLDTYP=C,VALUE=SCSCPA1,FLDLLEN=8,COND=E
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=6E,FLDLLEN=1,COND=M
OPTION COPY OFFSET=23,FLDTYP=X,VALUE=0002,FLDLLEN=2,COND=M
OPTION COPY OFFSET=47,FLDTYP=C,VALUE=SCSCPA2,FLDLLEN=8,COND=E
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=6E,FLDLLEN=1,COND=M
OPTION COPY OFFSET=23,FLDTYP=X,VALUE=0002,FLDLLEN=2,COND=M
OPTION COPY OFFSET=47,FLDTYP=C,VALUE=SCSCPA3,FLDLLEN=8,COND=E
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=6E,FLDLLEN=1,COND=M
OPTION COPY OFFSET=23,FLDTYP=X,VALUE=0002,FLDLLEN=2,COND=M
OPTION COPY OFFSET=47,FLDTYP=C,VALUE=SCSCPA4,FLDLLEN=8,COND=E
END
/*
//*****
//* STEP 3 - RUN DFHMOLS TO PRINT THE CICS MONITORING DATA, *
//* USING THE NEW DICTIONARY RECORD FROM STEP 1 *
//*****
//PRNTMND EXEC PGM=DFHSTUP
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//      DD DSN=CICS.V5R1M0.SDFHAUTH,DISP=SHR
//DFHSTATS DD DSN=&&TEMP2,DISP=(OLD,DELETE),UNIT=SYSDA
//DFHSTWRK DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK01 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK02 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK03 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK04 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK05 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SYSOUT DD SYSOUT=*
//DFHPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
SELECT APPLID=(SCSCPA1,SCSCPA2,SCSCPA3,SCSCPA4)
COLLECTION TYPE=ALL
/*

```

Figure 29. Sample Job to Print CICS Log Stream Statistics

5.5.2 MVS Log Streams Statistics

Because log streams can be shared across multiple MVS images, you may find it useful to examine the statistics generated by MVS.

The sample job shown in Figure 30 on page 82 extracts all system management facility records of type 88 from the SMF system data sets. These records are written by the MVS system logger and contain the statistics for each connected log streams. The sample program IXGRPT1, supplied in SYS1.SAMPLIB, processes the SMF data and produces a report. Step 2 has been added to the sample supplied by the SAMPLIB, to suppress the SMF records Type 2 and 3 added by the IFASMFDP utility.

```

//PRTSTALS JOB (999,POK),'CICS510',CLASS=A,MSGCLASS=T,
//          NOTIFY=&SYSUID
//*****
//* STEP 1 - UNLOAD THE SMF DATA SET CONTAINING CICS DATA      *
//*****
//SMFDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYS1.SC52.MAN1,DISP=SHR,AMP=(' BUFSP=65536')
//INDD2 DD DSN=SYS1.SC52.MAN2,DISP=SHR,AMP=(' BUFSP=65536')
//INDD3 DD DSN=SYS1.SC52.MAN3,DISP=SHR,AMP=(' BUFSP=65536')
//INDD4 DD DSN=SYS1.SC47.MAN1,DISP=SHR,AMP=(' BUFSP=65536')
//INDD5 DD DSN=SYS1.SC47.MAN2,DISP=SHR,AMP=(' BUFSP=65536')
//INDD6 DD DSN=SYS1.SC47.MAN3,DISP=SHR,AMP=(' BUFSP=65536')
//OUTDD1 DD DSN=&&TEMP,DISP=(NEW,PASS),
//        SPACE=(CYL,(12,5)),UNIT=SYSDA
//SYSPRINT DD DUMMY
//SYSIN DD *
        INDD(INDD1,OPTIONS(DUMP))
        INDD(INDD2,OPTIONS(DUMP))
        INDD(INDD3,OPTIONS(DUMP))
        INDD(INDD4,OPTIONS(DUMP))
        INDD(INDD5,OPTIONS(DUMP))
        INDD(INDD6,OPTIONS(DUMP))
        OUTDD(OUTDD1,TYPE(88))

/*
//*****
//* STEP 1A- DISPLAYS SMF RECORD TYPE                            *
//*****
//*TOOL1 EXEC PGM=ICETOOL,REGION=1024K
//*TOOLMSG DD SYSOUT=*
//*DFSMSG DD SYSOUT=*
//*TOOLIN DD *
//* DISPLAY FROM(INDD) ON(6,1,HEX) LIST(LISTDD)
//*
//*INDD DD DSN=&&TEMP,DISP=(OLD,PASS)
//*LISTDD DD SYSOUT=*
//*
//*****
//* STEP 2 - Copies SMF records type 88 only                    *
//*****
//COPYSEL EXEC PGM=SORT,REGION=1024K
//SYSOUT DD DUMMY
//SYSABEND DD SYSOUT=*
//SORTIN DD DSN=&&TEMP,DISP=(OLD,DELETE)
//SORTOUT DD DSN=&&TEMP1,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(12,5))
//SYSIN DD *
        OPTION COPY
        INCLUDE COND=(6,1,BI,EQ,X'58')
/*

```

Figure 30 (Part 1 of 2). Sample Job to Print MVS Log Stream Statistics


```

//*****
/* STEP 3 SORT EACH SMF INPUT BY TIMESTAMP AND LOGSTREAM NAME */
//*****
//SORT1 EXEC PGM=SORT */
//SYSOUT DD DUMMY */
//SORTIN DD DSN=&&TEMP1,DISP=(OLD,DELETE) */
//SORTOUT DD DSN=&&TEMP2,DISP=(NEW,PASS),UNIT=SYSDA, */
// SPACE=(CYL,(5,1)) */
//SYSIN DD *
OPTION VLSHRT
SORT FIELDS=(125,8,BI,A,
97,26,CH,A)

/* */
/* */
/* ***** */
/* STEP 4 */
/* COMPILE ASM SUBRTN IXGR1A. PGM IEV90 IS USED INSTEAD OF */
/* CATALOGED PROC HLASMC, BECAUSE CATALOGED PROC PLIXCG REQUIRES */
/* THAT THE ASM OBJ (ASMC.SYSLIN) BE A PDS. */
/* ***** */
//ASMC EXEC PGM=IEV90,REGION=180K,PARM=' OBJECT,NODECK' */
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR * LIBRARY DSN */
//SYSLIN DD DSN=&&OBJ(IXGR1A),SPACE=(TRK,(10,30,10),,ROUND), */
// UNIT=3390,DISP=(MOD,PASS), */
// DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB) * ASM OBJ */
//SYSPRINT DD DUMMY */
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(5,5)) */
//SYSIN DD DSN=SYS1.SAMPLIB(IXGR1A),DISP=SHR */
/* */
/* ***** */
/* COMPILE, LINK, AND EXECUTE PL/I PROGRAM IXGRPT1. */
/* ***** */
//PLIC EXEC PLIXCLG */
//PLI.SYSIN DD DSN=SYS1.SAMPLIB(IXGRPT1),DISP=SHR
//LKED.SYSLIB DD DSN=PLI.V2R3MO.PLIBASE,DISP=SHR
// DD DSN=PLI.V2R3MO.SIBMBASE,DISP=SHR
// DD DSN=*.ASMC.SYSLIN,DISP=(MOD,PASS) * ASM OBJ (IN) */
//LKED.SYSPRINT DD DUMMY
//PLI.SYSPRINT DD DUMMY
//GO.SMFDATA DD DSN=*.SORT1.SORTOUT,DISP=(MOD,PASS) */
//GO.RPORT88 DD SYSOUT=* * IXGRPT1 OUTPUT */
//GO.SYSPRINT DD SYSOUT=* * IXGRPT1 OUTPUT */
/* */

```

Figure 30 (Part 2 of 2). Sample Job to Print MVS Log Stream Statistics

Figure 31 on page 84 shows the output produced by the sample job shown in Figure 30 on page 82.

SYSTEM LOGGER ACTIVITY REPORT (IXGRPT1)

PAGE 49

--LOGSTREAM NAME-----	STRUCTURE NAME--	#	BYT WRITTN	BYT DELETD	BYT DELETD	#	%WRITES OF		---EVENTS---		
							TO COUPLNG	FRM CF W/O	FRM CF AFTR	WRITES	COMP-TYPE
		FLAGS	FACILITY	DASD WRITE	DASD WRITE	COMPLETD	1+2	3	FULL	BLD	SHFT
SCSCFWR.DFHLOG	LOG_USERBWF_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	----- 2	1536	0	0	2	100%	0%	0	0	0
SYSPLX.OPERLOG	SYSTEM_OPERLOG	----- 2	46592	0	0	76	100%	0%	0	0	0
07/11/96 11:10:00 PM (SMF INTERVAL TIMESTAMP 'AD26D025CA200000' X)											
SCSCFWR.DFHLOG	LOG_USERBWF_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	----- 2	2560	0	0	2	100%	0%	0	0	0
SYSPLX.OPERLOG	SYSTEM_OPERLOG	----- 2	80896	0	0	137	100%	0%	0	0	0
07/11/96 11:20:00 PM (SMF INTERVAL TIMESTAMP 'AD26D261FE800000' X)											
SCSCFWR.DFHLOG	LOG_USERBWF_P01	----- 1	10496	0	0	41	100%	0%	0	0	0
SCSCFWR.VSAMU	LOG_USERBWF_P01	----- 1	67328	0	0	54	100%	0%	0	0	0
SCSCFWR.VSAMV	LOG_USERBWF_P01	----- 1	49920	0	0	43	100%	0%	0	0	0
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	----- 1	109824	627487	6338	224	100%	0%	0	0	0
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	----- 1	27136	0	0	58	100%	0%	0	0	0
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	----- 1	70400	0	0	138	100%	0%	0	0	0
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	----- 1	54528	0	0	107	100%	0%	0	0	0
SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0

KEY FOR FLAGS: R=STRUCTURE REBUILD INITIATED. F=STRUCTURE FULL DETECTED. S=STRUCTURE SPACE CRITICAL DETECTED.

SYSTEM LOGGER ACTIVITY REPORT (IXGRPT1)

PAGE 50

--LOGSTREAM NAME-----	STRUCTURE NAME--	#	BYT WRITTN	BYT DELETD	BYT DELETD	#	%WRITES OF		---EVENTS---		
							TO COUPLNG	FRM CF W/O	FRM CF AFTR	WRITES	COMP-TYPE
		FLAGS	FACILITY	DASD WRITE	DASD WRITE	COMPLETD	1+2	3	FULL	BLD	SHFT
SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	----- 1	15360	0	0	10	100%	0%	0	0	0
SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	----- 2	3072	0	0	3	100%	0%	0	0	0
SYSPLX.OPERLOG	SYSTEM_OPERLOG	----- 2	141824	0	0	255	100%	0%	0	0	0
07/11/96 11:30:00 PM (SMF INTERVAL TIMESTAMP 'AD26D49E32E00000' X)											
SCSCFWR.DFHLOG	LOG_USERBWF_P01	----- 1	1792	0	10280	7	100%	0%	0	0	0
SCSCFWR.VSAMU	LOG_USERBWF_P01	----- 1	277504	0	152646	272	100%	0%	0	0	0
SCSCFWR.VSAMV	LOG_USERBWF_P01	----- 1	299776	0	155388	268	100%	0%	0	0	0
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	----- 1	489216	0	0	977	100%	0%	0	0	0
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	----- 1	145664	0	0	303	100%	0%	0	0	0
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	----- 1	365568	0	0	724	100%	0%	0	0	0
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	----- 1	217856	0	0	450	100%	0%	0	0	0
SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	----- 1	0	0	0	0	0%	0%	0	0	0
SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	----- 2	1024	0	0	1	100%	0%	0	0	0
SYSPLX.OPERLOG	SYSTEM_OPERLOG	----- 2	30720	0	0	47	100%	0%	0	0	0

Figure 31 (Part 1 of 3). Sample Log Stream Statistics

07/11/96 11:40:00 PM (SMF INTERVAL TIMESTAMP 'AD26D6DA67400000' X)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
SCSCFWR.DFHLOG	LOG_USERBWF_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMU	LOG_USERBWF_P01	-----	1	278272	0	319675	257	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMV	LOG_USERBWF_P01	-----	1	306944	0	357939	275	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	-----	1	496896	629741	10343	988	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	-----	1	195072	0	0	401	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	-----	1	304640	637911	6341	608	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	-----	1	255488	0	0	530	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	-----	2	2048	0	0	2	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SYSPLX.OPERLOG	SYSTEM_OPERLOG	-----	2	71680	0	0	124	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
07/11/96 11:50:00 PM (SMF INTERVAL TIMESTAMP 'AD26D9169BA00000' X)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
SCSCFWR.DFHLOG	LOG_USERBWF_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
KEY FOR FLAGS: R=STRUCTURE REBUILD INITIATED. F=STRUCTURE FULL DETECTED. S=STRUCTURE SPACE CRITICAL DETECTED.																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
SYSTEM LOGGER ACTIVITY REPORT (IXGRPT1)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
PAGE 51																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
<table border="1"> <thead> <tr> <th rowspan="2">--LOGSTREAM NAME-----</th> <th rowspan="2">STRUCTURE NAME--</th> <th rowspan="2">FLAGS</th> <th rowspan="2">#</th> <th rowspan="2">BYT WRITN</th> <th rowspan="2">BYT DELETD</th> <th rowspan="2">BYT DELETD</th> <th rowspan="2">#</th> <th rowspan="2">%WRITES OF</th> <th colspan="3">---EVENTS---</th> </tr> <tr> <th>TO COUPLNG</th> <th>FRM CF W/O</th> <th>FRM CF AFTR</th> <th>WRITES</th> <th>COMP-TYPE</th> <th>STRC RE-</th> <th>DAS</th> </tr> <tr> <th colspan="3"></th> <th>CONN</th> <th>FACILITY</th> <th>DASD WRITE</th> <th>DASD WRITE</th> <th>COMPLETD</th> <th>1+2</th> <th>3</th> <th>FULL</th> <th>BLD</th> <th>SHFT</th> </tr> </thead> <tbody> <tr> <td>SCSCFWR.VSAMU</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>150272</td> <td>0</td> <td>0</td> <td>136</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCFWR.VSAMV</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>150016</td> <td>0</td> <td>0</td> <td>133</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA1.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>319744</td> <td>0</td> <td>0</td> <td>642</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA1.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA2.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>68608</td> <td>0</td> <td>0</td> <td>139</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA2.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA3.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>186368</td> <td>0</td> <td>0</td> <td>375</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA3.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA4.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>56320</td> <td>0</td> <td>0</td> <td>102</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA4.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPFA1.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPFA1.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPTA1.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPTA1.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SYSPLX.LOGREC.ALLRECS</td> <td>SYSTEM_LOGREC</td> <td>-----</td> <td>2</td> <td>1024</td> <td>0</td> <td>0</td> <td>1</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SYSPLX.OPERLOG</td> <td>SYSTEM_OPERLOG</td> <td>-----</td> <td>2</td> <td>51712</td> <td>0</td> <td>0</td> <td>88</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td colspan="11">07/12/96 0:00:00 AM (SMF INTERVAL TIMESTAMP 'AD26DB52D0000000' X)</td> </tr> <tr> <td>SCSCFWR.DFHLOG</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCFWR.VSAMU</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>190464</td> <td>0</td> <td>189105</td> <td>166</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCFWR.VSAMV</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>167936</td> <td>0</td> <td>177719</td> <td>164</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA1.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>373248</td> <td>0</td> <td>0</td> <td>741</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA1.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA2.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>49664</td> <td>0</td> <td>0</td> <td>101</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA2.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA3.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>247552</td> <td>0</td> <td>0</td> <td>492</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA3.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA4.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>82176</td> <td>647917</td> <td>9196</td> <td>166</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA4.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPFA1.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPFA1.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPTA1.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPTA1.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SYSPLX.LOGREC.ALLRECS</td> <td>SYSTEM_LOGREC</td> <td>-----</td> <td>2</td> <td>1024</td> <td>0</td> <td>0</td> <td>1</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SYSPLX.OPERLOG</td> <td>SYSTEM_OPERLOG</td> <td>-----</td> <td>2</td> <td>41984</td> <td>0</td> <td>0</td> <td>69</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td colspan="11">07/12/96 0:10:00 AM (SMF INTERVAL TIMESTAMP 'AD26DD8F04600000' X)</td> </tr> <tr> <td>SCSCFWR.DFHLOG</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCFWR.VSAMU</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>135424</td> <td>0</td> <td>181784</td> <td>133</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCFWR.VSAMV</td> <td>LOG_USERBWF_P01</td> <td>-----</td> <td>1</td> <td>118784</td> <td>0</td> <td>160232</td> <td>114</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA1.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>265728</td> <td>640412</td> <td>8683</td> <td>534</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA1.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA2.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>31488</td> <td>0</td> <td>0</td> <td>60</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA2.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA3.DFHLOG</td> <td>LOG_DFHLOG_P01</td> <td>-----</td> <td>1</td> <td>192000</td> <td>0</td> <td>0</td> <td>378</td> <td>100%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>SCSCPAA3.DFHSUNT</td> <td>LOG_DFHSHUNT_P01</td> <td>-----</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0%</td> <td>0%</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td colspan="11">KEY FOR FLAGS: R=STRUCTURE REBUILD INITIATED. F=STRUCTURE FULL DETECTED. S=STRUCTURE SPACE CRITICAL DETECTED.</td> </tr> </tbody> </table>											--LOGSTREAM NAME-----	STRUCTURE NAME--	FLAGS	#	BYT WRITN	BYT DELETD	BYT DELETD	#	%WRITES OF	---EVENTS---			TO COUPLNG	FRM CF W/O	FRM CF AFTR	WRITES	COMP-TYPE	STRC RE-	DAS				CONN	FACILITY	DASD WRITE	DASD WRITE	COMPLETD	1+2	3	FULL	BLD	SHFT	SCSCFWR.VSAMU	LOG_USERBWF_P01	-----	1	150272	0	0	136	100%	0%	0	0	0	SCSCFWR.VSAMV	LOG_USERBWF_P01	-----	1	150016	0	0	133	100%	0%	0	0	0	SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	-----	1	319744	0	0	642	100%	0%	0	0	0	SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	-----	1	68608	0	0	139	100%	0%	0	0	0	SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	-----	1	186368	0	0	375	100%	0%	0	0	0	SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	-----	1	56320	0	0	102	100%	0%	0	0	0	SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	-----	2	1024	0	0	1	100%	0%	0	0	0	SYSPLX.OPERLOG	SYSTEM_OPERLOG	-----	2	51712	0	0	88	100%	0%	0	0	0	07/12/96 0:00:00 AM (SMF INTERVAL TIMESTAMP 'AD26DB52D0000000' X)											SCSCFWR.DFHLOG	LOG_USERBWF_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCFWR.VSAMU	LOG_USERBWF_P01	-----	1	190464	0	189105	166	100%	0%	0	0	0	SCSCFWR.VSAMV	LOG_USERBWF_P01	-----	1	167936	0	177719	164	100%	0%	0	0	0	SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	-----	1	373248	0	0	741	100%	0%	0	0	0	SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	-----	1	49664	0	0	101	100%	0%	0	0	0	SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	-----	1	247552	0	0	492	100%	0%	0	0	0	SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	-----	1	82176	647917	9196	166	100%	0%	0	0	0	SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	-----	2	1024	0	0	1	100%	0%	0	0	0	SYSPLX.OPERLOG	SYSTEM_OPERLOG	-----	2	41984	0	0	69	100%	0%	0	0	0	07/12/96 0:10:00 AM (SMF INTERVAL TIMESTAMP 'AD26DD8F04600000' X)											SCSCFWR.DFHLOG	LOG_USERBWF_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCFWR.VSAMU	LOG_USERBWF_P01	-----	1	135424	0	181784	133	100%	0%	0	0	0	SCSCFWR.VSAMV	LOG_USERBWF_P01	-----	1	118784	0	160232	114	100%	0%	0	0	0	SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	-----	1	265728	640412	8683	534	100%	0%	0	0	0	SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	-----	1	31488	0	0	60	100%	0%	0	0	0	SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	-----	1	192000	0	0	378	100%	0%	0	0	0	SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0	KEY FOR FLAGS: R=STRUCTURE REBUILD INITIATED. F=STRUCTURE FULL DETECTED. S=STRUCTURE SPACE CRITICAL DETECTED.										
--LOGSTREAM NAME-----	STRUCTURE NAME--	FLAGS	#	BYT WRITN	BYT DELETD	BYT DELETD	#	%WRITES OF	---EVENTS---																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
									TO COUPLNG	FRM CF W/O	FRM CF AFTR	WRITES	COMP-TYPE	STRC RE-	DAS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
			CONN	FACILITY	DASD WRITE	DASD WRITE	COMPLETD	1+2	3	FULL	BLD	SHFT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMU	LOG_USERBWF_P01	-----	1	150272	0	0	136	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMV	LOG_USERBWF_P01	-----	1	150016	0	0	133	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	-----	1	319744	0	0	642	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	-----	1	68608	0	0	139	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	-----	1	186368	0	0	375	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	-----	1	56320	0	0	102	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	-----	2	1024	0	0	1	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SYSPLX.OPERLOG	SYSTEM_OPERLOG	-----	2	51712	0	0	88	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
07/12/96 0:00:00 AM (SMF INTERVAL TIMESTAMP 'AD26DB52D0000000' X)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
SCSCFWR.DFHLOG	LOG_USERBWF_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMU	LOG_USERBWF_P01	-----	1	190464	0	189105	166	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMV	LOG_USERBWF_P01	-----	1	167936	0	177719	164	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	-----	1	373248	0	0	741	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	-----	1	49664	0	0	101	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	-----	1	247552	0	0	492	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA4.DFHLOG	LOG_DFHLOG_P01	-----	1	82176	647917	9196	166	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA4.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPFA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPFA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPTA1.DFHLOG	LOG_DFHLOG_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPTA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SYSPLX.LOGREC.ALLRECS	SYSTEM_LOGREC	-----	2	1024	0	0	1	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SYSPLX.OPERLOG	SYSTEM_OPERLOG	-----	2	41984	0	0	69	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
07/12/96 0:10:00 AM (SMF INTERVAL TIMESTAMP 'AD26DD8F04600000' X)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
SCSCFWR.DFHLOG	LOG_USERBWF_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMU	LOG_USERBWF_P01	-----	1	135424	0	181784	133	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCFWR.VSAMV	LOG_USERBWF_P01	-----	1	118784	0	160232	114	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHLOG	LOG_DFHLOG_P01	-----	1	265728	640412	8683	534	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA1.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHLOG	LOG_DFHLOG_P01	-----	1	31488	0	0	60	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA2.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHLOG	LOG_DFHLOG_P01	-----	1	192000	0	0	378	100%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
SCSCPAA3.DFHSUNT	LOG_DFHSHUNT_P01	-----	1	0	0	0	0	0%	0%	0	0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
KEY FOR FLAGS: R=STRUCTURE REBUILD INITIATED. F=STRUCTURE FULL DETECTED. S=STRUCTURE SPACE CRITICAL DETECTED.																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													

Figure 31 (Part 2 of 3). Sample Log Stream Statistics

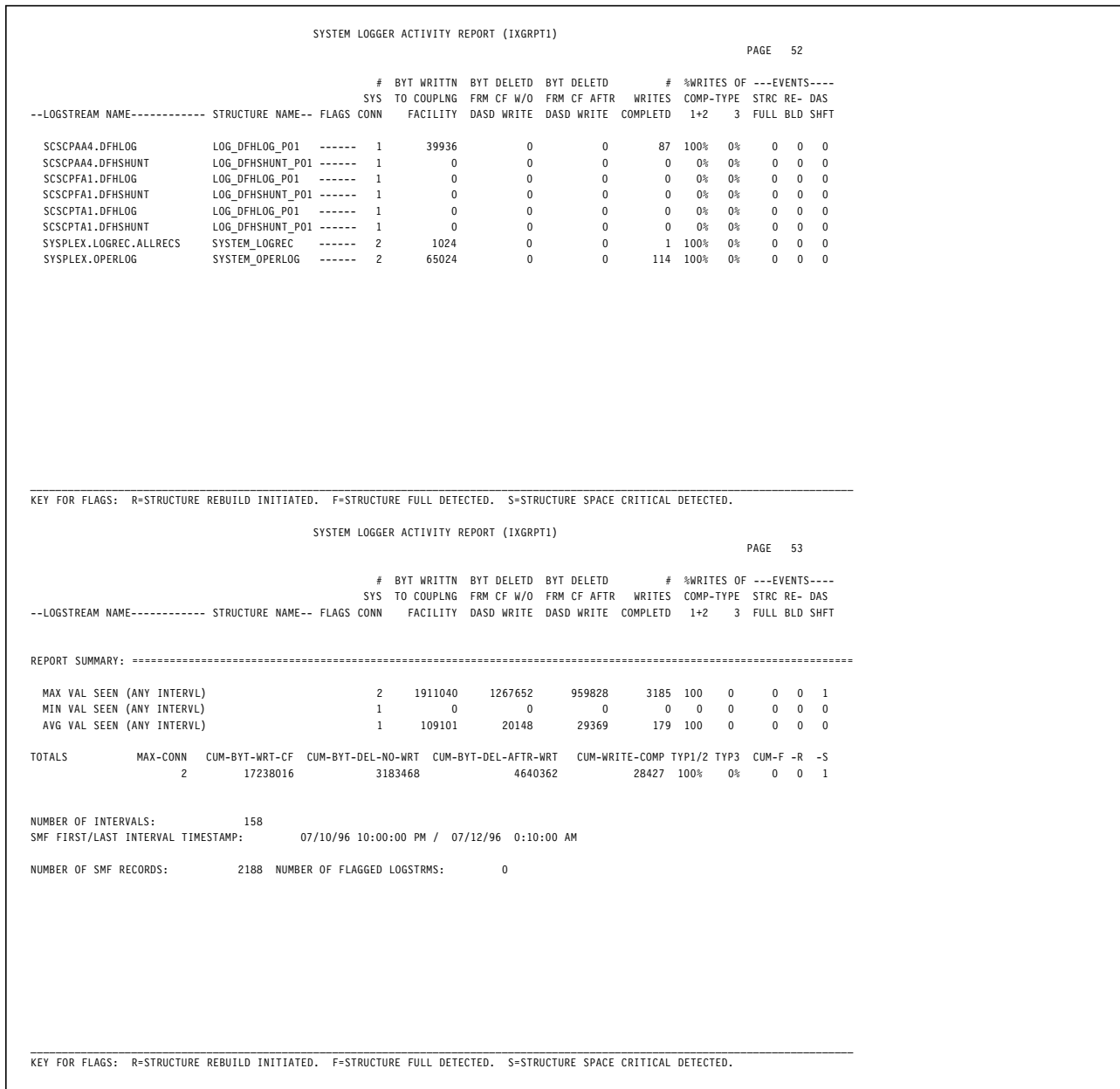


Figure 31 (Part 3 of 3). Sample Log Stream Statistics

IXGRPT1 does not report the average buffer size, but you can find this easily by dividing the BYTES WRITTEN value by the number of WRITES COMPLETED. For example, for the most active 10 minute interval (11:40:00 PM) for the system-log log stream SCSCPAA1 in Figure 31 on page 84, the average buffer size is calculated by dividing the BYTES WRITTEN (496896) by the WRITES COMPLETED (988) to give a value of 503. You should compare the actual average buffer size with the value you specified for AVGBUFSIZE when you defined the structure. An incorrectly specified AVGBUFSIZE can lead to unnecessary offloading of data from the coupling facility.

Routinely check the number of structure-full events, shown in the STRC FULL column. If structure full events occur frequently, it indicates that the MVS system logger cannot write data to DASD fast enough to keep up with incoming data, which causes CICS to wait before it can write more data. In Figure 31 on page 84 no structure-full events are reported.

Consider the following solutions to resolve structure-full events:

- Check that the correct log streams are connected to the structure
- Increase the coupling facility structure size to smooth out spikes in MVS system logger load.
- Reduce the data written to the log stream by not merging so many journals or forward-recovery logs to the same log stream.
- Reduce the HIGHOFFLOAD threshold percentage, the point at which the MVS system logger triggers the offload process data from the coupling facility to the DASD log stream data sets.
- Check that the logger data sets are not too small.
- Examine device I/O statistics for possible contention on the DASD log stream data sets.
- Use faster DASD devices.

For CICS system log with SYSLOG=NOKEEP, the best performance is achieved when CICS can delete the log tail data that is no longer needed and before the log stream is offloaded to DASD by the MVS system logger. To monitor that this is being achieved, review the numbers in the columns headed "BYT DELETED FRM CF W/O DASD WRITE" and "BYT DELETED FRM CF AFTR DASD WRITE" in the SMF Type 88 record report. These columns indicate (1) data deleted from the coupling facility without first being written to DASD and (2) data deleted from the coupling facility after being written to DASD. For a system log stream, the first column should be high while the second should be zero. However, a small amount of DASD offloading may be acceptable – for example, if coupling facility space is in short supply.

The amount of DASD offloading occurring for log stream SCSCPAA1.DFHLOG in the sample 11:40:00 interval is measured as follows:

$$DASD\ OFFLOADING = \frac{BYT\ DELETED\ FRM\ CF\ AFTR\ DASD\ WRITE}{TOTAL\ BYT\ DELETED} \times 100$$

$$DASD\ OFFLOADING = \frac{10303}{629741 + 10343} \times 100 = 1.6\%$$

Note: The performance cost of DASD offloading is more closely related to the rate at which offload events occur than to the actual amount offloaded to DASD. It is therefore also important to monitor DASD offload events (DASD SHFT column). The number of DASD shifts can be reduced by defining larger logger data sets. You should aim for DASD shifts to occur less frequently than 1 per hour.

If the "BYT DELETED FRM CF AFTR DASD WRITE" column frequently is unacceptably high for a CICS system log:

- Check that SYSLOG=KEEP is not specified as a system initialization parameter.
- Check that there are not any long-running transactions making recoverable updates without synch pointing.
- Consider increasing the size of the coupling facility structure.
- Consider increasing the HIGHOFFLOAD threshold value.

- Consider reducing the CICS activity keypoint frequency on the AKPFREQ system initialization parameter.
- If offloading is being triggered by the entry-full conditions (check the SMF88 NTRY-FULL field) the AVGBUFSIZE could be set too high.
- If offloading is being triggered by the threshold for the staging data sets, then increase the size of the staging data sets.

Note: NTRY-FULL and STG-THLD are not reported in the IXGRPT1 sample program.

For general log streams, the BYT DELETED FRM CF W/O DASD WRITE would normally be zero and the BYT DELETED FROM CF AFTR DASD WRITE would be large. For example, see the statistics for the log stream SCSCFWR.VSAMU which is the log stream that we used for forward-recovery logging.

Note: In any SMF interval, the Bytes-deleted columns may not match the Bytes-written column because data is only written to DASD and then deleted from the coupling facility when the HIGHOFFLOAD threshold limit is reached.

Chapter 6. Implementing RLS

In this chapter, we provide the information about the tasks you must perform in order to access your VSAM data sets (or some of them) in RLS mode. If you did not read the prerequisites for VSAM RLS, you should do so now. More information about migration to CICS TS is provided in Chapter 4, “CICS Migration” on page 35.

6.1 A Brief Summary of VSAM Record Level Sharing

In this section we give a short summary of the major topics related to VSAM. See 1.5, “VSAM Record Level Sharing” on page 8 for a technical overview.

- VSAM RLS allows multiple CICS TS application-owning regions (AORs) to access VSAM data sets for update while preserving data integrity.
- The updating regions can reside in any MVS image within one parallel sysplex.
- Each AOR that needs access to VSAM data in RLS mode connects automatically to the control ACB of the SMSVSAM server address space.
- SMSVSAM provides the locking on a record-level base. CICS provides the logging of before images on the CICS system log, and of the after images on a separate forward-recovery log if requested. See Chapter 5, “CICS Logging and Journaling” on page 57 for more information about logging.
- Locking information is stored in a coupling facility structure (IGWLOCK00) that is owned by SMSVSAM. Additionally, SMSVSAM stores some locking-related information in the share control data sets (SHCDS).
- SMSVSAM also owns a cache structure in the coupling facility to cache VSAM user data and perform buffer invalidation.
- RLS is a property of the file-open process, not of the data set. Therefore the first MVS task that opens a VSAM file determines the access mode for every other task that might also open the file. For example, if a CICS region is the first one to open a file in RLS mode, then all other CICS or batch regions have to open that file in RLS mode as well, otherwise the open will fail.
- You should avoid having to switch between RLS and non-RLS modes.
- A batch job that accesses recoverable RLS data sets concurrently with CICS, can only read the data set, not update it.
- Nonrecoverable RLS data sets can also be updated by a batch job concurrently to CICS. In this case, the applications have to understand the implications that multiple updaters can have.

Note: Do not switch the file definitions in CICS from recoverable to nonrecoverable and vice versa, in order to allow concurrent batch updates. See 6.2.6, “Sharing Data Sets between CICS and Batch” on page 95, for more information about batch updates.

6.1.1 Summary of Prerequisites

You need to consider the following list of prerequisites for VSAM RLS. See Chapter 3, “Prerequisite Products” on page 29, for details.

1. Consider hardware dependencies (a parallel sysplex with at least one coupling facility).
2. CICS TS
3. CICS VSAM Recovery 2.1
4. RACF 2.1
5. Appropriate levels of COBOL, PL/I, FORTRAN and Language Environment to support batch access to RLS files (see 3.9, “Programming Languages” on page 31).
6. SMS coexistence support

Within a sysplex, an SMS configuration can be shared by systems running DFSMS 1.3, DFSMS 1.2, DFSMS 1.1, and MVS/DFP 3.3. Toleration PTFs must be applied to all systems that have a pre-DFSMS 1.3 version installed and intend to keep it. The toleration maintenance will prevent a pre-DFSMS 1.3 system from opening a VSAM data set with an RLS subcell. It won't allow the pre-DFSMS 1.3 system to process the data set. See the *DFSMS/MVS 1.3 Planning For Installation Guide* for detailed information on coexistence.

Note: We recommend that you install DFSMS 1.3 on all systems in a sysplex if RLS is activated in one of the systems.

6.2 Implementing VSAM Record Level Sharing

In this section we list the major steps you must perform to migrate to VSAM RLS. We assume you have installed DFSMS 1.3 and the SMSVSAM server is available and running in all MVS images where you want RLS access. Information about DFSMS 1.3 is contained in Chapter 10, “Modifying SMS for RLS” on page 149.

6.2.1 Selecting Data Sets for RLS Access Mode

Basically any VSAM data set supported by CICS is eligible to be opened in RLS mode (including the CSD), subject to some restrictions. See 1.5, “VSAM Record Level Sharing” on page 8.

You need to decide which data sets you want to be accessed in RLS mode. The following considerations may help you make the decision:

- Data sets that are currently accessed via a file-owning region are candidates for RLS, because you probably created the FOR in order to be able to share the data sets among several application-owning regions in the first place.
- Data sets that are accessed by only a single CICS region should not be changed to RLS access mode. In this case you do not gain anything, and there is the additional overhead needed by SMSVSAM to maintain the locks on record level.
- You should consider defining a data set that is accessed only from CICS regions within a single MVS image as a shared data table, rather than migrating it to RLS access mode. The reason clearly is the better performance shared data tables gives you.

6.2.2 Review of DFSMS 1.3 Related Tasks

See Chapter 10, “Modifying SMS for RLS” on page 149, for details. The tasks are defined as follows:

- During the installation of SMSVSAM you need to define coupling facility cache and lock structures. The lock structure IGWLOCK00 must have global connectivity to all MVS images where an SMSVSAM server is running.

In order to size the cache structure, the existing space definitions for LSR pools and hyperspace are a good starting point. Defining more than one cache structure within a cache set allows data sets to be reassigned to the other cache structure in the event of a cache structure failure.

For more information on lock and cache structure definitions see the *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

- Define sharing control data sets (SHCDS)
Logically, there is one SHCDS per sysplex and a logical part of this linear data set is allocated for each MVS image. Physically, you must define and activate at least two active SHCDSs and one spare for recovery purposes.
- Examine SMS configuration changes, such as the parameter updates in member IGDSMSxx in the SYS1.PARMLIB (for example RLSINIT(Y)).
- Establish new authorization levels for users that must issue IDCAMS SHCDS commands. These users need permission to access the new RACF class STGADMIN.IGWSHCDS.REPAIR.

6.2.3 Enable RLS Access Mode

ih1.steps i2.enable RLS access mode ih1.RLS access mode i2.enable

These are the steps you must take to enable VSAM RLS support in CICS TS.

1. Set the SIT parameters related to RLS:

- RLS=YES, to make CICS use RLS and connect to the SMSVSAM control ACB.
- FTIMEOUT=30—SMSVSAM, SMSVSAM, not CICS, is responsible for detecting deadlocks for RLS files. However, SMSVSAM receives the deadlock timeout value from CICS, which is the value specified in FTIMEOUT if you did not specify a value for DTIMOUT in the transaction definition. Deadlocks will cause transactions to abend with either AFCV or AFCW. AFCV is the timeout abend on a transaction-wait for a record lock. AFCW means SMSVSAM has detected that the current request will cause a deadlock situation, and therefore the requesting transaction is abended to prevent the deadlock.

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

2. Change your file definitions using RDO or the DFHCSDUP utility. You may choose to change only the following attributes and leave the rest as they are because, for RLS, files they are either ignored or obtained from the ICF catalog:

- RLS access must be set to YES.
This attribute causes CICS to perform the next open against the file in RLS mode.

- The REMOTE attributes must be removed (if there are any), because RLS access is always local for each AOR.

Figure 32 shows a sample definition for the file CVRTEST:

