IBM

IMS in the Parallel Sysplex

Volume III: IMSplex Implementation and Operations

- Implement and operate IMS data sharing
- Implement and operate IMS shared message queues
- Implement and operate the new IMS CSL features

Jouko Jantti
Juan Jesús Iniesta Martínez
Knut Kubein
Bill Stillwell
Gary Wicks

Redbooks

ibm.com/redbooks

International Technical Support Organization

IMS in the Parallel Sysplex
Volume III: IMSplex Implementation and Operations

June 2003

SG24-6929-00

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

First Edition (June 2003)

This edition applies to IMS Version 8 (program number 5655-C56) or later for use with the OS/390 or z/OS operating systems.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

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

Trademarks

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

CICS®	Hiperspace™	Redbooks(logo)™
CUA®	IBM®	RACF®
DB2®	ibm.com®	RMF™
DFS™	IMS™	VM/ESA®
DFSMS/MVS®	Language Environment®	VSE/ESA™
DFSMSdss™	MVS™	VTAM®
ESCON®	OS/390®	WebSphere®
@server™	Parallel Sysplex®	z/OS®
FICON™	Redbooks™	zSeries®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This IBM® Redbook™ is the third volume of a series of redbooks called *IMS in the Parallel Sysplex*. These redbooks describe how IMS™ exploits the Parallel Sysplex® functions and how to plan for, implement, and operate IMS systems working together in a Parallel Sysplex. We use the term *IMSplex* to refer to multiple IMSs which are cooperating with each other in a Parallel Sysplex environment to process a common shared workload. Although we generally think of an IMSplex in terms of online environments, an IMSplex can include batch IMS jobs as well as IMS utilities.

This redbook encompasses two parts. Part 1, *Implementation*, is a description of the tasks an IMS installation must execute to enable each of the functions. Part 2, *Operations*, addresses the operational issues associated with running an IMSplex that include startup and shutdown, steady state operations, online change, and recovery from failure.

Both of the parts are divided into chapters addressing the sysplex services exploited by IMS to support block level data sharing, connecting to the IMSplex, shared queues, and the systems management functions enabled with the Common Service Layer introduced in IMS Version 8 — operations management and resource management. Additional topics are addressed in appendices or incorporated within these major topics.

The other volumes in this series are:

- *IMS in Parallel Sysplex, Volume I: Reviewing the IMSplex Technology,* SG24-6908
- *IMS in the Parallel Sysplex, Volume II: Planning the IMSplex,* SG24-6928

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.

Jouko Jäntti is a Project Leader specializing in IMS with the IBM International Technical Support Organization, San Jose Center. He holds a bachelor's degree in Business Information Technology from Helsinki Business Polytechnic, Finland. Before joining the ITSO in September 2001, Jouko worked as an Advisory IT Specialist at IBM Global Services, Finland. Jouko has been working on several e-business projects with customers as a Specialist in IMS, WebSphere®, and UNIX on the OS/390® platform. He has also been responsible for the local IMS support for Finnish IMS customers. Prior to joining IBM in 1997, he worked as a Systems Programmer and Transaction Management Specialist in a large Finnish bank for 13 years, and was responsible for the bank's IMS systems.

Juan Jesús Iniesta Martínez is a Senior IT Specialist with IBM Global Services in Spain. He has 15 years of experience in the IMS field. Prior to joining IBM in 1995, he worked 10 years as a Console Operator and eight years as an IMS Systems Programmer and Technical Support Specialist in the IMS team of a large Spanish bank. Since Juan joined IBM, he was assigned to "la Caixa", one of the largest financial entities in Spain. His areas of expertise and responsibilities include IMS installation and maintenance, IMS system management, IMS problem determination and related products, Parallel Sysplex architecture, IMS data sharing and IMS shared queues installation.

Knut Kubein is a Senior Advisory IT Specialist with IBM Global Services, Germany. He has 30 years of experience with IBM large systems products working as a Technical Support

Specialist, with 20 of those years devoted to IMS. He leads the IMS Support Team in EMEA. His areas of expertise include IMS data sharing, shared queues, and Parallel Sysplex architecture. He supports large bank accounts as well as other industrial IMS users. He has written extensively for the shared queues part of this book and about performance and monitoring of IMS.

Bill Stillwell is a Senior Consulting IT Specialist and has been providing technical support and consulting services to IMS customers as a member of the Dallas Systems Center for 20 years. During that time, he developed expertise in application and database design, IMS performance, Fast Path, data sharing, shared queues, planning for IMS Parallel Sysplex exploitation and migration, DBRC, and database control (DBCTL). He also develops and teaches IBM Education and Training courses, and is a regular speaker at the annual IMS Technical Conferences in the United States and Europe.

Gary Wicks is a Certified IT Specialist in Canada. He has 28 years of experience with IBM in Software Service and is currently on assignment as an IMS Development Group Advocate. He holds a degree in Mathematics and Physics from the University of Toronto. His areas of expertise include IMS enablement in Parallel Sysplex environments and has written several IBM Redbooks on the subject over the last six years.

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

Rich Conway
Bob Haimowitz
International Technical Support Organization, Poughkeepsie Center, USA

Rose Levin
Jim Bahls
Mark Harbinski
Claudia Ho
Rick Long
Judy Tse
Sandy Stoob
Pedro Vera
Dave Viguers
IBM Silicon Valley Laboratory, USA

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099

Part 1

IMSplex implementation

Part 1 addresses several topics relating to the implementation of an IMSplex:

- Data sharing implementation
- Connectivity implementation
- Shared message queue implementation
- CSL implementation

1

Data sharing implementation

Once the hardware and software environment for the sysplex is established, migration to use IMS block level data sharing (BLDS) can begin.

It is important to emphasize that although the order of the activities presented in this chapter need not be followed exactly, there is a logical installation path where prerequisite components and their features should be available for follow on installation elements.

We assume the OS/390 sysplex hardware and software, along with IRLM defined using the SCOPE=LOCAL option have been implemented already.

1.1 Migration steps to implement data sharing

There is a standard progression of activities associated with the implementation of a block level data sharing environment. We will map out those steps in the following sections.

Appendix A, “IMS Parallel Sysplex migration task lists” on page 323 contains a table that can be used to develop your own BLDS implementation checklist. This chapter describes the steps included in the checklist found in the appendix.

An example of that checklist is shown in Table 1-1, which states the task, owner, target date, and status.

Table 1-1 Sample implementation checklist

Task	Owner	Target Date	Status

1.2 Ensure that SHARECTL is set in the RECON

The SHARECTL status of the RECON is stored in the header record of the RECON and applies to all subsystems in the data sharing group. Only IMS Version 6 has an option to choose log control rather than SHARECTL. There is no SHARECTL option available with IMS Version 7 and 8, because it is mandatory.

1.3 Register databases at SHARELVL(1)

This is a typical registration level for installations not using block level data sharing yet. It allows Concurrent Image Copy (CIC) to be executed and allows two data sets in the same database to be image copied simultaneously.

1.4 Parameters to review in IMS procedures

There are several IMS procedures and parameters which need to be reviewed prior to migration. Except for RGSUF, which must be specified on the IMS procedure itself, these may be specified in the IMS system definition, the IMS procedure, or member DFSPBxxx (where RGSUF=xxx). The use of DFSPBxxx is recommended.

1.4.1 DL/I batch processing regions: DBBBATCH and DLIBATCH procedures

We assume that IMS batch jobs currently part of a data sharing group will continue to be so and that IMS batch jobs not using data sharing will not be changed to use it immediately. Batch jobs do not require changes to their execution parameters if they are not using data sharing. The following parameters in Table 1-2 are for block level data sharing jobs and also apply to executions of the batch backout utility. Default values can be set for IRLM= and IRLMNM= in IMS system definition to avoid changing JCL.

Table 1-2 DBBBATCH and DLIBATCH parameters related to BLDS

Parameter	Value
IRLM	Specifies that the IRLM is to be used. The value must be ‘Y’ for data sharing.

Parameter	Value
IRLMNM	The named IRLM must exist on the system used by the batch job.
IMSPLEX	This parameter has been introduced in IMS Version 8. Specifies a five character value that is passed to the SCI registration exit, DSPSCIX0. The sample version of DSPSCIX0 shipped with IMS will return the value you supply to DBRC as the IMSplex name.

We recommend that the CFNAMES statement is included in the data set assigned to DD statement DFSVSAMP for specifying the structure names for the IRLM lock structure and OSAM and VSAM cache structures. If the CFNAMES is included, the names must be identical to all other CFNAMES statements in the data sharing group. An alternative to CFNAMES CFIRLM value is to use the LOCKTABL= parameter in the IRLM start procedure, but if CFNAMES is used, it takes precedence.

1.4.2 DL/I Separate Address Space: DLISAS procedure

No parameters have changed in this procedure. The IMSID is provided by the control region when it starts the DLISAS address space.

1.4.3 IMS Transaction Manager and IMS DBCTL: IMS and DBC procedures

The following parameters in Table 1-3 should be reviewed when moving to a sysplex data sharing environment.

Table 1-3 IMS TM and IMS DBCTL parameters related to BLDS

Parameter	Value
DBRCNM	Identifies the name of the DBRC procedure which is automatically started by the control region. The value of this parameter should be different for each IMS.
DLINM	Identifies the name of the DLI SAS procedure which is automatically started by the control region. The value of this parameter should be different for each IMS.
IMSID	This is the OS/390 subsystem name for the IMS control region. The value of this parameter should be different for each IMS.
IRLM	Specifies that IRLM is to be used as the lock manager. The value of this parameter should be set to 'Y'.
IRLMNM	This is the OS/390 subsystem name for the IRLM.
PRDR	Defines the name of the IMSRDR procedure used by subsequent start region commands.
RGSUF	RGSUF must be hard coded in each IMS procedure. It identifies the suffix of the DFSPBxxx member in IMS.PROCLIB. The rest of the parameters should be specified in the DFSPBxxx member.
SSM	Identifies the external subsystem member containing the parameters for connection to DB2®. Because it identifies DB2 by its subsystem name, and because there may be multiple DB2 subsystems in the data sharing group, this should be different for each IMS.
SUP	Identifies the one character suffix for the control region in IMS.RESLIB. If the IMSs are sharing one RESLIB, and are not identical clones, then this parameter must be different for each IMS.

Parameter	Value
VSPEC	Identifies the PROCLIB member DFSVSMxx, which contains control statements. The default is 00. If the IMS systems are not exact clones, then this parameter should be different for each DFSVSMxx. All CFNAMES statements in a data sharing group must be identical.

1.4.4 Message processing region: DFSMPR procedure

The following parameters in Table 1-4 should be reviewed when moving to a sysplex data sharing environment.

Table 1-4 MPR parameters related to BLDS

Parameter	Value
IMSID	The value must match the IMSID of the control region to which this message processing region will be connected.
ALTID	Specifies a one- to four-character name for an alternate IMS system. When your IMS message processing region tries to link to an IMS system, it tries to link to the system specified by the IMSID= parameter. If no IMS system matches the name specified with IMSID, the message processing region tries the system specified with the ALTID= parameter.
PREINIT	Verify that the pre-initialization module member name (DFSINTxx) in IMS.PROCLIB has not changed.
SSM	Identifies the external subsystem member of IMS.PROCLIB, containing the parameters for connection to DB2. Since DB2 is identified by its subsystem name, and since there may be multiple DB2 subsystems in the data sharing group, the name should be different for each IMS.

1.4.5 Batch message processing: IMSBATCH procedure

The following parameters in Table 1-5 should be reviewed when moving to a sysplex data sharing environment.

Table 1-5 IIMSBATCH parameters related to BLDS

Parameter	Value
IMSID	This must match the IMSID of the control region to which this batch message processing region will be connected. Be sure this conforms to any BMP scheduling changes that may have occurred as a result of this migration.
ALTID	Specifies a one- to four-character name for an alternate IMS system. When your IMS message processing region tries to link to an IMS system, it tries to link to the system specified by the IMSID= parameter. If no IMS system matches the name specified with IMSID, the message processing region tries the system specified with the ALTID= parameter.
PREINIT	Verify that the pre-initialization module member name (DFSINTxx) in IMS.PROCLIB has not changed.
SSM	Identifies the external subsystem member containing the parameters for connection to DB2. Because it identifies DB2 by its subsystem name, and because there may be multiple DB2 subsystems in the data sharing group, this should be different for each IMS.

1.4.6 Fast Path application: IMSFP procedure

The following parameters in Table 1-6 should be reviewed when moving to a sysplex data sharing environment.

Table 1-6 IMSFP parameter related to BLDS

Parameter	Value
IMSID	Must match the IMSID of the control region to which this region is connected to. Be sure this matches any scheduling changes that may have occurred as a result of this migration.
ALTID	Specifies a one- to four-character name for an alternate IMS system. When your IMS message processing region tries to link to an IMS system, it tries to link to the system specified by the IMSID= parameter. If no IMS system matches the name specified with IMSID, the message processing region tries the system specified with the ALTID= parameter.
PREINIT	Verify that the pre-initialization module member name (DFSINTxx) in IMS.PROCLIB has not changed.
SSM	Identifies the external subsystem member containing parameters for connection to DB2. it identifies DB2 by its subsystem name, and because there may be multiple DB2 subsystems in the data sharing group, this should be different for each IMS.

1.4.7 Fast Path utility region: FPUTIL procedure

The following parameters in Table 1-7 should be reviewed when moving to a sysplex data sharing environment.

Table 1-7 FPUTIL parameters related to BLDS

Parameter	Value
IMSID	Must match the IMSID of the control region to which this Fast Path utility region will be connected.
ALTID	Specifies a one- to four-character name for an alternate IMS system. When the FPUTIL region tries to link to an IMS system, it tries to link to the system specified by the IMSID= parameter. If no IMS system matches the name specified with IMSID, the region tries the system specified with the ALTID= parameter.

1.4.8 IRLM: DXRJPROC procedure

The following parameters in Table 1-8 should be reviewed when moving to a sysplex data sharing environment.

Table 1-8 IRLM procedure parameters related to BLDS

Parameter	Value
DEADLOK	Specifies the local deadlock detection interval in seconds and the number of local cycles that are to occur before a global detection is initiated. This parameter must always be specified. The first value is a 1 to 4 digit number in the range 1-9999 that specifies the length in seconds of the IRLM local deadlock detection interval. The second value is a 1 to 4 digit number in the range 1-9999 that specifies the number of local deadlock cycles that must expire before global deadlock detection is performed. A reasonable starting value would be 1,1, which would be reduced if deadlocks need to be identified more quickly. The default set by IRLM Version 2.1 is 5,1. The first parameter is always changed to 5 if it is greater than 5 and the second parameter is always changed to 1 if specified otherwise. In data sharing environments, the IRLMs synchronize their DEADLOK parameters to the values present in the last IRLM to join the data sharing group. Via IRLM Version 2.1 APAR PQ44791, the ability to specify subsecond deadlock values is provided by allowing IRLMPROC DEADLOK value for the local deadlock frequency to be specified in milliseconds. The values can be between 100 and 5000 and represents the number of milliseconds used for the IRLM deadlock interval. The value displayed will be in even 100 millisecond increments. A new MODIFY irlmproc,SET,DEADLOCK= command is also provided to dynamically change the deadlock frequency.
IRLMGRP	Specifies the name of the XCF group to which this IRLM belongs. This use to be referred as the GROUP= parameter. The group name (IRLMDS by default) is used as the XCF group name. This name cannot start with SYS and cannot be the same as the LOCKTABL parameter.
IRLMID	Must be different for each IRLM in the same data sharing group and is entered as a decimal number between 1 and 255. The IRLM with the lowest ID number in the group becomes the global deadlock manager for the group. There is no default value.
IRLMNM	Specifies the four byte OS/390 subsystem name for the IRLM. All IRLMs in the data sharing group may use the same IRLM name to facilitate ease of moving IMS and FDBR regions without having to move the IRLM. Must be specified in the OS/390 subsystem name table and it must be unique on a given OS/390 image.
LOCKTABL	The parameter is not used when a CFNAMES CFIRLM statement is used by IMS. This parameter was available for pre IMS Version 5 systems which did not have the CFNAMES statement. Using the CFNAMES is recommended.
LTE	The number of Lock Table Entries goes from 0 to 1024 with each increment equal to 1048576 (must be a power of 2), in the Coupling Facility. If less than one unit, the value will be zero. This support was added with IRLM Version 2.1 APARs PQ44114 and PQ44791.

Parameter	Value
MAXCSA	Specifies the maximum amount of ECSA IRLM is to use for its dynamic control blocks. The value must be specified as a decimal number from 1 to 999, in multiples of 1 MB. A specification of 3 states that IRLM can use 3 MB of extended common service area. This parameter must always be specified. However, the parameter is ignored when the PC=YES parameter is specified. With PC=YES, the storage allocation amount is specified on the RGN= parameter in the IRLM startup procedure. Via IRLM V2.1 APAR PQ07327 - IRLM is able to provide warning message DXR175E to users when IRLM is not able to obtain storage and DXR176I when the IRLM short on storage situation has been relieved. Via IRLM V2.1 APAR PN92333 - IRLM provides 'MODIFY irlmproc,STATUS,STOR' command support for users to monitor IRLM storage usage. Via IRLM V2.1 APAR PQ12126 - console command ‘F irlmproc, set, CSA=nnn' can be used to change MAXCSA value without bringing IRLM down
MAXUSRS	Specifies the maximum numbers of users that you believe will connect to the data sharing group. A value from 2 to 32 can be specified. This initial allocation of locking resources depends on the number of entries and the maximum number of users, so some assessment of available resources is made when an IRLM joins the group. However, nothing is done to keep the number of users below the specified value, and structure rebuild may occur when the 7th connection occurs, if the MAXUSRS is set to 8, and when the 23th connection occurs if MAXUSRS was set 24. Table 1-9 presents the number of bytes used in each lock entry. With IRLM Version 2.1 APARs PN64702 and PN76352, the lock table and record list both receive 50% of the space allocated to the structure by default.
PC	If PC=YES is specified, the IRLM lock control blocks reside in the IRLM private address space storage, and the space switching program call (PC) instruction is used to obtain addressability to them. If the value NO is specified, the IRLM locks reside in ECSA. The default for this parameter is PC=NO.
PGPROT	This support was added with IRLM V2.1 APAR PQ35083. It specifies whether the IRLM is to place load modules that are resident in common storage into OS/390 page protected storage during initialization. If the load modules are in page protected storage, any application that attempts to overlay the modules is terminated. The default value is YES.
SCOPE	Must be GLOBAL or NODISCON for intersystem sharing. NODISCON performs the same function as GLOBAL but the IRLM will not disconnect from the lock structure. This is mostly used where batch jobs have a separate IRLM. By using NODISCON there is the additional benefit that database authorization is not lost in the case of multiple IMS failures allowing restart to obtain authorization before all IMSs have completed restart.
TRACE	Specifies whether the IRLM is to turn on traces. Tracing is initialized at IRLM startup. The default is TRACE=NO. **NO** Traces are not captured unless you use the MVS™ TRACE CT command. **YES** Initiates IRLM tracing.

Table 1-9 is useful in estimating the value you want to apply to the MAXUSERS IRLM startup parameter.

Table 1-9 Comparison of the number of sharing IRLMs to lock size entries

Maxusers	Lock entry width	Entry layout
2 - 7	2 bytes	1byte for global, 1 bit reserved, 6 bits for shared users
8 -23	4 bytes	1byte for global, 1 bit reserved, 22 bits for shared users
24 - 32	8 bytes	1byte for global, 1 bit reserved, 32 bits for shared users

To obtain IRLM ARM support, DXRRL0F1 must be in the LINKLIST. If not, then an error would occur when restart is attempted. Also the OS/390 dispatching priority of the IRLM address space should be set very high.

1.5 Comments on data sharing overhead

Before implementing IMS block level data sharing it is very important to produce a performance benchmark for both development and production IMS systems.

There are a few important data sharing overhead plateaus that you should be prepared to experience:

- **Moving from PI to IRLM**

  This should result in a small increase in CPU but running the IRLM trace will impact heavily utilized systems.

- **Use of the OSAM and VSAM cache structures**

  When the buffer invalidate structures are used the overhead depends on several factors:

  – Maintaining the OSAM and VSAM structures. The interest in a block or CI must be registered by creating or updating an entry in the directories.

  – Managing the buffer invalidation process. This is performed by the CF but the application owning region still run 100% busy.

  – The amount of extra DASD read I/O from the cross system invalidation processes.

  – The code path related to updating buffers in each system where invalidation refresh has occurred.

  – The management of common data in every subsystems buffer pools.

  Implement SHARELVL(2) or (3) with IRLM on one system before sharing across two systems. Changing SHARELVL to 3 will tell IMS that these databases can participate in N-way block level data sharing.

  The fact that you are still running only one system does not affect the amount or type of locking performed. This may create some lock conflicts for VSAM key sequenced data sets within one IMS system, which allows you to find these potential problems before you encounter other new lock conflicts. Block locks for VSAM entry sequenced data set and overflow sequential access method (OSAM) data sets do not conflict within one system. Block locks for VSAM key sequenced data sets may conflict.

- **IRLM SCOPE = GLOBAL and SHARELVL = 3**

  The utilization of the CF lock structure produces the largest jump in overhead associated with an IMSplex wide locking operation. Adding additional sharing systems beyond the second should result in minimal extra overhead.

Almost all of the data sharing overhead shows up the IMS dependent regions or DBCTL Database Resource Adapter (DRA) threads. The increase may be 100%, but this is not necessarily the “true” cost of data sharing to CPU cycles, because of the manner that Coupling Facility “synchronous” requests are internally managed. The IMS Monitor will show the increased CPU per transaction values in the Program Summary section (CPU/SCHED) and in the Call Summary section (non-iwait). Also the IMS type’07’ application terminated log record has accounting information of value here.

Overhead associated within the IRLM address space should be minimal unless there is contention for resources and very large lock control block chains have to be created and scanned.

The following facilities can be utilized to monitor for pre and post BLDS performance benchmarks:

- **HD Unload utility (DFSURGU0)**

  The number of roots in the database can be obtained from the statistics reported by the HD Unload utility (DFSURGU0). This might be useful in determining which databases may have locking conflicts due to database record lock conflicts.

- **DBD Source**

  The number of RAPs in a database can be calculated from information in the DBD. The RMNAME parameter on the DBD statement includes the specification of the number of RAPs per block and the number of blocks in the root addressable area.

- **LISTCAT reports**

  The statistics section of a LISTCAT report shows the number of records updated, inserted, deleted, and read. This is useful for determining whether a VSAM data set has high update activity.

- **IMS Monitor**

  The IMS Monitor contains a wealth of information about accesses to databases. The database buffer pool reports are useful in determining the amount of read and write activity to databases. The program IWAIT reports show activity to individual database data sets. Unfortunately, the data is not aggregated by database or data set. Most reports aggregate by PSB or region.

  Turning on the IMS Monitor trace should not degrade the performance of the system. Nevertheless, using a small block size for the IMS Monitor data set, or a small number of buffers for it, might cause performance problems. This degradation can be avoided by doing the following:

  - When not using dynamic allocation for the data set, always specify the BLKSIZE subparameter on the DD statement DCB parameter. For DASD, use half-track blocking sizes such as BLKSIZE=26624 for 3390s. For tape, use BLKSIZE=30720. When using an old data set without this specification, IMS will change the block size to the default. The default block size is calculated by IMS and will generally be from 1048 to 4096 bytes.
  - When using dynamic allocation for a tape data set, always specify BLKSIZE in the DFSMDA dynamic allocation statement. Use BLKSIZE=30720.
  - When using dynamic allocation for a DASD data set, always specify BLKSIZE in the DFSMDA dynamic allocation statement. For example, use BLKSIZE=26624 for 3390s. For more information review IMS Version 6 APAR PQ04670, which allows dynamic allocation of a preallocated monitor DASD data set (DISP=OLD).
  - Always specify the BUFNO parameter on the DD statement or on the DFSMDA dynamic allocation statement. Make the value at least 5. For systems doing over 100

transactions per second or with more than 10 concurrently executing BMPs, specify a larger value. BUFNO=20 should be adequate for any system. For example, one could use the following DD statement:

```
//IMSMON DD DSN=IMS.IMSMON,DCB=(BLKSIZE=27992,BUFNO=10),DISP=SHR
```

IMS Monitor buffers reside in ECSA. The use of 10 buffers with a 27992 block size would require less than 280 KB of ECSA.

Before running the monitor in block level data sharing environments, establish “normal” numbers using either Program Isolation (PI) or SCOPE = LOCAL IRLM. IRLM IWAITS are reported as PI IWAITS.

IMS Version 7 introduced facilities to constrain IMS Monitor data collection to include only specified resources over a user controlled period of monitoring duration. Also the IMS Monitor has been enhanced to obtain data on Fast Path resources although the IMS Performance Analyzer (PA) is required to report on the IMS Monitor gathered Fast Path data.

- **Deadlock reports**

  Log records of type X’67FF’ are created and automatically shipped to the OLDS after a deadlock event and a victim has been abended with a U0777. The File Print and Select utility DFSERA10 with exit DFSERA30 can be used to create formatted deadlock reports.

- **IMS lock trace**

  This trace is started via normal IMS /START and /STOP commands and the output can be externalized to either the OLDS or DFSTRAxx trace data sets. The trace can be formatted with DFSERA10 using exit DFSERA40.

- **The DBRC RECON listing**

  Each database has a global Data Management Block (DMB) assigned to it and it is needed to identify the database activity in traces.

- **The PSB trace**

  This trace is also started via normal IMS /START and /STOP commands and the trace goes to the OLDS as type x’5F’ records for formatting with DFSERA10 and exit DFSERA50. It can be used to analyze the DL/I call flow and results from a particular PSB related to an application without have to deal with any non-DL/I code.

- **RMF™ Coupling Facility reports**

  The RMF CF Activity reports show the CF call rates, access times to the CF, real and false contention statistics, and the number of buffers invalidated by cross invalidations.

- **IRLM commands**

  Some IRLM commands have been introduced via the IRLM Version 2.1 maintenance stream that assist in monitoring the system and changing specific settings:

  – F IRLMproc,STATUS,STOR as introduced with IRLM Version 2.1 APAR PN92333 to be able to snapshot storage consumption. Message DXR100I displays the current ECSA usage, and the ECSA high water mark since this IRLM started.

  – F IRLMproc,SET,CSA= as introduced with IRLM Version 2.1 APAR PQ12126 to change the MAXCSA allocation without bringing IRLM down.

  – F IRLMproc,SET,DEADLOCK= as introduced with IRLM Version 2.1 APAR PQ44791 to specify subsecond deadlock values.

1.6 Defining VSAM share options

When your database access method is VSAM, the declaration of the data set indicates to VSAM what degree of shared access is desired. The correspondence between the type of sharing and the SHAREOPTIONS parameters (in the DEFINE CLUSTER keyword) is shown in Table 1-10.

Table 1-10 SHAREOPTIONS parameter specifications

SHAREOPTION Value	Type of Sharing
(1 3)	No sharing, single updater, or multiple read.
(2 3)	Single updater ad multiple readers.
(3 3)	Multiple updaters and multiple readers. The first value indicates cross region and the second, cross system.

For databases that are to participate in block level data sharing and use the VSAM access method, you must include the SHAREOPTIONS (3 3) parameter when defining the data sets.

The RECON data set is also accessed as a KSDS and requires SHAREOPTIONS (3 3). Refer to *MVS/DFP V3 R3 Access Method Services for Integrated Catalog Facility*, SC26-4562, for preparation of access method services statements that declare and catalog the share options.

1.7 Set VSAM and OSAM data set dispositions to share

VSAM and OSAM data set definitions need their disposition changed to shared (DISP=SHR) via JCL DD Statements or the DFSMDA member. Remember to include the RECONs during this setup step as well.

For DEDBs, if the AREAs to be allocated are registered in DBRC, the information required to dynamically allocate the data sets is obtained from DBRC. You do not need to supply DFSMDA members for them. The data sets will be allocated with a DISP=SHARE or DISP=OLD depending on the following in DBRC:

- If SHARELVL(0), DISP=OLD is used.
- If SHARELVL(1,2 or 3), DISP=SHARE is used.

1.8 Adjust ACCESS=xx in the DATABASE macro

The ACCESS=xx values in the DATABASE macro has to be changed to reflect the sysplex data sharing environment.

The macro must be verified to make sure ACCESS=EX is not used. This sets the database access intent within a given subsystem to no sharing; first come, only served (is the default). Valid parameters are the following:

- ACCESS=UP (Update)
- ACCESS=RO (Read only)
- ACCESS=RD (Read)

For test purposes, the ACCESS=xx can be changed with the following command:

```
/START DATABASE dbx ACCESS=xx
```

The GLOBAL and ACCESS keywords are mutually exclusive. The /START command is rejected if both keywords are specified. Therefore this command only operates on each local IMS subsystem in the IMSplex when the ACCESS keyword is included.

1.9 Excluding a database from data sharing

A database whose data does not need to be shared with any other IMS system in the data sharing group can be exclusively owned by one IMS system, by specifying ACCESS=EX. The database should be registered with a share level of zero in DBRC.

1.10 Examine DBRC skeletal JCL

Skeletal JCL for archiving and Change Accumulation should be reviewed. Each IMS subsystem will have its own archive process; however, they will probably share the same skeletal JCL for archive. The subsystem ID (%ssid) is available to the GENJCL process and can be used to create different data set names for different IMS subsystems. The skeletal JCL supplied with IMS uses the following name for SLDSs:

```
IMS.SLDSP.%SSID.D%ARDATE.T%ARTIME.V%ARVERS
```

This provides unique names for different IMS subsystems. Because Change Accumulation is required for database recoveries with data sharing, Change Accumulation skeletal JCL must be in place.

1.11 Buffer pool use in a data sharing environment

If you have an IMS system using a certain amount of megabytes of buffer pools today, what size buffer pool will you need for a block level data sharing environment? The answer very much depends on the degree of real data sharing.

Presumably you would like to achieve a similar buffer hit ratio, so that the number of real I/Os is roughly the same on the Parallel Sysplex as it was on the single system. If you split a single IMS system into multiple systems, all accessing the data with the same access pattern, you need the same buffer size to achieve the same buffer hit ratio on each system. This is because, although each of the N number of systems you split the single system into processes 1/N times the original number of transactions per second, they access all the data with the same pattern. Therefore, to get the same hit ratio, you need the same buffer pool size. This means that in this (worst) case, you have N times the original buffer pool size in total.

It may be possible to reduce the total buffer pool size if you can direct transactions that use the same data to separate systems; in other words, minimize data sharing between systems. If you know that SET A transactions largely or exclusively use a subset A of data, and SET B transactions largely or exclusively use a subset B of data, then you may be able to set the systems up so that the actual data sharing is minimal. In this case, you could split up the original buffer pool assignment between the subsystems.

There is some loss of symmetry in this situation as not all transactions run on all systems. This may complicate workload balancing, transaction routing for scheduled outages, and recovery from unscheduled outages. Design your routing mechanism to allow rerouting to cover these situations. Also your buffer pool allocations should also be able to cope with unforeseen rerouted transactions database accesses.