```

OBJECT CHARACTERISTICS                                CICS RELEASE = 0510
CEDA View File( CVRTEST )
File          : CVRTEST
Group         : DSWRLSU
DEscription  : RLS Testfile with forward recovery on (see ICF)
VSAM PARAMETERS
DSName       : CICSDSW.VSAMU.TESTDB
Password     :                               PASSWORD NOT SPECIFIED
RLSaccess   : Yes                            Yes | No
Lsrpoolid    : 1                             1-8 | None
READInteg    : Repeatable                     Uncommitted | Consistent | Repeatable
DSNSharing   : Allreqs                        Allreqs | Modifyreqs
STRings      : 001                            1-255
Nsrgroup     :
REMOTE ATTRIBUTES
REMOTESystem :
REMOTENAME   :
RECORDSize   :                               1-32767

```

Figure 32. Example of File Definition for an RLS File

It might also be a good idea to keep your RLS files in a separate group list so that all CICS TS AORs can use the same definitions for RLS files.

3. Define a coupling facility structure and log streams for your forward-recovery logs. If you want a data set to be forward recoverable, you must tell the system where to write the log data, by defining a log stream name in the ICF catalog (see step Four on page :spotref=stp4. The log stream name, however, must first be defined in the LOGR policy, along with the definition of a coupling facility structure that receives the log data before it is written to the log stream data set.

The following is a list of secondary steps to follow, including examples from our test environment:

- a. Define a structure in your CFRM policy for the forward-recovery log streams. You may define one structure for all your general logs:

```

//STEP1 EXEC PGM=IXCMIAPU
//SYSIN DD *
DATA TYPE(CFRM) REPORT(YES)
DEFINE POLICY NAME(CFRM02) REPLACE(YES)
CF NAME(CF01)
etc...

STRUCTURE NAME(LOG_USERBWF_P01)
SIZE(2320)
INITSIZE(1510)
PREFLIST(CF02,CF01)
REBUILDPERCENT(1)

```

- b. Now define the structure and the corresponding log stream names that go into this structure using MVS system logger (LOGR) policy:

```

//IXCMIAPU EXEC PGM=IXCMIAPU
/** THIS JOB:-DEFINES A STRUCTURE WHICH MUST MATCH TO THE
/**          STRUCTURE IN THE CFRM POLICY FOR ALL FW
/**          RECOVERY LOGS OF ANY VSAM FILE USED, AND
/**          FOR THE LOG OF LOGS
/**          -DEFINES THE LOGSTREAMS TO GO IN THIS STRUCTURE
//SYSIN DD *
DATA TYPE(LOGR) REPORT(YES)

DEFINE STRUCTURE
  NAME(LOG_USERBWFP_P01)
  LOGSNUM(3)
  MAXBUFSIZE(64000)
  AVGBUFSIZE(2048)
DEFINE LOGSTREAM
  NAME(SCSCFWR.MODEL)
  STRUCTNAME(LOG_USERBWFP_P01)
  STG_DUPLEX(YES)
  DUPLEXMODE(COND)
  STG_SIZE(3000)
  LS_SIZE(1250)
  HLQ(CICS)
  HIGHOFFLOAD(80)
  LOWOFFLOAD(0)
  MODEL(YES)
DEFINE LOGSTREAM
  NAME(SCSCFWR.DFHJGLOG)
  STRUCTNAME(LOG_USERBWFP_P01)
  STG_DUPLEX(YES)
  DUPLEXMODE(COND)
  STG_SIZE(100)
  LS_SIZE(100)
  HLQ(CICS)
  HIGHOFFLOAD(80)
  LOWOFFLOAD(0)

```

This job fragment shows the LOGR definitions for a structure to be used by the general logs, and the two log streams that are currently directed to this structure. For non-RLS data sets and auto journals, you can provide journal-model resource definitions in CICS, as shown in Figure 33.

OBJECT CHARACTERISTICS	CICS RELEASE = 0510
CEDA View Journalmodel(BWFP#J90)	
Journalmodel	: BWFP#J90
Group	: C51LGMOD
Description	: BLOCK WRITE FORCED PERIODICALLY TYPE LOGS
Journalname	: DFHJ90
Type	: Mvs Smf Dummy
Streamname	: SCSCFWR.VSAMU

Figure 33. Journal Model for a Non-RLS File

Notes on the definitions in the examples:

- 1) Read more about such definitions in Chapter 5, “CICS Logging and Journaling” on page 57.
- 2) Note that the structure name in the LOGR policy must match the one defined in the CFRM policy.

- 3) The first log stream definition is a model definition for our forward-recovery log streams (SCSCFWR.MODEL). Thus the actual log stream name can be dynamically defined by CICS when it needs the log stream name. For RLS data sets, however, CICS derives the log stream name from the ICF catalog (see item 4). For the other log streams, you can define a journal model resource in CICS.
 - 4) The HLQ parameter in the log stream definition determines the high level qualifier for the log stream data-set name that is assigned by the MVS system logger whenever a new log stream data set is needed. Thus the MVS system logger will create for example a log stream name of "CICS.SCSCFWR.VSAMU.A0000023," where the last qualifier is an iteration value generated by the MVS system logger.
 - 5) The LS_SIZE parameter determines the size of your log stream data sets allocated by the MVS system logger whenever it needs a new log stream data set. The value you specify here influences how quickly you reach the logger inventory limit of 168 log stream data sets per log stream name.
4. Define the recovery-related parameters for each data set in the ICF catalog. The following example shows the SYSIN statements for an IDCAMS job to alter the catalog entry for one of our test data sets associated with file CVRTEST:

```
//IDCAMS EXEC PGM=IDCAMS,REGION=512K
//SYSIN DD *
ALTER CICSDSW.VSAMU.TESTDB LOG(ALL) LSID(SCSCFWR.VSAMU)
```

LOG=ALL means you want the file to be recoverable, including forward recovery. Therefore a log stream ID (LSID) has to be specified as well, giving the log stream name for the forward-recovery log, as defined in the LOGR policy. This is not checked in the catalog update, although CICS will check it during file open and return the following error if no log stream ID is specified:

```
+DFHFC0522 SCSCPFA1 884
RLS OPEN of file CVRTEST failed. IGWARLS call has returned that the
LOG parameter is set to ALL but LOGSTREAMID has not been specified.
Module DFHFCRO.
```

Other values for LOG are

NONE—defines the data set as nonrecoverable.

UNDO—defines the data set as recoverable, back-out only.

You may also specify the BWO parameter in the DEFINE or ALTER statements if you want to use BWO for a data set. You can do this only if LOG is defined as ALL. You may want to keep things simple and consider defining these parameters for non-RLS files as well having them in one place for maintenance.

Note: If CICS finds these parameters in the ICF catalog, the corresponding file resource definitions in the CSD are ignored.

6.2.4 Read Integrity

You can now request read integrity for the file operations. This can be achieved globally for all file requests by defining either consistent read or repeatable read in the READInteg parameter of the file definition. The following example defines repeatable read, which means CICS acquires shared locks when it reads records, and releases them at a synch point.

```
READInteg      : Repeatable          Uncommitted | Consistent | Repeatable
```

This ensures that the record cannot be changed by other users while it is read by the transaction that caused the lock request. Another way to request read integrity is to specify it in the individual API command. However, read integrity is recommended only for applications that cannot tolerate dirty reads. Before using it, you should review the application for increased risk of deadlock. See the *CICS Application Programmers Reference* for information about the increased risk of deadlocks, particularly when defining read integrity on file definitions.

6.2.5 Plan for Fallback

It is best to have a fallback plan ready before the migration to RLS is performed. When falling back from DFSMS 1.3 to a prior release, the RLS information must be deleted from the ICF catalog. You can use the SHCDS CFRESET command on a DFSMS 1.3 system to do this. See the *DFSMS 1.3 Planning For Installation Guide* for detailed information on fallback.

A short summary of fallback includes these steps:

1. Resolve all shunted UOWs and quiesce the data sets.
2. Because you need to change the file resource definitions back to non-RLS, it is a good idea to keep your original definitions and create a separate group list for RLS files as mentioned in 6.2.3, “Enable RLS Access Mode” on page 91.
3. Shut down CICS TS normally (CEMT P SHUT).
4. Take backups of important data sets.
5. Prepare the CICS/ESA 4.1 region:
 - a. Recreate CICS/ESA 4.1 logs.
 - b. Recreate CICS/ESA 4.1 catalogs.
 - c. Reactivate the CICS/ESA 4.1 tables and check your SIT overrides.
6. Start CICS/ESA 4.1. It must be a cold start.

The *CICS Migration Guide* contains a chapter about fallback planning.

6.2.6 Sharing Data Sets between CICS and Batch

VSAM ensures that a sphere cannot be opened in RLS mode at the same time as it is being opened in any of the existing VSAM modes (NSR, LSR and GSR). Batch programs cannot open recoverable data sets for update in RLS mode. Figure 34 on page 96 shows the support for sharing data sets between CICS and batch.

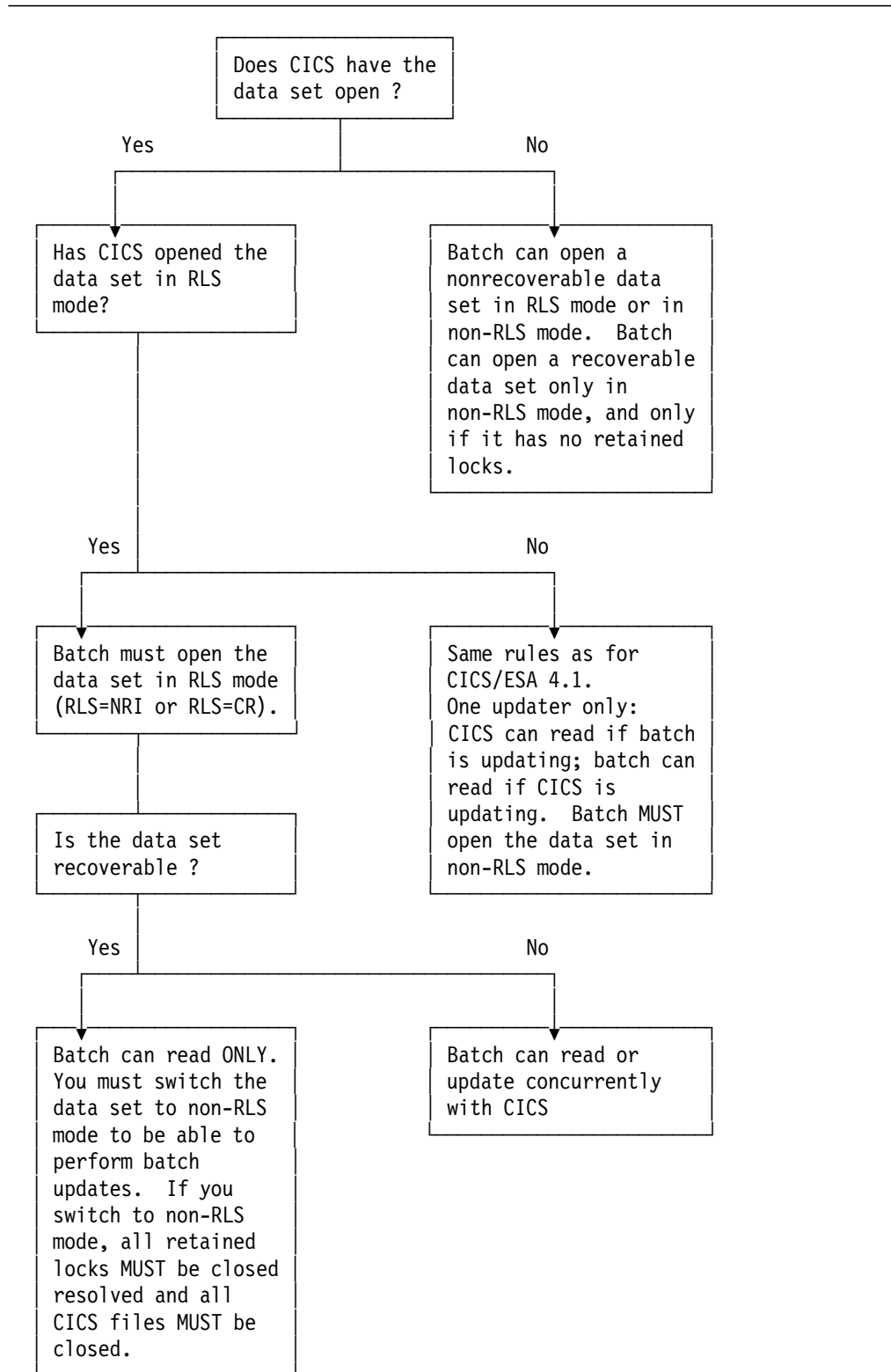


Figure 34. Sharing Data Sets among CICS and Batch Jobs

Switching between RLS mode and non-RLS mode is generally not recommended. You may want to consider converting the batch programs to run as CICS transactions or to use the external CICS interface (EXCI) API that allows

you to initiate a CICS transaction from a batch job. See the *CICS External CICS Interface* for more information.

If you cannot avoid switching, however, follow the procedure outlined here:

1. If a data set has retained locks, a batch job cannot open the data set for update! Retained locks can, for example, result from transactions that have failed to back out. The retained locks must be resolved prior to the batch run. *CICS and VSAM Record Level Sharing: Recovery Considerations* describes retained locks in detail, and gives test cases that allow you to learn how to resolve them without loss of data integrity.
2. Use the QUIESCE data set command for the data set to be switched to non-RLS mode, so that all files throughout the sysplex are closed in an orderly manner. All in-flight transactions are allowed to come to a synch point. Figure 35 shows the QUIESCE command for our test data set and the result of this operation.