It may be most practical to migrate to data sharing gradually, measuring the hit rates as the migration progresses, and reducing the size of the buffer pools. For example, if you migrate one transaction type to a different subsystem and the data reference pattern changes, you may be able to reduce the buffer pool size on the original system.

1.12 Calculate the size of the OSAM/VSAM XI structures

In this discussion, we examine the buffer invalidate cache structure rather than the OSAM database Coupling Facility data caching feature. Please refer to *IMS in Parallel Sysplex, Volume I: Reviewing the IMSplex Technology,* SG24-6908, for more information on OSAM database caching.

IMS uses two separate cache structures, one each for control over buffer validity for VSAM and OSAM local buffers. These cache structures contain no data and are used to maintain local buffer consistency between systems in the sysplex.

Minimum cache structure size is based on the total number of buffers in all IMS control region environments and IMS batch jobs that participate in data sharing and are registered to IRLM. If you define Hiperspaces for VSAM databases using the HSO/HSR and HSn parameters in the VSRBF subpool definition statement, calculate the Hiperspace™ buffers into the sizing.

The structures are identical in format, so to determine the size of each structure we use the same methodology. A buffer invalidate entry of 200 bytes is required for each individual block or control interval in a buffer anywhere in the sharing IMS subsystems.

For example:

- For VSAM, total up all the VSAM buffers specified in all sharing DFSVSMxx members (including Hiperspace buffers) or DFSVSAMP data sets for batch and multiply by 200.
- For OSAM, total up all the OSAM buffers for all IMS subsystems, including sequential buffers and multiply by 200.

The maximum amount of space calculation is based on the assumption that no local IMS buffer pools have common data.

1.13 Calculate the size of the IRLM lock structure

The IRLM lock structure is divided into two separate sections: the lock table and the record list. The IRLM structure size defined in the CFRM policy is the combined total of these two sections, rounded up to the next power of two.

1.13.1 Sizing the lock structure lock table

We calculate the approximate size of the lock table in three steps:

1. One-half of the structure space is devoted to the lock table. The sizing of the lock table is based on the average number of locks held at any one time by IRLM.

   The lock hold time is close to the transaction response time and the lock rate is the transaction rate times the number of locks per transaction. So the calculation is:

   Avg # of active locks = (transaction response time) * ((transaction rate) * (locks/transactions))

   Where, the transaction rate is the total of ALL the data sharing IMS subsystems.

Transaction response times and transaction rates can be determined through the IMS monitor and RMF reports. It is difficult to obtain accurate values of locks per transaction in a data sharing environment if the current (non-data-sharing) environment is the source of the ratio, due to the differences in the scope of the object that is locked. A rule of thumb is to double the number of locks for a data sharing environment with full-function databases.

Recommendation: If you are not currently using data sharing, double the locking values to estimate the number of data sharing locks for full function databases.

2. Since lock contention has a direct effect on throughput rates, it should be kept at a minimum. A percentage of contentions for the lock table slots will be false. A false contention of up to 0.1% of the total number of locks requests is tolerable. The recommended number of entries for the lock table is:

   # of entries per lock table = (avg # of locks held) / 0.001

   Recommendation: We recommend a 0.1% rate of false contentions compared to total lock requests as a suggested performance target.

3. Each entry in the lock table contains 1 byte plus 1 bit for each IRLM using the structure. The setting of these bits indicates whether the corresponding XES holds at least one lock that hashes to this entry in the table. But, as detailed in the MAXUSRS discussion in this chapter (1.4.8, “IRLM: DXRJPROC procedure” on page 7) as the number of IRLMs increases, so does the space allocated to each lock entry. Assuming you seek a 0.1% false lock contention rate and you have a 2-byte-wide lock table, then for each held lock, you need 2 * 1000 = 2000 bytes of lock TABLE.

   The last calculation in Step 3 is, therefore:

   Size of the lock table = # of entries in each lock table * the width of each entry (2, 4 or 8)

   If more IRLMs connect to the structure than originally specified in the MAXUSRS= parameter, IRLM rebuilds the lock structure as required. This effectively increases the width of a lock table entry, therefore reducing the number of lock-table entries and reducing production throughput during the rebuild phase.

   Recommendation: Multiply the number of entries in the lock table by the number of bytes per entry (2, 4, or 8) to obtain the size of the lock table.

1.13.2 Sizing the lock structure record list

The other half of the structure is used for lists of MODIFY locks. Modify locks are those that protect updates. Whenever an application program uses a DBPCB with a PROCOPT that allows updates, the lock request on the database record is flagged with a modify attribute. Beside the full function database record locks, all full function block locks, and some fast path control interval locks have the attribute modify set.

These locks are kept in the Coupling Facility to provide a backup copy to maintain integrity and availability. In the event of an IRLM failure, all surviving IRLMs read these record-list entries for the failed IRLM and construct in-storage retained locks. Should an application on another IMS request one of these retained locks associated with earlier activity on the failed IRLM, the lock request is rejected and the application is abended with a user 3303 abend code.

Each entry is approximately 140 bytes long and a minimum size of the record list should be large enough to list all the modify locks held, since too few entries cause application abends U3307.

With IRLM Version 2.1 APAR PN64702, the record list receives 50% of the allocated space, so that doubling the size of the lock table provides an estimated size for the lock structure. Using the example above, with a 2-byte-wide lock table entry and a 0.1% false contention rate, one needs 2000 bytes of lock table for each held lock, which is far larger than the 140 bytes per record list. There is therefore no need to size the record list table.

Recommendation: Use transaction rates from peak processing periods to determine the lock structure size. Overestimating sizes causes no problem, but you need to use a size that meets or exceeds the requirement for both the lock table and the record list.

With IRLM Version 2.1 APAR PQ44114, you have control over the allocation percentage of the physical CF storage that is to be used for the lock record list. By updating the LTE keyword at IRLM startup, other IRLMs in the group will update their control blocks during deadlock processing. This percentage allocation can also be modified via the MODIFY command. The value specified in the “MODIFY irlmproc, SET,LTE=” command does not take affect unless the IRLM is the first to join the IRLM group during normal group startup or during a structure rebuild.

1.13.3 Example of IRLM lock structure sizing

Assume we have the following data available from our use of traces and monitors:

- 256 transactions per second have been measured on the single IMS system.
- The average internal response time is 0.1 second
- There are, on average, 9.45 lock requests per transaction. This includes the data sharing locking rate uplift mentioned earlier.
- We have four data sharing subsystems accessing the coupling of the original IMS facility, with equal workload distribution. Therefore, we expect 256/4 = 64 transactions/second on each IMS data sharing partner. But, in reality, the total number of transactions active is 256, since all IMS subsystems share the same lock table.

Using Figure 1-1 as a guide, let’s calculate the estimated size of our lock table size, and therefore the total space allocation for the IRLM structure.

- The number of transactions active at any one time is:
  - (256 transactions/second * response time of 0.1 second) = 26 transactions
- Let's assume that the locks are held for the life of each transaction, (a worst case situation):
  - (26 active transactions * 9.45 lock requests/transaction) = 246 locks held at any point.
- In order to obtain a false contention rate of 0.1%,
  - (246 locks / 0.001) = 246 KB locks
- Since we have four data sharing subsystems accessing the CF lock structure, the lock table must have the following size:
  - (246,000 lock table entries * 2 (bytes/lock table entry) = 492,000
- Since the space allocation for the record list table is half of the total allocated lock structure:
  - (492 KB for the lock structure * 2) = 984 KB for both the lock table and record list tables
- MVS will round specified size to next 256 KB boundary so round up to 1 MB, and specify that as the SIZE value in the Structure statement in the CFRM policy for the IRLM structure.

Figure 1-1 Lock structure sizing calculation

Using the process laid out in the example, you can initially estimate the lock table size. If too big or too small (look at the RMF III CF Reports of false contention rates to determine this) just rebuild it. After monitoring, make the CFRM policy parameter INITSIZE equal to the most accurate value.

Currently, we recommend making the initial IRLM lock structure size 32 MB to 64 MB for large, active production environments.

New function IRLM V2.1 APAR PN67253 adds a feature for dynamic expansion and contraction of the lock structure allocation within the Coupling Facility, known as *structure alter processing*. XES dynamic reconfiguration support allows structure sizes to be altered without any subsystem or data outage.

1.14 Defining the OSAM, VSAM and IRLM CF structures

The OSAM and VSAM buffer invalidation structures are not persistent structures and do not have persistent connections. When all connections to them are removed, the structures are deleted by the system. The next time they are built, it is with the specifications in the current CFRM policy.

1.15 Update DFSVSMxx with CFNAMES parameters

You specify the structure names via control statements in one of the following:

- The IMS.PROCLIB member DFSVSMxx
- The DFSVSAMP data set in DLIBATCH or DBBATCH procedures

The following is an example of the CFNAMES statement needed in the DFSVSMxx PROCLIB member to define the data sharing structure names:

```
CFNAMES,CFIRLM=IMOA_IRLM,CFVSAM=IMOA_VSAM,CFOSAM=(IMOA_OSAM,10,1)
```

All keywords (CFIRLM, CFVSAM and CFOSAM) must be coded in the CFNAMES parameter, but null values for the CFOSAM and CFVSAM keywords are allowed. If you specify only the IRLM structure name, the environment defaults to two way data sharing using the notify protocol. If there is no CFNAMES statement, a SCOPE=GLOBAL or NODISCON IRLM will use its LOCKTABL = specification for its lock structure name.

Also, although the structure names are user defined, each name used in the CFNAMES parameter must be the same as the STRUCTURE NAME( ) in the DEFINE CFRM POLICY execution of PGM=IXCMIAPU.

IMS initialization includes a verification that all IMS subsystems in the data sharing group are using the same names for the three structures in their CFNAMES statement. If an IMS subsystem attempts to join a data sharing group using any name that differs, its initialization will fail. Once an IMS subsystem has identified to an IRLM, these values are retained by the IRLM. The IRLM must be stopped and restarted to change these values.

1.16 Registering databases at SHARELVL(2)

SHARELVL is an optional parameter in the database or AREA record in the RECON data set. It specifies a level of data sharing for which authorized subsystems can share a database.

SHARELVL(2) allows a database to be shared across IMS systems using the same IRLM. Some installations use SHARELVL(2) to find potential block level sharing lock conflicts before they implement a lock structure and SHARELVL(3).

The best time to register databases at SHARELVL(3) is after the IRLM has been implemented but before sharing with another IMS subsystem is needed. The combination of the use of the IRLM and SHARELVL(2) or SHARELVL(3) causes level 3 and 4 locks to be used for KSDS block locks. This could cause locking conflicts or deadlocks to occur within one subsystem. Using SHARELVL(2) or SHARELVL(3) for one database at a time is an easy way to discover if there are potential locking problems for these databases. It is easy to revert to SHARELVL(1) if a problem exists that must be addressed. SHARELVL(1) eliminates block locks. To change the SHARELVLs the DBRC command /RMCHANGE can be used, as follows:

```
/RMCHANGE DBRC='DB DBD(ORDERDB) SHARELVL(2)
```

DBRC replies with its corresponding command input and a series of DBRC messages:

```
CHANGE.DB DBD(ORDERDB) SHARELVL(2)
CONDITION CODE=nn
```

1.17 Update IRLM start procedures for global data sharing

Change the following parameters in the IRLM startup procedure and give them meaningful values:

- SCOPE=GLOBAL or SCOPE=NODISCON
- LOCKTABL=IM0A_IRLM (name of a sample lock structure, or leave blank). We recommend using the CFIRLM value in the CFNAMES statement.

- MAXUSRS=7 The MAXUSRS could be set to 7 if less than 7 subsystems are going to participate in BLDS.
- IRLMGRP=IM0AIRLM (This is the sample XCF group that this IRLM belongs to in the examples contained in this book).
- DEADLOK=(1,1) (Note with APAR PQ44791 on, you can specify value less than 1 second).

Refer to Table 1-8 on page 8 for a review of the parameters that are being changed now.

1.17.1 Startup the IRLM procedure

In our example, the procedure name is IM1AIRLM. From our preparation phase, the IRLM name would exist in the current IEFSSNxx member in SYS1.PARMLIB.

After starting the IRLM, you can check its status by issuing the following commands:

- F IM1AIRLM,STATUS

  Displays the IRLM status.
- F IM1AIRLM,STATUS,ALLI

  Displays other IRLMs in the group. At this point, only IM1AIRLM with ID=1 would be identified.
- F IM1AIRLM,STATUS,ALLD

  Displays lock status in any partner IMS subsystems (which, at this point, do not exist).
- F IM1AIRLM,STATUS,STOR

  Displays the current and maximum specified allocation for CSA and ECSA storage for this IRLM.
- F IM1AIRLM,STATUS,MAINT

  Displays the IRLM maintenance levels for (most of) IRLM load modules.

1.18 Starting the CFRM policy

Assuming the existence of a CFRM policy named POLICY1 defining OSAM and VSAM cache structures names IM0A_OSAM and IM0A_VSAM, you can start the policy with the following command:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=POLICY1
```

To list the policies, you can use the following command:

```
D XCF,POL
```

You can display the status of the new structures with the following commands:

```
D XCF,STR,STRNM=IM0A_OSAM
D XCF,STR,STRNM=IM0A_VSAM
```

When IMS is started, the CFRM policy associated with the above structure names should be active. The structures may be rebuilt while in use. When this is done, the size in the current CFRM policy is used. Since rebuilding the structure while it is in use is disruptive to the system, we do not recommend it as a normal procedure.

In Figure 1-2 the OSAM and VSAM buffer invalidate structures have been introduced to the Coupling Facility.

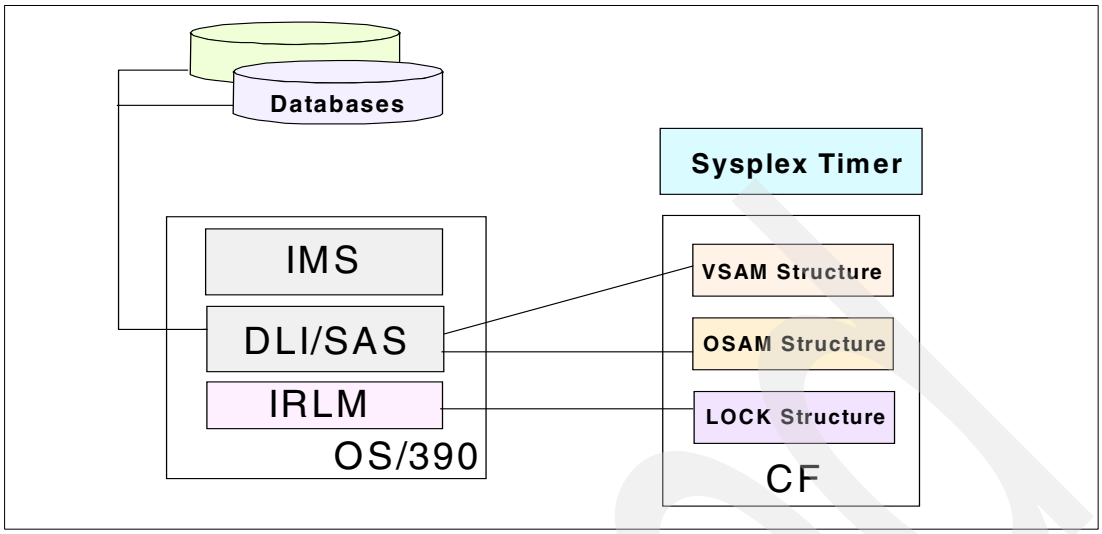

Figure 1-2 Addition of the OSAM and VSAM cache structures to the CF

The OSAM and VSAM XI structure disposition is DELETE, which indicates that when there are no defined connections to the structure, the structure is deallocated.

1.19 Registering databases at SHARELVL(3)

This will cause all locks for the databases to be placed in the lock structure. Although no new lock conflicts will occur, the overhead of the CF accesses will be added to the system.

The databases can be registered one at a time or in groups. This will limit the effects at any time.

To change the SHARELVLs the DBRC command /RMCHANGE can be used, as follows:

```
/RMCHANGE DBRC='DB DBD(ORDERDB) SHARELVL(3)
```

DBRC replies with its corresponding command input and a series of DBRC messages:

```
CHANGE.DB DBD(ORDERDB) SHARELVL(3)
CONDITION CODE=nn
```

Do not enter /RMCHANGE while a job is accessing the database. Stop the database using a /DBRECOVERY GLOBAL command before modifying any RECON record for that database. Otherwise, IMS rejects the command with an error message.

Check compatibility of SHARELVL(3) with the following parameters:

- DFSMDA
- PROCOPT=
- ACCESS=
- VSAM Shareoptions (must be set at (3 3))

1.20 Cloning your IMS subsystems

To clone your IMS subsystems across a sysplex, first determine how many unique IMS RESLIB data sets you need. That is, determine what IMS definitions if any need to be different for each IMS subsystem and cannot be overridden during execution. Then perform

IMS system definition to create your RESLIBs and put them on shared DASD. Use defaults or common coding during stage 1 as much as possible so you can override these definitions during execution.

As an example of a default definition overridden during execution, specify the APPLID as follows in the COMM macro:

APPLID=(IMS,IMS2,IMS3)

Then override “IMS” in the IMS procedure for IM2A using APPLID1=IM2A. See *IMS Installation Volume 2: System Definition and Tailoring*, GC27-1298, for more information on parameters that can be overridden at execution time.

Figure 1-3 shows the addition of another cloned IMS subsystem before it is connected to the IMSplex. Although the RECON is shared, the cloned system does not have any connections to CF structures and can access only databases that are exclusively aligned with it.

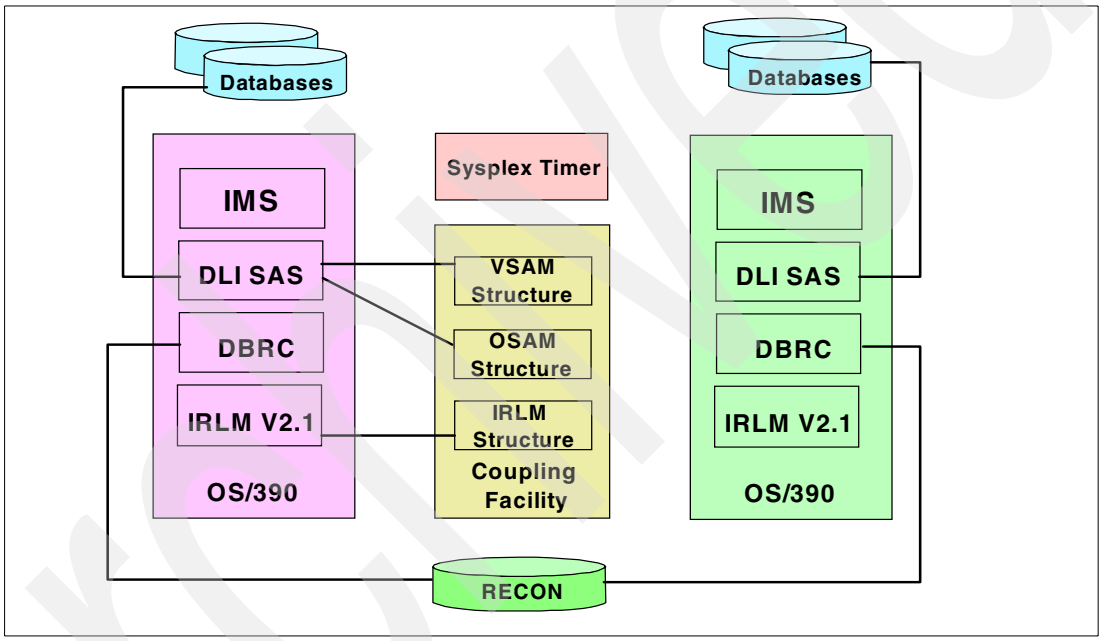

Figure 1-3 Introduction of a second IMS cloned subsystem before entry into the IMSplex

1.21 Starting the second IMS subsystem

Although some of the diagrams within this chapter may have led you to believe that multiple IMS data sharing subsystems are operational, it is only at this point we are ready to initialize the second data sharing partner.

The second IMS must have it own IRLM, using identical IRLMGRP and LOCKTABL names in its startup procedure to other IRLMs in the group. This is shown in Figure 1-4.

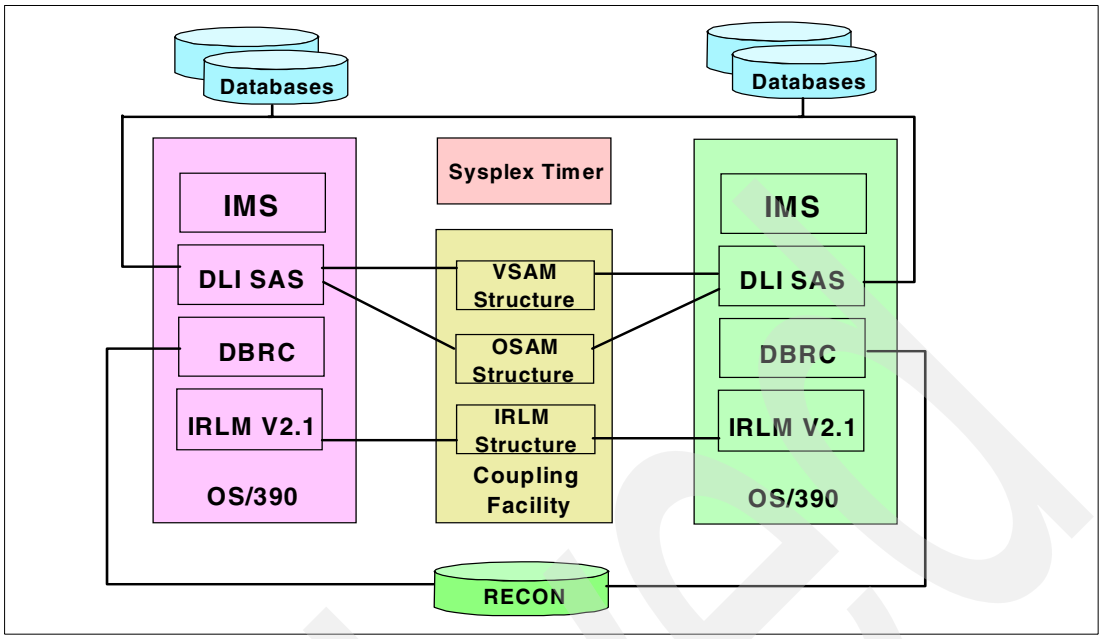

Figure 1-4 Initializing the second cloned IMS subsystem into the IMSplex

1.22 Testing data sharing with simple applications

At this point, two IMS subsystems with the capacity for block level data sharing are active and you should be able to exercise this facility using in-house test applications or members of the IMS IVP family.

1.23 IMS implementation for VSO data sharing

A series of steps are necessary to prepare for and perform VSO data sharing:

- **Define the cache structures in the CFRM policy**:

  Using the administrative data utility (IXCMIAPU), define the CFRM policy that will be stored in the CFRM couple data set. One or two store-in cache structures may be defined for each AREA, thereby providing a high availability characteristic equivalent to MADS. An example of the format for the structure definition is the following:

  ```
  DEFINE STR(name) SIZE(4K increments) PREFLIST(CFnames) EXCLLIST(CFnames)
  ```

  – **STR(name)**

    The 16 byte structure name. Since two structures may be defined for each AREA, we recommended including the AREA name in the structure name. The AREA name is used as the default if the structure name is not defined to DBRC for this AREA.

  – **SIZE**

    The size of the structure. For PRELOADed areas, it must be large enough for the direct portion of the AREA. The Reorg UOW, CI0 and CI1 (the DMAC) are not loaded. The size for non-PRELOADed AREAs is user determined. When the CI size is less than 4 KB, use 4 KB when calculating the structure size, as data elements in the structure are not shared by CIs. For CIs larger than 4 KB, the actual size, a multiple of 4 KB, should be used.

The sizing calculation that can be followed is shown in Figure 1-5.

DEDB AREA VSO Cache Structure Sizing

- If, from the DBDGEN the defined parameters are:

  UOW=(10,2),ROOT=(1000,100),SIZE=4096

  There are 10 CIs per unit of work and 1000 UOWs
  The 200 value is the size of each directory entry.
  The 4096 is the size of each data entry in the structure

- Then the size calculation will be:

  SIZE=(10*1000) * (4096+200) = 42,964,296 = 41,957KB

  41,957KB / 256KB = 163.9 and round to 164

- Multiply 164 by 256KB = 41984KB

This is the minimum required size for a preloaded VSO structure so SIZE(41984) should be specified

Figure 1-5 Sample DEDB AREA VSO cache structure sizing exercise

If multiple structures are defined for a particular area, the structure sizes must be identical.

– **PREFLIST**

  Define which CFs are preferred for containing the structures.

– **EXCLLIST**

  Define which structures are not to be on the same CF.

Via IMS Version 7 APAR PQ50661, support for the OS/390 system-managed rebuild and automatic altering of structures for VSO is available. Automatic alter is explicitly specified with the ALLOWAUTOALT(YES) statement in the CFRM policy.

- **Activate the policy using the SETXCF command**

  The following is an example of the SETXCF command:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=CFRMPOL1
```

- **Identify the structure names for data sharing AREAs to DBRC**

  The VSO AREA and its structures are identified to DBRC through the use of the INIT.DB and INIT.DBDS commands. Here is an example:

```
INIT.DB DBD(CUSTOMER) SHARELEVEL (2|3)
INIT.DBDS DBD(CUSTOMER)
AREA(CUSTAR01)
VSO
PRELOAD
CFSTR1(VSOCUSTAR01A)
CFSTR2(VSOCUSTAR01B)
LKASID
```

- **Allocate the private buffer pools for the local cache in DFSVSMxx**

  IMS builds default buffer pools for users of VSO, but it is best to specify VSO buffer pools in the DFSVSMxx member. They are not used unless VSO areas with SHARELVL(2) or SHARELVL(3) are opened.

  Figure 1-6 presents the buffer pools used in a SVSO environment.

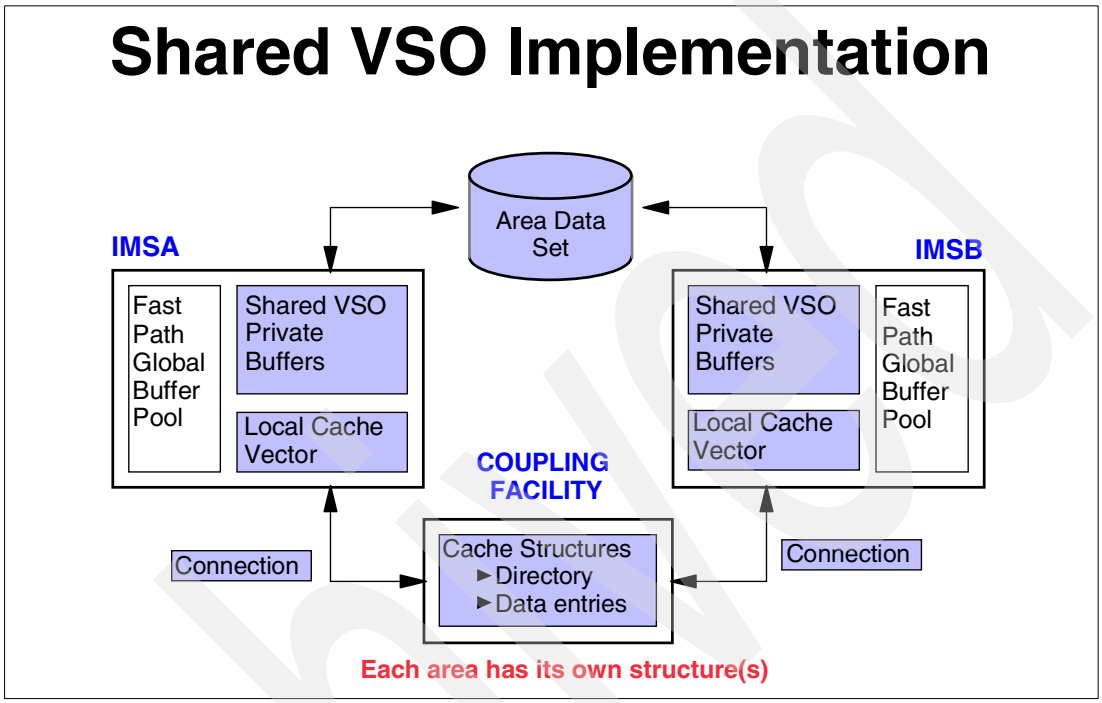

Figure 1-6 Buffer pools in a SVSO environment

Fast Path now uses two types of buffer pools for DEDBs:

– **The common buffer pool:**

  This pool has the traditional buffer pool function. It is defined in the FPCTRL macro, and is allocated in ECSA at control region initialization, and page fixed when needed.

– **The private buffer pool:**

  This is the IMS implementation of the local cache buffer pools, defined in DFSVSMxx with one DEDB statement for each private local pool. There can be multiple private pools and one pool can be shared by multiple AREAs if:

  - The AREAs have the same control interval size, or
  - The option chosen is LKASID or NOLKASID, and
  - A DEDB statement in DFSVSMxx does not assign them elsewhere

  If an AREA name or database description name is specified for any pool, that pool will be dedicated to the specified AREA or DEDB.

  These private buffer pools are allocated in ECSA during control region initialization and are page fixed when allocated. When VSO AREAs are initialized at SHARELVL (2 or 3), then the use of these pools is required. Here are the parameters of the DEDB statement in the DFSVSMxx member in PROCLIB:

  ```
  DEDB=(poolname,size,pbuf,sbuf,maxbuf,lkasid,areaname):
  ```

  poolname is the name of the private buffer pool.

  size specifies the buffer size (4096 bytes or 4 KB).

pbuf specifies the number of buffers in the primary.

sbuf specifies the number of buffers in the secondary.

maxbuf sets the maximum number of buffers.

lkasid | nolkasid sets lookaside buffer enablement ON or OFF for this particular pool.

areaname is the area name or dbd name using the pool.

- **Allocate the private buffer pools during IMS initialization:**

  Based on the each DEDB statement in DFSVSMxx, the private pools will be created in ECSA. The primary allocation as defined in the pbuf specification will be page fixed.

- **Connect to the CF structure:**

  The connection to the CF structure does not occur until the AREA is opened. The open occurs after initial system checkpoint for AREAs with the PREOPEN and PRELOAD options set. For AREAs without these options set, the opening occurs at the first call to the AREA.

- **Preload the AREA into the structure if requested:**

  During preload, a conditional lock is requested on each CI. If the lock is not successful because an application may own the lock, IMS bypasses the CI and it will be loaded by whoever owns the lock. Then the CI will be read into a private buffer.

- **Read and write from and to the CF structure**

  The data is written first to the cache at Phase 2 commit, then later to the DASD periodically.

- **Write to DASD (cast out) at system checkpoints:**

  The list of changed CIs is determined from the structure directory and it is through an XES cast-out request that only these CIs are written. The system obtains a cast-out lock on behalf of IMS that prevents any other IMS from casting out the same CI but does not prevent it from reading the data. This process is repeated for each unit of work until all changed CIs have been cast out.

In subsequent chapters we will discuss the SVSO operational topics, related commands and failure and recovery scenarios.

1.24 Fast Database Recovery implementation

Although Fast Database Recovery (FDBR) is an optional facility in the block level data sharing environment, we believe it is very useful and should receive attention. So FDBR is covered in this implementation chapter as well in the chapters about operations and recovery restarts.

There are no unique FDBR data sets and FDBR shares the IMS data sets in use by the active IMS that FDBR is tracking. Note that you should have one tracking FDBR address space per active IMS system in the sysplex. The tracking FDBR address space should be running on a different system image than the active IMS so that FDBR is able to perform its function in a situation when the LPAR where the active IMS is running, fails.

1.24.1 FDBR startup procedure

The FDBR startup procedure shown in Example 1-1 is very similar to the IMS subsystem startup procedure.

Example 1-1 The FDBR startup procedure

```
//       PROC RGN=OM,SOUT=A,DPTY='(14,15)',
//            SYS='IMSO.',SYS1='IM1A.',SYS2='IMOA.',
//            RGSUF=O1S,PARM1='FD1A',PARM2=
//IEFPROC EXEC PGM=DFSMVRCO,DPRTY=&DPTY,
//            REGION=&RGN,
//            PARM='FDR,&RGSUF,&PARM1,&PARM2'
//STEPLIB  DD DSN=IMSPSA.&SYS2.MDALIB,DISP=SHR
//         DD DSN=IMSPSA.&SYS2.SDFSRESL,DISP=SHR
//PROCLIB  DD DSN=IMSPSA.&SYS2.PROCLIB,DISP=SHR
//JCLOUT   DD SYSOUT=(A,INTRDR)
//JCLPDS   DD DSN=IMSPSA.&SYS2.PROCLIB,DISP=SHR
...........
```

In procedure FDBRA, the &RGSUF parameter with value '01S' identifies member DFSPB01S which contains a few parameters of FDBR interest:

- **FDRMBR=xx**

  Identifies a new PROCLIB member with FDBR related parameters (DFSRDRxx).

- **SUF=y**

  Should be the same as the active IMS.

- **IMSID=zzzz**

  May be the same as the IMSID of the active IMS or set differently. FDBR uses the IDs of the FDBR+IMSID for the IRLM identifier.

- **ARC=01**

  Allows the FDBR region to perform automatic log archiving, just as non FDBR control regions do, upon conclusion of its recovery process. This was introduced in IMS Version 6.1 with APAR PQ11478.

Other parameters have the same meaning as for the active IMS.

Via IMS Version 7 APAR PQ66507, and IMS Version 8 APAR PQ66531, IMS is changed to obtain the following DL/I related storage pools in FDBR extended private storage for an FDBR region:

- PSB work pool (PSBW - execution parameter: PSBW=)
- CSA PSB pool (DLMP - execution parameter: PSB= or the sum of CSAPSB= and DLIPSB=, whichever is greater)
- DMB pool (DLDP - execution parameter: DMB=)
- DB work pool (DBWP - execution parameter: DBWP=)

This will reduce the amount of ECSA storage that a FDBR region uses.

The IMS startup procedure and FDBR

If IMS is initialized with the FDRMBR=FD value, this will direct IMS to follow the settings in DFSRDRFD.

Use of the DFSVSMxx member and FDBR

Choose the DFSVSMxx member using VSPEC in JCL or member DFSPBxxx. Some specific activities in an FDBR region are associated with these values:

- The CFNAMES, OLDSDEF, and WADSDEF values must the same as for the active IMS.

- The IOBF, POOLID, VSRBF, RESVPOOL, OPTIONS(VSAM) and DBD values can be set differently for FDBR.
- The page fix options of IOBF and POOLID always are set to NO in FDBR.
- DLITRACE, NODYNALLOC, OPTIONS (for tracing) SBONLINE are always ignored by FDBR.

The DFSFDRxx member

This member of IMS.PROCLIB exists for use by both the active IMS and FDBR. The parameters in our example are shown in Example 1-2.

Example 1-2 FDBR startup parameters

```
ACTIVEIMSID=IM1A,
GROUPNAME=FDBRIM1A,
TIMEOUT=10,
AREA01=NORECOV
```

The parameters can be described as follows:

- **ACTIVEIMSID**

  This is the IMSID of the active IMS subsystem being tracked. The IMSID value in the IMS and DL/I startup procedures must be matched by this ACTIVEIMSID value if those started tasks are to be tracked by FDBR.

- **GROUPNAME**

  This is the name of the XCF group to which this active IMS and tracking FDBR belong. It is a unique combination for this IMS FDBR pair. If it is not specified, a default name is created by using the ACTIVEIMSID= parameter and the prefix FDBR. For example if the active IMSID is IM1A, the default XCF group name is FDBRIM1A.

- **TIMEOUT**

  TIMEOUT is specified in seconds and is the interval to time-out processing associated with XCF monitoring. FDBR will wait this number of seconds before determining a time-out status for the tracked IMS. Valid values are between 3 to 9999 with a default of 60. This value applies only to IMS time-out status from XCF. For log surveillance, FDBR uses a five second time-out value and a three second delay interval before calling DBRC to check whether an OLDS switch has occurred or there has been a problem related delay.

- **AREA01**

  This parameter is associated only with FDBR regions and specifies whether FDBR applies to Fast Path DEDB AREAs defined as SHARELVL=0 or 1. With a default value of NORECOV; unless RECOV is specified, SHARELVL=0 or 1 AREAs will not be recovered by FDBR. The reason for the existence of this parameter is to reduce the storage consumption by FDBR while tracking the active IMS.

  If this parameter is set, and RECOV and DEDBs with SHARELVL set at 0 or 1 are specified as VSO, then FDBR requires allocation of a data space and keeps the DB image there.

2

Implementing IMS shared queues

This chapter describes the tasks necessary to implement IMS shared queues:

- What needs to be done in IMS
- What needs to be done in CQS
- What needs to be done in OS/390 and in the Coupling Facility

Each of the tasks described in this chapter should have been identified in the planning phase.

We address each of these shared queues components, and the steps that must be taken for each to enable the shared queues environment.

2.1 What needs to be done in IMS

Figure 2-1 shows the components of a shared queues system. First we take a closer look at the IMS components, which only consists of IMS itself and the OLDS.

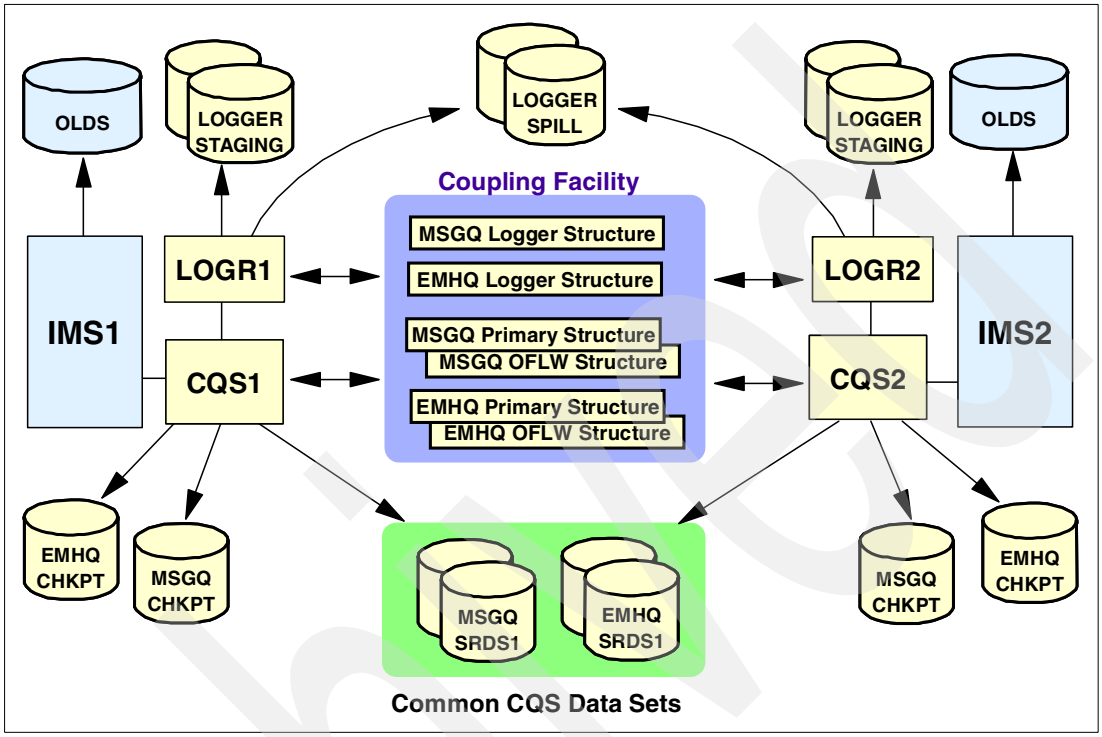

Figure 2-1 IMS shared queue components

2.1.1 IMS system definition

There are NO system definition requirements for IMS to run with shared queues. In the IMS system definition, the MSGQUEUE macro describes the conventional queue parameters. Although the message queue data sets are not used with SQ, and the QPOOL parameters may be specified separately for shared queues, this macro is still required for IMS system definition.

DD statements will be generated by stage 2 for the IMS control region JCL. These can be removed, although there is really no reason to remove them — IMS will not try to open them. If that same IMS is brought up without shared queues, then the only change that needs to be done is to an execution time parameter — no JCL changes (if the message queue data set JCL is still there).

The parameters defining the sizes of the queue buffers and the long and short message LRECLs become defaults for shared queues if the user does not specify them at startup time. The MSGQUEUE macro RECLNG minimum and default values have been modified for the short and long message queue data sets. The minimum values are now 392 and 1176, the default values are 504 and 2520.

Note: The new minimum and default values will not take effect until a minimum of a CTLBLKS system definition is performed.

Prior to IMS Version 8 and the use of Resource Manager, it is the IMS user's responsibility to ensure resources are defined consistently across multiple IMSs. A resource is defined as a transaction in one system and a logical terminal in another. IMS does not ensure that a resource defined or dynamically created in one subsystem has the same attributes as a resource of the same name in another subsystem. For example, a transaction may be defined as conversational in IMSA and as non-conversational in IMSB. An LTERM may be assigned to NODE1 in IMSA and to NODE2 in IMSB. With shared queues, users may log on to any of the IMSs. The physical and logical terminal definitions should be the same, no matter which IMS they sign on to. Each of above, as well as other inconsistencies, will cause problems in a shared queues environment. The Resource Manager in IMS Version 8 is introduced to address these kind of issues.

If the IMS systems are identical, they can be cloned. That is, a single IMS system definition can be used for all of the IMSs, and the SDFSRESL library can be shared. However, if the systems are not cloned, and separate system generations are done, then great care should be taken to ensure that resources that are common to each IMS are defined exactly the same way in each system.

2.1.2 IMS execution time parameters - DFSPBxxx

Shared queues is enabled simply by coding the IMS execution parameter SHAREDQ=xxx, either in the JCL directly or in IMS PROCLIB member DFSPBxxx. Figure 2-2 shows the definition needed in JCL. SHAREDQ=xxx identifies member DFSSQxxx in IMS.PROCLIB which contains the rest of the shared queue parameters for this IMS, such as the identify of the CQS subsystem to start and connect to. Existence of this parameter determines whether this execution of IMS will run with or without shared queues. SHAREDQ=, results in IMS using traditional queuing.

Within DFSPBxxx, there are some other shared queue parameters which may either be specified or defaulted to. We recommend that they be specified. The defaults are those defined in the MSGQUEUE macro and may not be the best for shared queues. If IMS is brought up with conventional queues, these parameters are just ignored. The other parameters in DFSPBxxx related to shared queues are the following:

- QBUFSZ
- QBUFMAX
- QBUFHITH
- QBUFLWTH
- QBUFPCTX
- LGMSGSZ
- SHMSGSZ

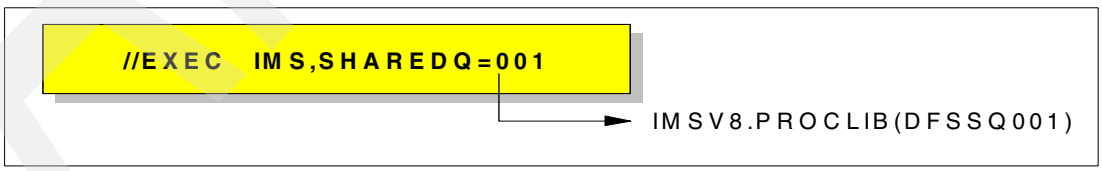

Figure 2-2 SHAREDQ=001 in JCL points to PROCLIB member DFSSG001

Some of these parameters are easily related to conventional queues. These parameters can be described as follows:

- **QBUF**

  Defines the initial number of queue buffers allocated in the QPOOL. Allowed values are from 3 to 9999 and the value for the BUFFERS parameter in the MSGQUEUE macro in the system definition is taken for the default.

- **QBUFSZ**

  Defines the size of the incore message buffers (QPOOL). Allowed value is from the value of LGMSGSZ to 30,632, and the default is the value of LGMSGSZ.

- **QBUFMAX**

  Defines the maximum number of queue buffers that the QPOOL can expand to. The maximum number that can be specified is 9999. However, if this parameter is not included, then the maximum is limited only by the virtual storage available. Since there are circumstances under which the QPOOL may be heavily used (for example, conversational processing, structure not available), it may be wise to NOT code this parameter and let IMS get what it needs.

  Note: This implies the following two things:

  - The QPOOL is still used with shared queues.
  - The QPOOL can be dynamically expanded.

- **QBUFHITH**

  This is the high threshold percentage for the QPOOL expansion. When the number of buffers in use reaches this percentage, the pool is expanded. The range for this value is from 1 to 100% and the default is 80%.

- **QBUFLWTH**

  This is the low threshold percentage for the QPOOL compression, if it had previously been expanded. When the number of buffers in use falls to this percentage, expanded buffers are compressed. The range for this values is from 1 to 100%, and the default is 50%.

- **QBUFPCTX**

  This is the percentage of the original QBUF specification that determines how many more buffers to acquire each time the pool is expanded. The range for this value is from 1 to 100% and the default is 20%.

- **LGMSGSZ**

  This is the size of the long message queue logical record (LRECL). Like with conventional queuing, IMS has long and short message logical records in the QPOOL. They must fit within a QBUF, so LGMSGSZ cannot be larger than QBUFSZ. If QBUFSZ is not specified, then this value is used as a default for QBUFSZ. Allowed values are from 1176 to 30632. The default value is taken from MSGQUEUE macro RECLNG=(...,size2) in IMS system definition.

- **SHMSGSZ**

  This is the size of the short message logical record. Allowed values are from 392 to LGMSGSZ value and the default value is taken from MSGQUEUE macro RECLNG=(size1,...) in IMS system definition.

Figure 2-3 shows the relationships between the buffers and the various parameters described.

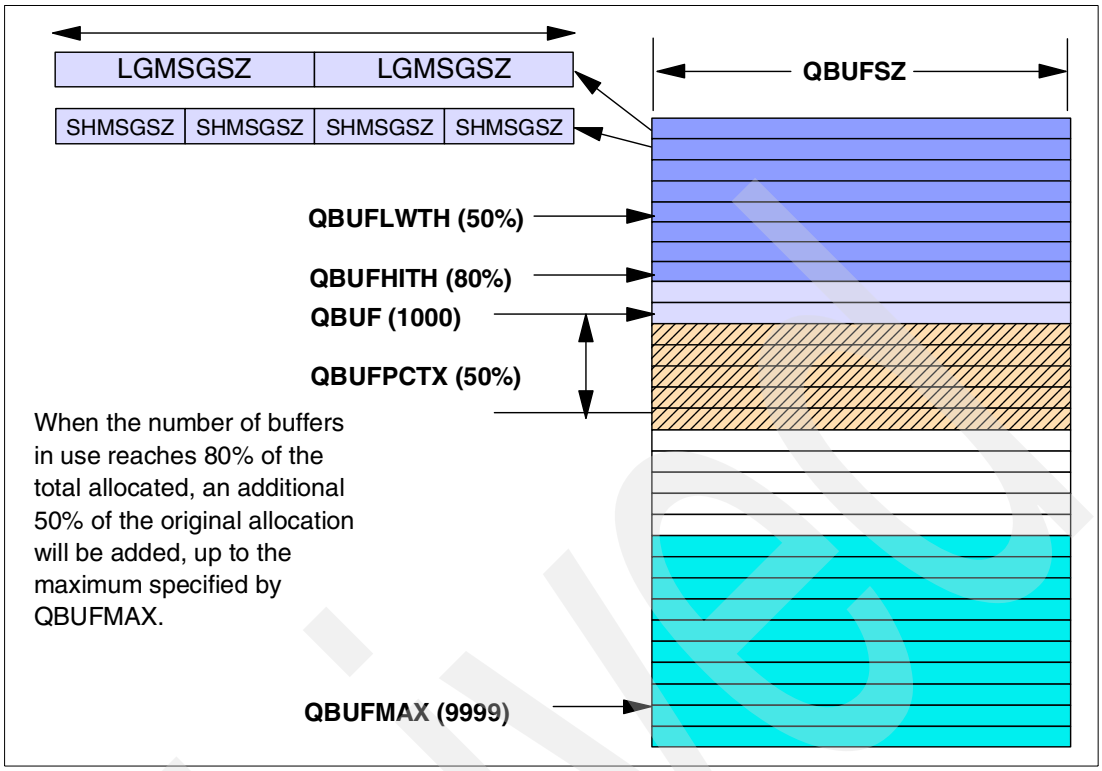

Figure 2-3 QBUFFER definition and expansion

2.1.3 DFSDCxxx PROCLIB member

Another execution time parameter which can be specified is DC=xxx. This parameter identifies PROCLIB member DFSDCxxx which, among other things, can be used to override the generated master terminal operator (MTO) and secondary master terminal operator (SMTO) definitions in the system generation. If the system generation is used to clone IMS systems, then this member needs to be used to make each IMS's MTO and SMTO unique.

Note: Each IMS must have its own unique DFSDCxxx parameter member.

The primary and secondary master terminal keyword options in DFSDCxxx:

- PMTO=name

  Overrides nodename of primary master terminal in system generation. For BTAM, PMTO keyword option is not required, and if defined it is ignored.

- PMTO1-PMTO8=(name,<MASTER>)

  Overrides 1st to 8th LTERM names of the primary master terminal in system generation. LTERM name with MASTER designation is the primary master. These LTERM names must be unique within a shared queues group.

- PMTOG=name

  Generic LTERM name for the primary master terminal. Default primary master LTERM name is DFSPMTO. This name should be the same for all IMSs in a shared queues group.

- SMTO, SMTO1-SMTO8, SMTOG
- These parameters are for the secondary master terminal accordingly. The LTERM name specified with the SECONDARY designation becomes the SMTO LTERM. The default LTERM name for the SMTOG is DFSSMTO.

Note: A cold start is required after changing the values in DFSDCxxx for the primary and secondary master terminals.

A new parameter is included, PMTOG and SMTOG. These are generic LTERM names for the PMTO and SMTO. Since an application program may be scheduled on any IMS system in the shared queues group, it may not know the name of the xMTO's LTERM. By coding this parameter, the application (or a message switch from a terminal) can send a message to the xMTO for the system it is currently running on (or logged on to).

Figure 2-4 shows an example of three IMSs all generated with the same xMTO definitions being made unique in DFSDC001/002/003 PROCLIB members. Note that they all have the same PMTOG and SMTOG values.

System defintion in ALL IMSs

```
TERMINAL    NAME=PMASTER,...
NAME        (PRIMSTR,MASTER)
TERMINAL    NAME=SMASTER,...      <Define in all IMSs>
NAME        (SECMSTR,SECONDARY)
```

DFSDCxxx members uniquely identify PMTO and SMTO names

```
DC=001      PMTO=PMAST1,PMTO1=(PMSTR1,MASTER),PMTOG=PMTO
            SMTO=SMAST1,SMTO1=(SMSTR1,SECONDARY),SMTOG=SMTO
DC=002      PMTO=PMAST2,PMTO1=(PMSTR2,MASTER),PMTOG=PMTO
            SMTO=SMAST2,SMTO1=(SMSTR2,SECONDARY),SMTOG=SMTO
DC=003      PMTO=PMAST3,PMTO1=(PMSTR3,MASTER),PMTOG=PMTO
            SMTO=SMAST3,SMTO1=(SMSTR3,SECONDARY),SMTOG=SMTO
```

Figure 2-4 Primary and secondary master definitions

2.1.4 DFSSQxxx PROCLIB member

SHAREDQ=xxx identifies a PROCLIB member DFSSQxxx. This member specifies information IMS needs to successfully connect to (and start if necessary) CQS and the shared queue structures. It contains the following parameters:

- **CQS**

  The name of the procedure to use to start CQS if CQS is not active when IMS comes up (S CQSPROC)

- **CQSSSN**

  This is the system name of CQS and must match the SSN specified in CQS's initialization parameters. Note that CQS is not really an OS/390 subsystem.

- **SQGROUP**

  This parameter may be from 1-5 characters. IMS will prefix it with the characters 'DFS™' and use the name to join an XCF group with other IMSs in the same shared queues group.

By doing this, IMS can communicate with other members of the same group using XCF signalling. All IMS members of the SQ Group must specify the same value for SQGROUP.

> **Note:** When CQS is defined, a CQSGROUP parameter must also be specified, but CQS prefixes the name with 'CQS' so that, even if these two parameters are the same, the XCF group names will be different.

- **MSGQ** and **EMHQ**

  Identify the names of the primary full function and EMH structures. Only the primary structures are defined to IMS. CQS knows which overflow structures (if any) are associated with each of the primary structures. These names must match those defined to CQS and must, of course, be defined in the CFRM policy.

  Note that, if Fast Path is included in the system by the FPCTRL macro in the system definition, an EMHQ structure must be defined, even if the user has no intent to use it. This is because it is possible for the user to add EMH transactions using online change and, if there were no structure defined, a cold start would be required to add one.

- **WAITRBLD**

  This parameter indicates whether Fast Path EMHQ activity is to be quiesced during structure recovery. This applies only to Fast Path, and only to recovery — not structure copy. If WAITRBLD=N is specified or defaulted to, then Fast Path can continue to function during recovery. Note that this will, however, extend the recovery time.

Example 2-1 shows the sample contents of DFSSQ01Q member in our test environment.

Example 2-1 Sample DFSSQxxx member DFSSQ01Q

```
CQS=IM1ACQS
CQSSSN=CQ1A
SQGROUP=IMSOG
MSGQ=IMOA_MSGP
EMHQ=IMOA_EMHP
```

The Figure 2-5 shows the relationships between the IMS execution time parameters and the library members. It also shows which parameters in the IMS libraries must match each other. If IMSs are sharing DFSPBxxx, then SHAREDQ=xxx can be specified in the JCL.

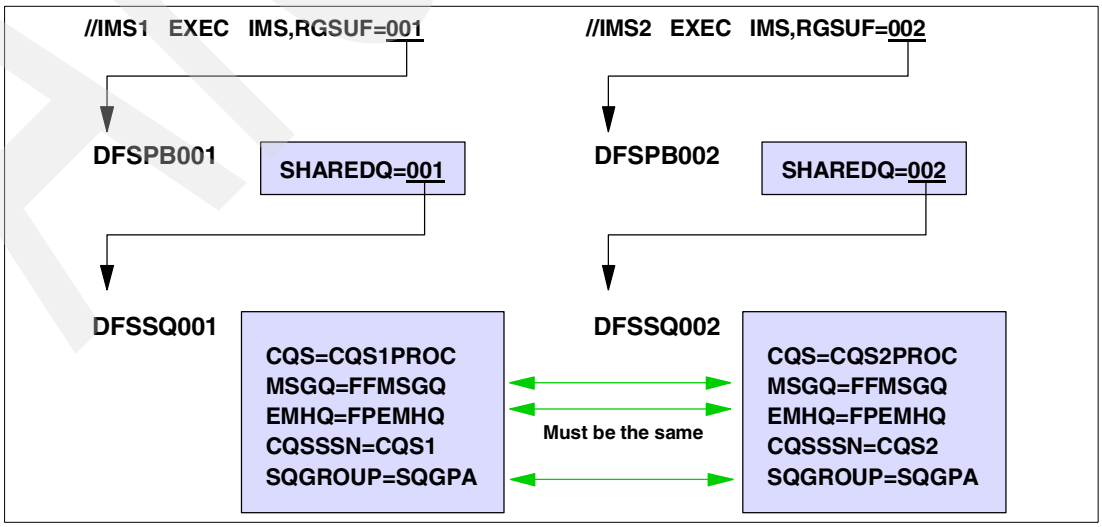

Figure 2-5 IMS JCL to library relationships

2.1.5 DFSVSMxx PROCLIB member

There are only two parameters in DFSVSMxx that have any significance for shared queues — these are the trace settings for the new QMGR and SQ traces.

> **Note:** QMGR=ON is the same as QMGR=MEDIUM. MEDIUM is the default for QMGR=ON. It is the same for SQTT.

2.2 Shared queues components

Figure 2-6 shows that there are some PROCLIB members which are used by CQS. Others are used by IMS. In this section we describe the PROCLIB members used by CQS. The CQS address space really contains both CQS code and Base Primitive Environment (BPE) code.

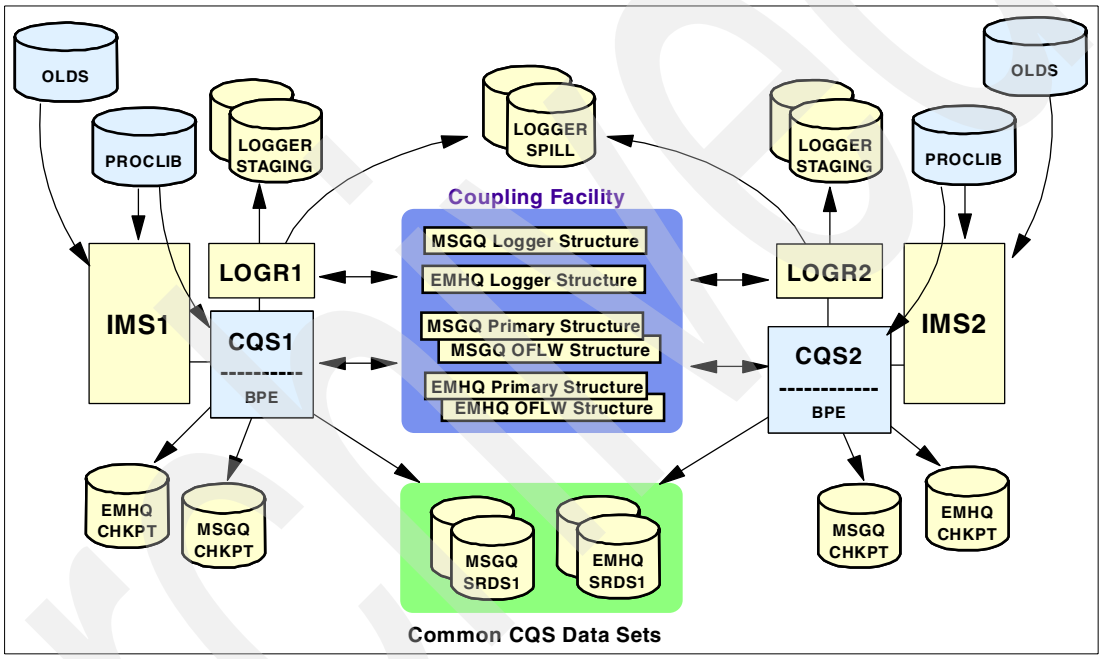

Figure 2-6 IMS shared queues components for CQS

2.2.1 BPE for CQS

BPE is intended to provide common services to not only CQS, but to other components of IMS that want to use these common services.

Currently, many of these functions are tightly integrated into the IMS code, often appearing multiple times in different modules. When it is necessary to change the way one of the base services works, it is difficult to find all the places and interfaces that may be affected by a simple conceptual change to how that function works. This makes both maintenance and enhancements much more difficult.

With the BPE architecture, these base services can be coded once, and an interface provided to any user that needs that service. Then when changes are required to the service, there is but one place to make the change, simplifying both maintenance and enhancements.

The CQS JCL contains one BPE parameter that may be coded by the user. *BPECFG=* identifies a PROCLIB member which contains BPE and BPE-user (in this case CQS)

execution parameters, exit names, etc. Example 2-2 shows part of the CQS JCL where BPECFG=BPECFG00 identifies the member in IMS.PROCLIB containing BPE execution options.

Example 2-2 Part of CQS JCL showing the BPECFG= parameter

```
//CQS  EXEC PGM=CQSINIT0,RGN=0M,
//     PARM='BPECFG=BPECFG00,....'
//STEPLIB DD  DSN=IMS.RESLIB,DISP=SHR
//PROCLIB DD  DSN=IMS.PROCLIB,DISP=SHR
```

The user sees some BPE externals, although there is little that needs or can be done. There are BPE modules in SDFSRESL, BPE messages, one CQS JCL parameter, some PROBLIB members, and the BPE dump formatter.

The BPE PROCLIB member identifies the language to be used for BPE and CQS messages (at this time only US English is supported), parameters to define trace types and levels, and another PROCLIB member containing a list of user exits. In most cases, it is not necessary for the user to write any user exit routines. The exception is the structure overflow exit. This is the exit which will be invoked when CQS performs overflow processing to approve or reject the candidates selected by CQS.

Note that, unless the user wants an overflow exit, it is not even necessary to code the BPECFG parameter. The BPE defaults are satisfactory for most situations. The most likely reason to change BPECFG is to satisfy a request from IBM service for a higher or different level of tracing, or to implement a new exit.

2.2.2 BPE PPT entry for CQS

The OS/390 program properties table must be updated. This is done in SYS1.PARMLIB(SCHEDxx). Prior to IMS Version 8 version of BPE (BPE 1.4), each IMS component using BPE had its own module for getting control in the address space when the job started. This module issued the a BPESTART macro to start BPE services and the module had to be added to the PPT to allow it to get control in key 7. For CQS, the module was CQSINIT0. Example 2-3 shows the entry needed for CQS before IMS Version 8.

Example 2-3 PPT entry for CQS prior to IMS Version 8

```
PPT PGMNAME(CQSINIT0)  /* MVS SUPPLIED VALUE IS - '6870FFFF00000000' */
        CANCEL         /* PROGRAM CAN BE CANCELLED         (DEFAULT) */
        KEY(7)         /* PROTECT KEY ASSIGNED IS 7                  */
        NOSWAP         /* PROGRAM IS NOT-SWAPPABLE                   */
        NOPRIV         /* PROGRAM NOT PRIVILEGED           (DEFAULT) */
        SYST           /* PROGRAM IS A SYSTEM TASK                   */
        DSI            /* DOES REQUIRE DATA SET INTEGRITY  (DEFAULT) */
        PASS           /* PASSWORD PROTECTION ACTIVE       (DEFAULT) */
        AFF(NONE)      /* NO CPU AFFINITY                  (DEFAULT) */
        NOPREF         /* NO PREFERRED STORAGE FRAMES    (NODEFAULT) */
```

With IMS Version 8, there are new address spaces that use BPE: Operations Manager (OM), Resource Manager (RM) and Structured Call Interface (SCI). To avoid having to add PPT entries for each of these components, IMS Version 8 has a new technique that can be used to start a BPE address space. BPE 1.4 provides a new initial startup module, BPEINI00 that can be used for any IMS component address space. This is true for the CQS too, CQS can be started using BPEINI00. It can also continue to be started using CQSINIT0 as the PGM= module, just as before BPE 1.4. The PPT entry needed for CQS and other address spaces

using the BPE services. It is similar to PPT entry for CQSINIT0, only the name of the module is now BPEINI00.