```
CEMT SET DSN(CICSDSW.VSAMU.TESTDB) Q

RESULT - OVERTYPE TO MODIFY
  Dsname(CICSDSW.VSAMU.TESTDB)
  Accessmethod(Vsam)
  Action(          )
  Filecount(0001)
  Validity(Valid)
  Object(Base)
  Recovstatus(Fwdrecoverable)
  Backuptype()
  Frlog(00)
  Availability( Available )
  Lostlocks(Nolostlocks)
  Retlocks(Noretained)
  Quiescestate( Quiesced )
  Uowaction(          )
  Basedsname(CICSDSW.VSAMU.TESTDB)
  Fwdrecovlsn(SCSCFWR.VSAMU)
```

Figure 35. Using CEMT to Quiesce an RLS File

See *CICS System Programming Reference* for details.

3. When all retained locks are resolved, make a backup of the data set so that the batch job can be repeated in case it ends abnormally.
4. Run the batch update jobs.
5. Make another backup so CICS has a new backup copy for forward recovery.
6. Unquiesce the data set to allow it to be reopened in RLS access mode.

CICS provides SPI commands (such as INQUIRE UOWDSNFAIL) and sample programs (DFH0BAT1, DFH0BAT2 and DFH0BAT3) to help with the investigation of retained locks.

You can allow CICS to read the files while the batch update is running, by changing the operation attributes to NO for all operations except read and browse using CEMT.

There is some support for emergency situations, where you cannot (or do not want to) go through the quiescing procedure. However, note that in this case you risk losing data integrity.

- If a batch must run, and you wish to remove the locks (with possible loss of data integrity), you can clear in-doubt failures without waiting for them to be resolved by issuing `CEMT SET DSNAME(dsn) COMMIT|BACKOUT|FORCE` and force the locks by issuing `CEMT SET DSNAME(dsn) RESETLOCKS`.
- In situations where it is not convenient to resolve locks using CICS commands, or if you know that the batch jobs never update the same records as CICS, you can use the SHCDS option `PERMITNONRLSUPDATE`. However, `PERMITNONRLSUPDATE` causes CICS to discard any shunted units of work that may exist for the particular data set because otherwise the CICS backouts could overwrite updates that had been made by batch. If you do not want the shunted UOWs to be discarded, you can use the global user exit `XFCBOVER`.

6.3 RLS Failure Scenarios

In this section we highlight some failure situations where the failure handling differs for files accessed in RLS mode. You find more details in *CICS and VSAM Record Level Sharing: Recovery Considerations*.

- Back-out failures

Unlike in earlier releases, a back-out failure does not cause a data set to be taken offline. CICS will continue to use the dataset with individual records being locked. Some of the failures can be automatically resolved, while others will need the data set to be taken offline, but this can be done later.

- SMSVSAM server failure

Should the SMSVSAM server abend, then it will automatically restart up to six times. When the SMSVSAM server restarts, it issues an MVS ENF signal which triggers CICS to use dynamic RLS restart to recover from the server failure. No manual action should be required.

- Cache structure failure

This is similar to a server failure in that no manual intervention should be required. SMSVSAM rebuilds the cache or switches data sets that were using the cache to another cache structure in the cache set.

- Lock structure failure

This is a more serious failure. When the lock structure fails, the SMSVSAM servers attempt to rebuild the structure from their locally held copies of the locks. If this rebuild fails, then all the servers terminate and can be used again only after they connect to a new lock structure. The sharing control data sets are updated to reflect all the data sets that have lost locks. A data set with lost locks cannot be used until all UOWs that were updating the data set at the time of the failure complete by commit or backout. This may require manual intervention for backout-failed and in-doubt UOWs.

- MVS failure

SMSVSAM servers on other MVS images convert active locks into retained locks. When you IPL the failed MVS image, CICS recovery is performed by emergency restart.

- Sysplex failure

The first SMSVSAM server to restart reconnects to the lock structure and converts all active locks to retained locks.

- Forward-recovery log failure

If CICS can no longer use the forward recovery log stream you defined because of a log stream data set failure, for example, then CICS quiesces the data set and issues a message advising you to make an immediate backup copy of the data set. You should then take the new backup and redefine the log stream.

Chapter 7. Coupling Facility Preparation

If you have not read *MVS/ESA Setting Up a Sysplex*, do so now. Consult with your MVS systems programmer. Establish responsibility for the coupling facility definitions because the CICS coupling facility definitions must be placed with other MVS and subsystem definitions.

7.1 Coupling Facility Data Sets and Policies

There are two types of coupling data sets and policies that concern CICS, the Coupling Facility Resource Manager (CFRM) and the MVS system logger (LOGR).

Coupling facility definitions can be compared as follows:

- The CFRM policy definitions are used to reserve space (structure) in the coupling facility.
- The LOGR policy definitions are used to define virtual data sets (log streams) within the reserved space (structure). The space occupied by each log stream within a structure is calculated by dividing the number of log streams into the size of the structure. Two other data sets, namely the staging and log stream, are also defined by the LOGR policy. Do not confuse the log stream data set with the log stream within the structure; they are separate entities.

Note: A structure that is to contain log streams has to be defined in both the CFRM and LOGR policies. Other structures (such as temporary storage server) have only a CFRM policy definition.

7.1.1 Coupling Facility Resource Manager

Use the following MVS command to obtain CFRM-related information.

```
DISPLAY XCF,COUPLE,TYPE=CFRM  
DISPLAY XCF,STRUCTURE,STRNAME=structure_name
```

Figure 36 on page 102 shows an example of using the `DISPLAY XCF,COUPLE,TYPE=CFRM` command from SDSF. The command has been abbreviated to `D XCF,CPL,TYPE=CFRM`

```

Display Filter View Print Options Help
-----
SDSF JOB DATA SET DISPLAY - JOB PRTSTALS (JOB29579)      COMMAND ISSUED
COMMAND INPUT ==> /D XCF,CPL,TYPE=CFRM                  SCROLL ==> CSR
RESPONSE=SC52
IXC358I 20.43.02 DISPLAY XCF 132
CFRM COUPLE DATA SETS
PRIMARY  DSN: SYS1.XCF.CFRM00
          VOLSER: TOTDS0      DEVN: OCEE
          FORMAT TOD          MAXSYSTEM
          01/02/96 10:34:58    16
ALTERNATE DSN: SYS1.XCF.CFRM01
          VOLSER: TOTDS1      DEVN: OFEE
          FORMAT TOD          MAXSYSTEM
          01/02/96 10:36:39    16
CFRM IN USE BY ALL SYSTEMS

```

Figure 36. Displaying the CFRM Data Sets from SDSF

Figure 37 shows an example of using the DISPLAY XCF,STRUCTURE,STRNAME=structure_name command from SDSF. The command has been abbreviated to D XCF,STR,STRNM=structure_name

```

SDSF OPERLOG DATE 07/03/1996    0 WTORS                8 RESPONSES NOT SHOWN
COMMAND INPUT ==> /D XCF,STR,STRNM=LOG_DFHLOG_P01      SCROLL ==> CSR
RESPONSE=SC52
IXC360I 15.53.30 DISPLAY XCF 941
STRNAME: LOG_DFHLOG_P01
STATUS: ALLOCATED
POLICY SIZE      : 24320 K
POLICY INITSIZE: 16128 K
REBUILD PERCENT: 1
PREFERENCE LIST: CF02      CF01
EXCLUSION LIST IS EMPTY

ACTIVE STRUCTURE
-----
ALLOCATION TIME: 07/03/96 15:52:42
CFNAME          : CF02
COUPLING FACILITY: 009672.IBM.02.000000040104
                  PARTITION: 1  CPCID: 01
ACTUAL SIZE     : 16128 K
STORAGE INCREMENT SIZE: 256 K
VERSION         : AD1C9520 A0B8A002

```

Figure 37. Example of MVS Command Displaying Structure Information

The following is an example of the IXCMIAPI utility that is used to define structures and their attributes to the CFRM policy:

```

//STRCTRS JOB ...
//STEP1 EXEC PGM=IXCMIAPI
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(CFRM) REPORT(YES)
STRUCTURE NAME(structure-name)
SIZE(size)

```

```

INITSIZE(initial-size)
PREFLIST(cfname1,cfname2,...cfname8)
REBUILDPERCENT(rebuild-percent)
/*

```

7.1.2 MVS System Logger

Use the following MVS command to obtain information about the LOGR couple data sets:

```
D XCF,CPL,TYPE=LOGR
```

```

  Display Filter View Print Options Help
-----
SDSF JOB DATA SET DISPLAY - JOB PRTSTALS (JOB29579)      COMMAND ISSUED
COMMAND INPUT ==> /D XCF,CPL,TYPE=LOGR                  SCROLL ==> CSR
RESPONSE=SC52
IXC358I 20.41.16 DISPLAY XCF 123
LOGR COUPLE DATA SETS
PRIMARY  DSN: SYS1.XCF.LOGR00
          VOLSER: TOTDS0      DEVN: OCEE
          FORMAT TOD          MAXSYSTEM
          06/11/96 21:28:19    32
ALTERNATE DSN: SYS1.XCF.LOGR01
          VOLSER: TOTDS1      DEVN: OFEE
          FORMAT TOD          MAXSYSTEM
          06/11/96 21:28:25    32
LOGR IN USE BY ALL SYSTEMS

```

Figure 38. Example of MVS Command Displaying the LOGR Data Sets

The following is an example of the IXCMIAPU utility that is used to define log streams and their attributes to the LOGR policy:

```

//LOGSTRMS JOB ...
//STEP1 EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(LOGR) REPORT(YES)
DEFINE STRUCTURE
    NAME(structure-name)
    LOGSNUM(logstreams-per-structure-name)
    MAXBUFSIZE(maximum-buffer-size)
    AVGBUFSIZE(average-buffer-size)
DEFINE LOGSTREAM
    NAME(log-stream-name)
    STRUCTNAME(structure-name)
    STG_DUPLEX(duplex-staging)
    DUPLEXMODE(duplex-mode)
    STG_DATACLAS(staging-dataset-sms-data-class)
    STG_MGMTCLAS(staging-dataset-sms-management-class)
    STG_STORCLAS(staging-dataset-sms-storage-class)
    STG_SIZE(staging-dataset-size)
    LS_DATACLAS(logstream-dataset-sms-data-class)
    LS_MGMTCLAS(logstream-dataset-sms-management-class)
    LS_STORCLAS(logstream-dataset-sms-storage-class)
    LS_SIZE(logstream-dataset-size)
    HLQ(high-level-qualifier-of-datasets)
    HIGHOFFLOAD(start-offload-percent)

```

```

LOWOFFLOAD(end-offload-percentage)
MODEL(is-definition-a-model)
/*

```

Note: The staging and logstream data sets are on DASD and not in the coupling facility.

The numbers of structures and log streams that can be defined to the LOGR policy data sets are established when the LOGR policy data sets are formatted. Use the above JCL example with only the DATA TYPE(LOGR) REPORT(YES) statement to produce a report. In the report, LSR specifies the maximum number of log streams that can be defined to the LOGR policy and LSTRR specifies the maximum number of structure names that can be defined to the LOGR policy. Make sure that the LOGR policy can hold all CICS-related specific and dynamic definitions. An example of the above-mentioned report section follows:

Type	Formatted	In-use
-----	-----	-----
LSR (Log Stream)	60	18
LSTRR (Structure)	60	5

7.2 Naming Convention

It is sensible to adopt a naming convention for your coupling facility structures that would help to identify the purpose of the structure.

7.2.1 Log Streams

A format such as LOG_*purpose_enn* is recommended for structures containing log streams, where:

- LOG identifies the content type of the structure.
- The word *purpose* identifies the type of use of the structure.
- The letter *e* identifies the type of environment that will use the structure, for example, P for production and D for development.
- *nn* is a sequence number to allow for more than one structure for each purpose within each environment.

For example, CICS log streams can be divided into the following coupling facility structures based on their type of use:

LOG_DFHLOG_P01

For the CICS system log. The structure should be large enough to avoid the need to write data to DASD. The average buffer size would be small. This structure is required.

LOG_DFHSUNT_P01

For the CICS secondary system log. The structure should be small but requires a large buffer size. This structure is required.

LOG_USERBWFP_P01

For user journals and forward-recovery logs where block writes are forced periodically (BWFP). This type of structure is optional.

LOG_USERBWNF_P01

For user journals where block writes are not forced (BWNF). The average and maximum buffer sizes of these structures would be the same. This type of structure is optional.

7.2.2 TS Pools

A format such as DFHXQLS_////ssss is recommended for structures containing temporary storage elements, where:

- DFHXQLS is required by the temporary storage server. It identifies the structure as a temporary storage server pool structure and cannot be changed.
- //// could be any literal or omitted. In the examples, TSQS is used.
- ssss identifies the system ID of the TS server region. The system ID could be from the queue-owning region that the temporary storage server replaced.

7.3 Log Stream Sizing

Use the DFHLSCU utility to help you in calculating the SIZE, INITSIZE, AVGBUFSIZE, STG_SIZE attributes of your coupling facility definitions. It is the easiest method. For more information on the DFHLSCU utility, refer to the *CICS Operations and Utility Guide*

The setting of the following structure and log stream attributes are general in nature for all types of CICS-related coupling facility definitions:

STRUCTNAME and NAME of structure

Use a naming convention as covered in Section 7.2.2. Share structures between MVS images so that automated rebuild recovery is possible when an MVS image or MVS logger address space fails. Remember that structures defined in the LOGR policy are usable only when they are also defined in a CFRM policy, and that CFRM policy is activated.

PREFLIST

Seek the assistance of your MVS systems programmer in setting PREFLIST. Use the MVS command DISPLAY XCF to obtain information.

REBUILDPERCENT

Use a low value, for example 1.

MAXBUFSIZE

Set value to 64000. This allows CICS to write the maximum size user record (62 KB) and allows coupling facility storage to be allocated in 256 byte units. If you allow MAXBUFSIZE to default (64 KB), coupling facility storage is allocated in 512 byte units, which can be wasteful. There is no significant advantage in setting MAXBUFSIZE lower than 64000 as far as the utilization of storage is concerned.

MODEL

Setting up models is worthwhile only if there will be several log streams to be defined using the model, each of the log streams has similar characteristics, and all should go into the same coupling facility structure. Use MODEL(YES) for DFHLOG and DFHSHUNT log streams and eliminate the requirement to predefine the DFHLOG and DFHSHUNT log streams for each new CICS region you start up.

LOGSNUM

Try not to exceed 20 log streams per structure. The speed of structure rebuild depends on structure size; smaller structures are faster to allocate and rebuild. 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 structure INITSIZE. A higher value for LOGSNUM can be used when defining the structure to the MVS System Logger, however, 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). It is not possible to change LOGSNUM (or AVGBUFSIZE and MAXBUFSIZE) after a structure has been defined, without deleting and redefining the log streams which are currently defined within the structure.

HLQ Set the high-level qualifier of the staging and log stream data sets of the log stream to a value other than the default (IGXLOGR), for example CICS.

STG_DUPLEX and DUPLEXMODE

If you have a nonvolatile, stand-alone coupling facility for normal logging, with a PR/SM LPAR configured as a backup coupling facility or only a single coupling facility, define STG_DUPLEX(YES) and DUPLEXMODE(UNCOND). Otherwise, define STG_DUPLEX(YES) and DUPLEXMODE(COND) for all log streams.

SMS attributes

The benefit of using the SMS attributes are that the staging and log stream data set sizes can be changed without any LOGR definition changes. The SMS ACS routines can override SMS attribute specifications depending on their logic.

7.3.1 DFHLOG Log Streams

For a more detailed explanation, refer to *CICS Installation Guide* and *CICS Performance Guide*.

NAME of log stream

Only one should exist per CICS region across the sysplex. For model definitions, there must be a model definition for each MVS image where CICS will run.

SIZE

Use the DFHLSCU utility or the following formula:

$$480 + \frac{(\text{LOGSNUM} * (2500 + (\text{no. entries} + 5) * (\text{AVGBUFSIZE} * 1.6821 + 289)))}{1024}$$

Round the result up to the next multiple of 256.

INITSIZE

Use the DFHLSCU utility or the following formula:

$$310 + \frac{(\text{LOGSNUM} * (2000 + (\text{no. entries} + 5) * (\text{AVGBUFSIZE} * 1.1289 + 195)))}{1024}$$

Round the result up to the next multiple of 256.

AVGBUFSIZE

Use the DFHLSCU utility or the following formula.

AVGBUFSIZE = (bytespersec / writespersec) + 48

HIGHOFFLOAD

If SIT parameter SYSLOG=NOKEEP then set HIGHOFFLOAD(95) else set HIGHOFFLOAD(80).

LOWOFFLOAD

Use the following formula.

$$\text{LOWOFFLOAD} = \frac{\text{trandur} * 90}{\text{akpintv1} + \text{trandur}} + 10$$

where SYSLOG = NOKEEP is specified

STG_SIZE

Use the DFHLSCU utility or the following formula.

$$\text{STG_SIZE} = (\text{NR} * \text{AVGBUFSIZE} \text{ rounded up to next unit of } 4096)$$

where NR is the number of records to fill the log stream structure. This can be calculated as follows:

$$\text{NR} = \text{logstream structure size} / (\text{AVGBUFSIZE} \text{ rounded up to next element})$$

LS_SIZE

Set to DFHJ01A journal size of previous CICS release.

Figure 39 shows a sample CFRM policy definition.

```
STRUCTURE NAME(LOG_DFHLOG_P01)
SIZE(14848)
INITSIZE(9728)
PREFLIST(CF02,CF01)
REBUILDPERCENT(1)
```

Figure 39. Sample CFRM Policy Definition

Figure 40 on page 108 shows sample LOGR policy definitions. SC47 and SC52 are the names of MVS sysnames.

```

DEFINE STRUCTURE
    NAME(LOG_DFHLOG_P01)
    LOGSNUM(6)
    MAXBUFSIZE(64000)
    AVGBUFSIZE(256)
DEFINE LOGSTREAM
    NAME(SC47.DFHLOG.MODEL)
    STRUCTNAME(LOG_DFHLOG_P01)
    STG_DUPLEX(YES)
    DUPLEXMODE(COND)
    STG_SIZE(4000)
    LS_SIZE(7000)
    HLQ(CICS)
    HIGHOFFLOAD(95)
    LOWOFFLOAD(60)
    MODEL(YES)
DEFINE LOGSTREAM
    NAME(SC52.DFHLOG.MODEL)
    STRUCTNAME(LOG_DFHLOG_P01)
    LIKE(SC47.DFHLOG.MODEL)
    MODEL(YES)

```

Figure 40. Sample LOGR Policy Definitions

Example DFHCSD definitions include these:

```

DEFINE JOURNALMODEL(C51LOG) GROUP(C51LGMOD)
DESCRIPTION(Primary CICS system log)
    JOURNALNAME(DFHLOG) TYPE(MVS)
    STREAMNAME(&APPLID..DFHLOG)

```

7.3.2 DFHSHUNT Log Streams

NAME of log stream

Only one log stream name should exist per CICS region across the sysplex. For model definitions, there has to be a model definition for each MVS image where CICS will run.

SIZE

Use the following formula:

SIZE=230*LOGSNUM+480.

LOGSNUM should be same as for DFHLOG.

INITSIZE

Use the following formula:

INITSIZE=150*LOGSNUM+310.

LOGSNUM should be same as for DFHLOG.

AVGBUFSIZE

Set to AVGBUFSIZE(4096).

HIGHOFFLOAD

If SIT parameter SYSLOG=NOKEEP then set HIGHOFFLOAD(95) else set HIGHOFFLOAD(80).

LOWOFFLOAD

If SIT parameter SYSLOG=NOKEEP then set LOWOFFLOAD(60) else set LOWOFFLOAD(10).

STG_SIZE

Use the following formula:

$$\text{STG_SIZE} = V * \text{AKPFREQ} * (\text{AVGBUFSIZE rounded up to multiple of 4096})$$

If SIT parameter SYSLOG=NOKEEP set V = 4 else set V = 1.25.

LS_SIZE

Set to DFHJ01A journal size of previous CICS release.

Example CFRM definitions include this:

```
STRUCTURE NAME(LOG_DFHSUNT_P01)
          SIZE(1860)
          INITSIZE(1210)
          PREFLIST(CF02,CF01)
          REBUILDPERCENT(1)
```

Example LOGR definitions include these:

```
DEFINE STRUCTURE
      NAME(LOG_DFHSUNT_P01)
      LOGSNUM(6)
      MAXBUFSIZE(64000)
      AVGBUFSIZE(4096)
DEFINE LOGSTREAM
      NAME(SC47.DFHSUNT.MODEL)
      STRUCTNAME(LOG_DFHSUNT_P01)
      STG_DUPLEX(YES)
      DUPLEXMODE(COND)
      STG_SIZE(4000)
      LS_SIZE(7000)
      HLQ(CICS)
      HIGHOFFLOAD(95)
      LOWOFFLOAD(60)
      MODEL(YES)
DEFINE LOGSTREAM
      NAME(SC52.DFHSUNT.MODEL)
      LIKE(SC47.DFHSUNT.MODEL)
      MODEL(YES)
```

Example DFHCSD definitions include this:

```
DEFINE JOURNALMODEL(C51SHUNT) GROUP(C51LGMOD)
DESCRIPTION(Secondary CICS system log, shunted tasks)
      JOURNALNAME(DFHSUNT) TYPE(MVS)
      STREAMNAME(&APPLID..DFHSUNT)
```

7.3.3 Log Streams of the Type Block Write Forced Periodically

NAME of log stream

Make sure that the log stream name is descriptive and unique throughout the sysplex.

SIZE

Use the DFHLSCU utility.

INITSIZE

Use the DFHLSCU utility.

AVGBUFSIZE

Use the DFHLSCU utility.

HIGHOFFLOAD

Set to HIGHOFFLOAD(80).

LOWOFFLOAD

Set to LOWOFFLOAD(0).

STG_SIZE

Use the DFHLSCU utility.

LS_SIZE

Set to the size of the journal in the prior release of CICS.

Example CFRM definitions include this:

```
STRUCTURE NAME(LOG_USERBWFP_P01)
          SIZE(2320)
          INITSIZE(1510)
          PREFLIST(CF02,CF01)
          REBUILDPERCENT(1)
```

Example LOGR definitions include these:

```
DEFINE STRUCTURE
      NAME(LOG_USERBWFP_P01)
      LOGSNUM(3)
      MAXBUFSIZE(64000)
      AVGBUFSIZE(2048)
DEFINE LOGSTREAM
      NAME(SCSCFWR.MODEL)
      STRUCTNAME(LOG_USERBWFP_P01)
      STG_DUPLEX(YES)
      DUPLEXMODE(COND)
      STG_SIZE(3000)
      LS_SIZE(1250)
      HLQ(CICS)
      HIGHOFFLOAD(80)
      LOWOFFLOAD(0)
      MODEL(YES)
DEFINE LOGSTREAM
      NAME(SCSCFWR.DFHLGLOG)
      STRUCTNAME(LOG_USERBWFP_P01)
      STG_DUPLEX(YES)
      DUPLEXMODE(COND)
      STG_SIZE(100)
      LS_SIZE(100)
      HLQ(CICS)
      HIGHOFFLOAD(80)
      LOWOFFLOAD(0)
```

Example DFHCSD definitions include these:

```
DEFINE JOURNALMODEL(BWFP#J90) GROUP(C51LGMOD)
DESCRIPTION(Forward Recovery Logstream for VSAMU files)
      JOURNALNAME(DFHJ90) TYPE(MVS) STREAMNAME(SCSCFWR.VSAMU)
DEFINE JOURNALMODEL(BWFP#J91) GROUP(C51LGMOD)
DESCRIPTION(Forward Recovery Logstream for VSAMV files)
      JOURNALNAME(DFHJ91) TYPE(MVS) STREAMNAME(SCSCFWR.VSAMV)
DEFINE JOURNALMODEL(C51LGLOG) GROUP(C51LGMOD)
DESCRIPTION(Log of logs for VSAM recovery products)
      JOURNALNAME(DFHLGLOG) TYPE(MVS) STREAMNAME(SCSCFWR.DFHLGLOG)
```

7.3.4 Log Streams of the Type Block Write Not Forced

NAME of log stream

Make sure that the log stream name descriptive and unique throughout sysplex.

SIZE

Use the DFHLSCU utility.

INITSIZE

Use the DFHLSCU utility.

AVGBUFSIZE

Use the DFHLSCU utility or set AVGBUFSIZE(64000).

HIGHOFFLOAD

Set to HIGHOFFLOAD(80).

LOWOFFLOAD

Set to LOWOFFLOAD(0).

STG_SIZE

Use the DFHLSCU utility.

LS_SIZE

Set to the size of the journal in the prior release of CICS.

7.4 Temporary Storage Pool Sizing

Only structure definitions in CFRM policy are required for TS pools. See the *CICS System Definition Guide* for sizing of the temporary storage pools.

7.5 Activating Definitions

To activate the coupling facility definitions, the following tasks should be done.

1. Run the job to define the LOGR definitions.
2. Run the job to define the CFRM policy.
3. Activate the CFRM policy by using the following MVS command:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=policyname
```

7.6 MVS Commands

Familiarize yourself with the following MVS commands by reading the *MVS/ESA System Commands* manual:

```
DISPLAY XCF,COUPLE,TYPE=CFRM|ALL
DISPLAY XCF,STRUCTURE,STRNAME=structure_name|ALL
DISPLAY XCF,CF,CFNAME=ALL
SETXCF FORCE,STRUCTURE,STRNAME=(structure_name)
SETXCF START,POLICY,TYPE=CFRM,POLNAME=policyname
SETXCF START,REBUILD,STRNAME=(structure_name)
SETXCF START,ALTER,STRNAME=structure_name
SETXCF STOP,POLICY,TYPE=CFRM
SETXCF STOP,REBUILD,STRNAME=(structure_name)
SETXCF STOP,ALTER,STRNAME=structure_name
```

Chapter 8. Temporary Storage Data Sharing Servers

This chapter describes the new temporary storage data sharing feature of CICS TS.

It discusses the following topics:

- 8.1, "Setting Up the Server"
- 8.2, "Monitoring" on page 123
- 8.3, "Customization" on page 126

CICS temporary storage (TS) data sharing provides multiregion access to the nonrecoverable temporary storage queues for CICS TS. Temporary storage data sharing allows your CICS applications to access nonrecoverable temporary storage queues from multiple CICS regions running on any MVS image within a parallel sysplex.

8.1 Setting Up the Server

This section describes the five areas that must be looked at to set up temporary storage data sharing servers:

- Defining the authorized cross-memory (AXM) subsystem
- Coupling facility structures
- Temporary-storage server regions
- Connecting CICS regions
- Security access

8.1.1 Defining the AXM Subsystem

CICS regions using temporary storage data sharing access the temporary storage data sharing servers using cross-memory connection services. Authorized cross-memory (AXM) server environment services are defined using the MVS subsystem interface.

The AXM subsystem is normally defined in the IEFSSNxx member of SYS1.PARMLIB. This ensures that AXM system services are automatically made available at IPL. If the member is in the keyword parameter format, the entry is as follows:

```
SUBSYS SUBNAME(AXM) INITRTN(AXMSI)
```

For the positional parameter format, the entry is:

```
AXM,AXMSI
```

The following command dynamically defines the subsystem for the AXM services, avoiding the need to wait for an IPL when AXM is first installed.

```
SETSSI ADD,SUBNAME=AXM,INITRTN=AXMSI
```

If initialization of the subsystem fails for any reason (for example, because of an error in the command, or because AXMSI is not in a linklist library) MVS does not allow another attempt because the subsystem is then already defined. In this case, use a different subsystem name, such as AXM1, because it does not

rely on a specific subsystem name. If you start AXM successfully the first time, further attempts will be ignored.

8.1.2 Defining the Temporary Storage Pool Structure

Using temporary storage data sharing means replacing 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 QOR. Each temporary storage pool is defined, using MVS cross-system extended services (XES), as a keyed list structure in a coupling facility. This means you must define coupling facility resource manager (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. See Chapter 7 on Coupling Facility Preparation for sizing information. For an example of this utility, see member IXCCFRMP in the SYS1.SAMPLIB library, in *MVS/ESA Setting Up a Sysplex*.

The name of the list structure for a temporary storage data sharing pool is created by appending the temporary storage pool name to the prefix DFHXQLS_, giving DFHXQLS_poolname.

For example, if the pool name is TSQSPQA1 with an initial size of 500 KB and maximum size of 1000 KB, and coupling facilities named FACIL01 and FACIL02, then the following IXCMIAPU utility statements will define the temporary storage pool structure on the named coupling facilities:

```
STRUCTURE NAME(DFHXQLS_TSQSPQA1)
          SIZE(1000)
          INITSIZE(500)
          PREFLIST(FACIL01,FACIL02)
```

When defined, you must activate the CFRM policy using the MVS operator command SETXCF START.

Defining the CFRM policy statements for a list structure does not actually create the list structure. This is done by a temporary storage server during its initialization.

8.1.3 Defining Temporary Storage Server Regions

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. This invokes the queue server region program, DFHXQMN, which resides in an APF-authorized library.

DFHXQMN requires some startup parameters, of which the temporary storage pool name is mandatory. A SYSPRINT DD statement is required for the print file, and a SYSIN DD statement for the server parameters.

You can specify the DFHXQMN initialization parameters either in a SYSIN data set defined in the job control language, or in the PARM parameter on the EXEC statement.

The temporary storage server REGION parameter job control language needs to specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests, and can be calculated as follows:

$$REGION = (32 \text{ KB} \times b) + (100 \text{ KB} \times r) + \frac{(32 \text{ KB} \times b) + (100 \text{ KB} \times r)}{10}$$

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated abend with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate, so it is best to ensure that the REGION size is large enough to eliminate the risk.

The following is an example of a temporary storage server job control language procedure to replace a queue-owning region that has a SYSIDNT of PQA1:

```
//TSQSRVR PROC SYSIDNT=PQA1,
//          INDEX1='CICS.V5R1M0',
//          REG='64M',
//          OUTC='*'
//TSSERVER EXEC PGM=DFHXQMN,REGION=&REG,TIME=1440,
//          PARM=('POOL=TSQS&SYSIDNT')
//STEPLIB DD DSN=&INDEX1..SDFHAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=&OUTC
//SYSIN DD DSN=&INDEX1..SYSIN(TSQS&SYSIDNT),DISP=SHR
//          PEND
```

8.1.3.1 Parameters

Parameters are specified in the form KEYWORD=value. You can optionally specify keywords in mixed case to improve readability.

If you specify more than one parameter in the PARM field, or on the same SYSIN input line, the parameters must be separated by commas. Any text following one or more spaces is taken as a descriptive comment. Any parameter line starting with an asterisk or a space is assumed to be a whole-line comment.

You can enter some parameter keywords in more than one form, such as an abbreviation. The standard form of each keyword is generally the longest form of the first word shown.

The main parameters used are listed on the server print file during start-up.

The following parameters are all valid as initialization parameters (in the SYSIN file, or the PARM field), and some can be modified by the server SET command.

Primary Parameters: Primary parameters are usually specified for all servers:

POOLNAME=pool_name

Specifies the name, of 1 to 8 characters, of the queue pool used to form the server name and the name of the coupling facility list structure DFHXQLS_poolname. This parameter is valid only at initialization, and must always be specified. This keyword can also be coded as **POOL**.

BUFFERS={100|number}

Specifies the number of queue buffers to allocate for the server address space. A queue index buffer holds a queue index entry plus up to 32 KB of queue data (for a small queue). When a READ or WRITE request completes, the queue index information is retained in the buffer. This can avoid the need to reread the queue index if the same queue is referenced from the same MVS image before the buffer has been reused.

If no buffer is available at the time of a request, the request is made to wait until one becomes free. The number of buffers should preferably be at least ten for each CICS region that can connect to the server in this MVS image. This avoids the risk of buffer waits.

Additional buffers may be used to reduce the number of coupling facility accesses by keeping recently used queue index entries in storage. In particular, if the current version of a queue index entry is in storage at the time a queue item is read, the request requires only one coupling facility access instead of two. If the current version of a queue index entry is in storage when a second or subsequent item is written to the same queue, the request requires only one coupling facility access instead of three.

It is not worth defining extra buffers beyond the point where this might cause MVS paging, as it is more efficient to reread the index entry than to page-in the buffer from auxiliary storage. This parameter is valid only at initialization. The valid range is from 1 to 999999. This keyword can also be coded as **BUF**.

FUNCTION={SERVER|UNLOAD|RELOAD}

Requests the server to perform the special functions UNLOAD or RELOAD. When the unload or reload processing has completed (normally or abnormally) the server program terminates. If this parameter is omitted, the server program initializes the cross-memory queue server environment.

STATSOPTIONS={NONE|SMF|PRINT|BOTH}

Specifies the statistics options that determine whether statistics are to be produced at intervals, and whether statistics are sent to SMF, the print file, or both. This keyword can also be coded as **STATSOPT**.

ENDOFDAY={00:00|hhmm}

Specifies the time when end-of-day statistics are to be collected and reset. If statistics options specify NONE, end-of-day statistics are written to the print file. The valid range is from 00:00 to 24:00. This keyword can also be coded as **EOD**.

STATSINTERVAL={3:00|hhmm}

Specifies the statistics interval, within the range of 1 minute to 24 hours. It is ignored if STATSOPTIONS=NONE. The valid range is from 00:01 to 24:00 (although it may be specified in seconds). This keyword can also be coded as **STATSINT**.

List Structure Parameters: The following parameters specify list structure attributes and are used only for initial allocation of the pool list structure, which occurs the first time a server is started for the pool:

POOLSIZE={ 0|number_of_bytes{K|M|G}}

Specifies the maximum amount of storage to be allocated for the list structure, expressed as kilobytes in the form nK, or megabytes in the form nM, or gigabytes in the form nG. This takes effect when the list structure is being created with a specified value of less than that specified for the list structure in the CFRM policy. The default value 0 specifies that no maximum limit is to be applied other than that specified in the CFRM policy. A nonzero value is generally rounded up by MVS to the next multiple of 256 KB. The valid range is from 0 to 2 GB.

MAXQUEUES={1000|number}

Specifies the maximum number of data lists to be reserved when the structure is allocated, which determines the maximum number of large

queues that can be stored in the structure. This number cannot be changed without reallocating the structure. Therefore, if the structure is being allocated at less than its maximum size, the MAXQUEUEUS value should be based on the maximum possible size of the structure rather than its initial size. The valid range is from 1 to 999999. This keyword can also be coded as **MAXQ**.

Debug Trace Parameters: These parameters are used only for intensive debug tracing. Using these options in a production environment may degrade performance significantly and cause the print file to grow very rapidly, using up spool space. Trace messages from cross-memory requests may be lost if they are generated faster than the trace print subtask can print them. In such cases, the trace indicates only how many messages were lost. The parameters are:

TRACECF={OFF|number}

Specifies OFF or ON for the coupling facility interface debug trace options. This option produces trace messages on the print file indicating the main parameters to the coupling facility request interface and the result from the IXLLIST macro. This keyword can also be coded as **CFTR** or **CFTRACE**.

TRACERQ={OFF|number}

Specifies OFF or ON for the queue request debug trace options. This option produces trace messages on the print file indicating the main parameters on entry to the shared queue request or shared queue inquire interface and the results on exit. This keyword can also be coded as **RQTR** or **RQTRACE**.

Tuning parameters: The following parameters are provided for tuning purposes. They are normally allowed to assume their default values:

ELEMENTSIZE={256|number}

Specifies the element size for structure space, which must be a power of 2. For current coupling facility implementations, there is no known reason to specify any other value than the default value of 256. This parameter is valid only at server initialization and is used only when the structure is first allocated. The valid range is 256 to 4096. This keyword can also be coded as **ELEMSIZE**.

ELEMENTRATIO={1|number}

Specifies the element side of the entry-to-element ratio when structure is first allocated. This determines the proportion of the structure space initially set aside for data elements. The ideal value for this ratio results from the average size of data for each entry being divided by the element size. However, the server automatically adjusts the ratio according to the actual entry and element usage. This parameter is valid only at server initialization, and is used only when the structure is first allocated. The valid range is from 1 to 255. This keyword can also be coded as **ELEMTRATIO**.

ENTRYRATIO={1|number}

Specifies the entry side of the entry-to-element ratio when the structure is first allocated. It determines the proportion of structure space initially to be set aside for list entry controls. It is not essential to specify this parameter because the server automatically adjusts the ratio based on actual usage to improve space utilization if necessary. This parameter is valid only at server initialization and is used only when the structure is first allocated. The valid range is from 1 to 255.

LASTUSEDINTERVAL={00:10|hhmm}

Specifies how often the last-used time for large queues is to be updated. For small queues, the time last used is updated on every reference. For large queues, updating the last-used time requires an extra coupling facility access, so that it is done only if the queue has not yet been accessed within the current time interval. This means that the last-used time interval returned by INQUIRE can be greater than the true value by an amount up to the value specified on this parameter. As the main purpose of the last-used time specification is to determine whether the queue is obsolete, an interval of a few minutes should be sufficient. The valid range is from 00:00 to 24:00 (although it may be specified in seconds). This keyword can also be coded as **LASTUSEDINT**.

SMALLQUEUEITEMS={9999|number}

Specifies the maximum number of items that can be stored in the small queue format in the queue index entry data area. This parameter can force a queue to be converted to large queue format if it has a large number of small items. It can be more efficient to write the items separately than to rewrite the whole small queue data area each time. The valid range is from 1 to 32767.

SMALLQUEUESIZE={32K|number}

Specifies the maximum data size for a small queue including the 2-byte prefix on each data item. Any queue exceeding this when writing the second or subsequent item to a queue is converted to the large queue format. This parameter can force queues to be converted to the large queue format at a size smaller than 32 KB. This is to prevent large amounts of data from being written to the small queue format. Performance improvements can result on systems where asynchronous coupling facility processing causes contention for hardware resources. On most systems, however, it is probably more efficient to defer conversion until the maximum size of 32 KB is reached. The valid range is from 4096 to 32768.

Warning parameters: These parameters modify the threshold at which warning messages and automatic ALTER actions occur when the structure becomes nearly full:

ELEMENTWARN={80|number}

Specifies the percentage of elements in use at which warnings and automatic ALTER actions should be first triggered. The valid range is from 1 to 100. This keyword can also be coded as **ELEMWARN**.

ELEMENTWARNINC={5|number}

Specifies the percentage increase (or decrease) of elements in use before the next warning should be triggered (reduced to 1 when next increase would otherwise reach 100). Additional messages are issued as the number of elements in use changes. The messages stop when the number of elements in use falls at least by this percentage below the initial warning level. The valid range is from 1 to 100. This keyword can also be coded as **ELEMWARNINC**.

ENTRYWARN={80|number}

Specifies the percentage of entries in use at which warnings and automatic ALTER actions should be first triggered. The valid range is from 1 to 100.

ENTRYWARNINC={5|number}

Specifies the percentage increase (or decrease) of entries in use before the next warning should be triggered (reduced to 1 when next increase would

otherwise reach 100). Additional messages are issued as the number of elements changes. The messages stop when the number of entries in use falls at by least the specified percentage below the initial warning level. The valid range is from 1 to 100.

Automatic ALTER parameters: Define the following parameters to modify the conditions under which the server attempts an automatic ALTER action when the structure becomes nearly full:

ALTERELEMMIN={100|number}

Specifies the minimum number of excess elements that must be present for an automatic ALTER to be issued to convert those elements to entries. The valid range is from 1 to 999999999.

ALTERELEMPC={1|number}

Specifies the minimum percentage of excess elements that must be present for an automatic ALTER to be issued to increase the proportion of entries. The valid range is from 0 to 100.

ALTERENTRYMIN={100|number}

Specifies the minimum number of excess entries that must be present for an automatic ALTER to be issued to convert those entries to elements. The valid range is from 0 to 999999999.

ALTERENTRYPC={1|number}

Specifies the minimum percentage of excess entries that must be present for an automatic ALTER to be issued to increase the proportion of elements. The valid range is from 0 to 100.

ALTERMININTERVAL={00:10|hhmm}

Specifies the minimum time interval to be left between automatic ALTER attempts when the structure is nearly full (above the element or entry warning level). The valid range is from 00:00 to 24:00. This keyword can also be coded as **ALTERMININT**.

8.1.3.2 Commands

Commands can be issued to control a queue server using the MVS MODIFY (F) command specifying the job or started task name of the server region. The general form of a queue server command is as follows:

```
F server,cmd parameter,parameter...  comments
```

The MVS STOP command is equivalent to issuing the server command STOP using the MVS MODIFY command.

The queue server supports the following commands:

SET keyword=value

Changes one or more server parameter values. This applies to all parameters other than those indicated as being for initialization only. The command can be abbreviated to T, as for the MVS SET command.

DISPLAY keyword

Displays one or more parameter values, or statistics summary information, on the console. The valid parameter keywords for DISPLAY and PRINT are described later in this section. The command can be abbreviated to D, as for the MVS DISPLAY command.

PRINT keyword

Produces the same output as DISPLAY but only on the print file.

STOP

Terminates the server, waiting for any active connections to terminate first, and preventing any new connections. The command can be abbreviated to P, as for the MVS STOP command.

CANCEL

Terminates the server immediately.

Figure 41 shows a STOP command issued to queue server SCSCPQA1 through SDSF.

```

Display Filter View Print Options Help
-----
SDSF DA SC52 SC52 PAG 0 SIO 0 CPU 8/ 8 COMMAND ISSUED
COMMAND INPUT ==> /F SCSCPQA1,STOP SCROLL ==> CSR
RESPONSE=SC52 DFHXQ0304I STOP command is waiting for connections to be
RESPONSE=closed. Number of active connections = 1.

```

Figure 41. STOP Command Issued to Server SCSCPQA1

The server also responds to XES events such as an operator SETXCF command to alter the structure size. If the server can no longer access the coupling facility, it automatically issues a server CANCEL command to close itself down immediately.

The DISPLAY or PRINT commands support the following keywords:

CONNECTIONS

Lists the job names and application IDs for the regions currently connected to this server. This keyword can also be coded as **CONN**.

BUFSTATS

Means queue index buffer pool statistics. This keyword can also be coded as **BUFST**.

CFSTATS

Refers to coupling facility interface I/O and response statistics. This keyword can also be coded as **CFST** or **STATSCF**.

POOLSTATS

Calls usage statistics for the pool list structure as a whole. This keyword can also be coded as **POOLST**.

STORAGESTATS

Means main storage allocation statistics for the server address space. This keyword can also be coded as **STGST**, **STGSTATS**, or **STORAGEST**.

PARAMETERS

Calls main parameter values. The keyword can also be coded **PARM**, **PARMS**, or **PARAM**.

ALLPARAMETERS

Refers to all parameter values. This keyword can also be coded as **ALLPARMS**.

STATISTICS

Refers to all available statistics. This keyword can also be coded as **STAT** or **STATS**.

INITIALIZED

Calls selected parameters and statistics whose values are usually displayed when initialization is complete. This keyword can also be coded as **INIT**.

Figure 42 shows a DISPLAY CONN command issued to queue server SCSCPQA1 from SDSF, and the subsequent response.

```
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY SCSCPQA1 STC28727 DSID 101 LINE COMMAND ISSUED
COMMAND INPUT ==> /F SCSCPQA1,DISPLAY CONN SCROLL ==> CSR
RESPONSE=SC52 DFHXQ0351I Connection: Job SCSCPAA1 App1 SCSCPAA1 Idle
RESPONSE=00:10:17
RESPONSE=SC52 DFHXQ0351I Connection: Job SCSCPAA2 App1 SCSCPAA2 Idle
RESPONSE=00:07:22
RESPONSE=SC52 DFHXQ0351I Connection: Job SCSCPFA1 App1 SCSCPFA1 Idle
RESPONSE=00:05:06
RESPONSE=SC52 DFHXQ0352I Queue pool TSQSPQA1 total active connections:
RESPONSE=3.
```

Figure 42. STOP Command Issued to Server SCSCPQA1

8.1.4 Connecting to CICS Regions

Enabling applications running in different CICS regions access to a temporary storage pool, you must define the temporary storage pool to those CICS regions, using the new DFHTST TYPE=SHARED macro. The DFHTST TYPE=SHARED macro specifies the remote system name by which the CICS region identifies a temporary storage pool in the coupling facility.

To define temporary storage pool TSQSPQA1 with a remote system name of PQA1, for example, the following temporary storage table provides the required connection mapping:

```
DFHTST TYPE=INITIAL,SUFFIX=A1
DFHTST TYPE=SHARED,SYSIDNT=PQA1,POOL=TSQSPQA1
DFHTST TYPE=FINAL
END
```

The connection between a CICS region and a server of a temporary storage pool is logged by means of a message. For example, if a CICS region connects to a server of a temporary storage pool named TSQSPQA1, the following message indicates that the connection has been opened.

```
AXMSC0031I Connection to server DFHXQ.TSQSPQA1 has been opened.
```

8.1.5 Defining Security Access

RACF, or an equivalent external security manager, controls permission for CICS regions to access a temporary storage pool.

The security checks are to ensure that:

- The temporary storage server is authorized to access the structure in the coupling facility.
- The temporary storage server is authorized to act as a server for the temporary storage pool.

- The application-owning region issuing the request is authorized to attach to the temporary storage server.

The temporary storage server does not perform security checks on individual requests. The AOR continues to be responsible for resource security checks if you need to control user access to temporary storage queues.

8.1.5.1 Authorizing Access to the Coupling Facility List Structure

You must authorize the temporary storage server region to use the coupling facility list structure used for its own temporary storage pool. This requires that the temporary storage server user ID have ALTER authority to a coupling facility resource management (CFRM) RACF profile called IXLSTR.structure_name in the FACILITY general resource class.

If the user ID of the server is DFHXQQA1, and the list structure is called DFHXQLS_TSQSPQA1, for example, the following RACF commands define the profile and provide the required access:

```
RDEFINE FACILITY IXLSTR.DFHXQLS_TSQSPQA1 UACC(NONE)
PERMIT IXLSTR.DFHXQLS_TSQSPQA1 CLASS(FACILITY) ID(DFHXQQA1) ACCESS(ALTER)
```

To reduce security administration, use the same temporary storage server user ID to start each temporary storage server that supports the same temporary storage pool.

8.1.5.2 Authorizing a Temporary Storage Server for a Temporary Storage Pool

Give the temporary storage server's user ID CONTROL access to the CICS RACF profile called DFHXQ.poolname in the FACILITY general resource class. You must do this to authenticate the temporary storage server as an authorized server for the named temporary storage pool.

If the user ID of the server is DFHXQQA1, and the pool name is TSQSPQA1, for example, the following RACF commands define the profile and provide the required access:

```
RDEFINE FACILITY DFHXQ.TSQSPQA1 UACC(NONE)
PERMIT DFHXQ.TSQSPQA1 CLASS(FACILITY) ID(DFHXQQA1) ACCESS(CONTROL)
```

8.1.5.3 Authorizing CICS Regions Access to the Temporary Storage Server

You can control access by CICS regions to the temporary storage servers. A security check is made against the CICS region user ID to authenticate the region's authority to use the services of a temporary storage server. This check is made the first time that a CICS region connects to a temporary storage server.

Each CICS region that connects to a temporary storage server user ID needs UPDATE access to the CICS RACF profile called DFHXQ.poolname in the FACILITY general resource class. Such access authenticates the CICS region's authority to use the services of the temporary storage server for the named temporary storage pool.

If the user ID of a CICS region is SCSCPAA1, and the pool name is TSQSPQA1, for example, the following RACF commands define the profile and provide the required access:


```
RDEFINE FACILITY DFHXQ.TSQSPQA1 UACC(NONE)
PERMIT DFHXQ.TSQSPQA1 CLASS(FACILITY) ID(SCSCPAA1) ACCESS(UPDATE)
```

8.2 Monitoring

In this section, we cover methods of acquiring information on temporary storage data sharing.

8.2.1 CEMT Transaction

You can inquire as to what queues exist in a temporary storage pool by using the CEMT transaction. Figure 43 shows a list of queues that exist in temporary storage pool TSQSPQA1 at the time of the CEMT transaction.

```
INQUIRE TSQUEUE ALL SYSID(PQA1)
STATUS: RESULTS
Tsq(TSQUEUE1) Num(00007) Len(0000000070) Aux Tra(CECI)
  Max(00010) Min(00010) Las(00003476)
Tsq(TSQUEUE2) Num(00003) Len(0000000099) Aux Tra(CECI)
  Max(00036) Min(00027) Las(00003449)
Tsq(TSQUEUE3) Num(00012) Len(0000096024) Aux Tra(CECI)
  Max(08002) Min(05502) Las(00003093)

RESPONSE: NORMAL
PF 1 HELP      3 END      5 VAR      7 SBH 8 SFH 9 MSG 10 SB 11 SF
SYSID=PAA2 APPLID=SCSCPAA2
TIME: 13.55.08 DATE: 06.24.96
```

Figure 43. Using CEMT to Inquire as to All Queues in a Pool

8.2.2 Temporary Storage Server Statistics

Statistics are recorded as specified by the STATSOPTIONS parameter, or by the SET command for a temporary storage server. Figure 44 on page 124 shows an example of using the DISPLAY command from SDSF to see coupling facility statistics.

```

Display Filter View Print Options Help
-----
SDSF DA SC52 SC52 PAG 0 SIO 15 CPU 8/ 8 COMMAND ISSUED
COMMAND INPUT ==> /F SCSCPQA1,DISPLAY CFSTATS SCROLL ==>
RESPONSE=SC52
DFHXQ0431I Access statistics for CF structure DFHXQLS_TSQSPQA1:
Index:  Wrt Adjs  Writes  Reads  Deletes  Rereads  Read Adjs
        7         0       2       0         0         0
Data:   Creates  Writes  Reads  Deletes  Rereads  Rewrites
        0         0       0       0         0         0
Responses:  Asynch  Normal  Len err  Not fnd  Vers chk  List chk
          0         12       0       0         0         1
          List full  Str full  I/O err
          0         0         0
DFHXQ0303I DISPLAY command has been processed.

```

Figure 44. Coupling Facility Statistics

Figure 45 shows the result of the DISPLAY command for pool statistics.

```

Display Filter View Print Options Help
-----
SDSF DA SC52 SC52 PAG 0 SIO 13 CPU 9/ 7 COMMAND ISSUED
COMMAND INPUT ==> /F SCSCPQA1,DISPLAY POOLSTATS SCROLL ==> CSR
RESPONSE=SC52
DFHXQ0432I Queue pool statistics for CF structure DFHXQLS_TSQSPQA1:
Structure:  Size  Max size  Elem size  Queues:  Current  Highest
           1024K  1024K    256        4         4
Lists:     Total  In use  Max used  Control  Data
           1003   4       4         3        1
           100%  0%     0%       0%       0%
Entries:   Total  In use  Max used  Free  Min free
           637   17     17       620   0
           100%  3%     3%      97%   0%
Elements:  Total  In use  Max used  Free  Min free
           623   433   433     190   0
           100%  70%   70%     30%   0%

```

Figure 45. Pool Statistics

Figure 46 on page 125 shows the result of the DISPLAY command for buffer statistics.

```

Display Filter View Print Options Help
-----10
SDSF DA SC52 SC52 PAG 0 SIO 7 CPU 12/ 12 COMMAND ISSUED
COMMAND INPUT ==> /F SCSCPQA1,DISPLAY BUFSTATS SCROLL ==> CSR
RESPONSE=SC52
DFHXQ0511I Queue index buffer pool statistics:
  Buffers:      Total  Max used  Active  On LRU  Empty
                100      1         0       1       0
  Requests:    Gets    Puts     Keeps   Frees   Purges
                8       8        0       0       0
  Results:     Got hit  Got free  Got new  Got LRU  No buff
  (Get)        7       0        1       0       0
  Errors:     Not freed  No purge  Not owned  Waits: Pool lock  Buf lock
                0       0         0         0       0
DFHXQ0303I DISPLAY command has been processed.

```

Figure 46. Buffer Statistics

Figure 47 shows the storage statistics brought up by the DISPLAY command.

```

Display Filter View Print Options Help
-----
SDSF DA SC52 SC52 PAG 0 SIO 13 CPU 10/ 8 COMMAND ISSUED
COMMAND INPUT ==> /F SCSCPQA1,DISPLAY STORAGESTATS SCROLL ==> CSR
RESPONSE=SC52
AXMPG0004I Usage statistics for storage page pool AXMPGANY:
  Size  In Use  Max Used  Free  Min Free
  62032K  3840K   3844K   58192K  58188K
  100%    6%     6%     94%    94%
           Gets    Frees    Retries  Fails
           131     11       0        0
AXMPG0004I Usage statistics for storage page pool AXMPGLOW:
  Size  In Use  Max Used  Free  Min Free
  496K  12K    12K     484K  484K
  100%  2%     2%     98%   98%
           Gets    Frees    Retries  Fails
           3      0        0        0
DFHXQ0303I DISPLAY command has been processed.

```

Figure 47. Storage Statistics

8.2.3 Programs DFHSTUP and DFH0STAT

The statistics programs DFHSTUP and DFH0STAT provide the following information in the temporary storage section of their reports:

```

Shared Pools defined . . . . . : 1
Shared Pools currently connected . . . . : 1
Shared temporary storage read requests . : 0
Shared temporary storage write requests. : 20

```

8.3 Customization

The XTSERREQ exit allows you to intercept temporary storage API requests before any action has been taken on the request. For example, you can redirect the request to a temporary storage server instead of a queue-owning region by changing the SYSID of the API request.

The API requests affected are:

- EXEC CICS WRITEQ temporary storage
- EXEC CICS READQ temporary storage
- EXEC CICS DELETEQ temporary storage

For more information see the *CICS Customization Guide*

Chapter 9. CICS Utilities

This chapter describes the new or changed utilities supplied by CICS. It discusses the following topics:

- 9.1, "Monitoring Log Streams" on page 128
- 9.2, "Transaction Affinities Utility" on page 128
- 9.3, "New Transactions" on page 128
- 9.4, "Shutdown Assist Program" on page 129
- 9.5, "Recovery Manager Utility Program" on page 131
- 9.6, "Batch Access to RLS Data Sets" on page 134
- 9.7, "Trace Utility Print Program (DFHTU510)" on page 137
- 9.8, "New Enqueue Function" on page 143
- 9.9, "Resolving Enqueue Deadlocks" on page 145

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

- *CICS Transaction Server for OS/390 Operations and Utilities Guide*
- *CICS Transaction Server for OS/390 Migration Guide*
- *CICS Transaction Server for OS/390 CICS-Supplied Transactions*
- *CICS Transaction Server for OS/390 Performance Guide*

9.1 Monitoring Log Streams

Chapter 5, “CICS Logging and Journaling” on page 57 describes the CICS logging and journaling, the MVS system commands to display the log streams, and the CICS and MVS statistics on the log streams usage.