Example 2-4 PPT entry for CQS with IMS Version 8

```
PPT PGMNAME(BPEINI00)  /* MVS SUPPLIED VALUE IS - '6870FFFF00000000' */
         CANCEL        /* PROGRAM CAN BE CANCELLED         (DEFAULT) */
         KEY(7)        /* PROTECT KEY ASSIGNED IS 7                  */
         NOSWAP        /* PROGRAM IS NOT-SWAPPABLE                   */
         NOPRIV        /* PROGRAM NOT PRIVILEGED           (DEFAULT) */
         SYST          /* PROGRAM IS A SYSTEM TASK                   */
         DSI           /* DOES REQUIRE DATA SET INTEGRITY  (DEFAULT) */
         PASS          /* PASSWORD PROTECTION ACTIVE       (DEFAULT) */
         AFF(NONE)     /* NO CPU AFFINITY                  (DEFAULT) */
         NOPREF        /* NO PREFERRED STORAGE FRAMES    (NODEFAULT) */
```

2.2.3 CQS start procedure

The CQS procedure must be created. Parameters include PROCLIB members used for execution. Other parameters can be used to override specifications in these members, such as the CQS subsystem name (SSN), the CQS group name, and the use of ARM. Usually there is one CQS address space per IMS when shared queues is used, but starting from IMS Version 7, it is possible to have one CQS address space serving multiple clients (IMSs) within the same LPAR.

Example 2-5 shows how the CQS procedure might be coded. When IMS initializes, if CQS is not already started, IMS will start it using this procedure. When using the IMS Version 8 Resource Manager, the CQS address space should be started separately before the start of RM and IMS control region. Only BPECFG and CQSINIT need be coded in the procedure. All other parameters can be specified in the PROCLIB member pointed to by CQSINIT=xxx, where xxx is the suffix for the PROCLIB member CQSIPxxx.

To invoke this procedure by IMS, the PROCLIB member DFSSQxxx should have the parameter CQS=xxxx, where xxxx is the member name of the CQS procedure in PROCLIB.

Example 2-5 Start JCL for CQS

```
//IM1ACQS   PROC RGN=0M,SOUT=A,
//          BPECFG=BPECONFG,
//          CQSINIT=01Q,
//          SYS2='IM0A.',
//          SYS='IMS0.',
//          PARM1=
//*
//IEFPROC   EXEC   PGM=CQSINIT0,
//          REGION=&RGN,
//  PARM='BPECFG=&BPECFG,CQSINIT=&CQSINIT,&PARM1'
//*
//STEPLIB   DD     DSN=IMSPSA.&SYS.SDFSRESL,DISP=SHR
//*
//PROCLIB   DD     DSN=IMSPSA.&SYS2.PROCLIB,DISP=SHR
//*
//SYSPRINT  DD     SYSOUT=&SOUT
//SYSUDUMP  DD     SYSOUT=&SOUT
//*
```

CQSINIT=01Q in our example points to IMSPSA.IM0A.PROCLIB member CQSIP01Q. PARM1 is used to override values in CQSIPxxx. With IMS Version 8, it is possible to use the

common BPE initialization module for all the components that are using BPE services. Example 2-6 shows the CQS start JCL that is changed to use the BPEINI00 module as the first module to get control.

Example 2-6 CQS start JCL using the common BPE initialization module

```
//IM1ACQS   PROC RGN=0M,SOUT=A,
//          BPECFG=BPECONFG,
//          CQSINIT=01Q,
//          SYS2='IM0A.',
//          SYS='IMS0.',
//          PARM1=
//*
//IEFPROC   EXEC   PGM=BPEINI00,
//          REGION=&RGN,
//  PARM='BPEINIT=CQSINI00,BPECFG=&BPECFG,CQSINIT=&CQSINIT,&PARM1'
//*
//STEPLIB   DD     DSN=IMSPSA.&SYS.SDFSRESL,DISP=SHR
//*
//PROCLIB   DD     DSN=IMSPSA.&SYS2.PROCLIB,DISP=SHR
//*
//SYSPRINT  DD     SYSOUT=&SOUT
//SYSUDUMP  DD     SYSOUT=&SOUT
//*
```

2.2.4 CQSIPxxx (initialization) PROCLIB member

The JCL parameter CQSINIT=xxx identifies the CQS initialization parameter PROCLIB member CQSIPxxx. This library member sets the parameters for this execution of CQS. Any of these parameters can be overridden in the CQS JCL. CQSIPxxx specifies the following initialization parameters:

- CQS name
- CQS shared queues group name
- Identifiers for local and global shared queues parameter member names
- ARM restart parameter (Y/N)
- IMSplex name (starting from IMS Version 8)

The parameters in this member can be described as follows:

- **ARMRST**

  When set to 'Y', if CQS abends, it will be restarted by the Automatic Restart Manager. CQS must be defined in the ARM policy for this function.

- **CQSGROUP**

  This is the XCF group that all the CQSs will join when they start. This parameter should be the same for all CQSs in the SQG. This parameter may be the same as the SQGROUP parameter in the IMS PROCLIB member DFSSQxxx. CQS prefixes the value of this parameter with the characters 'CQS' to form an XCF group name CQSxxxxx.

- **SSN**

  This is the subsystem name for this CQS. It is not necessary to define this name to OS/390 in IEFSSNxx, since CQS is not really an OS/390 subsystem. It must, however, match the CQSSSN name specified in DFSSQxxx.

- **STRDEFG**

  This identifies the global structure definition PROCLIB member (CQSSGxxx) which defines global parameters for the shared queues structures. Parameters in this library

must be the same for all CQSs. It is recommended that all CQSs use the same global PROCLIB member.

- **STRDEFL**

  This identifies the local structure definition PROCLIB member (CQSSLxxx) for CQS structures. This member should be different for each CQS.

- **IMSPLEX**

  IMSPLEX is an optional parameter that specifies the IMSplex to which CQS joins (starting from IMS Version 8). All OM, RM, SCI, IMS, CQS and similar address spaces must specify the same to be part of the same IMSplex.

Example 2-7 shows a sample contents of the CQSIP01Q member.

Example 2-7 Sample CQSIPxxx member CQSIP01Q

```
ARMRST=Y
CQSGROUP=IMS0G
SSN=CQ1A
STRDEFL=01L
STRDEFG=00G
IMSPLEX(NAME=PLEX1)
```

2.2.5 CQSSLxxx local structure definition PROCLIB member

STRDEFL=xxx in CQSIPxxx identifies member CQSSLxxx in PROCLIB. These are the local parameters to identify, for each structure pair (MSGQ and EMHQ), the checkpoint data sets name and the checkpoint frequency. This member must be different for each CQS, since each CQS is having its own checkpoint data sets.

CQSSLxxx defines the following local shared queues parameters for each structure (MSGQ and EMHQ):

- Structure name
- CQS system checkpoint data set name
- Checkpoint frequency (log record count)

The parameters in this member can be described as follows:

- **STRNAME**

  Identifies the primary structure name for which the checkpoint data set and checkpoint frequency identified below apply. It must match MSGQ (EMHQ) parameter in DFSSQxxx.

- **CHKPTDSN**

  1 to 44 character checkpoint data set name. They are dynamically allocated during CQS initialization.

- **SYSCHKPT**

  Number of CQS log records written between system-generated checkpoints. Valid range is 200 to 2,147,483,647. Each CQS address space can specify a different value. There is no default. If SYSCHKPT is not specified, system checkpoints are taken on a restart, shutdown, after structure checkpoint and when requested by IMS operator command (/CQC SYSTEM STRUCTURE xxx).

Example 2-7 shows a sample contents of the CQSSL01L member.

Example 2-8 Sample CQSSLxxx member CQSSL01L

```
STRUCTURE
 (STRNAME=IMOA_MSGP,CHKPTDSN=IMSPSA.IM1A.MSG.CHKPT,SYSCHKPT=10000)
STRUCTURE
 (STRNAME=IMOA_EMHP,CHKPTDSN=IMSPSA.IM1A.EMH.CHKPT,SYSCHKPT=10000)
```

2.2.6 CQSSGxxx global structure definition PROCLIB member

CQSSGxxx defines the following global shared queue parameters for each structure (MSGQ and EMHQ):

- Primary shared queue structure name
- Overflow shared queue structure name
- Threshold percentage for overflow processing
- Minimum structure size to which CQS will connect
- Structure recovery data set names (2)
- Log stream name
- Average size of a data object

All CQSs in the same shared queues group share this member. If this is a Fast Path system, the parameters in this library member are repeated for the EMHQ structures, SRDSs, and logs. The parameters in CQSSGxxx can be described as follows:

- **STRNAME**

  This is the name of the primary structure. It must have been defined in the CFRM policy. Note that this parameter does not indicate whether this is a MSGQ structure or an EMHQ structure. That relationship is identified in the IMS member DFSSQxxx (MSGQ= and EMHQ=).

- **SRDSDSN1** and **SRDSDSN2**

  These are the structure recovery data sets for the named structure.

- **LOGNAME**

  This is the name of the log stream to be used for logging CQS activity related to this structure. This name must match the log stream name defined in the LOGR policy.

- **OVFLWSTR**

  This is the name of the overflow structure to be used for the named primary structure. Only CQS knows this name — it is not defined to IMS as is the primary structure.

- **OVFLWMAX**

  This is the threshold percentage to be used as a trigger for overflow processing. It is based on a percentage of total data elements in the primary structure.

- **STRMIN**

  This is the minimum size structure that CQS will accept when connecting to a structure. If the Coupling Facility is storage constrained when the structure is first allocated, then the CF may allocate a structure smaller than acceptable by CQS. If it is smaller than this structure minimum specification, and if it is empty when CQS connects, then CQS will terminate with a U0014. If the structure contains data (for example, if STRMIN were changed between executions of CQS and the existing structure is smaller than the new STRMIN, CQS does not abend — it continues processing with the smaller structure. Specified in units of 4 KB blocks, the value may range from 0 to 524288 (a 2 GB structure) default is 0 (accept any size).

- **OBJAVGSZ**

  This is the average size of data object passed by IMS to CQS to put on the shared queue. CQS will add 16 bytes to this to account for its own prefix, so don't include it in your calculations.

  The message that IMS passes to CQS does include the IMS prefix plus the user data. It is not the full QBuffer, so if a 500 byte message is in a 1000-byte buffer, then only 500 bytes is passed. Also remember that only the contents of a single queue buffer are put into a list entry, so long messages don't count (except whatever part fits in a single QBuffer).

  A data object includes contents of 1 IMS QBuffer or 1 EMH buffer as shown in Figure 2-7. Do not include the 16-byte CQS prefix when specifying this size. One data object uses one list entry and one or more data elements.

  OBJAVGSZ is used by CQS to set list entry to data element ratio when creating structure. The sizes of list entry and the data element are as follows:

```
List entry    =    167  bytes
Data element  =    512  bytes
```

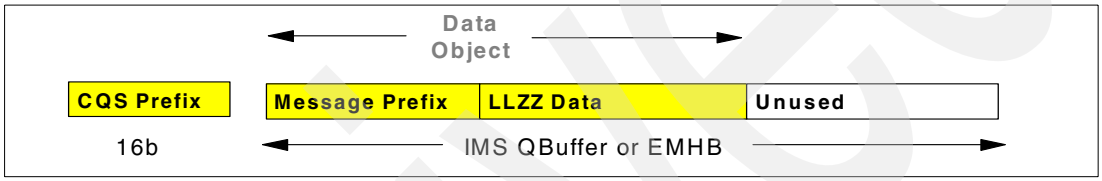

Figure 2-7 Data object

OBJAVGSZ is specified in decimal bytes and the valid range for the value is from 128 to LGMSGSZ. For EMH, OBJAVGSZ can never be larger than FPATH=size definition in TRANSACT macro in the IMS system definition.

Recommendation: Specify the OBJAVGSZ to be the average of the sizes of all the objects passed to CQS by a CQSPUT request. CQS adds its own prefix containing control information to every object placed on the structure. CQS adds the length of its prefix to the OBJAVGSZ value that you specify to get the true average object size. Therefore, OBJAVGSZ should reflect only the average size of the objects as they are passed to CQS, not the average size of the object on the Coupling Facility.

2.2.7 IMS and CQS library relationships

The Figure 2-8 shows the relationships between the various PROCLIB members and the IMSs and CQSs that use them. The CDS attached to the Coupling Facility is just to show that the structures are all defined in the CFRM policy in the CFRM couple data set. It does not mean that the CDS is connected to the CF.

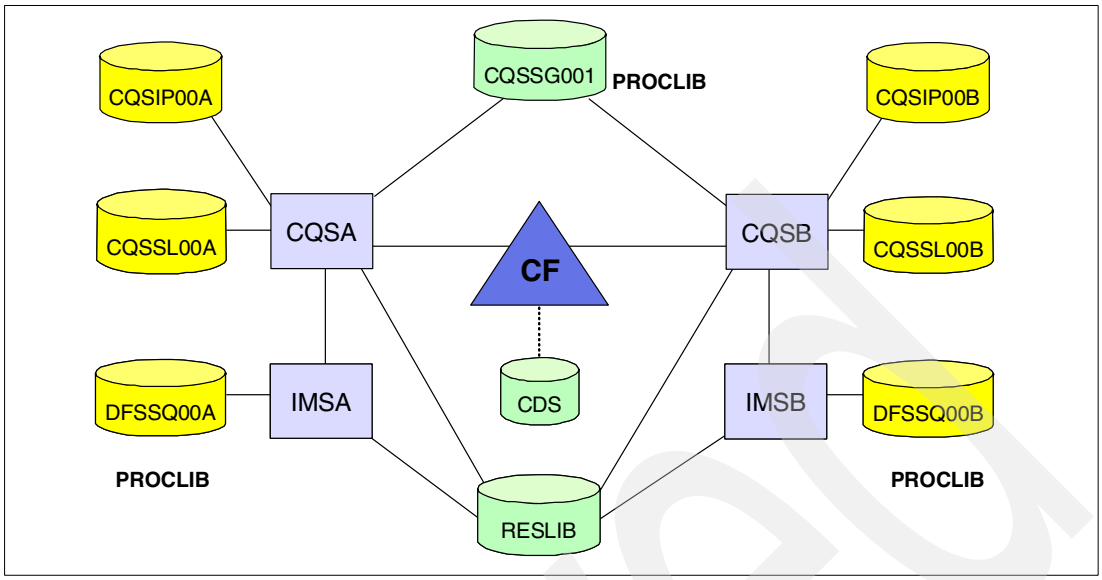

Figure 2-8 Library relationships between IMS and CQS

2.2.8 CQS PROCLIB member relationships

A sample PROCLIB member relationship is shown in Figure 2-9 and the connections to the IMS can look similar to Figure 2-10. The intent is to give you an idea how to start with two IMS shared queues members. If you have a need for more members on the shared queues group you need to draw a more complicated picture.

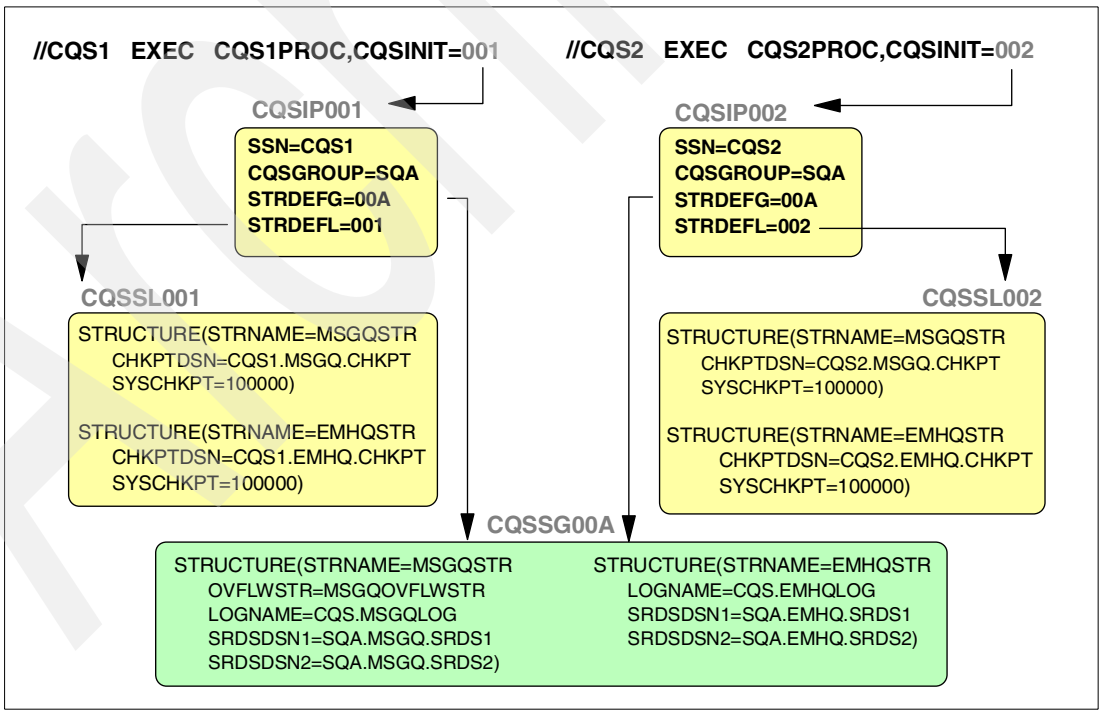

Figure 2-9 CQS PROCLIB member relationships

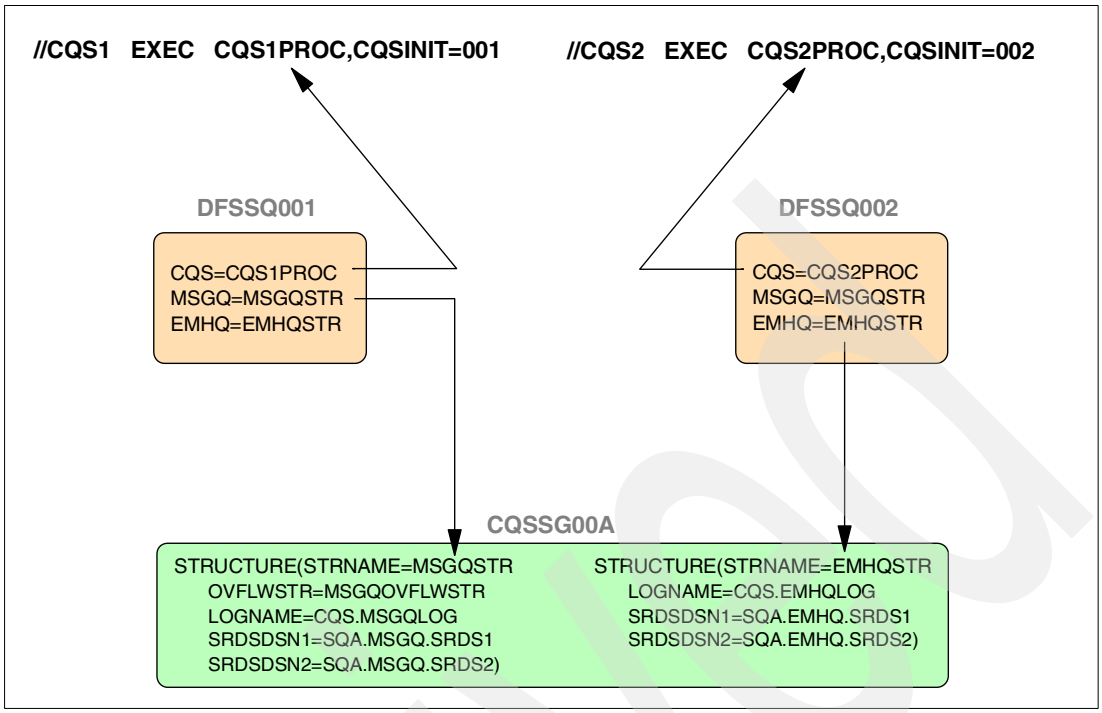

Figure 2-10 IMS and CQS PROCLIB member relationships

2.2.9 Shared queues structure sizing

In the CFRM policy we have to define the following structures for IMS shared queues:

- Primary and the optional overflow structure for full function messages (MSGQ).
- Primary and the optional overflow structure for Fast Path messages, if Fast Path is included in the system (EMHQ).
- System Logger structures for MSGQ and EMHQ.

The primary MSGQ structure is the main, and preferable, repository for IMS input and output messages. For Fast Path messages, it is the primary EMHQ structure. If the overflow structure is also defined, some messages may be moved by CQS to an overflow structure, when the user defined threshold is exceeded. If no overflow structure is defined, PUTs to the overflow queues are rejected. The overflow threshold percentage is defined in the PROCLIB member CQSSGxxx. This is determined by ratio of total number of data elements in use to total allocated elements. The maximum space allocated is determined by SIZE in the CFRM policy.

OS/390 System Logger is used by CQS to log all activity to and from shared queues structures. CQS writes log records to a log stream defined in the LOGR policy. The LOGR policy also identifies the System Logger structures that are used by the log streams. These structures are defined in the CFRM policy and looks like any other structure definition. One log stream and one structure each for MSGQ and EMHQ is needed.

Primary MSGQ structure sizing

Since overflow processing is not a general goal, sizing of the primary structure is important, at least to some reasonable value. Too small and we go into overflow processing or perhaps fill up; too large and we waste space in the CF. But, better to be too large than too small.

No matter how large you make it, you may not be able to make it large enough to completely avoid overflow processing. There will always be some queues which are not delivering messages for a variety of reasons. These are the ones which will go to overflow and may eventually fill overflow.

What we really need to know is, what is the maximum number of messages that may be on the queue at any one time, and how big are they? Although it is not easy to get accurate numbers, several sources of information may be used to get a good estimate. For example, if we know the maximum number of long and short QBuffers ever used by IMS, this might tell us how many QBuffers might be in use in a shared queues environment. Knowing how big the long and short LRECLs are would give us a maximum size of the data content of those LRECLs. This could be translated into the number of list entries and data elements we might need for the MSGQ structure. Of course, changes in transaction volumes would have to be taken into consideration, but this should give us a number to start with — then monitor it.

Remember that you can always code the INITSIZE parameter in the CFRM policy and, if more is needed, CQS will expand the size of the structure. So if you make a conservative guess and use it to set INITSIZE to conserve space, and then hedge your bet with a much larger SIZE, you should be alright until you get a better understanding of what the real requirements are, and then overflow will cover for any really bad guesses. Note also that the structure can be resized and rebuilt, all without taking down CQS or IMS.

Tip: If you don't code initsize, it defaults to 256k; even if you have autoalter, the increment steps are 256K as well. With the CFCC level 10 or later, you need to start with an initial size larger than 768K. That means even three increments of the default initsize 256K are not enough to fulfill the initial request and the result is that you cannot connect to the structure. This is due to the fact that the autoalter will only increment three times in a row, if this is not enough it stops increasing the storage.

So with CFCC Level 10 you must have INITSIZE large enough to cover this 768K plus EMC plus your message queue size needs. In our test we succeeded after we specified the INITSIZE to more than 1 MB. In most samples of CFRM policy, initsize is not coded.

Refer also to Flash 10118 at the following Web site:

http://www.ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/Flash10118

Overflow MSGQ structure sizing

The overflow structure is meant to allow IMS to continue processing while some queues become very long. These are shuttled off to the overflow structure but we would of course like to keep processing them as long as possible.

As with the primary structure, much will depend on experience as shared queues are used, and on the history of your IMS in a non-shared queues world. If your system has a tendency to fill up the queues, don't be too conservative.

Note that, at CQS initialization, the overflow structure is allocated, and then deleted. This is to determine whether there is space available for this structure should it be needed later. Of course, between the times CQS is initialized, and the time the overflow structure is actually needed, space may become insufficient in the CFs to support the overflow structure.

When queues are moved to overflow, about 20% of the data elements in the primary are moved each time overflow processing is invoked. So, the overflow structure must be at least 20% of the primary structure. If it happens again, there's another 20%. You should have additional space for more messages. Consider larger size if queues have history of frequent build-ups. Monitor the use and adjust the size accordingly. Overflow structure is not allocated

unless it is used. We recommend that the size of overflow be, at a minimum, the size of the threshold of the primary (default 70%) to accommodate the possibility that one queue name is causing overflow processing but maybe 100% is better.

EMHQ structure sizing

EMHQ structures should be much smaller than the MSGQs. Local processing avoids the queues altogether, and since all EMH input is response mode, there can never be more messages on the queue than there are terminals connected to the shared queues group, even if everyone had a global message on the queue at one time. Alternate PCB output (for example, output to printers) goes to the MSGQ structure.

A maximum size is required, this depends on number of TERMINALs and maximum size of input and output messages. Maximum size required is most likely to occur during IMS system checkpoint (IFP scheduling stops).

Overflow structure for EMHQ has the same function as overflow structure for MSGQ. The overflow structure may not be required if primary structure is set to maximum size or EMHQ structure checkpoint (all access to structure stops).

Logger structure sizing

Sizing the logger structure is an interesting exercise. Fortunately, it is not necessary to be even as accurate as the MSGQ structures. As a matter of fact, there is no accurate answer. The real answer is that is should be big enough to hold all the log data generated between the time off load begins and ends. Talk to the OS/390 systems programmer about performance implications of frequent off loads, but as long as the structure can continue to accept log data during off load, it will never fill up.

Note that it is impossible to make the structure large enough to hold the entire log stream, except perhaps for Fast Path where there is a predominance of local processing, and for a very small system, which would probably not be a shared queues system anyway.

2.2.10 Shared queues structure definitions in CFRM

Once we have decided how big the structures are going to be, they must be defined in the CFRM policy. In CFRM policy, a number of parameters are defined — some mandatory, some optional. The following parameters can be specified in CFRM policy:

- **NAME**

  Structure name that must match with the name specified in PROCLIB member CQSSGxxx. This parameter is required.

- **SIZE**

  Maximum size of structure in 1K units to which the structure can be altered or autoaltered. This parameter is required.

- **INITSIZE**

  Initial size allocated when the first client connects to the structure. It is increased incrementally to SIZE when OVFLWMAX reached.

- **MINSIZE**

  Minimum size of structure in 1K units to which the structure can be altered or autoaltered, defaulting to 75% of INITSIZE. If ALLOWAUTOALT(NO) is specified, the MINSIZE used will be 0.

- **ALLOWAUTOALT**

  Specifies if the automatic alter (autoalter) of the structure size is allowed to relieve the CF storage constraints. This parameter is optional, and the default is NO.

- **FULLTRESHOLD**

  This value is the percent full that will trigger autoalter, it is optional, and the default is 80%. This must be set to zero if you want to disable the structure full monitoring.

- **REBUILDPERCENT**

  When loss of connectivity reaches this percentage, all CQSs are notified by XES and structure rebuild is initiated. This parameter is optional and the default is 1.

- **PREFLIST**

  Identifies on which coupling facilities this structure may be allocated. This parameter is required. You should always specify at least two CF. The other CF is required for rebuild in case of failure of the preferred CF. XES will not build or rebuild a structure on a CF which is not in the PREFLIST.

- **EXCLLIST**

  Identifies structures which should not share the same CF. This parameter is optional and mutually exclusive with the ENFORCEORDER(YES) definition. This is rarely necessary with IMS structures - one usually tries to place structures in such a way as to balance the CF accesses across all available CFs. One consideration might be to separate the MSGQ structure from its System Logger structure so that if the CF fails, we don't lose both the structure and the logs. However, even this is not a requirement, since the logs are recoverable.

- **DUPLEX**

  Specifies if the system managed duplexing is DISABLED, ALLOWED or ENABLED. This parameter is optional and the default is DUPLEX(DISABLED).

Full description of the parameters can be found in the manual *z/OS MVS Setting Up a Sysplex,* SA22-7625. The Administrative Data Utility is used to define the CFRM policy.

Example 2-9 is an example of the IXCMIAPU JCL and CFRM policy definitions for the structures needed for IMS shared queues in our test environment. Note that we have defined only those parameters that are required for EMHQ and System Logger structures, and that we are using the default values for the other parameters. Note also that all the structures in the sysplex must be defined in the same CFRM policy. This example is showing only the structures needed for shared message queues.

Example 2-9 CFRM structure definitions for IMS shared queues

```
//CFRM093   JOB (999,POK),'CFRM',CLASS=A,REGION=4096K,
//            MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
/*JOBPARM SYSAFF=*
//STEP1    EXEC PGM=IXCMIAPU
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN    DD   *
     DATA TYPE(CFRM) REPORT(YES)
     DEFINE POLICY NAME(CFRM093) REPLACE(YES)

   ... other CFRM structure definitions


   /*-----------------------------------------------------------------*/
```

```
/* STRUCTURES FOR IMS SHARED QUEUES - FF MESSAGES                */
/*---------------------------------------------------------------*/

STRUCTURE NAME(IMOA_MSGP)
    SIZE(32768)
    INITSIZE(16384)
    MINSIZE(8192)
    ALLOWAUTOALT(YES)
    FULLTRESHOLD(60)
    PREFLIST(CF05,CF06)
    REBUILDPERCENT(10)

STRUCTURE NAME(IMOA_MSGO)
    SIZE(16384)
    INITSIZE(8192)
    MINSIZE(8192)
    ALLOWAUTOALT(YES)
    FULLTRESHOLD(60)
    PREFLIST(CF05,CF06)
    REBUILDPERCENT(10)
/*---------------------------------------------------------------*/
/* IMS SHARED QUEUES STRUCTURES FOR FAST PATH                    */
/*---------------------------------------------------------------*/

STRUCTURE NAME(IMOA_EMHP)
    SIZE(8192)
    PREFLIST(CF06,CF05)

STRUCTURE NAME(IMOA_EMHO)
    SIZE(8192)
    PREFLIST(CF06,CF05)
/*---------------------------------------------------------------*/
/* SYSTEM LOGGER STRUCTURES FOR IMS SHARED QUEUES                */
/*---------------------------------------------------------------*/

  STRUCTURE NAME(IMOA_LOGM)
      SIZE(4096)
      PREFLIST(CF06,CF05)

  STRUCTURE NAME(IMOA_LOGE)
      SIZE(12288)
   PREFLIST(CF05,CF06)
```

2.2.11 Security definitions for CQS structures

If access security for shared queues structures is desired, it must be defined in the security product. There are two ways to control access to CQS shared queues structures with a security product, such as RACF®. First, access by a CQS client can be controlled using profile name of CQSSTR.structure-name, where structure-name is the name of the primary structure. Note that only the primary structures need to be authorized. CQS will not request authorization for overflow structures. Example 2-10 shows an example of this. Second, access by any user of XES can be controlled by using a profile name of IXLSTR.structure-name, where structure-name is the name of the structure.