9.2 Transaction Affinities Utility

You may find it helpful to use the Transaction Affinities Utility in the following circumstances to determine whether any transactions in your applications use programming techniques that cause intertransaction affinity:

- In a CICSplex System Manager/ESA (CICSplex SM) environment, use it for workload balancing.
- In a dynamic transaction routing environment, use it for workload balancing.
- Use it with user application programs.
- Use it if you are planning to implement asynchronous processing using CICS function-request shipping or transaction isolation.

For detailed information about defining the resources needed by the utility, and about its individual components, see the *CICS Transaction Server for OS/390 Transaction Affinities Utility Guide*.

Note: The Transaction Affinity Utility detects transaction affinities in CICS TS regions only. If you want to detect affinities in earlier releases of CICS, such as CICS/ESA 4.1, you need to use the IBM CICS Transaction Affinities Utility MVS/ESA, program number 5696-582.

9.3 New Transactions

The CICS TS supplies the following new transactions:

CIND You can use CIND to:

- Produce a failure during synch point processing in the in-doubt window to test the effect of in-doubt conditions on applications.
- Cause shunted units of work to test programs that use SPI commands to inquire about, and change the characteristics of, shunted UOWs.
- Unshunt the units of work shunted by CIND.
- Test the effect of the new LOCKED condition on application programs.

CESD The purpose of CESD is to empty the system of user tasks in a controlled way for both normal and immediate shutdown, and to prevent CICS from becoming suspended for long periods when doing a shutdown because of long-running tasks that have not completed. CESD will ensure that the minimum number of RLS locks are retained by VSAM after CICS shutdown. See 9.4, “Shutdown Assist Program” on page 129 for more information.

- CFQS** This is the send transaction for handling quiesce and unquiesce requests. The transaction is started during CICS initialization as a long-running task. This is a system task which the user does not need to use directly.
- CFQR** This is the receive transaction for handling quiesce and unquiesce requests. The transaction is started during CICS initialization as a long-running task. This is a system task which the user does not need to use directly.

9.4 Shutdown Assist Program

You can use the shutdown assist transaction to help solve two of the problems that can arise when shutting down CICS:

- On a normal shutdown, CICS waits for all running tasks to finish before entering the second stage of shutdown. Long-running or conversational transactions can cause an unacceptable delay, or can require operator intervention.
- On an immediate shutdown, CICS does not allow running tasks to finish and back out is not performed until emergency restart. This can cause an unacceptable number of units of work to be shunted, with a consequent retention of locks.

The default shutdown assist program, DFHCESD, attempts to purge and back out long-running tasks. It ensures that as many tasks as possible commit or back out cleanly, enabling CICS to shut down in a controlled manner.

Tasks are purged in three steps; successive steps use increasingly stronger purge techniques and are invoked only if tasks refuse to disappear from the system. The three purge steps that DFHCESD moves through are:

1. Normal purge is issued for all remaining tasks.
2. VTAM is forced closed and IRC is closed immediately.
3. CICS is shut down using PERFORM SHUT IMMEDIATE. (This step does not cause the shutdown-assist transaction to run again.)

To check whether tasks are ending quickly enough, DFHCESD samples the number of tasks present in the system. It performs a purge operation, and moves on to the next step only if the number of tasks does not reduce to fewer than eight samples (normal shutdown) or four samples (immediate shutdown). After taking a sample, DFHCESD issues a delayed EXEC CICS START request for itself, passing the current sample count in a temporary storage queue record. The new invocation of DFHCESD also takes a sample, comparing it with the last sample from the temporary storage queue record. It then decides whether to carry out the purge operation and move to the next step, or to remain on the current step.

The information passed to DFHCESD in the temporary storage queue record is:

SDFN	Char(2)	Step to be performed (00,01,02,03)
SDXN	Char(4)	Task number of task that started shutdown
SDET	Bin(15)	Number of samples giving the value in SDNT
SDNT	Bin(31)	Number of tasks in the system at last sample

On the initial invocation, SDFN is '00', SDXN is set to the task number of the shutdown task, and SDNT and SDET are zero.

The DFHCESD processing sequence is as follows:

Initial step 00 – In step 00, if shutdown is NORMAL, DFHCESD puts out a message and waits for 2 minutes. It then issues a delayed start of CESD every 2 seconds.

- Every 2 seconds, the number of transactions in the system is sampled.
- If the number is unchanged over four or eight samples (depending on whether this is a normal or immediate shutdown), the first of the purge steps is taken.

Purge step 01 – The transaction dump data set is closed and the task purge command is issued for all remaining transactions, with messages giving details of each transaction still running and each unit of work shunted.

- Every 2 seconds, the number of transactions in the system is sampled.
- If the number is unchanged over four or eight samples, purge step 02 is taken.

Purge step 02 – Unless VTAM persistent-sessions support is being used (that is, unless the persistent-session delay interval is set to a value greater than zero), VTAM is forced closed and IRC is closed immediately.

- Every 2 seconds, the number of transactions in the system is sampled.
- If the number is unchanged over four or eight samples, purge step 03 is taken.

Purge step 03 – CICS is shut down abnormally, with messages giving details of each transaction still running and each unit of work shunted.

Figure 48 on page 131 presents a sample system log showing the CESD activity for a CICS shutdown.


```

+DFHTM1715 SCSCPA3 CICS for MVS/ESA is being quiesced by userid
CICSUSER in transaction CEMT at netname SJA2045D.
+DFHDM01021 SCSCPA3 CICS is quiescing.
+DFHTM1781 SCSCPA3 CICS shutdown cannot complete because some
non-system user tasks have not terminated.
+DFHCESD SCSCPA3 SHUTDOWN ASSIST TRANSACTION CESD STARTING. SHUTDOWN
IS NORMAL.
+DFHCESD SCSCPA3 THERE ARE NOW 0002 TASKS STILL IN THE SYSTEM.
+DFHDU03031 SCSCPA3 Transaction Dump Data set DFHMMPA closed.
+DFHCESD SCSCPA3 PURGING TRANID CEMT, TERMD 045C, USERID CICSUSER,
TASKNO 000027
+DFHZC2316 SCSCPA3 VTAM ACB is closed
+DFHCESD SCSCPA3 THERE ARE NOW 0001 TASKS STILL IN THE SYSTEM.
+DFHTM17821 SCSCPA3 All non-system tasks have been successfully
terminated.
+DFHTM17111 SCSCPA3 About to link to phase 2 PLT programs.
+DFHTM17121 SCSCPA3 Control returned from phase 2 PLT programs.
+DFHFC0563 SCSCPA3 The RLS control ACB has been successfully unregistere
d by CICS.
+DFHRM0204 SCSCPA3 There are no indoubt, commit-failed or backout-faile
d UOWs.
+DFHRM0130 SCSCPA3 Recovery manager has successfully quiesced.
+DFHKE1799 SCSCPA3 TERMINATION OF CICS FOR MVS/ESA IS COMPLETE.
IEF4041 SCSCPA3 - ENDED - TIME=15.59.34
-SCSCPA3 ENDED. NAME-
$HASP395 SCSCPA3 ENDED
+DFHTM17821 SCSCPA3 All non-system tasks have been successfully
terminated.
+DFHTM17111 SCSCPA3 About to link to phase 2 PLT programs.
+EYUNX0024E SCSCPA3 ASSIGN failed: RESP=000016, RESP2=000005
+EYUNX0012E SCSCPA3 RMAS initialization failed
+DFHTM17121 SCSCPA3 Control returned from phase 2 PLT programs.
+DFHFC0563 SCSCPA3 The RLS control ACB has been successfully unregistere
d by CICS.
+DFHRM0204 SCSCPA3 There are no indoubt, commit-failed or backout-faile
d UOWs.
+DFHRM0130 SCSCPA3 Recovery manager has successfully quiesced.
+DFHKE1799 SCSCPA3 TERMINATION OF CICS FOR MVS/ESA IS COMPLETE.
-SCSCPA3 ENDED. NAME-

```

Figure 48. Sample System Log Showing CESD Activity for a CICS Shutdown

9.5 Recovery Manager Utility Program

The recovery manager utility program (DFHRMUTIL) processes the global catalog data set. It can insert or modify the recovery manager autostart override record. Optionally, it can extract a subset of the catalog records to build a reduced new catalog for a cold start.

You can use the recovery manager utility program to:

- Set or reset the recovery manager autostart override record on the global catalog.
- Examine the setting of the autostart override record on the global catalog.
- Copy that part of the catalog needed for a cold start to a new global catalog.

If a new catalog is built using DFHRMUTL, CICS can perform only a cold start or an initial start with the new catalog. The performance of these starts will, however, be better than that of cold or initial start with a full catalog.

DFHRMUTIL sets a return code indicating if it has succeeded.

You can specify what you want the utility to do by supplying parameters in a single optional record in the input data set, SYSIN.

You may need to supply one or two CICS global catalog data sets:

DFHGCD The catalog from which a copy is extracted or, if no copy is being made, the one that contains the autostart override record.

NEWGCD The catalog that is cleared and receives the copy, if one is requested.

DFHRMUTL writes some or all of the following to the output data set, SYSPRINT:

- The input record from SYSIN
- Error messages
- A summary of the autostart override record found on the global catalog, DFHGCD
- Whether or not the DFHGCD catalog is a reduced copy from a previous run of DFHRMUTL
- The autostart override record that has been set on DFHGCD, or on NEWGCD if a copy is made

The catalogs DFHGCD and NEWGCD may be updated. If no copy is requested, DFHGCD may have an override record inserted or updated. If a copy is requested, then DFHGCD is unchanged, NEWGCD is cleared, and the copy and new override record are written to NEWGCD.

The sample job shown in Figure 49 on page 133 can be used to improve the performance of a cold start.

```

//DFHRMUTL JOB (999,POK),'CICS510',CLASS=A,MSGCLASS=T,
//          NOTIFY=&SYSUID
//*-----*
/* THIS JOB DELETES AND RECREATES
/* CICS GCD
//*-----*
//DELETE EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE CICS.V5R1M0.NEWPAA3.DFHGCD
SET MAXCC=0
/*
//DEFINE EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
/*-----*/
/* */
/* DEFINE A CICS GLOBAL CATALOG */
/* */
/*-----*/
DEFINE CLUSTER(NAME(CICS.V5R1M0.NEWPAA3.DFHGCD)-
INDEXED -
CYL(1 1)-
SHR(2)-
FREESPACE(10 10)-
REUSE -
VOLUMES(TOTCI2)) -
DATA(NAME(CICS.V5R1M0.NEWPAA3.DFHGCD.DATA)-
CISZ(8192)-
KEYS(28 0)) -
INDEX(NAME(CICS.V5R1M0.NEWPAA3.DFHGCD.INDEX))
/*-----*
/* THIS JOB INITIALIZES THE LCD DATA SET
/*-----*
//INITLCD EXEC PGM=DFHCCUTL
//STEPLIB DD DSN=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DFHLCD DD DSN=CICS.V5R1M0.PAA3.DFHLCD,DISP=SHR
/*
/*-----*
/* THIS JOB SETS THE OVERRIDE START FLAG
/*-----*
//DFHRMUTL EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSNAME=CICS.V5R1M0.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFHGCD DD DSNAME=CICS.V5R1M0.PAA3.DFHGCD,DISP=SHR
//NEWGCD DD DSNAME=CICS.V5R1M0.NEWPAA3.DFHGCD,DISP=SHR
//SYSIN DD *
SET_AUTO_START=AUTOCOLD,COLD_COPY
/*
/* IF (DFHRMUTL.RC=0) THEN
/* STEP TO BE PERFORMED IF RMUTIL SUCCEEDS
/*-----*
/* THIS JOB COPIES THE NEW GCD ON THE OLD GCD
/*-----*
//COPY EXEC PGM=IDCAMS
//DFHGCD DD DSN=CICS.V5R1M0.PAA3.DFHGCD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//NEWGCD DD DSNAME=CICS.V5R1M0.NEWPAA3.DFHGCD,DISP=SHR
//SYSIN DD *
REPRO INFILE(NEWGCD) OUTFILE(DFHGCD) REUSE
/*
// ENDIF

```

Figure 49. Sample Job to Improve Cold Start Performance

9.6 Batch Access to RLS Data Sets

Recoverable data sets opened by CICS in RLS mode can also be accessed by batch programs, provided the batch programs also open them in RLS mode and limit them to read-only processing. Batch programs cannot update recoverable data sets in RLS mode. SMSVSAM prevents batch programs from opening a recoverable data set for update in RLS mode. Thus, to update a recoverable data set from a batch program, you must first ensure that the data set is closed to all CICS regions. This allows the batch program to open the data set in non-RLS mode for update. To support this requirement, VSAM RLS provides a data-set quiesce function (and an unquiesce function).

The quiesce function enables you, with a single command, to close any data sets that are open in RLS mode throughout the sysplex and to do it in an orderly manner. It also prevents the data sets from being opened in RLS mode while they are in the quiesced state. This function is required in the data sharing environment because, unlike earlier CICS releases, many CICS TS regions can have the same data set open for update at the same time. Using the quiesce function, you can take a data set offline throughout the parallel sysplex in the following situations:

- To support switching between RLS and non-RLS VSAM access modes
- To prevent data set access during forward recovery.

The VSAM RLS quiesce mechanism causes all CICS regions in the parallel sysplex to close any RLS ACBs that are open against a specified data set. After they have been closed under the quiesce mechanism, data sets can be opened only in non-RLS mode. To reenable quiesced data sets to be reopened in RLS mode, all open non-RLS ACBs must be closed, and then the data sets must be unquiesced.

Note: The quiesce mechanism cannot inform batch programs whose data set is open in RLS access mode about the quiesce request. If you have such programs, you should use the DFSMS SHCDS LIST subcommands to check whether any non-CICS jobs have ACBs open in RLS mode against the data set. For information about the SHCDS LIST subcommand, see DFSMS/MVS Version 1, Release 3, *Access Method Services for the Integrated Catalog Facility*.

Quiescing a data set sets the quiesce flag in the ICF catalog so that the data set can be opened in non-RLS mode only. This is the recommended way of making data sets available for batch programs. However, even if a data set has been quiesced, you still cannot open it for update in non-RLS access mode if SMSVSAM is holding retained locks against the data set. This is because the locks are needed to preserve data integrity: they protect changes that are waiting to be either committed or backed out.

9.6.1 Sample Programs to Allow the Switch from RLS to Non-RLS

CICS-TS provides a suite of eight sample application programs that are designed to help you to automate your batch preparation procedures for data sets that are opened in RLS mode.

You can use these sample programs unmodified, or you can use them as a basis for writing your own programs. The programs are DFH0BAT1 through DFH0BAT8.

Before attempting to run your batch jobs, you should ensure that:

- No retained locks are held for the data sets.
- No files are open against the data sets in RLS mode.

The sample programs, using the INQUIRE DSNAME, INQUIRE UOWDSNFAIL, and SET DSNAME SPI commands, help you to deal with any retained locks. When you have successfully dealt with these, you can quiesce the data sets to the RLS ACBs using the SPI or CEMT commands.

Three of the programs are coordinating programs, which use CICS distributed program link (DPL) commands to run programs on a set of nominated CICS regions. The following is a summary of these three coordinating programs:

DFH0BAT1 is invoked by transaction BAT1 in the CICS region selected as the coordinator, controlling the disabling of specified transactions. It reads two extrapartition TD queues:

- BATX for the IDs of transactions to be disabled.
- BATA for the application IDs of the target CICS regions.

It issues DPL requests to DFH0BAT4 in each of the target regions to disable the named transactions. Any error returned by the DFH0BAT4 is displayed using BMS map DFH0BM1.

DFH0BAT2 is invoked by transaction BAT2 in the CICS region selected as the coordinator, controlling the gathering of retained lock information for the specified data sets. It reads two extrapartition TD queues:

- BATD for the names of data sets.
- BATA for the application IDs of the target CICS regions.

For each data set, DFH0BAT2 issues a DPL request to DFH0BAT7, in each target CICS region, to retry back-out failures associated with the data set. When the DPL requests to DFH0BAT7 for a data set are completed, the program issues DPL requests to DFH0BAT5 to gather retained-lock information from each of the target CICS regions for the same data set. The retained-lock information from DFH0BAT5 invocations is received in temporary storage queue (DFH0BQ2) and is displayed using BMS mapset DFH0BM2.

This process of issuing DPL requests to DFH0BAT7 and DFH0BAT5 is repeated for each of the data set names obtained from BATD.

DFH0BAT3 is invoked by transaction BAT3 in the CICS region chosen to be the coordinator region. It initiates the forced back-out of any in-doubt units of work, and the forced release of retained locks where back-out fails for specified data sets. It reads two extrapartition TD queues:

- BATD for the names of data sets.
- BATA for the application IDs of the target CICS regions.

For each data set, DFH0BAT3 issues a DPL request to DFH0BAT6, in each target CICS region, to force the back-out of in-doubt units of work associated with the data set. When the DPL requests to DFH0BAT6 for a data set are completed, the program issues DPL requests to DFH0BAT8 to force the release of backup-failed retained locks in each target region for the same data set. Messages from the

DFH0BAT6 and DFH0BAT8 invocations are displayed using BMS mapset DFH0BM3.

This process of issuing DPL requests to DFH0BAT6 and DFH0BAT8 is repeated for each of the data set names obtained from BATD.

DFH0BAT4 is linked by DPL request from DFH0BAT1 to disable specified transactions.

DFH0BAT5 is linked by DPL request from DFH0BAT2 to gather and return retained lock information to its caller.

DFH0BAT6 is linked by DPL request from DFH0BAT3 to force the back-out of in-doubt units of work.

DFH0BAT7 is linked by DPL request from DFH0BAT3 to retry any failed back-outs.

DFH0BAT8 is linked by DPL request from DFH0BAT3 to force the release of retained locks (back-out failures).

9.6.2 Installation and Customization

The resource definitions for the sample programs are supplied in the CSD in group DFH\$BAT. Add this group to one of the group lists you use at CICS startup on a cold start, or install the group while CICS is running using the CEDA install command.

The three coordinating programs require input from extrapartition transient data queues. These TD queues provide the parameters the sample programs need. The TD queues and the parameters they hold are:

BATA	The application IDs of the CICS regions involved in the quiesce operation
BATX	The transaction IDs of any transactions that are to be disabled
BATD	The data set names that are to be quiesced

To prepare these TD queues and the control information:

- Define the sequential data sets for the TD queues as fixed block data sets with an 80 byte block size. You can define these either in the CSD (the preferred method) or in the DCT.
 - If you define the queues in the CSD, specify the data set names for dynamic allocation; you do not need DD statements in the startup job control language. As a consequence of dynamic allocation, when a TD queue is closed, the underlying data set is deallocated, which means that it can then be modified by, say, a TSO editor. As a result, data sets can be modified without having to shut down, which is not possible if you use the DCT, where DD statements are required in the CICS startup JCL.
 - If you define the queues in the DCT, you must include the necessary DD statements for ddnames BATA, BATX, and BATD.
- Dynamic allocation allows you to use the TSO editor to enter the data into the data sets before they are dynamically allocated when the TD queues are opened.

Note: These definitions and TD queues need only be available to the CICS region you select to be the coordinator. They do not need to be defined to the

target CICS regions. The queue names are coded in the programs, but you can change them to conform to your own naming conventions.

If you have a requirement that CICS should maintain read-only access to recoverable data sets during the batch window, the only recommended procedure is as follows:

- Resolve retained locks.
- Quiesce the data sets.
- Redefine the files as non-RLS and read-only in all relevant CICS systems.
- Open the files as non-RLS, read-only in CICS.
- Concurrently, run batch non-RLS.

When the batch finishes:

- Close the read-only non-RLS files in CICS.
- Set the files back to RLS access and update.
- Unquiesce the data sets.
- Open the files in CICS, if not using open at first reference.
- Resume normal running.

In practice, there would also be a need to take data set copies for integrity purposes, but these steps have been omitted for simplicity. (Data set copies are, after all, required regardless of whether RLS is being used by CICS.)

9.7 Trace Utility Print Program (DFHTU510)

The trace utility print program is DFHTU510.

DFHTU510 extracts all or selected trace entries from the A or B auxiliary trace data set, and formats and prints the data.

9.7.1 Trace Format

The Trace utility program supplied by CICS supports the following way to format the trace entries:

ABBREV Indicates that you require the abbreviated form of trace print, one line per entry.

SHORT Indicates that you require the short formatted print of the data in each entry. This consists of the information in the abbreviated format entry, and the following elements from the interpretation string of the fully formatted entry:

- Interpreted parameter list
- Return address
- Time
- Interval

Note: This trace format has been introduced by CICS TS.

FULL Indicates that you want a fully formatted print of all the data in each entry. This is the default.

Figure 50 on page 138 shows a sample job to format the trace entries.

```
//PRTRRA JOB (999,POK),'CICS510',CLASS=A,MSGCLASS=T,  
//      NOTIFY=&SYSUID  
//PRTRR EXEC PGM=DFHTU510  
//STEPLIB DD DSN=CICS.V5R1MO.SDFHLOAD,DISP=SHR  
//DFHAUXT DD DSN=CICS.V5R1MO.PAA3.DFHAUXT,DISP=SHR  
//DFHAXPRT DD SYSOUT=*  
//DFHAXPRM DD *  
control statements  
/*
```

Figure 50. Sample Job to Format Trace Entries

Figure 51 shows a sample abbreviated trace output.

```

SELECTIVE TRACE PRINT PARAMETERS:

  ABBREV
  ALL

CICS FOR MVS/ESA - AUXILIARY TRACE FROM 07/01/96 - APPLID SCSCPA3 - TIME OF FIRST ENTRY ON THIS PAGE 12:45:04.7185783762 PAGE 00002

DM 1 KE 0101 KETI  ENTRY INQ_LOCAL_DATETIME_DECIMAL                                =000001=
DM 1 KE 0102 KETI  EXIT  INQ_LOCAL_DATETIME_DECIMAL/OK 07011996,124504,718590,MMDDYYYY    =000002=
DM 1 KE 0401 KEGD  ENTRY INQUIRE_KERNEL                                          =000003=
DM 1 KE 0402 KEGD  EXIT  INQUIRE_KERNEL/OK      SCSCPA3                          =000004=
DM 1 TR 0202 TRSR  EXIT  START_AUXILIARY_TRACE/OK                                =000005=
DM 1 TR 0002 TROM  EXIT  INITIALISE_DOMAIN/OK                                       =000006=
DM 1 DM 0009 DMDS  DOMAI CALL      RETURN INITIALISE_DOMAIN,OK                    =000007=
DM 1 LM 0003 LMLM  ENTRY LOCK      0623AA38,EXCLUSIVE                            =000008=
DM 1 LM 0004 LMLM  EXIT  LOCK/OK                                                =000009=
DM 1 DM 0010 DMWQ  ENTRY RESUME_PHASE_WAITERS  B,A00,300,06437610,06437630,06437628    =000010=
DM 1 DS 0004 DSSR  ENTRY RESUME      004F0001                                  =000011=
DM 1 DS 0005 DSSR  EXIT  RESUME/OK                                              =000012=
DM 1 SM 0301 SMGF  ENTRY FREEMAIN      06314860 , 00000027,06437910,28             =000013=
DM 1 SM 0302 SMGF  EXIT  FREEMAIN/OK                                           =000014=
DM 1 DS 0004 DSSR  ENTRY RESUME      003D0001                                  =000015=
DM 1 DS 0005 DSSR  EXIT  RESUME/OK                                           =000016=
DM 1 SM 0301 SMGF  ENTRY FREEMAIN      06314860 , 00000027,06437790,28             =000017=
DM 1 SM 0302 SMGF  EXIT  FREEMAIN/OK                                           =000018=
DM 1 DM 0011 DMWQ  EXIT  RESUME_PHASE_WAITERS/OK                                =000019=
DM 1 LM 0003 LMLM  ENTRY UNLOCK      0623AA38,EXCLUSIVE                            =000020=
DM 1 LM 0004 LMLM  EXIT  UNLOCK/OK                                             =000021=
DM 1 KE 0301 KEIN  ENTRY DELETE_TASKS                                          =000022=
DM 1 KE 0302 KEIN  EXIT  DELETE_TASKS/OK                                       =000023=
DM 1 SM 0301 SMGF  ENTRY FREEMAIN      06302D7C , 00000014,00054000,1000          =000024=
DM 1 SM 0302 SMGF  EXIT  FREEMAIN/OK                                           =000025=
DM 1 DS 0013 DSKE  EXIT  TASK_REPLY/OK                                         =000026=
DM 1 DS 0005 DSSR  EXIT  SUSPEND/OK                                           =000027=
DM 1 LM 0003 LMLM  ENTRY LOCK      0623AA38,EXCLUSIVE                            =000028=
DM 1 LM 0004 LMLM  EXIT  LOCK/OK                                               =000029=
DM 1 DS 0004 DSSR  ENTRY DELETE_SUSPEND  004F0001                                  =000030=
DM 1 DS 0005 DSSR  EXIT  DELETE_SUSPEND/OK                                       =000031=
DM 1 DM 0011 DMWQ  EXIT  SET_UP_WAIT/OK    0000000818600002                       =000032=
DM 1 LM 0003 LMLM  ENTRY UNLOCK      0623AA38,EXCLUSIVE                            =000033=
DM 1 LM 0004 LMLM  EXIT  UNLOCK/OK                                             =000034=
DM 1 DM 0002 DMDM  EXIT  WAIT_PHASE/OK                                         =000035=
DM 1 TI 0100 TISR  ENTRY INQUIRE_EXPIRATION_TOKEN                              =000036=
DM 1 TI 0101 TISR  EXIT  INQUIRE_EXPIRATION_TOKEN/OK 0649A000 , 00000000          =000037=
DM 1 DM 0001 DMDM  ENTRY WAIT_PHASE      400,INITIALISING                          =000038=
DM 1 LM 0003 LMLM  ENTRY LOCK      0623AA38,EXCLUSIVE                            =000039=
DM 1 LM 0004 LMLM  EXIT  LOCK/OK                                               =000040=
DM 1 DM 0010 DMWQ  ENTRY RESUME_PHASE_WAITERS  3,400,300,06437610,06437630,06437628    =000041=
DM 1 DM 0011 DMWQ  EXIT  RESUME_PHASE_WAITERS/OK                                =000042=
DM 1 DM 0010 DMWQ  ENTRY SET_UP_WAIT      3,0,400,300,0,06437610,06437630,06437628    =000043=
DM 1 DS 0004 DSSR  ENTRY ADD_SUSPEND  DMWQUEUE                                  =000044=
DM 1 DS 0005 DSSR  EXIT  ADD_SUSPEND/OK    004F0003                                  =000045=
DM 1 SM 0301 SMGF  ENTRY GETMAIN      06314860 , 00000027,28,NO                    =000046=
DM 1 SM 0302 SMGF  EXIT  GETMAIN/OK     06437760                                  =000047=
DM 1 LM 0003 LMLM  ENTRY UNLOCK      0623AA38,EXCLUSIVE                            =000048=
DM 1 LM 0004 LMLM  EXIT  UNLOCK/OK                                           =000049=
DM 1 DS 0004 DSSR  ENTRY SUSPEND      004F0003,YES,MISC                          =000050=
DM 1 DS 0005 DSSR  EXIT  SUSPEND/OK                                           =000051=
DM 1 LM 0003 LMLM  ENTRY LOCK      0623AA38,EXCLUSIVE                            =000052=
DM 1 LM 0004 LMLM  EXIT  LOCK/OK                                              =000053=
CICS FOR MVS/ESA - AUXILIARY TRACE FROM 07/01/96 - APPLID SCSCPA3 - TIME OF FIRST ENTRY ON THIS PAGE 12:45:04.7224326262 PAGE 00003

```

Figure 51. Sample Abbreviated Trace

Figure 52 shows a short trace output sample.

```

SELECTIVE TRACE PRINT PARAMETERS:

SHORT

ALL

CICS FOR MVS/ESA - AUXILIARY TRACE FROM 07/01/96 - APPLID SCSCPA3                 PAGE 00002

DM  1 KE 0101 KETI  ENTRY INQ_LOCAL_DATETIME_DECIMAL                RET-861B4792 12:45:04.7185783762 **.****** =000001=

DM  1 KE 0102 KETI  EXIT INQ_LOCAL_DATETIME_DECIMAL/OK DECIMAL_DATE(07011996) DECIMAL_TIME(124504) DECIMAL_MICROSECONDS
                          (718590) FULL_DATE_FORMAT(MMDDYYYY)        RET-861B4792 12:45:04.7189228762 00.0003445000 =000002=

DM  1 KE 0401 KEGD  ENTRY INQUIRE_KERNEL                          RET-861B482C 12:45:04.7189567512 00.0000338750 =000003=

DM  1 KE 0402 KEGD  EXIT INQUIRE_KERNEL/OK SPECIFIC_APPLID(SCSCPA3) RET-861B482C 12:45:04.7189717512 00.0000150000 =000004=

DM  1 TR 0202 TRSR  EXIT START_AUXILIARY_TRACE/OK                RET-861FB8DE 12:45:04.7189857512 00.0000140000 =000005=

DM  1 TR 0002 TRDM  EXIT INITIALISE_DOMAIN/OK                    RET-8614C9D4 12:45:04.7190001262 00.0000143750 =000006=

DM  1 DM 0009 DMDS  DOMAI CALL RETURN - FUNCTION(INITIALISE_DOMAIN) RESPONSE(OK)
                                                                    RET-8610F2DE 12:45:04.7190168762 00.0000167500 =000007=

DM  1 LM 0003 LMLM  ENTRY LOCK_LOCK_TOKEN(0623AA38) MODE(EXCLUSIVE) RET-8614CA9C 12:45:04.7190518762 00.0000350000 =000008=

DM  1 LM 0004 LMLM  EXIT LOCK/OK                                  RET-8614CA9C 12:45:04.7191107512 00.0000588750 =000009=

DM  1 DM 0010 DMWQ  ENTRY RESUME_PHASE_WAITERS DOMAIN_TOKEN(B) PHASE(A00) PHASE_STATE(300) WAIT_QUEUE_HEAD_ADDRESS(06437610)
                          LOCK_TOKEN_ADDRESS(06437630) SUBPOOL_TOKEN_ADDRESS(06437628)
                                                                    RET-8614D276 12:45:04.7191912512 00.0000805000 =000010=

DM  1 DS 0004 DSSR  ENTRY RESUME_SUSPEND_TOKEN(004F0001)          RET-8614F8FE 12:45:04.7192181262 00.0000268750 =000011=

DM  1 DS 0005 DSSR  EXIT RESUME/OK                                RET-8614F8FE 12:45:04.7192522512 00.0000341250 =000012=

DM  1 SM 0301 SMGF  ENTRY FREEMAIN SUBPOOL_TOKEN(06314B60 , 00000027) ADDRESS(06437910) FREE_LENGTH(28)
                                                                    RET-8614F67E 12:45:04.7192815012 00.0000292500 =000013=

DM  1 SM 0302 SMGF  EXIT FREEMAIN/OK                              RET-8614F67E 12:45:04.7193236262 00.0000421250 =000014=

DM  1 DS 0004 DSSR  ENTRY RESUME_SUSPEND_TOKEN(003D0001)          RET-8614F8FE 12:45:04.7193482512 00.0000246250 =000015=

DM  1 DS 0005 DSSR  EXIT RESUME/OK                                RET-8614F8FE 12:45:04.7193710012 00.0000227500 =000016=

DM  1 SM 0301 SMGF  ENTRY FREEMAIN SUBPOOL_TOKEN(06314B60 , 00000027) ADDRESS(06437790) FREE_LENGTH(28)
                                                                    RET-8614F67E 12:45:04.7193917512 00.0000207500 =000017=

DM  1 SM 0302 SMGF  EXIT FREEMAIN/OK                              RET-8614F67E 12:45:04.7194140012 00.0000222500 =000018=

DM  1 DM 0011 DMWQ  EXIT RESUME_PHASE_WAITERS/OK                RET-8614D276 12:45:04.7194295012 00.0000155000 =000019=

DM  1 LM 0003 LMLM  ENTRY UNLOCK_LOCK_TOKEN(0623AA38) MODE(EXCLUSIVE) RET-8614CB1E 12:45:04.7194526262 00.0000231250 =000020=

DM  1 LM 0004 LMLM  EXIT UNLOCK/OK                                RET-8614CB1E 12:45:04.7194715012 00.0000188750 =000021=

DM  1 KE 0301 KEIN  ENTRY DELETE_TASKS                            RET-861417F4 12:45:04.7195103762 00.0000388750 =000022=

DM  1 KE 0302 KEIN  EXIT DELETE_TASKS/OK                        RET-861417F4 12:45:04.7203905012 00.0008801250 =000023=

DM  1 SM 0301 SMGF  ENTRY FREEMAIN SUBPOOL_TOKEN(06302D7C , 00000014) ADDRESS(00054000) FREE_LENGTH(1000)
CICS FOR MVS/ESA - AUXILIARY TRACE FROM 07/01/96 - APPLID SCSCPA3                 PAGE 00003

```

Figure 52. Sample Short Trace

Figure 53 shows a full trace output sample.

```

SELECTIVE TRACE PRINT PARAMETERS:

FULL

ALL

CICS FOR MVS/ESA - AUXILIARY TRACE FROM 07/01/96 - APPLID SCSCPAA3                PAGE 00002

KE 0101 KETI ENTRY - FUNCTION(INQ_LOCAL_DATETIME_DECIMAL)

TASK-DM KE_NUM-001B TCB-009CC650 RET-861B4792 TIME-12:45:04.7185783762 INTERVAL-**.***** =000001=
1-0000 00580000 00000043 00000000 00000000 A7040000 00000000 05000100 00000000 *.....X.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*

KE 0102 KETI EXIT - FUNCTION(INQ_LOCAL_DATETIME_DECIMAL) RESPONSE(OK) DECIMAL_DATE(07011996) DECIMAL_TIME(124504)
DECIMAL_MICROSECONDS(718590) FULL_DATE_FORMAT(MDDYYYY)

TASK-DM KE_NUM-001B TCB-009CC650 RET-861B4792 TIME-12:45:04.7189228762 INTERVAL-00.0003445000 =000002=
1-0000 00580000 00000043 00000000 00000000 A7040000 00000000 05000100 00000000 *.....X.....*
0020 00000000 F0F7F0F1 F1F9F9F6 F1F2F4F5 F0F4F7F1 F8F5F9F0 00000000 00000000 *...07011996124504718590.....*
0040 00000000 00000000 00000000 00000000 00000500 00000000 *.....*

KE 0401 KEGD ENTRY - FUNCTION(INQUIRE_KERNEL)

TASK-DM KE_NUM-001B TCB-009CC650 RET-861B482C TIME-12:45:04.7189567512 INTERVAL-00.0000338750 =000003=
1-0000 00680000 00000000 00000000 00000000 B8000000 00000000 01000000 00000000 *.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0060 00000000 00000000 *.....*

KE 0402 KEGD EXIT - FUNCTION(INQUIRE_KERNEL) RESPONSE(OK) SPECIFIC_APPLID(SCSCPAA3)

TASK-DM KE_NUM-001B TCB-009CC650 RET-861B482C TIME-12:45:04.7189717512 INTERVAL-00.0000150000 =000004=
1-0000 00680000 00000000 00000000 00000000 B8000000 00000000 01000100 E2C3E2C3 *.....SCSC*
0020 D7C1C1F3 E2C3E2C3 D7C1C1F3 C4C6C8C3 D3E34040 C4C6C8E2 C9E3F65B D7C1C1F3 *PAA3SCSCPAA3DFHCLT DFHSIT6$PAA3*
0040 00004E20 00000000 00000000 00000102 80005FA0 0000005D D8E72202 F5024040 *..+.....)QX..5.*
0060 40404040 4040001E *.....*

TR 0202 TRSR EXIT - FUNCTION(START_AUXILIARY_TRACE) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-861FB0DE TIME-12:45:04.7189857512 INTERVAL-00.0000140000 =000005=
1-0000 00300000 00000027 00000000 00000000 B0000000 00000000 05000100 00000000 *.....*
0020 00000000 00000000 00000000 00000000 *.....*

TR 0002 TRDM EXIT - FUNCTION(INITIALISE_DOMAIN) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614C9D4 TIME-12:45:04.7190001262 INTERVAL-00.0000143750 =000006=
1-0000 00500000 00000002 00000000 00000000 B0000000 00000000 05000100 00000000 *.&.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 *.....*

DM 0009 DMDS DOMAIN CALL RETURN - FUNCTION(INITIALISE_DOMAIN) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8610F2DE TIME-12:45:04.7190168762 INTERVAL-00.0000167500 =000007=
1-0000 00500000 00000002 00000000 00000000 B0000000 00000000 05000100 00000000 *.&.....*
0020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 *.....*

LM 0003 LMLM ENTRY - FUNCTION(LOCK) LOCK_TOKEN(0623AA38) MODE(EXCLUSIVE)
CICS FOR MVS/ESA - AUXILIARY TRACE FROM 07/01/96 - APPLID SCSCPAA3                PAGE 00003

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614CA9C TIME-12:45:04.7190518762 INTERVAL-00.0000350000 =000008=
1-0000 00480000 0000001B 00000000 00000000 B4100000 00000000 03000100 00000000 *.....*
0020 00000000 0623AA38 00000000 00000000 00000000 00000000 00000100 00000000 *.....*
0040 00000000 00000000 *.....*
2-0000 C4D4D3D6 C3D2D5D4 *DMLOCKNM *

LM 0004 LMLM EXIT - FUNCTION(LOCK) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614CA9C TIME-12:45:04.7191107512 INTERVAL-00.0000588750 =000009=
1-0000 00480000 0000001B 00000000 00000000 B4100000 00000000 03000100 00000000 *.....*
0020 00000000 0623AA38 00000000 00000000 00000000 00000000 00000100 00000000 *.....*
0040 00000000 00000000 *.....*
2-0000 C4D4D3D6 C3D2D5D4 *DMLOCKNM *

DM 0010 DMWQ ENTRY - FUNCTION(RESUME_PHASE_WAITERS) DOMAIN_TOKEN(B) PHASE(A00) PHASE_STATE(300) WAIT_QUEUE_HEAD_ADDRESS(06437610)
LOCK_TOKEN_ADDRESS(06437630) SUBPOOL_TOKEN_ADDRESS(06437628)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614D276 TIME-12:45:04.7191912512 INTERVAL-00.0000805000 =000010=
1-0000 00500000 00000020 00000000 00000000 B38E0000 00000000 04000000 00000000 *.&.....*
0020 00000000 00000000 0000000B 0A000300 00000000 00000000 06437610 06437630 *.....*
0040 06437628 00000000 00000000 00000000 *.....*

```

Figure 53 (Part 1 of 2). Sample Full Trace Output

```

DS 0004 DSSR ENTRY - FUNCTION(RESUME) SUSPEND_TOKEN(004F0001)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F8FE TIME-12:45:04.7192181262 INTERVAL-00.0000268750 =000011=
1-0000 00580000 00000014 00000001 00000000 84000000 00000000 05000000 00000000 *.....*
0020 00000000 004F0001 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 00000000 00000000 *.....*

DS 0005 DSSR EXIT - FUNCTION(RESUME) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F8FE TIME-12:45:04.7192522512 INTERVAL-00.0000341250 =000012=
1-0000 00580000 00000014 00000001 00000000 84000000 00000000 05000100 00000000 *.....*
0020 00000000 004F0001 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 00000000 *.....*

SM 0301 SMGF ENTRY - FUNCTION(FREEMAIN) SUBPOOL_TOKEN(06314860 , 00000027) ADDRESS(06437910) FREE_LENGTH(28)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F67E TIME-12:45:04.7192815012 INTERVAL-00.0000292500 =000013=
1-0000 00400000 0000000E 00000000 00000000 8C800000 00000000 0219016D 06314860 *.....*
0020 00000027 06437910 D3D4D8E4 C5E4C540 00000028 00000000 00000000 00000000 *.....LMQUEUE .....*
2-0000 C4D4E2E4 C2D7D6D3 *DMSUBPOL *

SM 0302 SMGF EXIT - FUNCTION(FREEMAIN) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F67E TIME-12:45:04.7193236262 INTERVAL-00.0000421250 =000014=
1-0000 00400000 0000000E 00000000 00000000 8C800000 00000000 0219016D 06314860 *.....*
0020 00000027 06437910 D3D4D8E4 C5E4C540 00000028 00000000 00000000 00000000 *.....LMQUEUE .....*
2-0000 C4D4E2E4 C2D7D6D3 *DMSUBPOL *

DS 0004 DSSR ENTRY - FUNCTION(RESUME) SUSPEND_TOKEN(003D0001)

CICS FOR MVS/ESA - AUXILIARY TRACE FROM 07/01/96 - APPLID CSCSPAA3 PAGE 00004

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F8FE TIME-12:45:04.7193482512 INTERVAL-00.0000246250 =000015=
1-0000 00580000 00000014 00000001 00000000 84000000 00000000 05000100 00000000 *.....*
0020 00000000 003D0001 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 00000000 00000000 *.....*

DS 0005 DSSR EXIT - FUNCTION(RESUME) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F8FE TIME-12:45:04.7193710012 INTERVAL-00.0000227500 =000016=
1-0000 00580000 00000014 00000001 00000000 84000000 00000000 05000100 00000000 *.....*
0020 00000000 003D0001 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
0040 00000000 00000000 00000000 00000000 00000000 *.....*

SM 0301 SMGF ENTRY - FUNCTION(FREEMAIN) SUBPOOL_TOKEN(06314860 , 00000027) ADDRESS(06437790) FREE_LENGTH(28)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F67E TIME-12:45:04.7193917512 INTERVAL-00.0000207500 =000017=
1-0000 00400000 0000000E 00000000 00000000 8C800000 00000000 0219016D 06314860 *.....*
0020 00000027 06437790 D3D4D8E4 C5E4C540 00000028 00000000 00000000 00000000 *.....LMQUEUE .....*
2-0000 C4D4E2E4 C2D7D6D3 *DMSUBPOL *

SM 0302 SMGF EXIT - FUNCTION(FREEMAIN) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614F67E TIME-12:45:04.7194140012 INTERVAL-00.0000222500 =000018=
1-0000 00400000 0000000E 00000000 00000000 8C800000 00000000 0219016D 06314860 *.....*
0020 00000027 06437790 D3D4D8E4 C5E4C540 00000028 00000000 00000000 00000000 *.....LMQUEUE .....*
2-0000 C4D4E2E4 C2D7D6D3 *DMSUBPOL *

DM 0011 DMWQ EXIT - FUNCTION(RESUME_PHASE_WAITERS) RESPONSE(OK)

TASK-DM KE_NUM-001B TCB-009CC650 RET-8614D276 TIME-12:45:04.7194295012 INTERVAL-00.0000155000 =000019=
1-0000 00500000 00000020 00000000 00000000 838E0000 00000000 04000100 00000000 *.....*
0020 00000000 00000000 0000000B 0A000300 00000000 00000000 06437610 06437630 *.....*
0040 06437628 00000000 00000000 00000000 *.....*

```

Figure 53 (Part 2 of 2). Sample Full Trace Output

9.8 New Enqueue Function

i2.new enqueue function

The main addition to the level of enqueue function provided by CICS TS is the ability to hold enqueues in retained state. Retained enqueues are analogous to RLS retained locks in that they signify that the resource is locked to a shunted unit of work.

Retained enqueues are held by the shunted UOW until such time as the failure, such as an in-doubt failure, can be resolved. This resolution may take some time so requests for such enqueues are rejected with a “locked” response instead of allowing the caller to wait for the enqueue.

Non-RLS file control uses retained enqueues to cope with back-out failures as well as in-doubt failures.

It is important to realize that retained enqueues are used only to lock local resources over an in-doubt or back-out failure. Unlike RLS, these resources can only be accessed by that local system and thus aren't accessible when the system is down. Enqueues aren't logged, so it is the responsibility of the resource owner to reacquire the correct enqueues when CICS is restarted.

Statistics are now kept about enqueue requests for the first time. Apart from containing information about retained enqueues, statistics are also kept about delays incurred by tasks waiting for enqueues.

From a serviceability point of view, it is now far easier to diagnose enqueue deadlocks. This can be done online, using CEMT, instead of having to chase QEA control blocks around a dump. The CEMT INQUIRE UOWENQ command shows information about the owners of every enqueue held and, more important, which tasks are waiting for these enqueues. See 9.9, “Resolving Enqueue Deadlocks” on page 145, Example 1. An EXEC CICS INQUIRE UOWENQ command provides this information to application programs.

NQ domain provides better system dump formatting routines. A summary of enqueue owners and items waiting in a queue is now provided.

9.8.1 Enqueue Pools

Each enqueue is allocated from a distinct enqueue pool. Enqueues allocated from one pool are completely separate from those allocated in any other pool.

Each enqueue pool has its own hash table from which the enqueues are located. In this respect, NQ domain is similar to the DFHKC enqueue service whose hash table was in the QCA. An enqueue pool can be thought of as a QCA. However, there are several enqueue pools in CICS TS, whereas previously there was only the single QCA.

The best way to consider an enqueue pool is as a logical extension to the enqueue argument. For example, if the string 'DIFFERENT' was an enqueue held in one pool, it would not prevent another task holding an enqueue on the string 'DIFFERENT' in another enqueue pool.

Enqueue pools are created during CICS initialization by those components that use the new enqueue service. Each pool has an eight-character name associated with it. Pools used in CICS are as follows:

EXECADDR	Used by the EXEC layer for EXEC ENQ requests made on addresses.
EXECSTRN	Used by the EXEC layer for EXEC ENQ requests made on strings.
FCDSSEWR	Used by File Control for VSAM ESDS write locks.
FCDSLDM	Used by File Control for VSAM load mode locks.
FCDSRECD	Used by File Control to lock records in VSAM/CMT data sets.
FCDSRNGE	Used by File Control for VSAM range locks.
FCFLRECD	Used by File Control to lock records in BDAM/UMT files.
FCFLUMTL	Used by File Control for UMT load locks.
JOURNALS	Used by the Logger to lock journal entries.
KCADDR	Used by the DFHKC macro compatibility layer for address enqueues.
KCSTRING	Used by the DFHKC macro compatibility layer for string enqueues.
TDNQ	Used by Transient Data to lock recoverable TD queues.
TSNQ	Used by Temporary Storage to lock recoverable temporary storage queues.

Enqueue pools have different characteristics. For example, not all enqueues are retained when their owning unit of work becomes shunted. Indeed EXEC CICS enqueues allocated from the EXECADDR and EXECSTRN enqueue pools are released when a shunt occurs (subject to maximum lifetime considerations).

Enqueue pool names also serve to provide useful diagnostic information. When a task is suspended waiting for an enqueue, then the name of the pool containing that enqueue is used as the resource-name on the suspend request. This is reflected back on the CEMT INQUIRE TASK interface and indicates the type of request the suspended task last made.

For example, Task 35 is shown to be suspended after making a transient data request and Task 36 after making an EXEC CICS ENQ request:

```
INQUIRE TASK
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(0000025) Tra(CEMT) Fac(T773) Run Ter Pri( 255 )
                               Sta(T0) Use(CICSUSER)
Tas(0000034) Tra(CECI) Fac(T775) Sus Ter Pri( 001 )
Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000018) Sta(T0) Use(CICSUSER)
Tas(0000035) Tra(CECI) Fac(T774) Sus Ter Pri( 001 )
Hty(ENQUEUE ) Hva(TDNQ   ) Hti(000007) Sta(T0) Use(CICSUSER)
Tas(0000036) Tra(CECI) Fac(T776) Sus Ter Pri( 001 )
Hty(ENQUEUE ) Hva(EXECSTRN) Hti(000007) Sta(T0) Use(CICSUSER)
```

The new CEMT INQUIRE UOWENQ can be used to determine the actual resources that these tasks are waiting for.

9.9 Resolving Enqueue Deadlocks

Five examples are given.

9.9.1 Example 1 of Resolving Enqueue Deadlocks

Deadlocks involving enqueues can now be resolved online using CEMT. For example, consider the following where Task 32 has suspended waiting for a file control enqueue.

```
INQUIRE TASK
STATUS: RESULTS - OVERTYPE TO MODIFY
Tas(0000025) Tra(CEMT) Fac(T773) Run Ter Pri( 255 )
                                     Sta(TO) Use(CICSUSER)
Tas(0000028) Tra(TDUP) Fac(T774) Sus Ter Pri( 001 )
Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000018) Sta(TO) Use(CICSUSER)

Tas(0000032) Tra(FUPD) Fac(T775) Sus Ter Pri( 001 )
Hty(ENQUEUE ) Hva(FCDSRECD) Hti(000006) Sta(TO) Use(CICSUSER)

Tas(0000033) Tra(CEMT) Fac(T776) Sus Ter Pri( 255 )
Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000684) Sta(TO) Use(CICSUSER)
Tas(0000035) Tra(FUPD) Fac(T784) Sus Ter Pri( 001 )
Hty(ENQUEUE ) Hva(FCDSRECD) Hti(000006) Sta(TO) Use(CICSUSER)
Tas(0000036) Tra(CECI) Fac(T777) Sus Ter Pri( 001 )
Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000019) Sta(TO) Use(CICSUSER)
Tas(0000039) Tra(FUPD) Fac(T778) Sus Ter Pri( 001 )
Hty(ENQUEUE ) Hva(FCDSRECD) Hti(000006) Sta(TO) Use(CICSUSER)
Tas(0000040) Tra(CEMT) Fac(T779) Run Ter Pri( 255 )
Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000018) Sta(TO) Use(CICSUSER)
Tas(0000042) Tra(FUP2) Fac(T783) Sus Ter Pri( 001 )
Hty(ENQUEUE ) Hva(FCDSRECD) Hti(000005) Sta(TO) Use(CICSUSER)
Tas(0000051) Tra(CECI) Fac(T785) Sus Ter Pri( 001 )
Hty(ZCIOWAIT) Hva(DFHZARQ1) Hti(000018) Sta(TO) Use(CICSUSER)
```

Suppose user running Task 32 complains that his or her terminal is locked out.

The CEMT INQUIRE TASK display shows Task 32 to be waiting on an enqueue (Hty=ENQUEUE). Moreover, we know that the task is waiting for a lock on a record in a dataset (Hva=FCDSRECD).

FCDSRECD is the name of one of the enqueue pools mentioned earlier.

9.9.2 Example 2 of Resolving Enqueue Deadlocks

CEMT INQUIRE UOWENQ can be used to show enqueues held in the system:

```
INQUIRE UOWENQ
STATUS: RESULTS
Uow(AA8E9505458D8C01) Tra(CEMT) Tas(0000025) Act Exe Own
Uow(AA8E9505458D8C01) Tra(TDUP) Tas(0000028) Act Tdq Own
Uow(AA8E950545DAC004) Tra(FUPD) Tas(0000032) Act Dat Own
Uow(AA8E950545DBC357) Tra(FUPD) Tas(0000035) Act Dat Wai
Uow(AA8E97FE9592F403) Tra(FUP2) Tas(0000039) Act Dat Wai
Uow(AA8E9505458D8C01) Tra(TSUP) Tas(0000034) Ret Tsq Own
Uow(AA8E97FE9592F403) Tra(FUP2) Tas(0000039) Act Dat Own
Uow(AA8E950545DAC004) Tra(FUPD) Tas(0000032) Act Dat Wai
```

```
Uow(AA8E97FE9592F403) Tra(FUPD) Tas(0000042) Act Dat Own
```

In a busy system, the CEMT display could be filtered by our Task 32

```
INQUIRE UOWENQ TASK(32)
STATUS: RESULTS
Uow(AA8E950545DAC004) Tra(FUPD) Tas(0000032) Act Dat Own
Uow(AA8E950545DAC004) Tra(FUPD) Tas(0000032) Act Dat Wai
```

In order to determine why Task 32 is locked out waiting for an enqueue, the new CEMT INQUIRE UOWENQ command can be used. It displays all the enqueues owners and items waiting in a queue in the system. Those tasks waiting for enqueues are displayed immediately after the owner of the enqueue they are waiting for.

The initial line display shows only a limited amount of information.

- Unit of work ID (Uowid)
- Transaction ID (Transid)
- Task ID (Taskid)
- State
- Enqueue type
- Relation

As shown later, each entry can be expanded to provide more information.

Enqueues in the following pools aren't returned by the INQUIRE UOWENQ command:

- JOURNALS
- KCADDR
- KCSTRING

This omission occurs because these enqueues are internal to CICS. The new interface only returns information relating to resources that are directly accessible via the EXEC interface.

As with other CEMT commands, filtering can be used to show only those enqueues we are interested in. In the example shown, the display is filtered to show only those entries relating to Task 32. As can be seen, Task 32 owns one enqueue and is waiting for another.

9.9.3 Example 3 of Resolving Enqueue Deadlocks

Expanding the first entry shows the queue the task owns:

```
INQUIRE UOWENQ TASK(32)
RESULT
Uowenq
Uow(AA8E950545DAC004)
Transid(FUPD)
Taskid(0000032)
State(Active)
Type(Dataset)
Relation(Owner)
Resource(ACCT.CICS510.ACCTFILE)
Qualifier(BOLAM)
Netuowid(..GBIBMIYA.IYA2T774.n.....)
Enqfails(00000000)
```


Expanding the second entry shows the queue the task is waiting for:

```
INQUIRE UOWENQ TASK(32)
RESULT
  Uowenq
  Uow(AA8E950545DAC004)
  Transid(FUPD)
  Taskid(0000032)
  State(Active)
  Type(Dataset)
  Relation(Waiter)
  Resource(INDX.CICS510.ACIXFILE)
  Qualifier(BOLAM)
  Netuowid(..GBIBMIYA.IYA2T774.n.....)
  Enqfails(00000000)
```

An expanded display can be selected for any entry by highlighting it and pressing **Enter**. Among the further information that is displayed are the RESOURCE and QUALIFIER fields, which identify the resource that the enqueue relates to. Different data appears in these fields depending upon the type of the enqueue.