Example 2-10 CQSSTR Profile

```
RDEFINE FACILITY CQSSTR.IMOA_MSGP UACC(NONE)
PERMIT CQSSTR.IMOA_MSGP CLASS(FACILITY)ID(CQSUSER)ACCESS(UPDATE)
SETROPTS CLASSACT(FACILITY)
```

Since CQS clients use XES to access structures, the second way is recommended. It protects the structure from all potential users, not just CQS clients. Example 2-11 shows an example of this. To define a profile for a CQS primary structure named IM0A_MSGP, and to allow only user CQSUSER to connect to it, issue the RACF commands shown in the example.

Only users with USERID of IMSP are allowed to connect to the structure and manipulate it. Users would include IMSs, CQSs, and operators who issue SETXCF commands for the structure. If a user does not have authority to access a structure, the following message is issued.

```
IXL012I IXLCONN REQUEST FAILED, RETCODE: 00000008, RSNCODE: 0201084C
```

Example 2-11 XLSTR profile

```
RDEFINE FACILITY IXLSTR.IMOA_MSGP UACC(NONE)
PERMIT IXLSTR.IMOA_MSGP CLASS(FACILITY) ID(IMSP) ACCESS(ALTER)
SETROPTS CLASSACT(FACILITY)
```

2.2.12 Allocation of CQS data sets

For CQS you need to allocate the following system data sets:

- CQS system checkpoint data sets - two (MSGQ and EMHQ) for each CQS
- Structure recovery data sets - two (MSGQ and EMHQ) pairs for shared queues group

CQS system checkpoint data sets

The CQS system checkpoint data sets are very small, just a single CI in a VSAM ESDS; see Example 2-12. The data set names are identified in member CQSSLxxx (since each CQS has its own, they are defined in the local member). They are dynamically allocated by CQS when needed.

Example 2-12 VSAM definition for the system checkpoint data sets

```
DEFINE CLUSTER                                          -
    (NAME (IMSPSA.IM3A.MSG.CHKPT)                      -
    TRK(1,1) VOL (TOTIMK) NONINDEXED SHAREOPTIONS (2,3) -
    RECSZ(505,505) REUSE  CISZ (512))
DEFINE CLUSTER                                          -
    (NAME (IMSPSA.IM3A.EMH.CHKPT)                      -
    TRK(1,1) VOL (TOTIML) NONINDEXED SHAREOPTIONS (2,3) -
    RECSZ(505,505) REUSE  CISZ (512))
```

Structure recovery data sets

The SRDSs are the image copies of the message queues. They must be able to hold all of the messages in a structure pair at the time a structure checkpoint is taken, including both the primary and overflow structures.

Two data sets needs to be defined for each structure pair, as they are used alternately for structure checkpoint. Example 2-13 shows the sample control statements for defining these data sets. SRDSs are dynamically allocated when needed for structure checkpoint or

structure recovery, and then deallocated at the completion of the operation so that another CQS can allocate them if needed.

The data set names of these data sets is specified in the global PROCLIB member CQSSGxxx, since all CQSs must use the same SRDSs. These data sets should be sized large enough to hold all the data that would be in both primary and overflow structures if they were both full.

Example 2-13 Sample structure recovery data sets definitions

```
DEFINE CLUSTER                                          -
    (NAME (IMSPSA.IMOA.MSG.SRDS1)                      -
    CYL(5,1) VOL (IMS002) NONINDEXED SHAREOPTIONS (2,3) -
    RECSZ(32761,32761) REUSE  CISZ (32768))
DEFINE CLUSTER                                          -
    (NAME (IMSPSA.IMOA.MSG.SRDS2)                      -
    CYL(5,1) VOL (IMS003) NONINDEXED SHAREOPTIONS (2,3) -
    RECSZ(32761,32761) REUSE  CISZ (32768))
DEFINE CLUSTER                                          -
    (NAME (IMSPSA.IMOA.EMH.SRDS1)                      -
    CYL(5,1) VOL (IMS002) NONINDEXED SHAREOPTIONS (2,3) -
    RECSZ(32761,32761) REUSE  CISZ (32768))
DEFINE CLUSTER                                          -
    (NAME (IMSPSA.IMOA.EMH.SRDS2)                      -
    CYL(5,1) VOL (IMS003) NONINDEXED SHAREOPTIONS (2,3) -
    RECSZ(32761,32761) REUSE  CISZ (32768))
```

2.2.13 System Logger definitions in LOGR policy

At this point we have already defined the OS/390 System Logger structures for IMS shared queues in CFRM policy. The next task is to define the log streams in the LOGR policy.

The LOGR policy is defined in the Logger Inventory Couple Data Set. Unlike the CFRM policy, of which we can have many defined and activate the one we want, there is never more than one LOGR policy per sysplex, it has no name, and it is automatically activated.

The LOGR policy defines all the log streams in use within the sysplex, and makes the relationship between the log streams and the logger structures.

The logger structures are defined in the CFRM policy and the definitions look exactly like any other structure definition. The LOGR policy defines the log streams and the structures that log streams are using.

Example 2-14 shows the sample JCL for defining the log streams for IMS shared queues. JCL for defining the shared queues log streams in LOGR policy.

Example 2-14 Sample JCL for defining the log streams

```
//DEFLOGRP  JOB (999,POK),'LOGR POLICY',CLASS=A,REGION=4M,
//             MSGCLASS=X,TIME=10,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//STEP1    EXEC PGM=IXCMIAPU
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN    DD   *
     DATA TYPE(LOGR) REPORT(YES)

       DEFINE STRUCTURE NAME(IMOA_LOGM)
              LOGSNUM(1)
              MAXBUFSIZE(65272)
```

```
        AVGBUFSIZE(4096)
 DEFINE LOGSTREAM NAME(IMOA.MSG.LOGSTR)
        STRUCTNAME(IMOA_LOGM)
        HLQ(IXGLOGR)
        STG_DUPLEX(YES)
        DUPLEXMODE(COND)
        STG_SIZE(125000)
        LS_SIZE(125000)
        HIGHOFFLOAD(50)
        LOWOFFLOAD(0)

DEFINE STRUCTURE NAME(IMOA_LOGE)
        LOGSNUM(1)
        MAXBUFSIZE(65272)
        AVGBUFSIZE(4096)
 DEFINE LOGSTREAM NAME(IMOA.EMH.LOGSTR)
        STG_DUPLEX(NO)
        STRUCTNAME(IMOA_LOGE)
        HLQ(IXGLOGR)
        LS_SIZE(4096)
        HIGHOFFLOAD(50)
        LOWOFFLOAD(0)
```

Within the LOGR policy, you will find a *DEFINE STRUCTURE* statement. This is not really defining the structure as it is defined in the CFRM policy, it is really just naming the structure that will be referred to by the log stream definition, and providing some parameters for the use of the structure.

Note that *MAXBUFSIZE* is not quite 64K (65536). If 64K were used, the logger would build a log structure with 512-byte elements. When 65272 is used, the elements are only 256 bytes, making better use of space in the structure. You must specify the MAXBUFSIZE parameter value that is large enough to contain the largest log record written by CQS.

Recommendation: Specify the MAXBUFSIZE parameter as 65272 bytes.

The *LOGSNUM* parameter identifies the maximum number of log streams that can share this structure. For CQS, this should always be set to 1, or at least there should only ever be 1 log stream per structure.

DEFINE LOGSTREAM statements are for the definition of the log streams themselves. The parameters are described as follows:

- **STRUCTNAME**

  The name of the logger structure this log stream will use. It must match the name in the CFRM policy and in the DEFINE STRUCTURE part of the LOGR policy.

- **LOGSTREAM**

  Defines the name of the log stream. This name must match the LOGNAME parameter value specified in CQSSGxxx.

- **STG_DUPLEX**

  Defines whether we want to use the staging data sets to duplex the list entries in the logger structure. *NO* means that staging data sets are not used under any conditions. *YES* means that staging data sets may be used, depending on the next DUPLEXMODE parameter.

- **DUPLEXMODE**

  Determines under what conditions we will use the staging data sets. *UNCOND* means that they are used always. *COND* means that the staging data sets are used only if the logger structure is on a volatile CF. The user needs to be aware that use of the staging data sets will cause a performance degradation in the logging activity. The log write is synchronous, and both the write to the CF and the write to the staging data set must complete before the log write is considered complete.

- **STG_SIZE**

  This parameter is used to set the size of the staging data sets (in 4K increments - not 1K like the structure sizes). Although these data sets only hold the log data inserted by single CQS, if that CQS is the only one running, then it could possibly be as large as the structure. So, if you want to use staging data sets, make them as large as the logger structure.

- **LS_SIZE**

  Defines the size of the spill data sets. These should be at least at large as the structure, but should probably be as large as the user has DASD to support. The larger they are, the more space is available for the log stream. Each will be filled before the next is dynamically allocated.

- **HIGHOFFLOAD**

  The percentage of data elements in use that will trigger the off load process. The default is 80% but we recommend 50% — it gives System Logger more time to complete the off load before the structure fills up. This could possible happen if there was a period of high activity and the off load process didn't begin until 80%. There is no additional work done with the smaller number, it is just that less data is off load more frequently.

- **LOWOFFLOAD**

  Determines how much to offload. By setting this parameter to zero, we are asking System Logger to off load everything that is in the structure at the time the offload process begins.

There are additional SMS parameters which the OS/390 systems programmer will most definitely want to set.

2.3 Additional System Logger considerations

There is a lot more to be defined for the System Logger, and there are many users of the System Logger. So there are additional topics which, if you are interested in, you can talk to the OS/390 systems programmer about:

- SYS1.PARMLIB
- COUPLExx
- Logger inventory couple data set which contains LOGR Policy
- Security
- System Logger data set allocation
  - Spill data sets
  - Staging data sets

System Logger data sets

The logger data sets will be defined by the OS/390 systems programmer, probably without the help of IMS people, but here are some of the considerations they will have.

Be sure that enough space is allocated to hold all the log data between two structure checkpoints. Because the MSGQ structures are quiesced during structure checkpoint, you

may be able to take one only occasionally, such as once a day. So the log stream staging and spill data sets may need to be able to hold two days worth of log data. A lot depends on how frequently structure checkpoints can be taken.

Spill data sets

System Logger spill data sets are used to off load MSGQ and EMHQ log streams when structure reaches HIGHOFFLOAD threshold. They are single extent VSAM linear data sets. If it is SMS managed, which is recommended, consider defining the following SMS parameters in LOGR policy:

- DATACLAS
- STORCLAS
- MGMTCLAS
- SIZE

System Logger dynamically allocates up to 168 volumes for the spill data sets when needed. Starting from OS/390 Release 3, the user may specify the maximum number of volumes to be used. When spill data sets are full, structure is full. This should never occur in an IMS environment, because the log space is freed at structure checkpoint.

Staging data set

It is the same with the staging data sets. The OS/390 systems programmer will define them and System Logger dynamically allocates them if needed. They just need to know how big the logger structures are so that the staging data sets can be allocated large enough. Be sure to remind them that if only a single CQS is logging, then the staging data set will need to be as big as the logger structure.

2.4 Other Parallel Sysplex policies

These policies will have meaning for a lot more than just IMS, and so will be set by the OS/390 systems programmer, but they may need some input from IMS, especially with regard to ARM and where IMS and CQS can be restarted if either or both fail, or if OS/390 fails.

- System Failure Management (SFM) policy

  Enables automatic Sysplex recovery actions. Loss of connectivity, XCF status update missing, specify CONNFAIL(YES) to enable automatic rebuild.

- Automatic Restart Management (ARM) policy

  Defines how OS/390 is to handle IMS, CQS, IRLM, and OS/390 failures. Should they be restarted? If so, where? IMS, IRLM, and CQS can be automatically restarted by ARM. System Logger is not ARM-eligible. Each CQS should belong to the same restart group as its partner IMS.

  For more information of ARM, see *IMS in Parallel Sysplex, Volume I: Reviewing the IMSplex Technology,* SG24-6908.

- Workload Manager (WLM) policy

  WLM sets system parameters for performance. It is available to all members of XCF groups. It probably has no direct impact on shared queues, but it is used by VTAM® Generic Resources.

3

Implementing the Common Service Layer (CSL)

This chapter describes the tasks necessary to implement an IMSplex Common Service Layer:

- Setting up the Base Primitive Environment (BPE)
- Updating the CFRM couple data set (CDS)
- Setting up the Structure Call Interface (SCI)
- Setting up the Operations Manager (OM)
- Setting up the Resource Manager (RM)
- Setting up the IMS control region for CSL

3.1 Before the first IMS joins the IMSplex

There are a number of definitional requirements before an IMSplex can be established for the Common Service Layer. They include:

- Define the resource structure in the CFRM policy and activate the policy.
- Define all the (new) CSL address spaces, including the JCL and PROCLIB members; the name of the IMSplex must be defined to each.
- Update or create the CQS JCL and PROCLIB members to provide services for both RM and IMS shared queues; the IMSplex name must be defined for the RM and CQS function.
- Update the IMS PROCLIB members to include CSL parameters.

Before any IMS joins the IMSplex, no resource structure will exist.

3.2 Setting up a CSL environment

Figure 3-1 highlights the major requirements for defining the IMS Version 8 Common Service Layer (CSL) environment. These are:

- Create JCL for the address spaces.
- Create (or update) PROCLIB members.
- Define the shared queues and resource structures.

Four manuals contain information for defining this environment:

- *IMS Version 8: Installation Volume 2: System Definition and Tailoring,* GC27-1298
- *IMS Version 8: Common Queue Server Guide and Reference,* SC27-1292
- *IMS Version 8: Base Primitive Environment Guide and Reference,* SC27-1290
- *IMS Version 8:Common Service Layer Guide and Reference,* SC27-1293

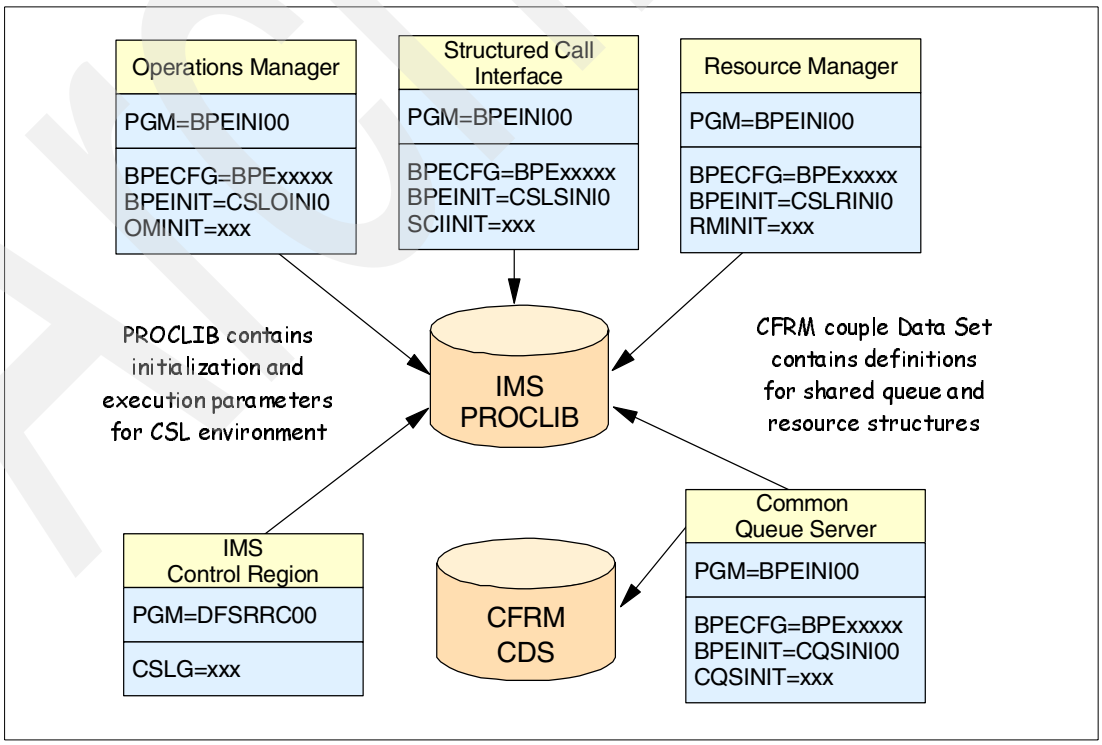

Figure 3-1 High level view of CSL definition requirements

3.2.1 CSL definitional requirements

Each of the following components has some definitional requirements:

- Base primitive environment (BPE)
- CSL address spaces
  - SCI (one per OS/390 image)
  - OM (one or more per IMSplex)
  - RM (one or more per IMSplex)
- IMS control region definitions for CSL
- For sysplex terminal management
  - CQS address space (one per RM)
  - Resource structure (one per IMSplex)
  - Shared queues (required for STM resource status recovery)
    - CQS address space
    - Shared queues structures
- OS/390 or z/OS®
  - Add BPEINI00 to program properties table (PPT)
- CFRM
  - Update CFRM couple data set
  - Add RM structure to CFRM policy

3.2.2 Checking the requirements

Setting up CSL requires new PROCLIB definitions and there are also some requirements for the hardware and software levels.

3.2.3 Basic rules

When setting up the CSL environment, the following rules might be helpful to simplify the process.

- All IMSplex members (IMS, CQS, SCI, OM, and RM) can share the same PROCLIB data set.
- All IMSplex members based on BPE (CQS, SCI, OM, and RM) can share the same BPE configuration PROCLIB member (BPECFG=). However, the parameters defined here all have defaults and it is not a requirement to even create this member except to change the defaults or to define some component exits.
- All BPE components can share the same BPE exit PROCLIB member (EXITDEF= in BPECFG). None of these exits are required.
- Each CSL component requires its own initialization PROCLIB member(s).

3.2.4 Hardware and software currency

If the CF itself is an IBM @server™ zSeries® machine (for example, it is an LPAR on a zSeries CPC), then CF Level 12 is required. Otherwise, CF Level 11 is required.

z/OS 1.2 or later is required on the CPC regardless of the CF Level:

- OS/390 V2 R10 (base IMS Version 8 requirement)
- z/OS 1.2 (SM duplexing and VTAM Generic Resources session level affinity)
- CF Level 9 (SM rebuild)

- CF Level 11 or 12 (SM duplexing)
- CF-to-CF links (SM duplexing)

3.3 Updating the CFRM couple data set (CDS)

Resource structure is defined in the CFRM policy and the policy must be activated. New enhancements in this area (also available to IMS Version 7) are:

- The autoalter support
- System-managed rebuild support
- System-managed duplexing support

For rebuild and duplexing, the CDS itself has be to reformatted; for duplexing, the CFRM policy needs to be updated. Here are the following requirements for this step.

- Update the CFRM Couple Data Set (CDS) to allow system managed rebuild and system managed duplexing.
- Define the resource structure.
- Define the shared queue structures, including the message queue primary and overflow structures, the Fast Path shared EMH structures, and the system logger structures. This may already have been done if the system is already using shared queues. This definition is not shown here.

You should also review the definitions of the Automatic Restart Manager (ARM) and System Failure Management (SFM) policies to be sure they accurately describe your requirements.

3.3.1 Reformatting the CFRM control data set

If you intend to use either system managed rebuild or structure duplexing, the CFRM control data set (CDS) must be reformatted. This can be done using the couple data set format utility. The following statements should be included:

```
ITEM NAME(SMREBLD) NUMBER(1)
ITEM NAME(SMDUPLEX) NUMBER(1)
```

You should reference the appropriate sysplex documentation for the exact requirements for this step, including any hardware and software prerequisites.

3.3.2 Sample resource structure definition

The resource structure is not a requirement for running IMS in a CSL environment. It is only required if sysplex terminal management (STM) is to be enabled. The other features utilizing CSL in IMS Version 8, automatic RECON loss notification, global online change, and the SPOC do not require a resource structure.

Define RM structure in CFRM policy:

- DUPLEX(ENABLED) or DUPLEX(ALLOWED)
- ALLOWAUTOALT(YES)
- FULLTHRESHOLD(value) - defaults to 80%
- MINSIZE(size) - defaults to 75% of INITSIZE

Example 3-1 shows how the resource structure might be defined in the CFRM policy.

Example 3-1 Sample resource structure definition in CFRM policy

```
STRUCTURE NAME(IMOA_RSRC)
   SIZE(8192)
   INITSIZE(4096)
   MINSIZE(3072)
   ALLOWAUTOALT(YES)
   FULLTHRESHOLD(60)
   DUPLEX(ENABLED)
   REBUILDPERCENT(10)
   PREFLIST(CF05 CF06)
```

3.3.3 Resource structure failure

CQS does not log resource updates or support a checkpoint of the resource structure. However, CQS supports automatic duplexing of the resource structure for backup in case of structure failure. You must define the resource structure in the Coupling Facility resource management (CFRM) policy to automatically be duplexed if you want the resource structure to be recoverable.

If the resource structure and its duplex (if applicable) fail, and CQS can allocate a new structure, CQS notifies RM to repopulate the structure. RM repopulates the structure from information in its local control blocks. RM then issues a directive to its clients to populate the structure.

If the resource structure fails, and CQS cannot allocate a new structure, CQS notifies RM that the structure failed. RM then issues a directive to its clients that the structure failed. Any RM or IMS resource that existed only on the resource structure is lost. When a new resource structure is allocated, the clients can choose to coordinate the repopulation of the resource structure.

Resource structures are defined with system-managed rebuild, so OS/390 automatically rebuilds the structure when no CQS is up to build the structure. OS/390 cannot rebuild the structure if the structure fails or if OS/390 loses connectivity to the structure. If any CQS is up and rebuild is initiated with the `SETXCF START,REBUILD` command, then CQS copies the structure. If the structure fails, no structure recovery is initiated because resource structures do not support structure checkpoint.

3.3.4 Shared queue structures

There are several structures required for shared queues, including both the shared queue structures themselves and the system logger structures used by CQS to log updates to the shared queues structures. Definitions of these structures are not included here. Refer to Chapter 2, “Implementing IMS shared queues” on page 29.

3.4 Setting up the Base Primitive Environment (BPE)

The base primitive environment is a requirement for CQS, SCI, OM, and RM. For the most part, this is transparent to the user, but there are some definitional requirements.

The IMS Base Primitive Environment (BPE) is a common system service base upon which many other IMS components are built. BPE provides services such as tracing, message formatting, parsing, storage management, sub-dispatching, and serialization. In IMS Version 8, the following components use BPE:

- Common Queue Server (CQS)
- Operations Manager (OM)
- Resource Manager (RM)
- Structured Call Interface (SCI)

When an IMS component that uses BPE is started, the component loads a copy of the BPE service modules into its address space from the IMS Version 8 program libraries.

The IMS component's modules are specific to that component; however, the BPE service modules are common across the various address spaces. The base system service functions are therefore identical in every address space that uses BPE.

3.4.1 Base Primitive Environment PROCLIB members

All CSL components use the Base Primitive Environment (BPE). There are two BPE PROCLIB members that are applicable to each CSL component address space:

- BPE configuration PROCLIB member - contains statements that configure the BPE execution environment parameters
- BPE user exit PROCLIB member - associates user exit points with one or more user exit modules
- Neither are required:
  - Can let BPE configuration parameters use defaults
  - Do not need user exits

Each CSL address space can have its own BPE configuration and user exit definition members, or they can share common members.

3.4.2 Update the OS/390 program properties table (PPT)

All of the BPE-based CSL address spaces can execute program BPEINI00. This is true even for CQS which, in earlier releases, executed CQSINIT0. BPEINI00 must be added to the OS/390 program properties table (PPT). Refer to 2.2.2, "BPE PPT entry for CQS" on page 37.

Add program name BPEINI00 into the OS/390 PPT Example 3-2 by editing SCHEDxx member in SYS1.PARMLIB.

Example 3-2 PPT input

```
PPT PGMNAME(BPEINI00)          /* IMS */
      CANCEL
      KEY(7)
      NOSWAP
      NOPRIV
      DSI
      PASS
      SYST
      AFF(NONE)
```

3.4.3 BPE user-supplied exit routines

In BPE 1.4, BPE defines two user exit types that are available to any address space that is using BPE. This is in contrast with component-defined user exits, which are only available for address spaces of that component type (for instance, the CQS-defined overflow user exit can only be used in a CQS address space).

BPE user exits enable you to customize and monitor address spaces built on the Base Primitive Environment. BPE-defined user exit types are available to all IMS component address spaces that are running with BPE. You write these exit routines. No sample exits are provided. The BPE user exits are given control in the address space in an authorized state.

Here is a list of the user exit routines and their functions.

- INITTERM (BPE Initialization-Termination)

  Called during BPE initialization and normal BPE termination.It is not called for abnormal termination.

- STATS (BPE Statistics)

  Called at regular intervals during the life of a BPE address space, and a final time at normal address shutdown, to gather address-space related statistics. The frequency of the STATS exit call is determined by the new STATINTV= parameter in the BPE configuration PROCLIB member dataset.

3.4.4 BPE configuration parameter PROCLIB member

You can use the BPE configuration parameter PROCLIB member to define BPE execution environment settings for the address space being started. You specify the PROCLIB member name by coding BPECFG=member_name on the EXEC PARM= statement in the address space startup JCL, as shown in the following example:

```
EXEC CQSINIT0,PARM='BPECFG=BPECFGCQ'
```

These are the keywords that are available for the BPE configuration parameter PROCLIB member:

- LANG=

  The language used for BPE and IMS component messages. ENU (English) is the default.

- TRCLEV=

  The trace level settings for BPE and IMS component internal trace tables. BPE 1.4 has added a new trace table type of "HASH". This table traces events in the BPE hash table service. Therefore, you can now specify "HASH" as the trace table type on TRCLEV statements and on BPE TRACETABLE commands.

  Note that BPE hash table services is an optional service, and not all IMS component address spaces make use of them. If an address space does not request hash table services, BPE will not load them, and will not create the HASH trace table. For that reason, BPE will ignore TRCLEV statements for the HASH trace table if hash table services are not present.

- EXITMBR=

  The name of a BPE exit list PROCLIB member where configuration information for IMS component user exits is stored.

  There is a new parameter, COMP=, which you can specify on the EXITDEF statement in the BPE user exit definition PROCLIB member. COMP= specifies the IMS component name for the component that owns the user exit. With this parameter, you can share a single user exit PROCLIB member among different IMS component address space types.

- STATINTV=

  The time interval between calls to the BPE statistics exits. STATINTV is specified in seconds. The default value if STATINTV is not coded is 600 (10 minutes).

Note: Avoid coding statements in the BPE configuration member that specify definitions for the same resources more than one time. For example, avoid multiple TRCLEV statements for the same trace table type, or multiple EXITMBR statements for the same IMS component type. BPE uses the *last* statement it encounters in the member. Any values that are specified on earlier duplicate statements are ignored. A message, BPE0017I is issued for each duplicate found.

Sample BPECFG member

Each BPE-based CSL component may specify a BPE configuration PROCLIB member which identifies such things as trace levels and user exits. All of the parameters specified in this member have defaults, and it is not even necessary to define one. However, if you want to change the trace levels for various components, or define user exits (such as the OM command security exit), then a BPE Configuration (BPECFG=) member must be defined and a BPE User Exit List (EXITMBR=) member must be defined as shown in Example 3-3.

Example 3-3 Sample BPE configuration PROCLIB members

```
LANG=ENU
STATINV=600

TRCLEV=(*,LOW,BPE)
TRCLEV=(*,LOW,CQS)
TRCLEV=(*,LOW,RM)
TRCLEV=(*,LOW,OM)
TRCLEV=(*,LOW,SCI)

EXITMBR=(SHREXIT0,BPE)
EXITMBR=(SHREXIT0,CQS)
EXITMBR=(SHREXIT0,RM)
EXITMBR=(SHREXIT0,OM)
EXITMBR=(SHREXIT0,SCI)
```

IBM recommends trace levels of at least LOW. These are incore traces and should have very little overhead.

The '*' turns on all trace types and allows for future types without having to change or add TRCLEV statements.

3.4.5 BPE user exit PROCLIB member

Use the PROCLIB member(s) specified by the EXITMBR= parameter in the BPE configuration parameter PROCLIB member to define user exit modules to BPE.

BPE Exit List PROCLIB members are IMS-component specific. You specify one EXITMBR statement for each IMS component that provides user exits through BPE services. Each EXITMBR statement specifies the name of a PROCLIB member that contains the definitions for exits for that IMS component. You can have a separate exit list PROCLIB member for each IMS component, or you can share one exit list PROCLIB member among several IMS components.

All user exits can be in same user exit list PROCLIB member.

Sample user exit list member

Example 3-4 shows a sample user exit list member shared by all components. The new COMP= parameter allows the user to define all exits in one member and then specify for which components the exits should be called.

Example 3-4 Sample BPE user exit list PROCLIB member (SHREXIT0)

```
EXITDEF=(TYPE=INITTERM,EXITS=(RMITEXIT),COMP=RM)
EXITDEF=(TYPE=STATS,EXITS=(BPESTATX),COMP=BPE)
EXITDEF=(TYPE=SECURITY,EXITS=(OMSECYX),COMP=OM)
```

Each component will also define, in its JCL, the name of an initialization module specific to that component type.

3.4.6 BPEINIT parameter in BPE execution

Prior to IMS Version 8 regarding BPE (BPE 1.4), an address space that used BPE services was started by specifying a module provided by the IMS component as the first module to run in the address space. This was the module whose name was coded on the PGM= parameter of the JCL EXEC statement for the job.

In IMS Version 8, there are three new address spaces that use BPE: OM, RM, and SCI. BPE provides an initial startup module BPEINI00. This module can now be passed on PGM= instead of the IMS component's main module. BPEINI00 can be used for any IMS component address space. See Figure 3-2.

The JCL to start a job using BPEINI00 must include one additional parameter on the PARM= statement. BPE has to know what kind of an address space it is that is being started.

Using this approach, only one module (BPEINI00) needs to be added to the z/OS PPT. Any BPE-using address space can use BPEINI00 to start itself. In IMS Version 8, this is the only way to start the new address spaces OM, RM, and SCI.

CQS can be started using BPEINI00; it can also continue to be started using CQSINIT0 as the PGM= module, just as it does today.

IMS components supporting BPEINI00:

- CQS: BPEINIT=CQSINI00 (previous PGM=CQSINIT0 still works too)
- OM: BPEINIT=CSLOINI0
- RM: BPEINIT=CSLRINI0
- SCI: BPEINIT=CSLSINI0

Example:

To start an SCI address space:

```
//SCI     EXEC  PGM=BPEINI00,PARM='BPEINIT=CSLSINI0,...'
```

To start a CQS address space:

```
//CQS     EXEC  PGM=BPEINI00,PARM='BPEINIT=CQSINI00,...'
```

Figure 3-2 BPE execution

3.5 Setting the Structured Call Interface (SCI)

SCI allows IMSplex members to communicate with one another. The communication between IMSplex members can happen within a single OS/390 image or among multiple OS/390 images. Individual IMS components do not need to know where the other components reside or what communication interface to use.