Expanding the first entry shows that Task 32 owns the enqueue on record identifier BOLAM in the ACCT.CICS510.ACCTFILE dataset.

The second entry when expanded shows that Task 32 is also waiting for an enqueue, this time for record identifier BOLAM in the INDX.CICS510.ACIXFILE dataset.

9.9.4 Example 4 of Resolving Enqueue Deadlocks

Filtering by the RESOURCE and QUALIFIER of the enqueue being waited upon shows who owns that enqueue:

```
INQUIRE UOWENQ RESOURCE(INDX.CICS510.ACIXFILE) QUALIFIER(BOLAM)
STATUS: RESULTS
  Uow(AA8E97FE9592F403) Tra(FUP2) Tas(0000039) Act Dat Own
  Uow(AA8E950545DAC004) Tra(FUPD) Tas(0000032) Act Dat Wai
```

Filtering by Task 39 shows it, too, is waiting on an enqueue:

```
INQUIRE UOWENQ TASK(39)
STATUS: RESULTS
  Uow(AA8E97FE9592F403) Tra(FUP2) Tas(0000039) Act Dat Wai
  Uow(AA8E97FE9592F403) Tra(FUP2) Tas(0000039) Act Dat Own
```

Since we now know the resource that Task 32 is waiting for, we can use filtering again to determine which task has that resource locked. This time, filtering by the RESOURCE and QUALIFIER of the enqueue being waited upon, we determine that it is Task 39 that owns the resource.

The next step would be to determine why Task 39 has been holding this enqueue for such a long time. Filtering INQUIRE UOWENQ to display only those enqueues in which Task 39 has an involvement shows that it, too, is waiting for an enqueue.

9.9.5 Example 5 of Resolving Enqueue Deadlocks

Expanding the “waiter” entry shows the enqueue that Task 39 is waiting for:

```
INQUIRE UOWENQ TASK(39)
RESULT
  Uowenq
  Uow(AA8E97FE9592F403)
  Transid(FUP2)
  Taskid(0000039)
  State(Active)
  Type(Dataset)
  Relation(Waiter)
  Resource(ACCT.CICS510.ACCTFILE)
  Qualifier(BOLAM)
  Netuowid(..GBIBMIYA.IYA2T776.p.nk4..)
  Enqfails(00000000)
```

Filtering by the RESOURCE and QUALIFIER of this enqueue shows the deadlock to be between Tasks 32 and 39;

```
INQUIRE UOWENQ RESOURCE(ACCT.CICS510.ACCTFILE) QUALIFIER(BOLAM)
STATUS: RESULTS
  Uow(AA8E950545DAC004) Tra(FUPD) Tas(0000032) Act Dat Own
  Uow(AA8E950545DBC357) Tra(FUPD) Tas(0000035) Act Dat Wai
  Uow(AA8E97FE9592F403) Tra(FUP2) Tas(0000039) Act Dat Wai
```

Expanding the entry that shows Task 39 waiting for an enqueue shows it to be waiting for the lock on record identifier BOLAM in the ACCT.CICS510.ACCTFILE dataset. This was the enqueue that Task 32 owned, so it is Tasks 32 and 39 that are deadlocked.

This can be confirmed by filtering by the RESOURCE and QUALIFIER of the enqueue that Task 39 is waiting for. Apart from showing Task 32 to be the owner, it also indicates that another task has been drawn into the deadlock. Task 35 is also waiting for this enqueue.

All of the information available to the CEMT INQUIRE UOWENQ command is also available to other applications via the EXEC CICS INQUIRE UOWENQ browse command. Customers wishing to automate deadlock detection and resolution can use this command.

Chapter 10. Modifying SMS for RLS

In this chapter, we describe the tasks the Storage Administrator should perform to enable and maintain VSAM RLS. We assume that you have already installed DFSMS 1.3 and are familiar with the SMS constructs and classes. We do not provide a guide for implementing SMS.

10.1 Sharing Control Data Set (SHCDS)

The SHCDS is a logically partitioned VSAM linear data set. It 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, such as a SMSVSAM address space restart, or a CF lock structure failure.

An SHCDS contains the following information:

- The name of the CF lock structure in use
- The system status for each system or failed system instance
- The time that the system failed
- A list of subsystems and their status
- A list of open data sets using the CF
- A list of data sets with unbound locks
- A list of data sets in permit non-RLS state

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

- Allocate SHCDSs so that the number of active and spare data sets ensures the data is always duplexed. At a minimum, define and activate two SHCDSs and at least one spare SHCDS for recovery purposes. You should ensure that there are enough spare SHCDSs, since these are used when I/O errors occur on the active SHCDSs.
- Place the SHCDSs on volumes with global connectivity. VSAM RLS processing is only available on those systems that currently have access to the active SHCDS. The share options for SHCDSs must be set to (3,3) so that each system in the sysplex can properly share the data sets. See *DFSMS/MVS V1R3 Access Method Services for ICF* for more information on share options.

Use the following naming convention when defining your SHCDSs:

`SYS1.DFPSHCDS.qualifier.Vvolser`

where

qualifier is a 1 to 8 character qualifier.

volser is the volume serial number. The V prefix allows you to specify numeric volume serial numbers.

Use the following formula to calculate the size of your SHCDSs:

$$\text{Space} = 16 + \text{number_of_systems} \times \left(16 + \frac{\text{number_of_OPENS}}{10}\right) \text{KB}$$

where

Space is the space required for the SHCDS.

number_of_systems is the number of systems.

number_of_OPENS is the number of concurrent OPENS to the coupling facility that you expect.

So, if you have three systems and expect to have 2000 concurrent OPEN requests, your SHCDSs requires 664 kB of space for each one.

You can create an SHCDS using JCL, or using access method services or TSO subcommands. When you use the DEFINE command of access method services to create an SHCDS, specify SHAREOPTIONS(3,3) to ensure that the SHCDS can be written to and read from any system. Figure 54 is an example of how to create SHCDS using IDCAMS DEFINE.

```
//STEP01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE CLUSTER( -
    NAME(SYS1.DFPSHCDS.PRIMARY.VTOTSMA) -
    VOLUMES(TOTSMA) -
    MEGABYTES(10 10) -
    LINEAR -
    STORAGECLASS(GSPACE))
  DEFINE CLUSTER( -
    NAME(SYS1.DFPSHCDS.SECONDARY.VTOTCAT) -
    VOLUMES(TOTCAT) -
    MEGABYTES(10 10) -
    LINEAR -
    SHAREOPTIONS(3,3) -
    STORAGECLASS(GSPACE))
  DEFINE CLUSTER( -
    NAME(SYS1.DFPSHCDS.SPARE.VTOTSMS) -
    VOLUMES(TOTSMS) -
    MEGABYTES(10 10) -
    SHAREOPTIONS(3,3) -
    LINEAR -
    STORAGECLASS(GSPACE))
```

Figure 54. SHCDS Allocation Example

Place your SHCDSs in such a way as to maximize availability in the event of the loss of a volume. They can be either SMS or non SMS-managed. To make them SMS-managed, use storage classes defined with the guaranteed space attribute. Avoid placing SHCDSs on volumes for which there might be extensive volume reserve activity.

Ensure that the space allocation for active and spare SHCDSs is the same.

After you have defined the SHCDS data sets, you must activate them for SMSVSAM:

- Use the VARY SMS,SHCDS(SHCDS_name),NEW for primary and secondary SHCDSs.
- Use VARY SMS,SHCDS(SHCDS_name),NEWSPARE for the spare SHCDS.

The SHCDS data sets are independent of device type. The key to allocating them is that they are available to every SMSVSAM in the sysplex and defined with the proper VSAM share options (3,3). The size of the data sets is checked only when a new data set is added. The first data set sets the criteria. No data set will be allowed to be added that has less allocated space than the space currently in use. That means that if the first data set is allocated in 5 MB spaces, all of the rest of the data sets added must have at least 5 MB available for use in the primary allocation. Data sets can be larger, but not smaller. The device type is not considered. Space is processed and calculated in terms of pages and kilobytes.

For additional details on the VARY SMS commands, refer to 10.8, “New Storage Management Commands” on page 158 of this book.

10.2 Coupling Facility Structures

For DFSMS support of RLS, you need two types of coupling facility structures, cache and lock:

- Coupling facility cache structures provide a level of storage hierarchy between local memory and DASD cache. Each coupling facility cache structure is contained in a single coupling facility. You may have multiple coupling facilities and multiple coupling-facility cache structures.

You must define cache structures to MVS/SP and also define it in the SMS base configuration.

- The coupling facility lock structure is used to enforce the protocol restrictions for VSAM RLS data sets, and to maintain the record-level locks and other DFSMSdfp serializations.

You must define a single, nonvolatile, master coupling facility lock structure with the required name of IGWLOCK00.

Use CFRM policy definitions to specify an initial and maximum size for each coupling facility structure. DFSMS uses the initial structure size you specify in the policy each time it connects to a coupling facility cache structure. Figure 55 on page 152 is an example of the job control language used to define the lock and cache structure when CFRM policy CFRM08M is placed in the CFRM couple dataset.

```

//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSIN DD *
DATA TYPE(CFRM) REPORT(YES)
DEFINE POLICY NAME(CFRM08M ) REPLACE(YES)

CF NAME(CF01) DUMPSPACE(2048) PARTITION(1) CPCID(00)
TYPE(009672) MFG(IBM) PLANT(02) SEQUENCE(000000040104)

CF NAME(CF02) DUMPSPACE(2048) PARTITION(1) CPCID(01)
TYPE(009672) MFG(IBM) PLANT(02) SEQUENCE(000000040104)

STRUCTURE NAME(DSNDSGA_GBPO) SIZE(16000)
INITSIZE(8000)
PREFLIST(CF02, CF01)

STRUCTURE NAME(IEFAUTOS) SIZE(640)
REBUILDPERCENT(20)
PREFLIST(CF01, CF02)

STRUCTURE NAME(IRRXCF00_B001) SIZE(332)
PREFLIST(CF02, CF01)
EXCLLIST(IRRXCF00_P001)

STRUCTURE NAME(IGWLOCK00)
INITSIZE(14300) SIZE(28600)
PREFLIST(CF01,CF02) REBUILDPERCENT(75)

STRUCTURE NAME(CICS_CACHE)
INITSIZE(50000) SIZE(70000)
PREFLIST(CF01,CF02) REBUILDPERCENT(75)

STRUCTURE NAME(ISTGENERIC) SIZE(328)
PREFLIST(CF02, CF01)

STRUCTURE NAME(LOG_DFHLOG_001)
INITSIZE(3072) SIZE(4352)
PREFLIST(CF01,CF02) REBUILDPERCENT(75)

```

Figure 55. Policy CFRM08M Defined and Placed in the CFRM Coupling Data Set

Note: The IXCMIAPU utility does not have an UPDATE function. This means that each time you want to modify an existing policy, you have to run the IXCMIAPU utility respecifying all the coupling facilities and the structures, and adding or deleting definitions as needed.

To switch to the updated policy, issue the SETXCF START POLICY,TYPE=name,POLNAME=polname command. This command causes the system to copy your updated policy to the active policy.

For more information and examples on the IXCMIAPU utility, refer to *MVS/ESA Setting Up a Sysplex*.

10.3 SMS Configuration Changes

You must add coupling facility cache structures to the SMS base configuration for DFSMSdfp to use the coupling facility for VSAM RLS.

Associate these structures with a cache set name in the base configuration. The cache set name is also specified in a storage class definition. When a storage class associated with a data set contains a cache set name, the data set becomes eligible for VSAM record-level sharing and can be placed in a coupling facility cache structure associated with the cache set. The system selects the best cache structure within the cache set defined for the storage class.

To modify the SMS configuration use the Interactive Storage Management Facility (ISMF) panels:

1. Select **Option 8** from the ISMF Primary Option Menu for Storage Administrators, to invoke the Control Data Set (CDS) Application Selection panel.

Figure 56 shows the CDS Application Selection panel

```
Panel Utilities Help
-----
                                CDS APPLICATION SELECTION
Command ==>

To Perform Control Data Set Operations, Specify:
  CDS Name . . 'SYS1.SCDSRLS'
                                (1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:

7  1. Display      - Display the Base Configuration
   2. Define       - Define the Base Configuration
   3. Alter        - Alter the Base Configuration
   4. Validate     - Validate the SCDS
   5. Activate     - Activate the CDS
   6. Cache Display - Display CF Cache Structure Names for all
                       CF CacheSets
   7. Cache Update - Define/Alter/Delete CF Cache Sets

If CACHE Display is chosen, Enter CF Cache Set Name . . *
                                (1 to 8 character CF cache set name
                                or * for all)

Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 56. CDS Application Selection Panel for VSAM RLS

2. Specify the name of the SCDS that is to contain the base configuration for VSAM RLS in the CDS NAME field.
3. Select **Option 7**, Cache Update, and press ENTER. ISMF displays the CF Cache Set Update panel, shown in Figure 57 on page 154


```

Panel Utilities Help
-----
                                STORAGE CLASS APPLICATION SELECTION
Command ==>

To perform Storage Class Operations, Specify:
CDS Name . . . . . 'SYS1.SCDRSL'
                                (1 to 44 character data set name or 'Active')
Storage Class Name . . *      (For Storage Class List, fully or
                                partially specified or * for all)

Select one of the following options :
3 1. List          - Generate a list of Storage Classes
  2. Display       - Display a Storage Class
  3. Define        - Define a Storage Class
  4. Alter         - Alter a Storage Class
  5. Cache Display - Display Storage Classes/Cache Sets
List Option is chosen,
  Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

If Cache Display is Chosen, Specify Cache Structure Name . .

Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 58. Defining a Storage Class for VSAM RLS

2. Specify the name of the SCDS you defined for VSAM RLS in the CDS NAME field.
3. Select **Option 3, Define**, and press ENTER. ISMF displays the Storage Class Define panel.
4. Press the DOWN key to view the second Storage Class Define panel, shown in Figure 59

```

Panel Utilities Scroll Help
-----
                                STORAGE CLASS DEFINE                                Page 2 of 2
Command ==>

SCDS Name . . . . . : SYS1.SCDRSL
Storage Class Name : CICSRLS

To Storage Class, Specify:

CF Cache Set Name . . . . . CICS1      (up to 8 chars or blank)
CF Direct Weight   . . . . . 5        (1 to 11 or blank)
CF Sequential Weight . . . 3         (1 to 11 or blank)

Use ENTER to Perform Verification; Use UP Command to View previous Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

```

Figure 59. Defining Storage Class Attributes for VSAM RLS

5. Enter the name of the coupling facility cache set you defined in the base configuration.
6. Specify a weight attribute for the data in the CF DIRECT WEIGHT or the CF SEQUENTIAL WEIGHT fields.

You can specify a value from 1 to 11, with 11 indicating the highest relative importance. Use the CF DIRECT WEIGHT field for direct data; use the CF SEQUENTIAL WEIGHT field for sequential data.

The greater the weight value, the more important it is that the data be assigned more cache resources, thereby improving cache hit rates for the data.

If you leave either field blank, a special partition is used within the coupling facility cache structure designated as the default buffer pool. This pool is assigned a weight value of 6.

Note: Be sure to change your Storage Class ACS routines so that RLS data sets are assigned the appropriate storage class.

10.5 Parameter Library (Parmlib) Changes

The changes to SYS1.PARMLIB for RLS are limited to member IGDSMSxx.

Figure 60 is an example of the IGDSMSxx member with the new parameters.

```

*****
SMS ACDS(SYS1.ACDSRLS) COMMDS(SYS1.COMMDS10)
  INTERVAL(10) TRACE(OFF) TYPE(ALL) DESELECT(OPCMD,MSG,DISP)
  PDSESHARING(EXTENDED) DEADLOCK_DETECTION(15,4)
  SMF_TIME(YES) CF_TIME(1800) RLSINIT(YES)
  RLS_MAX_POOL_SIZE(100)
*****

```

Figure 60. IGDSMSxx New Parameters

The new parameters are:

- **{DEADLOCK_DETECTION(iiii|15,kkkk|4)}** specifies the deadlock detection intervals used by the storage management locking services:
 - iiii* specifies the length in seconds of the local deadlock detection interval, as a one- to four-digit numeric value in the range 1-9999. The default is 15 seconds.
 - kkkk* specifies the number of local deadlock cycles that must expire before global deadlock detection is run, as a one- to four-digit numeric value in the range 1-9999. The default is 4 cycles.
- **{SMF_TIME(YES|NO)}** specifies that the DFSMS SMF Type 42 records are to be created at the SMF interval time, and that all of the indicated 42 records are to be synchronized with SMF and RMF data intervals. This allows the customer to merge these SMF records for a specified time period and obtain both the system view and the user view of activity in the interval. YES is the default.

DFSMS creates the specified SMF record when the interval period expires and SMF sends the event notification signal. If you specify YES, SMF_TIME overrides the following IGDSMSmm parameters: BMFTIME, CACHETIME, CF_TIME.

- **{CF_TIME(nnn|3600)}** indicates the number of seconds between recording SMF records for the CF (both cache and lock structures). You can specify a value from 1 to 86399 (23 hours, 59 minutes, 59 seconds). The default is 3600 (one hour).
- **{RLS_MAX_POOL_SIZE(nnnn|100)}** specifies the maximum size in megabytes of the SMSVSAM local buffer pool. SMSVSAM attempts to not exceed the buffer pool size you specify, although more storage might be used temporarily. Because SMSVSAM manages a buffer pool space dynamically, this value does not set a static size for the buffer pool. The minimum accepted value is 10, and the maximum is 1500. If you specify a value greater than 1500, SMSVSAM assumes there is no maximum limit. The default is 100 MB.
- **{RLSINIT(NO|YES)}** specify YES if you want the SMSVSAM address space started as part of system initialization or the V SMS,SMSVSAM,ACTIVE command. This value applies only to the system accessed by the parameter library member and is acted upon when SMSVSAM is next started. The default is NO.

Note: SMSVSAM is started only if the RLSINIT parameter is set to YES and a value, even if the default, is specified for RLS_MAX_POOL_SIZE.

10.6 Resource Access Control Facility (RACF)

RACF 2.2.0 provides some new profiles you can use to establish authorization to restrict access to certain VSAM RLS capabilities:

- You can establish that a user must be authorized for the facility class **STGADMIN.IGWSHCDS.REPAIR** profile to use the AMS SHCDS command,
- You can establish that a user must be authorized for the facility class **STGADMIN.VSAMRLS.FALLBACK** profile to issue the V SMS,SMSVSAM,FALLBACK command to fall back from VSAM RLS processing.

10.7 SMSVSAM

The SMSVSAM server is a new system-address space used for VSAM RLS. Each MVS system has its own SMSVSAM address space. The SMSVSAM server owns a data space that contains most of the VSAM control blocks and the system-wide buffer pool used for data sets opened for record-level sharing. The amount of real storage used by this buffer pool dynamically increases and decreases based on RLS activity rates and other real storage usage across the system. This space adjustment is performed through interfaces between SMS and the MVS Real Storage Manager (RSM) component.

SMSVSAM assumes responsibility for synchronizing the VSAM control block structure across the sysplex. The address space is created and the server is started when you perform an initial program load for MVS. VSAM internally performs cross-address space accesses and linkages between requestor address spaces and the SMSVSAM server address space.

Figure 61 on page 158 is an example of the messages issued during SMSVSAM initialization.

```
.
.
.
IGW415I SMSVSAM SERVER ADDRESS SPACE HAS FAILED AND IS RESTARTING
IEF402I SMSVSAM FAILED IN ADDRESS SPACE 0058
.
.
.
IGW619I ACTIVE SHARE CONTROL DATA SET 536
SYS1.DFPSHCDS.PRIMARY.VTOTSMA ADDED.
IGW619I ACTIVE SHARE CONTROL DATA SET 538
SYS1.DFPSHCDS.SECONDARY.VTOTCAT ADDED.
IGW619I SPARE SHARE CONTROL DATA SET 540
SYS1.DFPSHCDS.SPARE.VTOTSMS ADDED.
.
.
.
IGW467I DFSMS RLS_POOL_SIZE PARMLIB VALUE SET DURING 327
SMSVSAM ADDRESS SPACE INITIALIZATION ON SYSTEM: SC54
CURRENT VALUE: 100 1
IGW467I DFSMS SMF_TIME PARMLIB VALUE SET DURING 329
SMSVSAM ADDRESS SPACE INITIALIZATION ON SYSTEM: SC54
CURRENT VALUE: YES 1
IGW467I DFSMS DEADLOCK_DETECTION PARMLIB VALUE SET DURING 328
SMSVSAM ADDRESS SPACE INITIALIZATION ON SYSTEM: SC54
THIS SYSTEM IS OPERATING AS A LOCAL DEADLOCK PROCESSOR.
CURRENT VALUE: 15 4 1
IGW467I DFSMS CF_TIME PARMLIB VALUE SET DURING 330
SMSVSAM ADDRESS SPACE INITIALIZATION ON SYSTEM: SC54
CURRENT VALUE: 1800 1
.
.
.
IGW414I SMSVSAM SERVER ADDRESS SPACE IS NOW ACTIVE. DRIVER 4.0.2P26A
```

Figure 61. SMSVSAM Starting Messages

10.8 New Storage Management Commands

DFSMS/MVS 1.3 introduces a set of new commands to monitor and control the VSAM RLS environment. In this section, we show the command syntax and examples of the output they produce:

- **SETSMS DEADLOCK_DETECTION(...)**
 - SMF_TIME(...)**
 - CF_TIME(...)**
 - RLSINIT(...)**
 - RLS_MAX_POOL_SIZE(...)**

You can change the values for these parameters at any time during RLS processing. The new values are then used by all systems in the sysplex, except for RLSINIT parameter values which are used only by the system

accessing the changed PARMLIB member when SMSVSAM is next started.
 The command is:

```
SETSMS CF_TIME(3600)
IGW467I DFSMS CF_TIME PARMLIB VALUE
CHANGED ON SYSTEM: SC52
OLD VALUE: 1800 1
NEW VALUE: 3600 1
```

- **DISPLAY SMS,SMSVSAM{,ALL}** — Displays the status of the SMSVSAM server. Specify ALL to see the status of all the SMSVSAM servers. The command is

```
D SMS,SMSVSAM
IGW420I DISPLAY SMS,SMSVSAM ISSUED FOR SC52
IGW420I SMSVSAM ACTIVE IN ASID 10
IGW421I DISPLAY SMS,SMSVSAM ISSUED FOR SC52
IGW422I CONNECTED SUBSYSTEMS - ONLINE 0 368
BATCH 1
```

- **DISPLAY SMS,CFCACHE(CF_cache_structure_name|*)** — Displays information about coupling facility cache structures. Specify CFCACHE(*) to request information for all coupling facility cache structures. Specify a specific cache structure name to display information about only that cache structure:

```
D SMS,CFCACHE(*)
IEE932I 271
IGW530I DFSMS CF STRUCTURES

DFSMS CF CACHE STRUCTURE TO SYSTEM CONNECTIVITY

SYSTEM          ==> 000000000111111111222222222333
IDENTIFIER      ==> 12345678901234567890123456789012

CACHECICS      +.....
CICS_CACHE     +.....
.....         .....
.....         .....
```

- **DISPLAY SMS,CFLS** — Displays information about the coupling facility lock structure. This information includes the lock rate, lock contention rate, false contention rate, and average number of requests waiting for locks:

```
IGW320I 14:58:51 Display SMS,CFLS 122
System  Interval  LockRate  ContRate  FContRate  WaitQLen
SC52    1 Minute   3.7       0.000    0.000     0.00
SC52    1 Hour     1.2       0.000    0.000     0.00
SC52    8 Hour     0.1       0.000    0.000     0.00
SC52    1 Day     -----  -----  -----  -----
(02)   1 Minute   1.9       0.000    0.000     0.00
(02)   1 Hour     0.6       0.000    0.000     0.00
(02)   8 Hour     0.1       0.000    0.000     0.00
(02)   1 Day     0.0       0.000    0.000     0.00
```

*** No other systems provided data
 ***** LEGEND *****

LockRate = number of lock requests per second
 CONTRATE = % of lock requests globally managed
 FCONTRATE = % of lock requests falsely globally managed
 WaitQLen = Average number of requests waiting for locks

- **DISPLAY SMS,SHCDS** — Displays information about SHCDSs. This information includes SHCDS names, sizes, and the amount of free space for all the active SHCDSs and their status. It also includes the names of all the spare SHCDSs. The command is:

```
IGW612I 15:35:47 DISPLAY SMS,SHCDS
```

NAME	SIZE	%UTIL	STATUS	TYPE
PRIMARY.VTOTSMA	10800Kb	1%	GOOD	ACTIVE
SPARE.VTOTSMS	10800Kb	1%	GOOD	ACTIVE
SECONDRY.VTOTCAT	10800Kb	1%	GOOD	ACTIVE
-----0Kb		0%	N/A	N/A
-----0Kb		0%	N/A	N/A
-----0Kb		0%	N/A	N/A
-----0Kb		0%	N/A	N/A
-----0Kb		0%	N/A	N/A
-----0Kb		0%	N/A	N/A

- **DISPLAY SMS,CFVOL(volser)** — Displays a list of coupling facility cache structures containing data for the volume specified. Also displays the CF_VOLUME status.

```
D SMS,CFVOL(TOTSMS)
IEE932I 292
IGW531I DFSMS CF VOLUME STATUS

VOLUME = TOTSMS

DFSMS VOLUME CF STATUS = CF_ENABLED

CICS_CACHE      = CF_ENABLED
CACHECICS       = CF_ENABLED
```

- **DISPLAY SMS,MONDS(specification_mask*)** — Specify MONDS(*) to view all the data set specifications eligible for coupling-facility statistics monitoring. Use a specification mask to view only a subset of those specifications (a specification mask is a fully or partially qualified data set name). You can specify a full or partial data set name, and you must specify least one high-level qualifier. A wild card in the data set name cannot be followed by additional qualifiers. The example is:

```
D SMS,MONDS(CICSDSW.VSAMU.*)

IGW650I DISPLAY DFSMS MONITOR DATASET MASK STATUS

TOTAL NUMBER OF DATASET MASK SPECIFICATIONS: 1

DATASET MASK SPECIFICATION:          CURRENT STATUS:

CICSDSW.VSAMU.*                      SMF          0
```

- **VARY SMS,MONDS(data_set_name{data_set_name,...}},{ON|OFF}** — Use this command to specify which data sets are eligible for coupling facility statistical monitoring. This command is required to gather SMF 42 subtype 16 records. Select OFF to indicate that the specified data sets are no longer eligible for statistical monitoring. The example is:

```
VARY SMS,MONDS(CICSDSW.VSAMU.*),ON
```

```
IGW663I DFSMS CF MONITORING HAS BEEN ACTIVATED  
FOR VSAM/RLS DATASETS WHICH MATCH THE FOLLOWING  
MASK SPECIFICATION: CICSDSW.VSAMU.*
```

```
VARY SMS,MONDS(CICSDSW.VSAMU.*),OFF
```

```
IGW665I DFSMS CF MONITORING HAS BEEN DEACTIVATED  
FOR VSAM/RLS DATASETS WHICH MATCH THE FOLLOWING  
MASK SPECIFICATION: CICSDSW.VSAMU.*
```

- **VARY SMS,CFCACHE(CF_cache_structure_name),ENABLE|QUIESCE** — Use this command to alter the state of the specified coupling-facility cache structure. When a coupling facility cache structure is enabled, VSAM RLS data can be stored in the cache structure. This is the normal state of operations, and is the state the coupling facility cache structure is in after sysplex initial program load. When a coupling facility cache structure is quiesced, no VSAM RLS data can be stored in it. The example follows:

```
V SMS,CFCACHE(CACHECICS),QUIESCE  
IGW464I DFSMS CF CACHE REQUEST TO QUIESCE 769  
STRUCTURE CACHECICS          IS ACCEPTED  
.  
.  
.  
D SMS,CFCACHE(CACHECICS)  
IEE932I 771  
IGW530I DFSMS CF STRUCTURES  
  
DFSMS CF CACHE STRUCTURE TO SYSTEM CONNECTIVITY  
  
SYSTEM          ==> 0000000001111111111222222222333  
IDENTIFIER      ==> 12345678901234567890123456789012  
  
CACHECICS      .....  
  
DFSMS CF CACHE STRUCTURE STATUS:  
CACHECICS      = CF_QUIESCING
```

- **VARY SMS,CFVOL(volser),ENABLE|QUIESCE** — The VARY SMS command is used to alter the state of the specified volume as it relates to all coupling facility cache structures. When a volume is coupling-facility-enabled, data contained on this volume can be stored in a CF cache structure. This is the normal state of operations. When a volume is CF-quiesced, no data contained on it can be stored in a coupling facility cache structure. The example is:

```
VARY SMS,CFVOL(TOTSMS),QUIESCE  
IGW462I DFSMS CF CACHE REQUEST TO QUIESCE 626  
VOLUME TOTSMS IS ACCEPTED  
.  
.  
.  
D SMS,CFVOL(TOTSMS)  
IEE932I 632  
IGW531I DFSMS CF VOLUME STATUS
```

VOLUME = TOTSMS

DFSMS VOLUME CF STATUS = CF_QUIESCED

VOLUME TOTSMS IS NOT BOUND TO ANY DFSMS CF CACHE STRUCTURE

- **VARY SMS,SHCDS(SHCDS_name),NEW** — Use the NEW option on the VARY SMS command to add a new primary or secondary active SHCDS:

```
V SMS,SHCDS(PRIMARY.VTOTSMA),NEW
IGW619I ACTIVE SHARE CONTROL DATA SET
SYS1.DFPSHCDS.PRIMARY.VTOTSMA ADDED.
```

- **VARY SMS,SHCDS(SHCDS_name),NEWSPARE** — Use the NEWSPARE option on the VARY SMS command to add a new spare SHCDS:

```
V SMS,SHCDS(SPARE.VTOTSMS),NEWSPARE
IGW619I SPARE SHARE CONTROL DATA SET
SYS1.DFPSHCDS.SPARE.VTOTSMS ADDED.
```

- **VARY SMS,SHCDS(SHCDS_name),DELETE** — Use the DELETE option on the VARY SMS command to logically delete either an active or spare SHCDS:

```
V SMS,SHCDS(SECONDRY.VTOTCAT),DELETE
IEF196I IGD104I SYS1.DFPSHCDS.SECONDRY.VTOTCAT          RETAINED,
IEF196I DDNAME=SYS00003
```

Note: In the above VARY SMS,SHCDS command examples, the SHCDS name is not fully qualified. SMSVSAM takes as a default the first two qualifiers which must always be SYS1.DFPSHCDS. You must specify only the last two qualifiers as the SHCDS names.

- **SETXCF START,ALTER,STRNAME=CF_cachestructurename,SIZE=newsize** — Use the SETXCF command to alter the size of a coupling facility cache structure:

CF_cache_structure_name

is the name of the cache structure being altered.

newsize

is the new structure size in megabytes.

This new size can be larger or smaller than the size of the current CF cache structure, but it cannot be larger than the maximum size specified in the CFRM policy.

Note: The SETXCF START,ALTER command will not work unless the structure's ALLOW ALTER indicator is set to YES. You can verify this by issuing D XCF,STR,STRNAME=structure_name,CONNAME=ALL. If at least one of the connections has ALLOW ALTER NO, then the command fails. Connector indicates that it can or cannot support structure alter being initiated for this structure.

MVS automatically starts the alter process in place, without disruptions to the application using the coupling facility cache structure.

The alter function does not cross CF boundaries and does not take the place of the rebuild function. If you require a larger structure size than that specified in the CFRM policy, you must activate a new CFRM policy and rebuild the structure.

For more information about the new commands, please refer to *MVS/ESA System Product: JES2 Version 5 System Commands* (GC28-1442) and to

10.8.1 Access Method Service (AMS) Commands

A new AMS IDCAMS command, the SHCDS command, is available. The SHCDS command lists SMSVSAM recovery related to online applications and spheres accessed in RLS mode. It also manages that recovery. This command is supported in batch and TSO foreground environments.

See Section 3.29, “SHCDS,” of *DFSMS/MVS Version 1 Release 3 Access Method Services for ICF* (SC26-4906-02), for additional details on the SHCDS command.

10.9 Fallback Procedure

If you want to fall back from VSAM RLS processing, there are rules, considerations, and specific procedures to follow. These are documented in Chapter 14, “Falling Back from VSAM RLS Processing” of *DFSMS/MVS Version 1 Release 3 DFSMSdfp Storage Administration Reference*.

10.10 New Catalog Information

Two new parameters support VSAM RLS: LOG and LOGSTREAMID. These parameters are set using the AMS IDCAMS DEFINE CLUSTER and ALTER commands. Figure 62 shows an example of the new DEFINE CLUSTER and IDCAMS ALTER parameters for RLS:

```
DEFINE CLUSTER
    .
    .
    .
    LOG(none|undo|all)
    LOGSTREAMID(dsname)

ALTER
    .
    .
    .
    LOG(none|undo|all)
    LOGSTREAMID(dsname)
```

Figure 62. Examples of IDCAMS DEFINE and ALTER Parameters for RLS

The parameter specifications shown in Figure 62 are these:

LOG(none|undo|all)

Establishes whether the sphere to be accessed with VSAM record-level sharing (RLS) is recoverable or nonrecoverable. It also indicates whether a recovery log is available for the sphere. LOG applies to all components in the VSAM sphere.

none indicates that neither an external backout nor a forward recovery capability is available for the sphere accessed in RLS mode. If you use this, RLS considers the sphere to be nonrecoverable.

undo specifies that changes to the sphere accessed in RLS mode can be backed out and forward recovered using external logs. RLS considers the sphere recoverable when you use LOG(undo).

all specifies that changes to the sphere accessed in RLS mode can be backed out and forward recovered using an external log. RLS considers the sphere recoverable when you use LOG(all).

LOGSTREAMID(dsname)

Gives the name of the CICS forward-recovery log stream. It applies to all components in the VSAM sphere.

dsname is the name of the forward-recovery log stream.

Figure 63 is an example of the RLS DATA portion of the output for a LISTC ENT(dsname) ALL command for a VSAM cluster defined for RLS processing:

```
RLSDATA
LOG -----ALL    RECOVERY REQUIRED --(NO)
VSAM QUIESCED -----(NO)    RLS IN USE -----(NO)
LOGSTREAMID-----CDSWV.FWRLOG.UHOTEL1
RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
RECOVERY TIMESTAMP GMT-----X'0000000000000000'
```

Figure 63. RLS Catalog Information

Note: The RLS IN USE indicator is set **on** by a successful VSAM open for RLS. The indicator remains on after the data set is closed. The RLS IN USE indicator is turned off by a successful VSAM open for NON-RLS.

If you rename, alter, move, recover, or change the size of a basic catalog structure when using VSAM RLS, you must use the AMS SHCDS CFREPAIR command to correct RLS information in the catalog. If you are using VSAM RLS support and decide to stop using it, then the AMS SHCDS CFRESET command is needed to reset applicable RLS indicators in the catalog. For more details, please refer to *DFSMS/MVS Version 1 Release 3 Access Method Services for ICF*.

10.11 Maintenance and Messages

Before you start using RLS, make sure that all maintenance is applied on your system. See 3.3, “DFP Environment” on page 30 for a list of the maintenance requirements at the time this book was written. You should check to see if there are any new requirements before you implement VSAM RLS. VSAM RLS introduces a new set of messages. They are IGW015, IGW320, IGW311, IGW313, IGW314, IGW400 through IGW700, IDC01880 through IDC01895, IDC31880 through IDC31959, IGD016I, IGD06200I, IGD17209, IGD002I, IGD029I, and IGD031I.

Make sure that you obtain the latest version of the *MVS/ESA SP JES2 Version 5 System Messages*.

10.12 Step by Step Guide

In this section, we give you this logical sequence to follow to implement VSAM record level sharing:

1. Install DFSMS 1.3.
2. Order the PSP bucket for MVSDFSMS130/RLS.
3. Check that you have the latest version of *MVS/ESA SP JES2 Version 5 System Messages*
4. Define the SHCDS data sets.

Check with an LISTC command that the share options of the data sets are set to (3,3). If they are set to (1,3), issue the IDCAMS ALTER command to set them to (3,3).

5. Define the coupling facility cache and lock structures with the IXCMIAPU utility.
6. Alter the base SMS configuration, adding the Cache Set information.
7. Create a storage class that contains the cache set information.
8. Alter the storage class ACS routine to select the new SC defined.
9. Alter SYS1.PARMLIB, member IGDSMSxx with the new RLS-related parameters.
10. Define the RLS-related new profiles in the RACF FACILITY class.
11. Make an IPL of the MVS and check if SMSVSAM address space is active.
12. Activate the SHCDS data sets using the command:

```
V SMS,SHCDS(...),NEW|NEWSPARE
```
13. Test the new commands introduced for RLS and verify existing procedures.

Appendix A. System Measurement Facility and RLS

DFSMS uses SMF record type 42 to record its statistical information. VSAM writes record type 64 when it closes a VSAM component or cluster.

SMF record type 42 has many subtypes. You can record SMF type 42 records, subtypes 15-18 to gather coupling facility information, and you can specify SMF_TIME in IGDSMSxx to synchronize SMF type 42 data with SMF and RMF data intervals.

Cache structure statistics are also recorded by selecting SMF type 42 records, subtypes 15, 16, 17, and 18:

- Subtype 15 can be created on a timed interval or whenever the SMF timer ends to collect data about VSAM RLS storage class response time. This data includes information for each system and a sysplex-wide summary.
- Subtype 16 can be created on a timed interval or whenever the SMF timer ends to collect data about VSAM RLS data set response time. This data includes information for each system and a sysplex-wide summary.
- Subtype 17 can be created on a timed interval or whenever the SMF timer ends to collect data about VSAM RLS coupling facility lock structure usage. This data includes information for each system and a sysplex-wide summary.
- Subtype 18 can be created on a timed interval or whenever the SMF timer ends to collect data about VSAM RLS coupling facility cache partition usage. This data includes information for each system and a sysplex-wide summary.

For Subtypes 15, 16, 17, and 18, specify the time interval in the IGDSMSx parameter library member and activate recording on one system in the sysplex. Because data is collected across the sysplex, it is unnecessary to merge SMF records from all the systems in the sysplex.

SMF writes a record type 64 if a cluster is closed, and writes one record for each component in the cluster. Record type 64 indicates why the record was created. It describes the device and volumes on which the object is sorted, and gives the extents of the object on the volumes. It gives statistics about various processing events that have occurred since the object was defined, such as the number of records in the data component, the number of records inserted, and the number of control intervals that were split.

SMF record type 64 has no subtypes. New sections were added to give CF cache informations and some fields were modified to reflect RLS use of the cluster or component. In the SMF record type 64, you can also find information about OPEN time, in the 'Statistics at OPEN Time Section.'

Note: Field SMF64RLS tells you if the VSAM cluster or component was opened for RLS.

For more detailed information about SMF records, please refer to *MVS/ESA JES2 Version 5: System Management Facilities (SMF)*

Appendix B. Special Notices

This publication is intended to help those who are responsible for implementing 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.

This book refers to CICS. When the term "CICS" is used, it refers to Customer Information Control System, an element of the IBM CICS Transaction Server for OS/390 program product. It is also used to refer to the predecessor products, CICS/MVS and CICS/ESA. When the CICS Transaction Server is required, the reference is specifically to that product. Citations are to the CICS Transaction Server for OS/390 library; CICS/MVS and CICS/ESA users should use the corresponding book in those libraries. Please see *CICS Transaction Server for OS/390 Up and Running!*, GC33-1789, for more information.

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.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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

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, other countries, or both:

IBM

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.

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 173.

- *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: Recovery Considerations*, SG24-4768 (available April 1997)
- *OS/390 Parallel Sysplex Configuration Cookbook*, SG24-4706

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
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	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
- *CICS Transaction Server for OS/390 Up and Running*, GC33-1789
- *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 DFSMSshm 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
- *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**—Use 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 BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks:

```
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.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS 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.ibm.link.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:

	IBMAIL	Internet
In United States:	usib6fpl at ibmail	usib6fpl@ibmail.com
In Canada:	caibmbkz at ibmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmail	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 Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **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)001-408-256-5422 (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.ibm.com/pbl/pbl

- **Internet Listserver**

With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserv. To initiate the service, send an e-mail note to announce@webster.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

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 shows consistency if it 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 is executed either before it or after it.

Durability A transaction shows durability if, after it 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 synch 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 synch 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.
- Written a log record of its response to signify that it has entered the in-doubt state.
- Not yet learned of 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).

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 expiration 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 synch 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 synch point in a distributed UOW. At synch 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 synch 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 synch points.
- Start of task and a synch point
- A synch point and end of task
- Two synch points

Thus a UOW is completed when a transaction takes a synch point, which occurs either when a transaction issues an explicit synch point request or when CICS takes an implicit synch point at the end of the transaction. In the absence of user synch 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

ABARS	aggregate backup and recovery	HLL	high-level language
ACB	access control block	HURBA	hi-used relative byte address
ACS	automatic class selection	IBM	International Business Machines Corporation
AOR	application-owning region	ICF	integrated catalog facility
APF	authorized program facility	ICMF	Integrated Coupling Migration Facility
API	application programming interface	IMS	Information Management System
APPC	Advanced Program-to-Program Communication	IPL	initial program load
ARM	Automatic Restart Manager	ISMF	Interactive Storage Management Facility
ATI	automatic transaction initiation	ITSO	International Technical Support Organization
BMP	batch message processing	JCL	job control language
BWO	Backup While Open	KSDS	key-sequenced data set
CBIC	control blocks in common	LP	logical partition
CF	coupling facility	LPA	link pack area
CFCC	coupling facility control code	LSR	local shared resources
CFRM	coupling facility resource management	MCT	monitoring control table
CI	control interval	MLPA	modified link pack area
CICS	Customer Information Control System	MRO	multiregion operation
CICSVR	CICS VSAM Recovery	MVS	multiple virtual storage
CICS TS	Customer Information Control System Transaction Server	NSR	nonshared resources
CMOS	complementary metal-oxide semiconductor	NRI	no read integrity
CMT	CICS-maintained data table	ONC RPC	open network computing remote procedure call
CP	central processor	PR/SM	processor resource/system manager
CPC	central processor complex	QCA	queue control area
CR	consistent read	QEA	queue element area
CSD	CICS system definition	QOR	queue-owning region
DASD	direct access storage device	RBA	relative byte address
DBCTL	database control	RCT	resource control table
DB2	DATABASE 2	RDO	resource definition online
DDM	Distributed Data Management	RLS	record level sharing
DOR	data-owning region	RMI	resource manager interface
DRA	database resource adapter	RRDS	relative record data set
DTR	dynamic transaction routing	SCDS	source control data set
ENF	event notification facility	SDT	shared data table
ESDS	entry-sequenced data set	SFM	sysplex failure management
FCT	file control table	SHCDS	sharing control data set
FOR	file-owning region	SIT	system initialization table
GRS	global resource serialization	SPI	system programming interface
GSR	global shared resources		
HARBA	hi-allocated relative byte address		

SMS	system managed storage	URM	user replaceable module
STC	started tasks	VRRDS	variable length relative record data set
TCB	task control block	VSAM	Virtual Storage Access Method
TOR	terminal-owning region	VVDS	VSAM volume data sets
UBF	user buffering	WLM	workload management
UMT	user-maintained data table	XCF	cross-system coupling facility
UOW	unit of work	XES	cross-system extended services
UPS	uninterruptible power supply	XPI	exit programming interface

Index

A

- ABEND codes 55
- access method service
 - commands 163
- ACS routines 26
- activate
 - SHCDS 150
- allocation
 - SHCDS 150
- AMS
 - commands 163
- AXM subsystem 113
 - defining 113

B

- batch
 - access to RLS data sets 134
 - sample program 134
 - sharing data sets 95

C

- cache
 - coupling facility structures 151
- cache set
 - SMS 153
- catalog information
 - LOG parameter 163
 - LOGSTREAMID 163
 - RECOVERY REQUIRED 164
 - RLS IN USE 164
 - RLSDATA 164
 - VSAM QUIESCED 164
- CEMT transaction
 - monitoring 123
- CF DIRECT WEIGHT
 - storage class changes 156
- CF lock structure
 - DISPLAY SMS command 159
- CF SEQUENTIAL WEIGHT
 - storage class changes 156
- cf_time
 - parmlib changes 157
- cfcache
 - DISPLAY SMS command 159
 - VARY SMS command 161
- CFLS
 - DISPLAY SMS command 159
- CFRM
 - activating 111
 - policy 151
 - preparation 101

- CFRM policy
 - IXCMIAPU example 152
 - JCL example 151
- cfvol
 - cfvol 160
 - DISPLAY SMS command 160
 - VARY SMS command 161
- checklist
 - implementing VSAM RLS 165
- CICS log manager
 - log streams 3
 - log structures 3
 - overview 2
- CICS recovery manager
 - overview 4
- CICS regions
 - connecting 121
- CICS TS
 - CICS log manager 2
 - CICS recovery manager 4
 - components 1
 - enqueue domain 6
- commands
 - MVS commands 111
- configuration changes
 - SMS 153
- coupling facility
 - data sets and policies 101
 - preparation 101
- coupling facility resource manager
 - See CFRM
- coupling facility structures 151
 - cache 151
 - IGWLOCK00 151
 - lock 151
- CSD (CICS system definition file)
 - compatibility groups 38
 - restrictions for RLS 38
 - upgrade 38

D

- deadlock
 - example 148
 - examples 145, 146, 147
 - resolving 145
- deadlock_detection
 - parmlib changes 156
- DFH0STAT 20, 125
- DFHCSD (CICS system definition file)
 - compatibility groups 38
 - restrictions for RLS 38
 - upgrade 38
- DFHCSDUP 38

- DFHLOG log streams
 - sizing 106
- DFHRMUTL 38
- DFHSHUNT log streams
 - sizing 108
- DFHSTUP 20, 125
- DFSMS environment 23
- DFSMS/MVS 23
- DFSMSdfp 23
- DFSMSdss 23
- DFSMShsm 23
- DFSMSrmm 23
- DISPLAY SMS
 - new commands list 159

E

- enqueue
 - enqueue pools 143
 - resolving deadlocks 145
- enqueue domain
 - overview 6
- enqueue pools 7
- enqueue requests 7

F

- failure scenarios 98
- fallback
 - plan 95
 - procedure 163

G

- general logs 3

I

- IDCAMS
 - new commads 163
 - SHCDS command 163
- igdsmsxx
 - cf_time 157
 - deadlock_detection 156
 - parmlib changes 156
 - rls_max_pool_size 157
 - RLSINIT 157
 - SMF_TIME 156
- IGWLOCK00
 - coupling facility structures 151
- implementing VSAM RLS
 - step by step guide 165
- INITIAL start 38
- integrity
 - READ integrity 95
- IXCMIAPU
 - CFRM policy 152

L

- lock
 - coupling facility structures 151
- LOCKED condition 10
- LOG parameter
 - catalog information 163
- log stream 3
- log stream storage
 - primary 3
 - secondary 3
 - tertiary 4
- log streams
 - naming convention 104
 - overview 3
 - sizing 105, 109, 111
- log structures
 - overview 3
- logging
 - DUMMY log 58
 - setting up
 - DUMMY log 58
 - log data sets 63
 - log stream 61
 - MVS system logger 59
 - offloading 64, 66
 - overview 58
- LOGSTREAMID
 - catalog information 163
- LSN (log stream name) 3

M

- maintenance
 - checklist 164
- messages
 - new set of 164
- monds
 - DISPLAY SMS command 160
 - VARY SMS command 160
- monitoring
 - temporary storage data sharing 123
- MVS commands 111
- MVS system logger 103

N

- naming convention
 - log streams 104
 - TS pools 105
- new commands
 - DISPLAY SMS 159
 - IDCAMS 163
 - SETSMS 158
 - SETXCF START 162
 - SHCDS IDCAMS command 163
 - VARY SMS 160
- NQ domain
 - See enqueue domain

O

offloading
causes
 HIGHOFFLOAD 66
 structure list entry limit 66
HIGHOFFLOAD
 list structure 66
 staging data set 66
monitoring 66
OPEN time information
 smf type 64 167

P

parmlib changes
 cf_time 157
 deadlock_detection 156
 igdsmsxx 156
 rls_max_pool_size 157
 RLSINIT 157
 SMF_TIME 156
performance data
 changes 19
policies
 coupling facility 101
prerequisites
 RLS 90
primary storage 3

R

RACF
 new profiles 157
 STGADMIN.IGWSHCDS.REPAIR profile 157
 STGADMIN.VSAMRLS.FALLBACK 157
RDO support
 transient data 14
read integrity
 CONSISTENT 9
 REPEATABLE 9
 UNCOMMITTED 9
RECORDBUSY condition 10
recovery manager
 utilities 131
RECOVERY REQUIRED
 catalog information 164
resource owners 7
RLS
 implementing 89, 90
 selecting data sets 90
 smf type 64 167
 SMS required 28
 summary 89
RLS IN USE
 catalog information 164
rls_max_pool_size
 parmlib changes 157

RLSDATA
 catalog information 164
RLSINIT
 parmlib changes 157

S

SCDS 26
secondary storage 3
security access
 authorizing 122
 defining 121
SETSMS
 new commands 158
sharing control data set
 allocation example 150
 JCL example 150
 naming convention 149
sharing data sets
 between CICS and batch 95
SHCDS
 activate command 150
 allocation criteria 151
 allocation example 150
 DISPLAY SMS command 160
 JCL example 150
 naming convention 149
 new commands 162
 sizing formula 150
 VARY SMS 151
 VARY SMS command 162
shunt 6
sizing
 DFHLOG log streams 106
 DFHSHUNT log streams 108
 log streams 105, 109, 111
 temporary storage pool 111
smf
 type 42 167
 type 42 subtypes 167
 type 64 167
smf 42
 subtype 15 167
 subtype 16 167
 subtype 17 167
 subtype 18 167
smf type 64
 description 167
 new sections 167
 OPEN time information 167
 RLS 167
SMF_TIME
 parmlib changes 156
SMS
 cache set 153
 cache set update panel 153
 configuration changes 153
 ISMF panels 153
 new commands 158

- SMS (*continued*)
 - storage class changes 154
- SMS (Storage Management Subsystem)
 - ACS routines 26
 - base configuration 26
 - benefits 24
 - changes for RLS 28
 - classes 25
 - classes list 25
 - configuration 26
 - data class 25
 - goals 24
 - groups 25
 - introduction 23
 - management class 25
 - required for RLS 28
 - storage class 25
 - storage group 25
- SMSVSAM 157
 - DISPLAY SMS command 159
 - initialization messages 158
- SMSVSAM server
 - SMSVSAM server 8
- source control data set 26
- starting CICS
 - INITIAL 38
- statistics
 - changes 19
 - sample program 20
 - utility program 20
- STGADMIN
 - RACF profiles 157
- storage class changes
 - CF DIRECT WEIGHT 156
 - CF SEQUENTIAL WEIGHT 156
 - example 155
 - SMS 154
- storage hierarchy 27
 - levels 27
- Storage Management Subsystem
 - See SMS
- summary
 - prerequisites 90
 - RLS 89
- sys1.parmlib changes 156
- SYSLOG 68
- system logger 103

T

- tasks
 - implementing RLS 91
- temporary storage
 - authorizing 122
 - customization 126
 - data sharing servers 113
 - setting up 113
 - defining server regions 114
 - commands 119
 - parameters 115

- temporary storage (*continued*)
 - monitoring 123
 - CEMT transaction 123
 - server statistics 123
 - pool structure
 - definig 114
 - security access 121
- temporary storage domain
 - See TS domain
- temporary storage pool
 - sizing 111
- tertiary storage 4
- trace utility
 - DFHTU510 137
 - DFHUT510 137
- transaction affinities 128
- transient data
 - RDO support 14
- TS domain
 - overview 11
- TS pools
 - naming convention 105

U

- unshunt 6
- UOW control 4
- UOW shunting 6
- utilities
 - monitoring log streams 128
 - new transactions 128
 - recovery manager 131
 - shutdown assist program 129
 - trace utility 137
 - transaction affinities 128
- utility programs
 - DFHCSDUP 38
 - DFHRMUTL 38

V

- VARY SMS
 - new commands list 160
 - SHCDS 151
- VSAM QUIESCED
 - catalog information 164
- VSAM RLS
 - cache structures 28
 - fallback procedure 163
 - overview 8
 - processing restrictions 10
 - SMSVSAM server 8



Printed in U.S.A.

SG24-4766-00