The main functions of SCI are:

- Routes any requests or messages between the IMS control region, OM, RM, and other members of the IMSplex.
- Registers and deregisters IMSplex members.
- Notifies IMSplex members when a member joins or leaves the IMSplex.

Any IMSplex member that requires SCI services must have an SCI active on its OS/390 image:

- OM, RM, CQS, and IMS register with SCI.
- SPOC and user automation register with SCI.
- SCI provides the communications interface between components within the same IMSplex:
  - Components use CSL (SCI) macros to communicate.
  - Macros invoke code in SCI address space.
- SCI initialization module CSLSINI0 must be found from the STEPLIB.
- SCI initialization PROCLIB member CSLSIxxx pointed by SCIINIT=xxx must be in PROCLIB.

3.5.1 SCI startup procedure

SCI address space as all CSL address spaces execute BPEINI00. Each CSL component has its own initialization module and PROCLIB member, but they can share a common BPE configuration member. Each instance of SCI (and OM, RM, and CQS) will have different initialization PROCLIB members, because they require unique SCI (or OM, RM, CQS) names.

You can use the SCI startup procedure to dynamically override the settings in the SCI initialization parameters PROCLIB member. The startup procedure is required, but setting values for the execution parameters is optional. A sample startup procedure is called CSLSCI and can be found in IMS.PROCLIB. Example 3-5 shows a sample JCL for SCI.

Example 3-5 SCI started task JCL

```
//*  SCI   PROCEDURE
//*
//*     PARAMETERS:
//*     BPECFG  - NAME OF BPE MEMBER
//*     SCIINIT -  SUFFIX FOR YOUR CSLSIxxx MEMBER
//*     PARM1  - OTHER OVERRIDE PARAMETERS e.g.
//*               ARMRST   - Indicates if ARM should be used
//*               SCINAME  - Name of the SCI being started
//*
//*               EXAMPLE:
//*               PARM1 ='ARMRST=Y,SCINAME=IM1ASC'
//*
//*******************************************************************
```

```
//IM1ASC PROC RGN=3000K,SOUT=A,
// RESLIB='IMSPSA.IMS0.SDFSRESL',
// BPECFG=BPECFGSC,
// SCIINIT=001,
// PARM1=
//*
//IM1ASC EXEC PGM=BPEINI00,REGION=&RGN,
//  PARM=('BPECFG=&BPECFG,BPEINIT=CSLSINI0,SCIINIT=&SCIINIT,&PARM1')
//*
//STEPLIB  DD  DSN=&RESLIB,DISP=SHR
//                 DD DSN=SYS1.CSSLIB,DISP=SHR
//*
//PROCLIB  DD  DSN=IMSPSA.IM0A.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
```

3.5.2 SCI execution parameters

You can specify the following parameters as execution parameters on the EXEC statement in the SCI startup procedure. Certain parameters that are required for SCI address space initialization can also be specified in the SCI initialization parameters PROCLIB member.

- ARMRST=Y | N

  Specifies whether Automatic Restart Manager (ARM) is or is not used to restart the SCI address space after an abend.

- BPECFG=

  Defines the BPE configuration parameters PROCLIB member. If omitted, BPE defaults are used:

  – No BPE user exits
  – BPE trace level = error
  – Error message language = English

- BPEINIT=CSLSINI0

  Must be CSLSINI0 for SCI.

- SCIINIT=

  Defines the 3 character suffix for the SCI initialization parameters PROCLIB member (CSLSIxxx).

- SCINAME=

  Defines the SCI address space name.

3.5.3 SCI initialization PROCLIB member CSLSIxxx

Use the CSLSIxxx PROCLIB member to specify parameters related to initialization of the SCI address space. Certain parameters within CSLSIxxx can be overridden via SCI execution parameters.

- ARMRST=Y | N

  Specifies whether ARM is or is not used to restart the SCI address space after an abend.

- IMSPLEX()

  Specifies definitions for an IMSplex managed by SCI. IMSPLEX is a required parameter. Only one IMSPLEX keyword can be specified and it must precede the left parenthesis.

The IMSPLEX definition parameters *NAME*= specifies a 1-5 character name that specifies the IMSplex group name. SCI concatenates this name to `CSL` to create the IMSplex group name. All OM, RM, SCI, IMS, and other address spaces that are in the same IMSplex must specify the same name.

- SCINAME=

  Specifies the name for the SCI address space. This is an optional 1-6 character name. You must specify this parameter either as an execution parameter or in the CSLSIxxx PROCLIB member.

- FORCE=()

  Specifies that SCI is to clean up the global interface storage. FORCE is an optional parameter and has no default. The valid keywords are *ALL* and *SHUTDOWN*.

  ALL means SCI should delete all of the global storage, including control blocks and routines. This keyword is required. This causes all "stubs" to be deleted from ECSA and the new, updated ones to be loaded.

  SHUTDOWN SCI should shut down after cleaning the global storage. This keyword is optional.

  No local IMSplex members can be active when the FORCE keyword is used. Use the FORCE keyword in the following situations:

  – When an IMSplex managed by an SCI on one image will be managed by a different SCI. For example, PLEX1 is managed by SCI1. Before SCI2 is started, use FORCE(ALL,SHUTDOWN) on SCI1 to clean the global storage.

  – When an SCI will not be reactivated on an image. To clean the global storage, reactive that SCI one final time using the FORCE(ALL, SHUTDOWN) keyword.

Example 3-6 shows a sample SCI initialization PROCLIB member. The optional parameter 'FORCE=(ALL,SHUTDOWN)' is not used in this example. It would be used to cleanup SCI's global blocks from ECSA, and it would be also useful if you want to change the SCINAME without an IPL. During the normal operations, you don't need it.

Example 3-6 Sample SCI initialization PROCLIB member

```
ARMRST=N,                    /* ARM should restart OM on failure  */
SCINAME=IM1A,                /* SCI Name (SCIID = SCI1SCI)        */
IMSPLEX(NAME=PLEX1)          /* IMSplex Name (CSLPLEX1)           */
* same identifier must be used for IMSPLEX= parameter
* in CSLOIxxx,CSLRIxxx, DFSCGxxx PROCLIB members*/
*-------------------------------------------------------------------*
```

3.5.4 SCI security

When a client issues the CSLSCREG request to register with SCI, SCI first determines whether the address space is authorized to register with SCI. It does this by issuing a `RACROUTE REQUEST=AUTH` call. RACF (or an equivalent security product) checks the user ID of the address space issuing the CSLSCREG request; the user ID must have at least update authority to register to SCI.

Authorize (RACF) connection to IMSplex for all SCI connectors:

- IMS, DBRC, CSL, DBRC utility, batch and utilities w/DBRC, AOPs

The security administrator can define profiles in the FACILITY class to control SCI registration. Profile names must be of the following form:

```
CSL.imsplex_name
```

`In this profile name` *imsplex_name* is the name of the IMSplex as defined on the IMSPLEX parameter in the CSLSIxxx PROCLIB member, with `CSL` added as a prefix to the name.

At a minimum, if security is desired, you should permit OM, RM, CQS, and IMS (online, batch and utility regions) to access IMSplex through SCI.

The needed RACF definitions are:

- Add group and users (including batch DBRC users)
- Define started tasks (only if not using JOBs with USER=userid)
- Define facility (CSL.plexname)
- Permit group update access to facility

Example 3-7 is an example of the RACF commands required to define a RACF FACILITY class for SCI and to permit the IMS and TSO user IDs to that facility.

Example 3-7 RACF for SCI

```
RDEFINE FACILITY CSL.CSLPLEX1 UACC(NONE)
PE CSL.CSLPLEX1 CLASS(FACILITY) ID(IMSuserid) ACCESS(UPDATE)
PE CSL.CSLPLEX1 CLASS(FACILITY) ID(TSOuserid) ACCESS(UPDATE)
SETR RACLIST(FACILITY) REFR
```

3.6 SCI user exit routines

SCI user exits allow you to customize and monitor the SCI environment. They are written and supplied by the user. all the exits are optional, and no sample exits are provided. The following exit points are available:

- SCI client connection user exit

  This exit is defined as TYPE=CLNTCONN in the EXITDEF statement in the BPE user exit list PROCLIB member.

- SCI Initialization/Termination user exit

  This exit is defined as TYPE=INITTERM in the EXITDEF statement in the BPE user exit list PROCLIB member.

- SCI statistics available through BPE Statistics user Exit

  The exit is The BPE Statistics user exit routine is defined as TYPE = STATS, COMP=BPE in the EXITDEF statement in the BPE user exit PROCLIB member. The interval between successive statistics exit calls is specified on the STATINTV parameter in the BPE configuration PROCLIB member.

3.7 Setting up the Operations Manager (OM)

At least one Operation Manager must be defined in the IMSplex to use OM functions on any operating system image (LPAR) where CSL is active. However, an OM on every LPAR is recommended for availability.

Each OS/390 image can have more than one OM. If multiple OMs are defined in the IMSplex, any OM defined can perform work from any OS/390 image in the IMSplex. OM registers with SCI and RM and IMS register with OM. OM provides a standard entry point into the IMSplex from SPOC or any user-written automation programs. It routes commands to various members of IMSplex and consolidates responses from multiple responders.

OM initialization module is CSLOINI0 and it must be found in the STEPLIB. OM has initialization member CSLOIxxx in PROCLIB.

3.7.1 OM startup procedure

You can start OM as a started procedure or with JCL. OM as all other CSL address spaces execute BPEINI00. Each CSL component has its own initialization module and PROCLIB member, but they can share a common BPE configuration member.

A sample startup procedure is called CSLOM and can be found in IMS.PROCLIB.

Example 3-8 shows the JCL required for the Operations Manager (OM) started task.

Example 3-8 OM started task JCL

```
//*******************************************************************
//*  OM  PROCEDURE
//*
//*     PARAMETERS:
//*     BPECFG  - NAME OF BPE MEMBER
//*     OMINIT  - SUFFIX FOR YOUR CSLOIxxx MEMBER
//*     PARM1   - OTHER OVERRIDE PARAMETERS
//*               ARMRST     - Indicates if ARM should be used
//*               CMDLANG - Language for command description text
//*               CMDSEC     - Command security method
//*               OMNAME   - Name of the OM being started
//*
//*               EXAMPLE:
//*               PARM1='ARMRST=Y,CMDSEC=R,OMNAME=IM1AOM,CMDLANG=ENU'
//*
//*********************************************************************
//IM1AOM PROC RGN=3000K,SOUT=A,
// RESLIB='IMSPSA.IMS0.SDFSRESL',
// BPECFG=BPECFGOM,
// OMINIT=001,
// PARM1=
//*
//IM1AOM PROC EXEC PGM=BPEINI00,REGION=&RGN,
// PARM=('BPECFG=&BPECFG,BPEINIT=CSLOINI0,OMINIT=&OMINIT,&PARM1')
//*
//STEPLIB  DD  DSN=&RESLIB,DISP=SHR
//                 DD DSN=SYS1.CSSLIB,DISP=SHR
//*
//PROCLIB  DD  DSN=IMSPSA.IMOA.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
```

3.7.2 OM execution parameters

Here are the parameters that you can specify as execution parameters on the startup procedure for OM.

- BPECFG=

  Specifies an 8-character name for the BPE configuration parameters PROCLIB member. This parameter can only be specified as an execution parameter.

- BPEINIT=CSLOINI0

  Specifies the name of the module that contains OM start up values required by BPEINI00 to start an OM address space.

- OMINIT=000

  Specifies a 3-character suffix for the OM initialization parameters PROCLIB member, CSLOIxxx. This parameter can only be specified as an execution parameter. The default suffix is 000.

- OMNAME=

  Specifies the name for the OM address space. This is an optional 1-6 character name. If specified, it overrides the value specified in the CSLOIxxx PROCLIB member.

The following parameters that are required for OM initialization can also be specified in the initialization parameters PROCLIB member, and they are described in the next paragraph:

- ARMRST=Y | N
- CMDLANG=ENU
- CMDSEC=
- OMNAME=

3.7.3 OM initialization parameters PROCLIB member CSLOIxxx

Use the CSLOIxxx PROCLIB member to specify parameters that initialize the OM address space. Certain parameters within CSLOIxxx can be overridden with the OM execution parameters.

- ARMRST=Y | N

  Specifies whether the OS/390 Automatic Restart Manager (ARM) is to be used to restart the OM address space after an abend.

- CMDLANG= ENU

  The language to be used for IMS command text that is distributed to OM automation clients upon request. This affects only the command descriptions that are displayed on a workstation SPOC that requests command text from OM. This value defaults to ENU for US English.

- CMDSEC=

  Specifies the security method to be used for OM command security.

  – A

    Specifies that both RACF (or an equivalent security product) and the exit are to be called (options E and R). RACF is called first.

  – E

    Specifies that the OM Security user exit routine is to be called for command authorization.

  – N

    Specifies that no authorization checking is to be done. This is the default.

  – R

    Specifies that RACF (or an equivalent security product) is to be called for command authorization.

- CMDTEXTDSN=

  Specifies the data set name for the PDS that contains the command syntax translatable text. This keyword is required.

- IMSPLEX()

  Specifies definitions for an IMSplex managed by OM. IMSPLEX is a required parameter. Only one IMSPLEX keyword can be specified. The IMSPLEX definition parameter *NAME*= specifies a 1-5 character identifier that specifies the IMSplex group name. OM concatenates this identifier to `CSL` to create the IMSplex group name. All IMSplex member address spaces that are in the same IMSplex group sharing either databases or message queues must specify the same identifier.

- OMNAME=

  Specifies the name for the OM address space. This is an optional 1-6 character name. Specify this parameter either as an execution parameter or in the CSLOIxxx PROCLIB member. This name must be unique for each OM instance within the IMSplex.

Example 3-9 shows a sample Operations Manager initialization PROCLIB member CSLOIxxx.

Example 3-9 OM initialization member

```
**************************************
* OM INITIALISATION PARAMETERS       *
* PROCLIB MEMBER - CSLOI001          *
**************************************
CMDSEC=R              /* R for RACF */
IMSPLEX(NAME=PLEX1)
OMNAME=IM1A
CMDTEXTDSN=IMS.SDFSDATA
```

3.7.4 OM command security

Command security for commands entered through the OM interface. This can be RACF or an Exit or both (or neither). If OM is doing the security, then you (probably) don't need IMS to do it also. If an exit is used, it must be coded and then identified in BPE.

OM command security is optionally performed during command processing. Command security allows:

- The user to control which user IDs can enter IMS commands through OM
- The user ID to be associated with an application program address space
- The user ID to be the end user logged onto TSO SPOC

Note: Use OM command security rather than IMS command security to improve system performance. By allowing OM to perform the security checks, commands that fail security authorization are not routed to IMS, which reduces processing overhead and network traffic.

OM command security is controlled with the CMDSEC= parameter. If RACF (or equivalent) security checking is specified (CMDSEC=R|A), OM command security uses the OPERCMDS resource class:

- If CMDSEC=A or E in CSLOIxxx
  - Define OM security exit in BPE user exit list PROCLIB member:

    `EXITDEF(TYPE=SECURITY,EXITS=(OMSECYX),COMP=OM)`

- Write security exit OMSECYX.
- Link it the exit into authorized library in OM steplib.

- If CMDSEC=A or R in CSLOIxxx
  - Define OM commands and security to RACF for IMSplex and classic command security for commands entered through OM. The new LE QRY requires READ access, UPD, INIT, TRM, and DEL require UPDATE access.

When you register the command with OM, specify the access authority with the CSLOMBLD request, as shown in Example 3-10.

Example 3-10 CSLOMBLD request

```
CSLOMBLD FUNC=DEFKEY SEC=READ|UPDATE
```

Defining RACF authorization for commands entered through OM

All OM security checking uses the RACF OPERCMDS class. The resource name is in the following format:

```
IMS.imsplexname.commandverb.commandkeyword.
```

In the resource name above:

- *IMS* is the high level qualifier for all IMS commands entered to OM.
- *imsplexname* specifies the name of the IMSplex for which the command is authorized.
- *commandverb* specifies the name of the command verb.
- *commandkeyword* specifies the primary command keyword or resource type for the command.

The access authority on the RACF PERMIT command must match the access authority with which the command was registered.

When CMDSEC=R in the OM initialization member, RACF will be called to authorize a SPOC user to enter the command. Security for these commands is defined in the OPERCMDS class in RACF.

Example 3-11 is an example of RACF command security for a few of the OM-entered commands, where some IMSplex commands are defined and users are permitted to have READ or UPDATE access to these commands.

Note that the IMSplex name is part of the definitions and that generic substitution is allowed.

Example 3-11 RACF security statements

```
RDEFINE   OPERCMDS IMS.CSLPLX0.UPD.TRAN UACC(NONE)
RDEFINE   OPERCMDS IMS.CSLPLX1.UPD.TRAN UACC(NONE)
RDEFINE   OPERCMDS IMS.*.QRY.* UACC(NONE)

PERMIT    IMS.CSLPLX0.UPD.TRAN CLASS(OPERCMDS) ID(JOHN) ACCESS(UPDATE)
PERMIT    IMS.CSLPLX1.UPD.TRAN CLASS(OPERCMDS) ID(HENRY) ACCESS(UPDATE)
PERMIT    IMS.*.QRY.* CLASS(OPERCMDS) ID(JOUKO) ACCESS(READ)
```

3.8 OM user exit routines

You can write OM user exits to customize and monitor the OM environment. All exits are optional and no sample exits are provided.

OM uses BPE services to call and manage its user exits. BPE enables you to externally specify the user exit modules to be called for a particular user exit type by using EXITDEF= statements in the BPE user exit list PROCLIB members. BPE also provides a common user exit runtime environment for all user exits.

- OM client connection user exit

  Defined in BPE user exit list PROCLIB member via EXITDEF statement specified with TYPE=CLNTCONN.

- OM initialization and termination user exit

  Defined in BPE user exit list PROCLIB member via EXITDEF statement specified with TYPE=INITTERM.

- OM input user exit

  Defined in BPE user exit list PROCLIB member via EXITDEF statement specified with TYPE=INPUT.

- OM output user exit

  Defined in BPE user exit list PROCLIB member via EXITDEF statement specified with TYPE=OUTPUT.

- OM security user exit

  Defined in BPE user exit list PROCLIB member via EXITDEF statement specified with TYPE=SECURITY. This exit can override RACF and it can accept or reject command. This exit is invoked when the CMDSEC= parameter on the OM procedure is specified as A or E:

  – A: Both this exit and RACF (or equivalent) are used for OM command security
  – E: Only this exit is called for OM command security

- BPE Statistics user exit for OM

  Defined in BPE user exit list PROCLIB member via EXITDEF statement specified with TYPE=STATS.

3.9 Setting up the Resource Manager (RM)

Resource manager (RM) is an IMS address space that manages global resources and IMSplex-wide processes in a sysplex on behalf of its clients. IMS is one example of an RM client that uses RM to manage global message destination and terminal resources and global online change.

With RM, the system administrator can manage resources that are shared by multiple IMS systems in an IMSplex. RM provides an infrastructure for managing global resources and coordinating processes across the IMSplex.

RM uses Common Queue Server (CQS) to maintain global resource information in a *resource structur*e, which is a Coupling Facility list structure that all CQSs in the IMSplex can access.

Note: Although a resource structure is optional in an IMSplex, we recommend that you define a resource structure for improved recovery capabilities.

RM also supports coordinated processes across the IMSplex. For example, IMS uses this support to coordinate global online change across the IMSplex. For this activity, OM is also required.

3.9.1 RM infrastructure

At least one RM must be defined in an IMSplex to use RM functions. You can have one or more RMs on each OS/390 image if a resource structure is defined. If no resource structure is defined, you can have only one RM. Figure 3-3 shows the RM infrastructure.

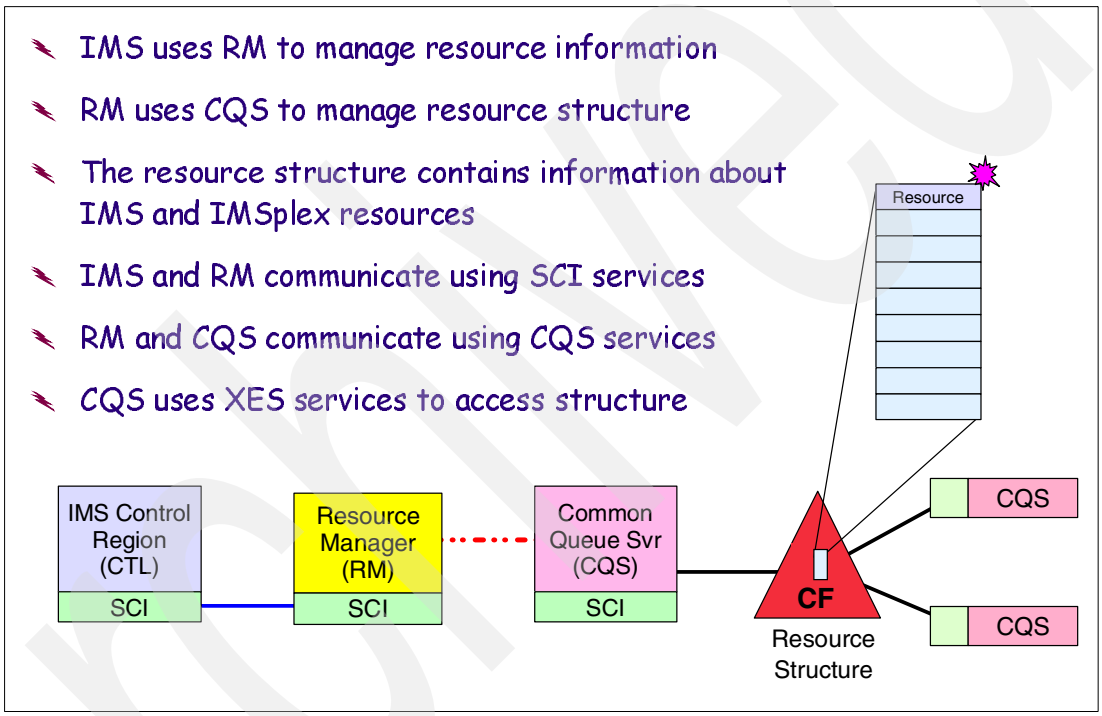

Figure 3-3 RM infrastructure

Any RM can process work from any OS/390 image within an IMSplex. An RM on every LPAR is recommended for performance, However, if no RM structure is defined, then only one RM per IMSplex is allowed. RM registers with SCI, OM, and CQS and IMS registers with RM. IMS uses RM services for sysplex terminal management and for global online change. RM uses CQS services to manage RM structure. RM initialization module is CSLRINI0, and it must be found in the STEPLIB. RM has an initialization PROCLIB member CSLRIxxx.

3.9.2 Maintaining global resource information in the resource structure

RM uses CQS to store global resource information on a Coupling Facility list structure that all RMs can access. This structure is called a *resource structur*e.You can use RM to create, update, query, or delete resources in the resource structure.

CQS does not log resource updates or support a checkpoint of the resource structure. However, CQS supports automatic duplexing of the resource structure for backup in case of structure failure.

If the resource structure and its duplex (if applicable) fail, and CQS can allocate a new structure, CQS notifies RM to repopulate the structure. RM repopulates the structure from information in its local control blocks.

Resource structures are defined with system-managed rebuild, so OS/390 automatically rebuilds the structure when no CQS is up to build the structure. OS/390 cannot rebuild the structure if the structure fails or if OS/390 loses connectivity to the structure.

Authorize (RACF) RMs to access RM structure. Note that authorization is for RM, not for CQS.

3.10 RM definitions

The following execution parameters and PROCLIB members must be defined before you bring up RM:

- RM startup procedure
- RM execution parameters
- CSLRIxxx RM initialization PROCLIB member
- RM BPE configuration parameters PROCLIB member
- RM BPE user exit list PROCLIB member
- RM user exits
- (Optional) resource structure (refer to 3.3.2, “Sample resource structure definition” on page 58)

3.10.1 RM startup procedure

Use the RM startup procedure to dynamically override the settings in the RM initialization parameters PROCLIB member. You can start RM as a started procedure or with JCL. RM as all CSL address spaces executes BPEINI00. Each CSL component has its own initialization module and PROCLIB member, but they can share a common BPE configuration member.

A sample startup procedure is called CSLRM and can be found in IMS.PROCLIB.

Example 3-12 shows the JCL for the Resource Manager (RM) started task.

Example 3-12 RM started task JCL

```
//*****************************************************************
//*  RM  PROCEDURE
//*
//*     PARAMETERS:
//*     BPECFG  - NAME OF BPE MEMBER
//*     RMINIT  - SUFFIX FOR YOUR CSLRIxxx MEMBER
//*     PARM1   - OTHER OVERRIDE PARAMETERS e.g.
//*               ARMRST  -  Indicates if ARM should be used
//*               RMNAME - Name of RM being started
//*
//*               EXAMPLE:
//*               PARM ='ARMRST=Y,RMNAME=IM1ARM'
//*
//******************************************************************
//IM1ARM PROC RGN=3000K,SOUT=A,
// RESLIB='IMSPSA.IMS0.SDFSRESL',
// BPECFG=BPECFGRM,
// RMINIT=001,
// PARM1=
```

```
//*
//IM1ARM EXEC PGM=BPEINI00,REGION=&RGN,
// PARM=('BPECFG=&BPECFG,BPEINIT=CSLRINI0,RMINIT=&RMINIT,&PARM1')
//STEPLIB  DD  DSN=&RESLIB,DISP=SHR
//                 DD DSN=SYS1.CSSLIB,DISP=SHR
//*
//PROCLIB  DD  DSN=IMSPSA.IM0A.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
```

3.10.2 RM execution parameters

The parameters that can be specified as execution parameters on the RM startup procedure are described below. Some parameters that are required for RM initialization can also be specified in the RM initialization parameters PROCLIB member.

- BPECFG=

  Specifies an 8-character name for the BPE configuration parameters PROCLIB member. This parameter can be specified only as an execution parameter.

- BPEINIT=CSLRINI0

  Specifies the name of the module that contains the RM start up values required by BPEINI00 to start an RM address space.

- RMINIT=

  Specifies a 3-character suffix for the RM initialization parameters PROCLIB member, CSLRIxxx.

The following parameters that are required for RM initialization can also be specified in the initialization parameters PROCLIB member, and they are described in the next paragraph:

- ARMRST=Y | N
- RMNAME=

3.10.3 RM initialization parameters PROCLIB member CSLRIxxx

Use the CSLRIxxx PROCLIB member to specify parameters that initialize RM. ARMRST and RMNAME parameters within CSLRIxxx can be overridden with RM execution parameters.

- ARMRST=Y | N

  Specifies whether or not the OS/390 Automatic Restart Manager (ARM) should be used to restart RM after an abend.

- CQSSSN=

  Specifies the 1-4 character subsystem name of the CQS. RM uses this name to connect to the proper CQS.

- IMSPLEX()

  Specifies definitions for an IMSplex managed by RM. IMSPLEX is a required parameter. Only one IMSPLEX keyword can be specified. The IMSPLEX definition parameters are as follows:

  – NAME=

    Specifies a 1-5 character identifier that specifies the IMSplex group name. RM concatenates this identifier to `CSL` to create the IMSplex group name. All OM, RM, SCI,

IMS, and IMSplex members that are in the same IMSplex sharing group sharing either databases or message queues must specify the same identifier.

– RSRCSTRUCTURE()

  Specifies definitions for a resource structure managed by RM. This keyword construct is optional. RSRCSTRUCTURE must be specified to make use of RM's global resource services.

  - STRNAME=

    Specifies the 1-16 character name of a resource structure that IMS connects to, which contains IMS resource information.

- RMNAME=

  Specifies the name for the RM address space. This is an optional 1-6 character name. You must specify this parameter either as an execution parameter or in the CSLRIxxx PROCLIB member. This name is used to create the RMID, which is used in RM processing.

Example 3-13 shows a sample Resource Manager PROCLIB member CSLRIxxx.

Example 3-13 RM PROCLIB member

```
*-------------------------------------------------------------------*
* RM INITIALIZATION PROCLIB MEMBER.                                 *
*-------------------------------------------------------------------*
ARMRST=N,                /* SHOULD ARM RESTART RM ON FAILURE     */
IMSPLEX(
  NAME=PLEX1,RSRCSTRUCTURE(STRNAME=IMOA_RSRC)),
*       IMSPLEX NAME (CSLPLEX1) & RESOURCE MANAGER STRUCTURE NAME *
CQSSSN=CQ1A,             /* NEEDS TO MATCH SSN= PARM OF CQSIPXXX */
RMNAME=IM1A              /* RM NAME (RMID = IM1ARM)              */
*-------------------------------------------------------------------*
* END OF MEMBER CSLRI001                                            *
*-------------------------------------------------------------------*
```

3.10.4 RM security

There are no specific RM security requirements other than to permit RM register with SCI (IMSplex name).

3.11 RM user exit routines

You can write RM user exits to customize and monitor the RM environment. All the exits are optional and no sample exits are provided.

RM uses BPE services to call and manage its user exits. BPE enables you to externally specify the user exit modules to be called for a particular user exit type by using EXITDEF= statements in the BPE user exit list PROCLIB members.

BPE also provides a common user exit runtime environment for all user exits.

- RM client connection user exit

  This exit is defined as TYPE=CLNTCONN in the EXITDEF statement in the BPE user exit list PROCLIB member.

- RM initialization and termination user exit

  This exit is defined as TYPE=INITTERM in the EXITDEF statement in the BPE user exit list PROCLIB member.

- BPE statistics user exit for RM

  This exit is defined in BPE user exit list PROCLIB member via EXITDEF statement specified with TYPE=STATS. BPE statistics user-supplied exit parameter list (BPESTXP) field BPESTXP_COMPSTATS_PTR contains a pointer to RM statistics header. RM statistics header is mapped by CSLRSTX macro.

3.12 Setting up the IMS control region for CSL

IMS is responsible for sysplex terminal management and global online change. IMS that joins to IMSplex, requires an SCI address space on same LPAR. RM and OM address spaces can be on any LPARs. IMS registers with SCI, RM, and OM, and IMS registers also with CQS for shared queues. Some IMS control region JCL and parameter changes may be required for CSL. The changes required can be summarized as follows:

- Changes in control region JCL only if using global online change
  - Remove local //MODSTAT DD statement
  - Add //OLCSTAT DD statement and data set
  - May need to initialize OLCSTAT (if this is first IMS in IMSplex)
- Two new DFSVSMxx parameters
  - OCMD and CSLT
- Several new DFSDCxxx parameters
  - SRM and RCVYxxxx
- One new execution or DFSPBxxx parameter (identifies DFSCGxxx)
  - CSLG=xxx
- One new PROCLIB member
  - DFSCGxxx

3.13 Setup IMS PROCLIB members

Adding the CSLG parameter to DFSPBxxx instructs IMS to initialize itself for the CSL environment.

In addition to DFSPBxxx, DFSDCxxx and DFSVSMxx also have CSL-related parameters. Be sure that the SGN= parameter is consistent across all IMSs. If they are different, the first IMS to join the IMSplex sets the value and all others are stuck with it.

There is also a new PROCLIB member (DFSCGxxx) which is pointed to by CSLG in DFSPBxxx.

3.13.1 DFSPBxxx PROCLIB member

One new parameter in DFSPBxxx activates CSL by identifying the suffix of a new PROCLIB member DFSCGxxx.

- CSLG=xxx

  Identifies new PROCLIB member DFSCGxxx. It can also be coded in control region procedure, or start command.

- SGN=

  Must be consistent in all IMSplex members. It is ignored if first IMS to join IMSplex has different value.

Example 3-14 shows the change required to the IMS initialization PROCLIB member DFSPBxxx to activate the CSL for the IMSplex. Our example points to the PROCLIB member DFSCG001.

Example 3-14 CSL parameter in DFSPBxxx

```
CSLG=001              /* suffix of member DFSCGxxx */
```

3.13.2 DFSCGxxx PROCLIB member

Use this PROCLIB member to specify parameters related to the CSL, OM, and RM. The suffix is specified on the CSLG= parameter.

All IMSplex members that are in the same IMSplex group sharing databases and/or message queues must specify the same values except OLC=, which specifies either LOCAL or GLOBAL. The parameters are:

- CMDSEC=A | E | N | R

  Specifies whether or not IMS should perform security checking on commands routed from OM. Pertains to *classic* commands entered through OM only. For commands in the enhanced IMSplex command syntax, all security checking is performed by OM and controlled by the OM parameter CMDSEC=.

  IBM recommends that OM command security be used instead of IMS command security to prevent unnecessary processing overhead, reduce network traffic, and so on.

  – A - RACF or equivalent and the IMS command security exit, DFSCCMD0, are called in that order
  – E - Only DFSCCMD0 is called
  – N - No IMS command security checking is to be done
  – R - Only RACF or equivalent is to be called

- IMSPLEX=

  Specifies a 1-5 character identifier. IMS concatenates this ID to "CSL" to create the CSL IMSplex group name. All IMSplex members (OM, RM, SCI, CQS, and so on) that are in the same IMSplex sharing group must specify the same ID. IMS has no way to enforce this rule. It is the user's responsibility.

- LEOPT=Y | N

  Specifies whether LE runtime options can be dynamically overridden.

- NORSCCC=(...)

  Specifies that no resource consistency checking is to be performed for the specified resources. Resources must be enclosed in parenthesis and separated by commas. Optional. If a resource structure is defined for the IMSplex, the default is that consistency checking is performed for ACBLIB, FORMAT, MODBLKS. Otherwise, no resource definition consistency checking is performed. Specify one or more of the following resources:

– ACBLIB - the ACBLIB dataset names not checked. The IMSs in the IMSplex don't need to define the same datasets. Only applies if OLC=GLOBAL is specified.
– FORMAT - the FORMAT dataset names not checked. The IMSs in the IMSplex don't need to define the same datasets. Only applies if OLC=GLOBAL is specified.
– MODBLKS - the MODBLKS and MATRIX dataset names not checked. The IMSs in the IMSplex don't need to define the same datasets. Only applies if OLC=GLOBAL is specified.

- OLC= LOCAL | GLOBAL

  LOCAL is default. LOCAL means the online change applies locally to each IMS. *GLOBAL* means the online change is coordinated across the IMSplex.

- OLCSTAT=datasetname

  Specifies a 1-44 character name of the OLCSTAT dataset. It is optional. All IMSs in an IMSplex must define the same physical OLCSTAT dataset.

Example 3-15 shows an example of the DFSCGxxx PROCLIB member, which specifies the name of the IMSplex that this IMS belongs to, and whether or not global online change is active for this IMS.

Example 3-15 IMS PROCLIB member DFSCGxxx

```
CMDSEC=N
IMSPLEX=PLEX1          /* Note - different format of IMSPLEX= */
OLC=LOCAL              /* Local OLC - default */
*OLC=GLOBAL            /* GLOBAL OLC */
*OLCSTAT=              /* OLC data set, replaces MODSTAT  if OLC=GLOBAL */
*NORSCCC=              /* Defines whether resource consistency checking to be */
                       /* bypassed for ACBLIB/FORMAT/MODBLKS */
```

Note: Make sure all IMS systems in the IMSplex are using the same OLCSTAT data set and ideally the same DFSCGxxx member.

3.13.3 DFSDCxxx PROCLIB member

Use the IMS DC execution parameters PROCLIB member DFSDCxxx to define the status recovery mode of resources for *sysplex terminal management*. This member, while not new, does have some new parameters to define the SRM and RCVYxxxx system defaults when sysplex terminal management is enabled. Refer to Chapter 4, “Implementing the sysplex terminal management (STM)” on page 83 for the description of this parameters.

3.13.4 DFSVSMxx PROCLIB member

DFSVSMxx PROCLIB member merely allows the user to specify whether or not tracing of IMSplex command activity and CSL activity is turned on, and at what level. Use the OCMD= and CSLT= parameters of the OPTIONS statement to activate traces related to IMSplex activity as follows:

```
OPTIONS,OCMD=option,CSLT=option
```

The options are the same as for other traces: on, out, (out)low, (out)medium, (out)high, and so on. OCMD is for tracing IMSplex command activity, CSLT is for tracing CSL interaction activity. If the options are not included in DFSVSMxx, traces are not automatically activated at IMS initialization. These traces can be turned on and off with the following trace commands:

```
/TRACE SET ON TABLE OCMD
```

```
/TRACE SET ON TABLE CLST
/TRACE SET OFF TABLE OCMD
/TRACE SET OFF TABLE CLST
```

3.14 DBRC address spaces

All of the address spaces which invoke DBRC are eligible to participate in automatic RECON loss notification function. To do this, they must join the IMSplex either by coding IMSPLEX=plexname in their execution JCL, or by using exit DSPSCIX0. The exit is easier, otherwise you have to change all the following jobs or procedures:

- Online DBRC
- FDBR
- DBRC batch utilities (DSPURX00)
- Batch jobs with DBRC=Y
- Database utilities with DBRC=Y

The RECON header record includes an IMSPLEX name, so if a DBRC job tries to join the plex using a set of RECONs for another IMSplex, it will not be able to access the RECONs.

3.15 IMSplex security

The way you establish resource security for the IMS online system does not change for an IMS in an IMSplex. The resources that can be protected and the facilities available to protect them are similar.

3.15.1 Security issues

The two types of security issues for a single IMS also exist for an IMS within an IMSplex. You must address one or both types of security issues:

- Securing the kind of resource to which a user has access
- Securing what the user can do to the resource after the user has access to it

If commands are entered through OM API, we recommend that you perform all security checking in OM. To authorize classic commands from OM, you must set the appropriate value for the CMDSEC= parameter in DFSCGxxx.

If SCI registration security is desired, include IMS control region in RACF definitions.

3.16 Miscellaneous considerations when implementing CSL

There are few additional things you may want to be aware of when implementing the CSL:

- All new address spaces default to ARM=Y. This is usually good.
- There are many new and changed messages and abends (and some deleted ones too)
  - See Chapter 6 of *IMS Version 8: Release Planning Guide,* GC27-1305
- Log record format changes
  - See Chapter 29 of *IMS Version 8: Release Planning Guide,* GC27-1305

3.16.1 RSR

If using RSR, the tracking site does not have to have CSL; however, if it is active, then the tracking site can take advantage of OM for command entry and SCI for automatic RECON loss notification (ARLN).

RSR tracking system can connect to a CSL to enable automatic RECON loss notification and to receive OM commands. If CSL available is available on the tracking site, tracking IMS registers with SCI and registers commands with OM. CSL is not required for tracking function, but it should be available if takeover is required. If using shared queues, takeover site must cold start, which means no message or status recovery is available at takeover.

3.16.2 Features not affected by CSL or STM

Many features are not affected by the implementation of CSL and STM (described in detail in the next chapter). Some are listed here:

- Application programs
  - Except CMD and ICMD entered commands may have local or global scope (same as terminal entered commands)
- Databases
- MSC, SPOOL API, FES, associated printers, time-controlled operations (TCO)
- APPC (except for APPC descriptor name consistency)
- OTMA, BTAM
- Definitional consistency of attributes across IMSs in IMSplex not verified for:
  - IMS system defined resources
  - APPC descriptors, ETO descriptors

4

Implementing the sysplex terminal management (STM)

This chapter describes the tasks necessary to implement the sysplex terminal management (STM):

- Setting SRMDEF and RCVYxxxx values
- Setting the resource structure
- IMS exits
- Considerations when STM is active

4.1 Enabling sysplex terminal management (STM)

Sysplex terminal management is enabled whenever IMS is running in a CSL environment with a resource structure and shared queues. Resource status recovery requires that all IMSs in the IMSplex belong to the same shared queues group. Without shared queues, resource status recovery is not functional. Section 3.2, “Setting up a CSL environment” on page 56 describes the steps required to enable the Common Service Layer environment.

This section provides a bit more detail on those parameters related directly to STM.

4.1.1 Prerequisites for STM

Here are the requirements for the sysplex terminal management function:

- Common Service Layer must be set up
  - Resource Manager (RM) to maintain global resource information using CQS
  - Structured Call Interface (SCI) to communicate between IMSplex address spaces
  - Operations Manager (OM) to manage global resource operations
- Common Queue Server (CQS) must be set up
  - Maintains global resource information on CF resource structure
  - Shared queues is not required unless SRMDEF=GLOBAL is used
- Coupling Facility (CF)
  - Contains resource structure with global resource information
- OS/390 V2 R10
  - Supports programmable LEIDs to enforce name uniqueness for IMS resources
- z/OS R1 V2 (optional)
  - Required to exploit VTAM Generic Resource enhancements, such as session level affinities

4.1.2 Migration and coexistence

Only IMS systems at Version 8 or later can participate in resource sharing. IMS Version 6 and IMS Version 7 systems may coexist for message processing in the shared queues group, but cannot participate in the systems management capabilities (particularly sysplex terminal management) of the IMSplex.

Care must be taken if terminals or users move between systems in an IMS Version 8 IMSplex, and Version 6 or Version 7 systems, because the terminal or user status cannot be shared between systems, but messages can, and this can lead to unpredictable results.

Important: When using sysplex terminal management, ensure that all the front-end (network-connected) IMS systems are at Version 8.

4.2 Recovery and recoverability status

Status recovery mode and status recoverability apply to each terminal/user session and, with one exception, are set when the terminal logs on. The exception is for dynamic non-STSN terminals, in which case the SRM is set when the ETO user signs on.

Each IMS can specify a default, and then the defaults can be overridden at logon or signon time by the Logon Exit (DFSLGNX0) or the Signon Exit (DFSSGNX0). The following paragraphs identify how to set SRMDEF and RCVYxxxx values, and how to override them using IMS exits.

4.2.1 System defaults in DFSDCxxx

In a CSL environment, there are four new parameters in PROCLIB member DFSDCxxx which determine the system defaults for SRMDEF and RCVYxxxx. One of these is SRMDEF, which is the IMS system default to use if not overridden at logon or signon time. Note that this is not an IMSplex-wide default. Each IMS can have different defaults, although why one would do this is unclear.

Three other parameters determine the recoverability of each type of end-user significant status: RCVYCONV, RCVYSTSN, and RCVYFP. These also are IMS system defaults to use when not overridden.

- SRMDEF=GLOBAL | LOCAL | NONE

  Specifies the status recovery mode for user and node resources.

- RCVYCONV=YES | NO

  Specifies whether the status of a conversation can be recovered.

- RCVYSTSN=YES | NO

  Specifies whether the status of STSN (SLUP, FINANCE, and ISC) can be recovered.

- RCVYFP=YES | NO

  Specifies whether the status of Fast Path can be recovered. RCVYFP applies to conversation status and Fast Path output.

- RCLASS=name

  Override value on SECURITY macro. Specifies an identifier of one to seven alphanumeric characters that is to be used to identify the IMS system as a resource class to RACF for transaction authorization and user ID verification. *IMS* is the default. The specification is valid only if one of the following is true in the IMS system definition:

  – TYPE=RACFTERM is specified on the system definition SECURITY macro; or RCF=A,T, or Y is specified on the IMS procedure.

  – TYPE=RACFAGN is specified on the system definition SECURITY macro, or ISIS=1 is specified on the IMS procedure.

If not specified in this member, defaults are determined by whether or not this IMS is running in a CSL with shared queues, an RM, and a resource structure. If so, then the SRMDEF default is GLOBAL, otherwise it is LOCAL. RCVYxxxx always defaults to YES unless SRMDEF is set to NONE.

These defaults can again be overridden on a session basis by:

- Logon exit routine (DFSLGNX0) - all but dynamic non-STSN
- Signon exit routine (DFSSGNX0) - dynamic non-STSN

Example 4-1 shows a DFSDCxxx sample definition member.

Example 4-1 DFSDCxxx member showing SRMDEF and RCVYxxxx parameters

```
SRMDEF=GLOBAL
RCVYCONV=YES
RCVYSTSN=YES
RCVYFP=YES
```

ETO descriptors can be updated for SRM and RCVY values, but it is not necessary. APPC descriptors should be examined to make sure they do not conflict with LTERM names defined on other IMS systems.

4.2.2 Setting status recovery mode (SRM)

Status recovery mode (SRM) identifies if, and how, the three types of end-user significant status is to be maintained.

There are three choices, SRMDEF=GLOBAL, LOCAL or NONE:

- SRMDEF=GLOBAL

  End-user significant status is to be maintained on the resource structure. Status is also maintained locally while the resource is active. When the resource becomes inactive, local status is deleted.

- SRMDEF=LOCAL

  End-user significant status is to be maintained in local IMS control blocks only. When the resource becomes inactive, its status is maintained in local control blocks and in the IMS system checkpoint log records.

- SRMDEF=NONE

  End-user significant status is to be maintained in local IMS control blocks only as long as the resource is active on that IMS. When the resource is no longer active, for any reason, that status is deleted.

Note: When a resource structure exists, resource entries with their SRM are maintained in that structure regardless of the SRMDEF setting. When no resource structure exists, the SRMDEF setting is kept in the local control blocks.

There is no SRM option for command significant status. It is always maintained globally if there is a resource structure, and always maintained locally if there is no structure.

4.2.3 Setting status recoverability (RCVYxxxx)

This parameter allows the user to determine whether specific end-user statuses should be recoverable.

The value is set by the RCVYxxxx parameter, where xxxx is CONV, STSN, or FP.

- RCVYCONV=YES | NO

  If *YES*, information required to recover a conversation will be kept across a session termination/restart and IMS termination/restart.

  If *NO,* conversational information is kept locally and only as long as the session is active.

- RCVYFP=YES | NO

  If *YES*, Fast Path (EMH) response mode is recoverable.

  If *NO*, Fast Path response mode is not restored after session termination and restart.

- RCVYSTSN=YES | NO

  If *YES*, STSN sequence numbers for both input and output messages will be saved and are recoverable.

  If *NO*, they are not recoverable. When a session terminates and then is restarted, the sequence numbers will revert to zero (cold start).

4.2.4 SRMDEF and RCVxxxx rules

There are three rules that apply when specifying these values:

- SRMDEF=GLOBAL cannot be specified without a resource structure
- RCVYxxxx=YES cannot be specified if SRMDEF=NONE
- RCVYSTSN=YES cannot be specified if RCVYFP=NO

If SRMDEF is not specified in DFSDCxxx, IMS will determine the system default according to the following rules:

- SRMDEF=GLOBAL when using a resource structure and shared queues
- SRMDEF=LOCAL when not using a resource structure and shared queues
- RCVYxxxx=YES unless SRMDEF=NONE

Note: Other than SRMDEF=GLOBAL, these parameters can be set even when not using the Common Service Layer (CSL). For example, if RCVYSTSN=NO, then STSN sequence numbers would be discarded no matter how a STSN session is terminated.

Table 4-1 describes the default values for RCVYxxxx as they relate to the values specified on the SRMDEF keyword.

Table 4-1 RCVYxxxx values related to SRMDEF values

	SRMDEF=GLOBAL	**SRMDEF=LOCAL**	**SRMDEF=NONE**
RCVYCON=YES	Valid (default)	Valid (default)	Invalid
RCVYCON=NO	Valid	Valid	Valid (default)
RCVYFP=YES	Valid (default)	Valid (default)	Invalid
RCVYFP=NO	Valid	Valid	Valid (default)
RCVYSTSN=YES	Valid (default)	Valid (default)	Invalid
RCVYSTSN=NO	Valid	Valid	Valid

4.2.5 ETO descriptors

For ETO, there is an addition to the user descriptor definitions that allows each of these parameters to be specified as defaults for that session. They are coded the same as in DFSDCxxx. They override the values specified in DFSDCxxx but can be overridden by the logon or signon exit. For example:

```
U HENRY LTERM=(HENRYSLT) SRMDEF=GLOBAL RCVYCONV=YES RCVYSTSN=NO RCVYFP=NO ...
```

4.3 Overriding SRM and RCVYxxxx defaults

The system defaults can be overridden for individual sessions at logon or signon time by the logon exit (DFSLGNX0) or signon exit (DFSSGNX0). If a resource entry already exists with

end-user significant status (and therefore already has SRM and RCVYxxxx settings), and when an end user logs on or signs on, then the exits cannot override them.

For example, if a session has SRMDEF=GLOBAL and RCVYCONV=YES, and then that session terminates with conversational status, the resource entry is not deleted (resource entries are not deleted when a session terminates with significant status).

When that user logs back on, the logon or signon exit cannot override these settings. Any attempt to override them is ignored. If, however, the logon exit (or signon exit) enters an *invalid* value, then the logon (or signon) will be rejected.

For example, the logon exit cannot specify SRMDEF=GLOBAL if there is no resource structure. This is invalid and the logon would be rejected.

4.3.1 Logon exit (DFSLGNX0)

The logon exit, when present in SDFSRESL, is driven whenever a *static or dynamic* terminal logs on. This exit knows what the system defaults are and can override them if they have not already been set in a previous session and still exist. If a resource already has an SRM and end-user significant status, the exit cannot change it.

4.3.2 Signon exit (DFSSGNX0)

The Signon exit is driven *only for ETO terminals* when the user signs on. For non-STSN terminals, the exit can override the SRMDEF and RCVYxxxx values if they have not already been set in a previous session and still exist.

For STSN terminals, since the STSN numbers are set at logon time and are associated with the VTAM session, the defaults for these settings (SRMDEF and all RCVYxxxx) can only be overridden by the Logon Exit - not by the Signon Exit.

> **Important:** SRMDEF and RCVYxxxx apply only to end-user significant status. They do not apply to command significant status. Command significant status is always kept globally if a resource structure exists, and locally if no resource structure exists. It is always recoverable.

4.4 Resource structure

IMS may (optionally) use a Coupling Facility list structure, called the resource structure, to share information about IMSplex resources among all members of the IMSplex. Transactions, MSNAMEs, NODEs, LTERMs, (static NODE) USERs, USERIDs, APPC descriptors, and IMSplex global resources may all have entries on the resource structure. Each entry may contain zero, one, or multiple 512 byte data elements, depending on the resource type and its status.

You don't need a resource structure, but if you don't have one, you are limited in what features are available. Basically, without a resource structure, you don't get sysplex terminal management.

Most resource structure management functions are done by the Resource Manager, CQS, or the system. The installation's primary responsibility is to define the structure in the CFRM policy, occasionally (perhaps) issuing some structure commands to alter or rebuild the structure, and then manage its size and location.

The resource structure must be defined in the CFRM policy with a maximum size (SIZE) and (optionally) an initial size (INITSIZE) and a minimum size (MINSIZE). If defined with INITSIZE and ALLOWAUTOALT(YES), the allocated size and/or the entry-to-element ratio will be adjusted automatically by the system in increments up to the maximum size when the FULLTHRESHOLD (default is 80%) is reached. Determining the correct size specifications for the resource structure depends on the ability to accurately estimate the number and size of the entries.

4.4.1 CFSIZER tool for resource structure definition

The CFSIZER tool provides sizing recommendations based on user-provided input. The user must estimate the *number of resource entries* on the structure, and the *number of data elements* associated with those entries, and provide those numbers to the tool. The tool then returns a recommendation for the SIZE (or INITSIZE) parameter for the resource structure definition in the CFRM policy.

The user may want to run this tool several times, once with the maximum number of resources and data elements possible and once with the expected average number of resources and data elements. The two values returned can then be used as the SIZE and INITSIZE values for the CFRM policy definition.

Calculating an average may be especially useful when a large number of the potential terminals/users are not logged on at the same time. By defining the CFRM policy with ALLOWAUTOALT(YES), the size of the structure and the entry-to-element ratio can be dynamically altered by the system as conditions warrant.

The CFSIZER tool is available on the Internet at URL:

http://www.ibm.com/servers/eserver/zseries/cfsizer/ims.html

Refer to 3.3.2, “Sample resource structure definition” on page 58 for an example of resource structure definition.

4.5 Resource types

The following are the types of resources on the resource structure, including when they are created and deleted, and whether or not they have data elements associated with them. Use these descriptions along with the resource table shown later when providing input to the CFSIZER tool.

4.5.1 IMSplex global resources

IMSplex global resources contain information about the IMSplex itself or its individual members and are created as the IMSplex is initialized, as members join the IMSplex, or as global processes are initiated. Once these resource list entries are created, they are not deleted. CFSIZER factors IMSplex global resources into its calculations.

4.5.2 Static transaction and MSNAME resources

Static transaction and MSNAME resources are created during IMS initialization or when added by online change. They are never deleted and remain on the resource structure as long as the structure exists. Transaction and MSNAME resources are easy to count — count the number of “unique” TRANSACT and MSNAME statements in the IMS system definition STAGE1 input. These resource list entries never have data elements.

4.5.3 CPI-C transaction resources

CPI-C transaction resources are created when the CPI-C transaction is first entered and are deleted when all IMSs defining that transaction have terminated. CPI-C transaction resources can be counted by counting the number of “unique” TRANCODEs in the TP_PROFILE data sets. These resource list entries only have data elements when the number of IMSs defining them is greater than two.

4.5.4 APPC descriptor resources

APPC descriptor resources are created during IMS initialization and are deleted when all IMSs defining that resource terminate. APPC descriptors can be counted by counting the number of “unique” APPC descriptors in DFS62DTx. These resource list entries only have data elements when the number of IMSs defining them is greater than two.

4.5.5 Sysplex terminal resources

Sysplex terminal resources such as NODEs, LTERMs, USERs, and USERIDs are created when the resource first becomes active (for example, terminal logon or user signon). They are deleted from the resource structure when the resource becomes inactive (for example, terminal logoff or user signoff) and the resource has no recoverable significant status. Note that USERIDs are always deleted when the user becomes inactive.

These resource list entries are the most variable in number and size, since they exist only when a resource is active, or when inactive but with significant status in the resource entry. The number and size of sysplex terminal resources depends on a number of factors:

- What is the resource type (NODE, LTERM, USER, USERID)?

  USERIDs have list entries only if single signon is being enforced. LTERMs have list entries but no data elements. Parallel session ISC NODEs are the only NODEs that have data elements.

  The most significant of the list entries in terms of size is the USER (or static NODE USER) entry which contains the majority of the significant status. End-user significant status, when it exists, is always kept in a data element in the (static NODE) USER entry. A (static NODE) USER resource entry always has at least one data element.

- Status Recovery Mode and Recoverability settings of the terminal

  Defaults for Status Recovery Mode (SRM) and Recoverability (RCVYxxxx) settings are defined in DFSDCxxx, and can be overridden by the Logon Exit or Signon Exit. They determine whether end-user significant status is maintained, and if so, where it is maintained:

  - If SRMDEF=GLOBAL and RCVYxxxx=YES, then end-user significant status will be kept in the USER entry data element in the resource structure.
  - If SRMDEF=LOCAL or NONE, end-user status is not kept in the USER entry.

  Command significant status is always kept in the resource structure when the structure exists, but (usually) doesn’t require a data element.

- Terminal or user significant status and the type of significant status

  Does the terminal or user have significant status and what type of significant status is it? This determines whether the resource list entry has data elements with it and if it is deleted when the resource becomes inactive.

4.6 Resource number

The resource number is the total number of resources expected to be in the resource structure at one time. One resource is stored on the resource structure as one list entry, which may contain zero, one, or more data elements, depending upon the amount of resource data.

4.6.1 Methods to calculate the resource number

One of the following methods can be used to calculate the value to use:

- Define a resource number of 1, if your installation is DBCTL-only and you only plan to use the resource structure for global online change.
- Calculate an initial resource number by using the resource table below and summing all of the numbers in the resource number column.
- Tune (adjust) the resource number by running workloads with the resource structure enabled and querying CQS statistics periodically to extract the list entry high water mark. You may want to size your structure large enough to accommodate the list entry high water mark, or some amount larger than the high water mark. The list entry high water mark field is SS3ENTHI in mapping macro CQSSSTT3. Structure statistics may be gathered through the CQS statistics user exit or the CSLZQRY macro interface. This requires a user-written program to issue the CQS macro and to return the results.

4.6.2 QUERY STRUCTURE command

The QUERY STRUCTURE command also displays the list entries allocated, the list entries in use, the data elements allocated, and the data elements in use. Figure 4-1 shows a QUERY STRUCTURE command entered through TSO SPOC.

```
  File  Display  View  Options  Help
 ____________________________________________________________________
 PLEX1                    IMS Single Point of Control
 Command ===> ______________________________________________________
 ____________________________________________________________________
  ________________________________ Plex . PLEX1  Route . __________  Wait . _____
 Response for: QUERY STRUCTURE NAME(*) SHOW(ALL)
StructureName    MbrName    CC Type     LeAlloc LeInuse ElmAlloc ElmInuse LE/EL>
IM0A_RSRC        IM1ARM      0 RSRC        1410      63      195       23 0467/0
IM0A_RSRC        IM4ARM      0 RSRC        1410      63      195       23 0467/0
IM0A_RSRC        IM3ARM      0 RSRC        1410      63      195       23 0467/0
```

Figure 4-1 QUERY STRUCTURE command

You can issue the QUERY STRUCTURE command periodically to get an idea of the maximum list entries that have been in use over a period of time. QUERY STRUCTURE output may also be used to determine how many more resources the resource structure can accommodate.

4.7 Data element number

A data element is a piece of storage on the Coupling Facility associated with a (resource) list entry. Resource entries may have zero, one, or more 512-byte data elements, depending on the amount of resource data stored.

4.7.1 Methods to calculate the data element number

One of the following methods can be used to calculate the value to use for the data element number.

- Define a data element number of 1, if your installation is DBCTL-only and you only plan to use the resource structure for global online change.
- Calculate an initial data element number by using the resource table below and summing all of the numbers in the data element number column that apply to your installation. The data element number is rarely more than 1, unless there is an exceptional amount of significant status associated with the resources (such as hundreds of LTERMs assigned to a USER, hundreds of held conversations associated with a USER, and so on), or a very large number of IMSs (more than 34), neither of which is likely.
- Tune the data element number by running workloads with the resource structure enabled and querying CQS statistics periodically to extract the data element high water mark. The data element high water mark field is SS3ELMHI in mapping macro CQSSSTT3. Structure statistics may be gathered through the CQS statistics user exit or the CSLZQRY macro interface.

4.7.2 QUERY STRUCTURE command

The QUERY STRUCTURE command can be used periodically to get an idea of the maximum data elements that have been in use over a period of time. Figure 4-1 on page 91 shows a QUERY STRUCTURE command entered through TSO SPOC.

QUERY STRUCTURE output may also be used to determine how many more resource data elements the resource structure can accommodate.

4.8 Resource table

Use the Table 4-2 to calculate the resource number and the data element Number needed to define an initial resource structure.

4.8.1 Calculating an initial resource structure

The number of entries and data elements used for IMSplex resources is estimated by the tool and not a required input.

Table 4-2 Resource table

Resource Type	Resource Number	Data Element Number
APPC descriptor	Total number of uniquely defined APPC descriptors in entire IMSplex	0 (if #IMSs<3) -or- if #IMSs > 2 #APPC descriptors * (#IMSs-2)/32 (rounded up)
LTERM	Total number of uniquely generated LTERMs + maximum number of dynamic LTERMs	0
MSNAME	Total number of uniquely generated MSNAMEs in entire IMSplex	0
NODE	Total number of uniquely generated NODEs + maximum number of dynamic NODEs	#ISC NODEs with multiple parallel sessions * (#IMSs-1)/32 (rounded up)

Resource Type	Resource Number	Data Element Number
Transaction (static)	Total number of unique static (generated) transactions in entire IMSplex	0
Transaction (CPI-C)	Total number of unique CPI-C transactions invoked by APPC	0 (if #IMSs < 3), -or- if #IMSs > 2 #CPI-Ctrans * (#IMSs-2)/32 (rounded up)
USERID	Maximum number of USERIDs signed on, if single signon enforced with SGN= not G, M or Z	0
USER	Maximum number of dynamic USERs + total number of unique static ISC subpools	#USERs * 1
USER (for static NODE)	Total number of unique static single-session terminals	#staticUSERs * 1

4.9 Defining the resource structure

Like all Coupling Facility structures, the resource structure must be defined in the CFRM policy and that policy must be activated.

4.9.1 CFRM policy definition options

When defining the structure, the user must consider the following definitions:

- SIZE, INITSIZE, MINSIZE

  Sets initial, maximum, and minimum size for structure.
- DUPLEX(ALLOWED | ENABLED | DISABLED

  Determines whether structure is to be duplexed.
- ALLOWAUTOALT(YES | NO)

  Determines whether the system can alter the size and entry-to-element ratio of the structure.

  FULLTHRESHOLD(80 | nn)

  Sets percent full which will invoke autoalter.
- REBUILDPERCENT(nn)

  Determines percentage of lost connectivity to invoke system initiated rebuild.
- PREFLIST(cfnn cfnn cfnn)

  Identifies candidate coupling facilities where structure can be allocated.

An example of the resource structure definition can be found in 3.3.2, “Sample resource structure definition” on page 58.

4.10 Adjusting the size of the resource structure

Once the structure has been sized and is in production, it may become necessary to change the maximum size upward due to unexpected volumes or initial miscalculation.

This can be done by changing the SIZE parameter in the CFRM policy, activating the new policy, and rebuilding the structure using the command:

```
SETXCF START,REBUILD,STRNM=structure-name
```

The size can be adjusted downward in the same way, but the alter command can also be used to make the structure smaller.

This might be done following an unusual period where the structure was autoaltered upward due to a high number of concurrent logons which are not expected to continue. Use the command:

```
SETXCF START,ALTER,STRNM=structure-name,SIZE=size
```

4.11 IMS exits using STM

There are three IMS exits which must be considered when using STM. The reader should refer to *IMS Version 8: Customization Guide,* SC27-1294, for a complete description of these exits.

4.11.1 Logon exit (DFSLGNX0)

The logon exit (DFSLGNX0), if it exists, is driven whenever a terminal logs on to IMS. With STM enabled, the logon exit has some additional capabilities when any static or dynamic STSN terminal logs on. It is able to:

- Override the SRMDEF and RCVYxxxx defaults for this logon session. This applies to all static and dynamic STSN terminals. It does not apply to dynamic non-STSN.
- Allow NODEs which are owned by an inactive IMS to be *stolen*. This function applies to all terminals for which this exit is driven. It should only be necessary when SRMDEF=LOCAL and the user does not want to wait for the failed IMS to be restarted. User data could be included on the logon request which could then be used by the exit as a signal that the user really means it. That is, that the user wants the NODE to be stolen. When a NODE is stolen, the logon is accepted, ownership is reset to the new IMS, and any end-user status will be lost. Command significant status will be restored on the new IMS.

4.11.2 Signon exit (DFSSGNX0)

The signon exit (DFSSGNX0), if it exists, is driven when a user signs on from a *dynamic (ETO) terminal*. It has capabilities similar to the logon exit, except that it only applies to the User signon. It can:

- Override the SRMDEF and RCVYxxxx defaults for this USER. This function applies only to dynamic non-STSN terminals. For all others, these values can only be overridden in the logon exit.
- Allow USER resource to be stolen if it is owned by an inactive IMS. Like with the logon exit, all end-user status will be lost. Command significant status will be restored.

4.11.3 Output creation exit (DFSINSX0)

Prior to IMS Version 8 and without shared queues, DFSINSX0 can be used only to create dynamic LTERMs. With shared queues, DFSINSX0 could also be used to create dynamic transactions. The exit must specify transaction characteristics, conversational, response mode, Fast Path, and so on. DFSINSX0 is used only for queuing to the shared queue structure when the transaction is not defined in local IMS. The exit cannot be sure it exists in

another IMS, and if the transaction is not defined anywhere, the message stays on the shared queue structure until manually dequeued or structure deleted.

With IMS Version 8 and RM structure, when a message arrives in IMS, and IMS does not know the destination (FINDDEST fails), then that IMS will query RM to determine whether that destination is already defined by another IMS. *If RM finds the resource, then IMS will dynamically create the control blocks (requires ETO for LTERM creation) and queue the message accordingly.*

However, without further knowledge, IMS does not know the characteristics of the resource, and so chooses the defaults. For example, if the destination were a transaction, the IMS would queue it as non-response mode, non-conversational, non-Fast Path, and so on. This could cause problems if the default characteristics are not correct.

When a message with an unknown local destination is received, IMS will query RM to see if it exists in the RM structure. If the structure has this message destination registered as a transaction, then the output creation exit (if it exists) will be called. The exit can allow the message to be queued as a transaction with the IMS default characteristics, it can change the characteristics of the transaction (for example, conversational, Fast Path, and so on), or it can reject the message, in which case IMS will discard it as an unknown destination.

The user should consider writing an output creation exit to do just this. It is unlikely that the default characteristics of a message are correct. To avoid queuing messages invalidly just because some other IMS has them defined, this exit should reject all messages unless it has explicit knowledge of the characteristic of the destination.

If RM identifies the message destination as an LTERM, then the exit will not be called and IMS will queue the message according to its definitions in the resource structure.

4.11.4 Command authorization exit (DFSCCMD0)

The command authorization exit routine (DFSCCMD0) can be used to verify that a command is valid from a particular origin. DFSCCMD0 is an optional exit routine for commands entered from IMS terminals, including LU 6.2 and OTMA.

DFSCCMD0 is a required exit routine if it is specified to authorize commands entered from:

- ICMD DL/I calls (from automated operator applications)
- OS/390 MCS or E-MCS consoles

This exit routine verifies that the user is authorized to issue a particular command. IMS does not call this exit routine for internally generated or auto-restart commands. Some classic commands may come from OM.

4.12 Considerations when STM is active

There are several considerations and changes when STM is active.

4.12.1 Global callable services

Callable services are services available to most IMS exits, including the logon exit and signon exit. There is one change that applies when STM is in effect. When the exit is using callable control block services, then a new default *global* option is in effect. Global resources accessible are NODEs, USERs, and LTERMs.

If the resource is active in or owned by the local system, the local control block will be returned regardless of global or local option. If the resource is not active in or owned by the local system, but exists in the RM structure, global information is returned and the local control block is ignored. A *hidden* control block will be returned which reflects the global status in RM. If the resource exists locally, but does not exist in RM, the local control block will be returned, even if not active and not owned.

When using the FIND or SCAN option, if the control block is not active in the local system, then RM will be queried.

- If the resource is found on the structure, then the address returned will be that of hidden control blocks with global information contained in the structure. These are mapped the same as the real control blocks, but contain only that information known to the structure.
- If the resource is active on another IMS, that IMS is NOT queried for more information. One example of the use of this capability is for a signon exit. The exit may issue a FIND or SCAN for an LTERM name for purposes of assigning a name to an ETO user during the signon process. If that LTERM is registered anywhere in the IMSplex, the exit will find it and can use a different name to build a CNT control block.

Note: The default is global. If the user wants only to FIND or SCAN local control blocks, the exit must be changed to set a flag in the function specific parameter list. This is documented in *IMS Version 8: Customization Guide,* SC27-1294.

4.12.2 Extended Recovery Facility (XRF)

Here are the following considerations when running with Extended Recovery Facility (XRF) and STM:

- Both the active and alternate IMSs must have a local SCI address space.
- The owner of a resource is the RSENAME, not the IMSID. This is so that if the alternate has to take over, it will own those resources owned by the failing IMS.
- XRF and VTAM generic resources are mutually exclusive, so only the RM affinity will force a user to log back on to the “right” IMS.
- Terminal/use status recovery depends on the terminal class:
  - Class 1 terminals

    A backup session is maintained with a class 1 terminal. When the alternate takes over, the session is automatically re-established with the new active. Any session status that existed on the old IMS is determined from the log records, not from the structure. The SRMDEF option only applies if a session is terminated, not if IMS fails. If global and local status differ, then local status prevails, even if SRMDEF=NONE, terminal/user status is recovered (based on log records, not on the resource entry).
  - Class 2 terminals

    For class 2 terminals, the new active will reacquire the session (OPNDST). In this case, status recovery is determined by the SRMDEF value. Whatever is in the resource entries prevails. If SRMDEF=NONE, then the status will be deleted.
  - Class 3 terminals

    Class 3 terminals are treated like any terminal in a non-XRF environment. Ownership (RM affinity) is released only if SRMDEF=GLOBAL or NONE. Status is recovered globally or locally, depending on SRMDEF. If SRMDEF=LOCAL, the user must log on to the new active.

4.12.3 Rapid Network Reconnect (RNR)

Rapid Network Reconnect (RNR) is intended to hold a user's VTAM session information in a data space for single node persistent sessions (SNPS) or in the Coupling Facility for multiple node persistent session (MNPS) until that IMS is restarted, opens its VTAM ACB, and issues /START DC (logons accepted).

At this time, the user is back in session with the original IMS (assuming RNR=ARNR). Any attempt to log on to another IMS would fail, since that NODE is already in session (discuss this with your VTAM expert to see how to get around this).

This defeats the purpose of STM, since the user is always logging back on to the same IMS. So it is unlikely that RNR would be used in conjunction with STM, at least not with SRMDEF=GLOBAL. RNR keeps the session alive for the surviving IMS to be restarted then re-establishes the session. This would prevent the user from logging on to another IMS.

5

Other Common Service Layer (CSL) functionalities

This chapter describes the tasks necessary to implement other functionalities made possible by the Common Service Layer (CSL):

- Setting up the global online change
- Setting up a single point of control (SPOC)
- Setting up an user written interface to Operation Manager (OM)
- Setting up the automatic RECON loss notification (ARLN)
- Language Environment® (LE)

5.1 Global online change

Global online change is sometimes called coordinated online change, or even coordinated global online change. We call it global online change in this book.

Global online change requires a CSL environment with all the address spaces, but does not require a structure. If one exists, it will be used but is only beneficial for restarting global online change after certain types of failures.

The following topics discuss how to implement and manage the global online change feature included with IMS Version 8.

Global online change is activated for an IMS when OLC=GLOBAL is specified in the DFSCGxxx member which it is using.

5.1.1 Setting up the global online change

In an IMSplex, global online change allows you to perform online change to resources across the IMSplex.

To enable online change, a new data set pointed to by the OLCSTAT DD statement must be created and initialized. It only needs to be initialized once. As each IMS joins the IMSplex, it will update OLCSTAT. Although all IMSs must use the same OLCSTAT, they do not have to use the same OLC data sets (ACBLIB, CTLBLKS, FORMAT, MATRIX). You may decide that you want (for example) each IMS to have its own copy of ACBLIB. In this case, it is a user responsibility to make sure the copies are in sync (they all use the same A or B suffix). And DFSCGxxx should specify NORSCCC for whatever libraries are not the same data set.

To enable global online change for an IMSplex, the system programmer must:

- Define the global OLCSTAT data set.
- Remove local MODSTAT definitions.
- Define the OLCSTAT data set name in the DFSCGxxx PROCLIB member.

5.1.2 Requirements for the global online change

Global online change requires the OLCSTAT data set and the use of the Common Service Layer (CSL). In IMS PROCLIB member DFSCGxxx, OLC=GLOBAL and OLCSTAT=data set name, must be defined.

Common Service Layer (CSL) requirements:

- Structured Call Interface address space on each system in the IMSplex
- Operations Manager
- Resource Manager
  - At least one RM in the IMSplex
  - Resource structure required for resource consistency checking

5.1.3 Preparation for the global online change

Preparation for global online change is the same as that of local online change. The IMS resources are generated with the appropriate utility process:

- DBDGEN, PSBGEN, and ACBGEN
- MODBLKS system generation
- MFS utility
- Security generation using the security maintenance utility (SMU)

Before the online change may be done, the updated staging libraries must be copied to the inactive libraries. This is done with the online change copy utility (DFSUOCU0). It is the same utility that is used with local online change. It is available with previous IMS releases, and has been enhanced to support global online change. It reads OLCSTAT to determine inactive libraries.

5.1.4 Overview of execution

Global online change is invoked using the new IMSplex commands, also called *enhanced commands*.

Online change is started invoking the INIT OLC PHASE(PREPARE) command. This is similar to the /MODIFY PREPARE command used with local online change. It also coordinates the prepare processing across all of the IMSs.

After the prepare has completed, the actual changes are invoked with the INIT OLC PHASE(COMMIT) command. This is similar to the /MODIFY COMMIT command used with local online change.

An online change may be aborted by issuing a TERMINATE OLC command. It coordinates the online change abort phase across all the IMSs in the IMSplex. TERMINATE OLC is similar to the /MODIFY ABORT command used with local online change.

A QUERY MEMBER command may be used to show online change status of IMSs. It reports the current status of each IMS participating in the online change.

5.1.5 DFSCGxxx PROCLIB member

The following parameters in DFSCGxxx PROCLIB member are used for global online change:

- OLC=LOCAL | GLOBAL

  GLOBAL indicates that global online change is enabled. It is alright to have some IMSs global and others local.

- OLCSTAT=

  Data set name, required with OLC=GLOBAL and it must be the same for all IMSs in IMSplex. If OLC=GLOBAL is specified, an OLCSTAT specification is required. OLCSTAT is the data set name for the online change status data set (OLCSTAT). All IMSs in an IMSplex must define the same physical OLCSTAT data set. The data set is dynamically allocated. MODSTAT is not required for global online change but is still used for local online change.

- NORSCCC=

  The NORSCCC parameter is used to specify that consistency checking will not be done for some libraries. Consistency checking verifies that all IMS systems are using the same data set names for their libraries affected by global online change.

  Up to three values may be specified for NORSCCC. ACBLIB indicates that ACBLIB data set names are not checked for consistency. FORMAT indicates that the FORMAT data set names are not checked for consistency. MODBLKS indicates that MODBLKS and MATRIX data set names are not checked for consistency. There is no "ALL" value for NORSCCC. If you want to turn off consistency checking for all of the libraries, specify:

  ```
  NORSCCC=(ACBLIB,FORMAT,MODBLKS)
  ```

  The default is that all IMSs will use the same OLC data sets (MODBLKS, ACBLIB, FORMAT). If you don't want to use the same data sets (same DSNs) then specify

NORSCCC= for whichever ones are different. It is the user's responsibility to be sure that even though the data sets are different, the contents are the same.

If NORSCCC is omitted, IMS ensures that the ACBLIB, FMTLIB, MODBLKS, and MATRIX dataset names are consistent across the IMSplex. All dataset concatenations are validated. The dataset concatenations must be defined in the same order across the IMSplex. Resource consistency checking does not ensure whether the contents of these datasets are consistent.

NORSCCC must be used if you don't have cloned systems and your datasets are different in the different IMSs.

Figure 5-1 is and example of a DFSCGxxx member used to enable online change.

Example 5-1 DFSCGxxx for global online change

```
*-------------------------------------------------------------------*
* IMS COMMON SERVICE LAYER PROCLIB MEMBER                           *
*-------------------------------------------------------------------*
CMDSEC=N,                    /* NO CMD AUTHORIZATION CHECKING     */
IMSPLEX=PLEX1,               /* IMSPLEX NAME                      */
OLC=GLOBAL,                  /* GLOBAL ONLINE CHANGE              */
OLCSTAT=IMSPSA.IMA0.OLCSTAT
* NORSCCC=(ACBLIB,FORMAT,MODBLKS)
*-------------------------------------------------------------------*
* END OF MEMBER DFSCG000                                            *
*-------------------------------------------------------------------*
```

Note: Make sure all IMS systems in the IMSplex are using the same OLCSTAT data set and ideally the same DFSCGxxx member. Also make sure you take a backup of your OLCSTAT data set following every successful online change, (the same as you would for existing MODSTAT data sets).

5.1.6 Mixed environment

You can have a mixed online change environment where some IMSs use global online change and some use local online change. Of course, you would have to coordinate the changes between the systems using global online change and those using local online change.

Some members of the IMSplex can be defined with global online change (OLC=GLOBAL in DFSCGxxx), and the INITIATE OLC commands would effect all instances.

Some members could be defined to used local online change (the default or OLC=LOCAL in DFSCGxxx), and would use the **/MODIFY** commands on each IMS instance. You would then have to coordinate the changes.

5.2 Resource consistency checking

Resource consistency checking may be used to ensure that each IMS system using global online change is using the same data sets. This capability requires the use of a resource structure.

If a resource structure is defined, the default is that resource consistency checking is performed for all data sets. Resource consistency checking is optional, but as it is the default,

you have to explicitly disable it if you don't want it to happen. With consistency checking enabled, IMS ensures that the OLCSTAT data set names are consistent. If they are not consistent, IMS initialization fails.

5.2.1 Checking function

The checking may be turned off by use of the NORSCCC parameter in the DFSCGxxx member for an IMS system.

```
NORSCCC=(ACBLIB,FMTLIB,MODBLKS)
```

Specifying MODBLKS indicates that neither MODBLKS nor MATRIX data set names will be checked for consistency.

If resource consistency checking is used for ACBLIB, initialization for an IMS system will fail if it is not using the same ACBLIB data set names that are being used by other members of the IMSplex which are also invoking consistency checking.

5.2.2 Checking choice

Resource consistency checking is optional. An installation should decide if it is beneficial for its environment.

Consistency checking creates a single point of failure. If a data set is lost, it is lost for all members of the IMSplex using this capability. An advantage of consistency checking is that only one execution of the online change copy utility is required to copy the staging data set to the inactive data set. It also ensures that the same datasets are used for all IMS systems in the IMSplex.

Turning off consistency checking creates some operational exposures. If the data sets used by each of the IMSs differ, unexpected and unwanted results could occur. To avoid this, the online change copy utility must be executed multiple times, once for each IMS system. Each time the same input data set must be used, but the output data set must be different.

5.3 OLCSTAT data set

The OLCSTAT data set is new for IMS Version 8. One OLCSTAT data set is shared by all of the IMS systems in the IMSplex. Its function is similar to that of the MODSTAT data set used with local online change. It contains the online change status. This includes an indication of which DDNAMEs are the active libraries. It also contains a list of the IMS systems which participate in global online change.

5.3.1 Definition of OLCSTAT data set

The OLCSTAT data set is BSAM. It contains one record of variable size. IBM suggests the data set attributes shown in Example 5-2, to support an IMSplex of up to 65 IMSs.

Example 5-2 OLCSTAT allocation attributes

```
DSORG     Sequential
RECFM     V
LRECL     5200
BLKSIZE   5204
```

The OLCSTAT data set is initialized by using the new global online change utility.

The OLCSTAT data set is dynamically allocated by IMS using the data set name defined in the DFSCGxxx PROCLIB member (OLCSTAT=data set name). An OLCSTAT DD statement should not be defined in the IMS procedure.

5.3.2 OLCSTAT data set header record

The OLCSTAT dataset contains a header, after being initialized by the global online change utility. The header contains control information, and information pertinent to global online change, such as the active online change libraries, the modify id, and whether or not a library was changed by the last online change.

The OLCINP is a literal that represents a lock word, indicating that the OLCSTAT dataset is locked by online change. If OLCINP is set and no IMS is up, then a severe error occurred. This lock word will have to be cleaned up by the global online change utility.

The MODBLKS, IMSACB, and FORMAT DDnames for the A and B libraries are defined in the IMS control region JCL. The suffix (A or B) indicates which library is the active library.

Each library has an associated last online change indicator after the library name and active suffix. The last online change indicator is set to Y if the associated library was changed during the last online change. The last online change indicator is set to N if the associated library was not changed during the online change library. If all libraries show an indicator of N, that means that no global online changes have taken place.

5.3.3 OLCSTAT data set IMS record

The OLCSTAT data set contains one IMS record for each IMS current with the online change libraries, and zero IMS records if no IMS has been cold started yet.

The OLCSTAT data set IMS record format is the following:

- IMS record length
- IMS record version
- IMSid
- Online change prepare timestamp on this IMS
- Online change commit timestamp on this IMS

5.3.4 OLCSTAT data set examples

Example 5-3 shows the OLCSTAT dataset contents for two IMSs that have been cold started, but no global online change has been performed yet.

Example 5-3 OLCSTAT dataset content

```
810                         1 MODBLKSA N IMSACBA N FORMATA N
810 IM1A
810 IM2A
```

Based on this data set, we can tell the following:

- DFSUOLC0 have been run on IMS at the 810 level initializing the OLCSTAT data set
- Modify id is initialized to 1
- MODBLKS DD name suffix is initialized to A
- IMSACBA DD name suffix is initialized to A
- FORMAT DD name suffix is initialized to A
- IM1A has been restarted
- IM2A has been restarted

Example 5-4 shows the OLCSTAT dataset contents for two IMSs that have successfully participated in the 9th global online change for MFS FMTLIB.

Example 5-4 OLCSTAT dataset content

```
810 IM2A            10 MODBLKSB N IMSACBB N FORMATA Y
810 IM1A 2001073 1334121 -07:00 2001073 1334132 -07:00
810 IM2A 2001073 1334125 -07:00 2001073 1334131 -07:00
```

Based on this data set, we can tell the following:

- DFSUOLC0 have been run on IMS at the 810 level initializing the OLCSTAT data set
- IM2A was master of the commit phase of the last successful online change
- Modify id is now 10 (9 online changes done)
- MODBLKS DD name suffix remains as B
- IMSACBA DD name suffix remains as B
- FORMAT DD name suffix is now A
- FORMAT last OLC indicator is Y for yes
- IM1A record indicates IM1A participated in the last online change
- IM2A record indicates IM2A participated in the last online change
- IM1A and IM2A may be warm started

5.4 Global online change utility (DFSUOLC0)

The global online change utility (DFSUOLC0) is a new utility in IMS Version 8. It is used for the following four functions:

- To initialize the OLCSTAT data set.
- To add IMS systems to the list of IMSs using global online change. This is usually not required. This function may be needed if you lose the OLCSTAT data set and must rebuild it.
- To delete IMS systems from the list of IMSs using global online change. You may need to do this if you decide not to use an IMS system again. This will be explained later.
- To unlock the OLCSTAT data set. This is only required in the rare case that every IMS system participating in global online change abends while doing online change.

The global online change utility must be run to initialize the OLCSTAT dataset before the first IMS coldstarts the first time, to define the suffix of the active ACBLIB, MODBLKS, and FMTLIB libraries.

Warning: If the global online change utility is run by mistake, it can destroy the OLCSTAT dataset contents. IMSs that are up will not be able to do global online change. IMSs that you try to bring up will not be able to initialize.

An installation should have a procedure in place for recreating the OLCSTAT dataset after severe error (including user error where the global online change utility was run in error).

5.4.1 DFSUOLC procedure

IMS supplies the global online change utility procedure in PROCLIB. Example 5-5 is an example of the DFSUOLC procedure.

Example 5-5 DFSUOLC procedure

```
//       PROC FUNC=INI,ACBS=,MDBS=,FMTS=,MDID=,PLEX=,SOUT=A
//STEP1  EXEC PGM=DFSUOLC0,PARM=(&FUNC,&ACBS,&MDBS,&FMTS,&MDID,&PLEX)
//STEPLIB  DD DSN=IMSPSA.IMA0.&SYS2.SDFSRESL,DISP=SHR
//OLCSTAT  DD DSN=IMSPSA.IMA0.OLCSTAT,DISP=OLD
//SYSUDUMP DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT
//SYSIN    DD DUMMY
```

The procedure contains the following symbolics and in turn program parameters:

- FUNC

  Defines the function:

  - *ADD* to add one or more IMSs (after severe error)
  - *DEL* to delete one or more IMSs (after severe error)
  - *INI* to initialize the OLCSTAT dataset before the first IMS is cold started the first time
  - *UNL* to unlock the OLCSTAT dataset (after severe error)

- ACBS

  Defines the IMS JCL IMSACB DD statement suffix for the active ACBLIB library.

- MDBS

  Defines the IMS JCL MODBLKS DD statement suffix for the active MODBLKS library.

- FMTS

  Defines the IMS JCL FORMAT DD statement suffix for the active MFS format library.

- MDID

  Specifies initial modification ID (FUNC=INI). The modification ID is a number which is incremented with each online change. It is similar in function to the MODSTAT identifier used by local online change.

- PLEX

  Specifies IMSplex name, used only for unlock function (FUNC=UNL)

- SOUT

  Defines the SYSOUT class

//SYSIN DD is used to specify IMS systems for the add and delete functions.

5.4.2 Initializing OLCSTAT data set

Example 5-6 is an example of using the global online change utility to initialize an OLCSTAT data set. The DFSUOLC PROC is used.

Example 5-6 Initializing the OLCSTAT data set

```
//JOUKO3X  JOB (999,POK),'OLC',NOTIFY=&SYSUID,
//    CLASS=A,MSGCLASS=T,
//    MSGLEVEL=(1,1)
//*
//S EXEC DFSUOLC FUNC=INI,ACBS=A,MDBS=A,FMTS=A,MDID=1
```

This shows how a typical user would initialize the OLCSTAT data set. The initialization function (INI) is specified. All DDNAME suffixes are set to "A". This means we will begin with

IMSACBA, MODBLKSA, MATRIXA, and FORMATA as the active library DDNAMEs. The initial modification id number is set to 1.

5.5 Online change utility (DFSUOCU0)

The online change utility (DFSUOCU0) has a new name in IMS Version 8. It is the online change copy utility. The new name more clearly defines its function. Typically, it copies a staging library to an inactive library, but there are several other options for input and output data sets.

5.5.1 OLCUTL procedure

IMS supplies the online change utility procedure in PROCLIB. Example 5-7 contains an example of the OLCUTL procedure.

Example 5-7 OLCUTL procedure

```
//        PROC TYPE=,IN=,OUT=,SOUT=A,SYS=,SYS2=,
//         OLCGLBL=,OLCLOCL='DUMMY,'
//S       EXEC PGM=DFSUOCU0,PARM=(&TYPE,&IN,&OUT)
//STEPLIB  DD DSN=IMSPSA.IMA0.SDFSRESL,DISP=SHR
...
//MODSTAT  DD &OLCLOCL.DSN=IMSPSA.&SYS.MODSTAT,
//         DISP=SHR
//OLCSTAT  DD &OLCGLBL.DSN=IMSPSA.IMA0.OLCSTAT,
//         DISP=OLD
...
```

To support global online change the utility can read the OLCSTAT data set to determine the inactive library. A new value for the OUT= parameter (G) allows this. The OLCSTAT DD statement has been added to the procedure for this utility.

- TYPE

  Specifies the library to be copied. It can be the ACB, FORMAT, MATRIX, or MODBLKS library.

- IN

  Defines the library DDNAMEs to be used as input: If *S*, the IMS staging library (IMSACB, FORMAT, MATRIX, or MODBLKS); If *I*, a user input library (IMSACBI, FORMATI, MATRIXI, or MODBLKSI).

- OUT

  Defines the library DDNAMEs to be used for output: If *A*, the IMS A library (IMSACBA, FORMATA, MATRIXA, or MODBLKSA); If *B*, the IMS B library (IMSACBB, FORMATB, MATRIXB, or MODBLKSB); If *O*, a user output library (IMSACBO, FORMATO, MATRIXO, or MODBLKSO).

  If *U*, the target library (inactive) is determined by the utility, using the MODSTAT data set. The target will be the library not currently in use by the IMS online system. This is the recommended value when local online change is used.

  If *G*, (new in IMS Version 8) the target library (inactive) is determined by the utility, using the OLCSTAT data set. The target will be the library not currently in use by the IMS online system. This is the recommended value when global online change is used.

5.5.2 Migration to global online change

IMS systems may be migrated to global online change one system at a time. The process is shown here.

1. Define OLCSTAT data set.
2. Run the DFSUOLC0 utility to initialize OLCSTAT data set, before the IMS is cold started for the first time.
3. Shut down an IMS.
4. Remove MODSTAT DD statements from the IMS control region JCL.
5. If you are using XRF, also remove MODSTAT2 DD.
6. Define the IMS's DFSCGxxx with OLC=GLOBAL & OLCSTAT=dsname.
7. Cold start the IMS.
8. Perform coordinated online change on all IMSs defined with DFSCGxxx OLC=GLOBAL. IMSs defined with OLC=LOCAL or at a previous release of IMS must perform local online change.

It is advisable to limit the time that some systems are using local online change and some are using global online change. The support for the mixed environment is provided to facilitate the migration of systems one at a time.

Important: Note that a change to global online change requires a cold start of the IMS system.

5.5.3 Fallback to local online change

Fallback from global online change to local online change is also supported. This may also be done one system at a time. The process is:

1. Define DFSCGxxx OLC=LOCAL.
2. Shut down IMS.
3. Add MODSTAT DD statement to IMS control region JCL.
4. If you are using XRF, also add MODSTAT2 DD.
5. Remove the OLCSTAT DD statement from the control Region JCL.
6. Run INITMOD job to initialize MODSTAT data set.
7. Cold start IMS.
8. Perform local online change on each IMS and manually coordinate online change with the other IMSs.

Important: Note that fallback to local online change from global also requires an IMS cold start.

5.6 Setting up the single point of control (SPOC)

A single point of control (SPOC) is a terminal where you can manage operations of all IMSs within an IMSplex instead of using a master terminal.

On a workstation, the SPOC is an IMS system-management application — the Control Center. On a 3270-type TSO terminal, a TSO SPOC is controlled by an IMS

system-management application using an ISPF panel interface. The REXX SPOC API allows automation programs to use SPOC functions.

SPOC means that you can issue commands to all members of an IMSplex at one time. There can be any number of SPOC users active at any time. See Figure 5-1.

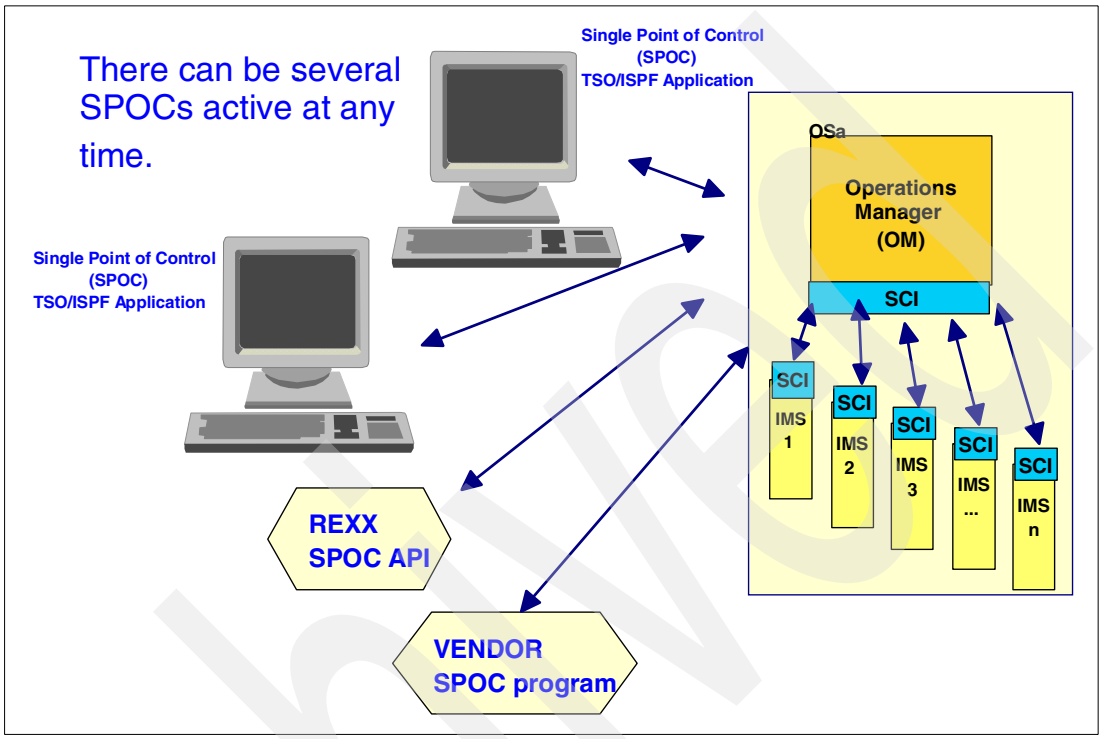

Figure 5-1 Several SPOCs at any time

A SPOC communicates with a single OM. Through SCI, OM then communicates with all of the other IMS control regions in the IMSplex.

Using a SPOC application, you can:

- Issue commands to all the IMS subsystems in an IMSplex.
- Display consolidated responses from those commands.
- Send a message to an IMS terminal that is connected to any IMS in the IMSplex using the BROADCAST command.

The use of a SPOC application is not required. The existing command interfaces for the WTOR, MTO, and E-MCS console continue to be supported.

5.7 Setting up the TSO SPOC program

There are a couple of methods for setting up the TSO SPOC program.

5.7.1 Updating the logon procedure

One method is to edit the logon procedure to include copies of the IMS distribution libraries or to issue ALLOCATE commands to make the data sets available to the TSO user. The data sets are shown in Table 5-1 on page 110.

After the data sets are added to your ISPF environment, you can start the SPOC program by entering:

```
TSO DFSSPOC
```

Table 5-1 Data sets for TSO SPOC setup

Usage	Data Set
TSOLIB command or STEPLIB	IMS.SDFSRESL
FILE(ISPPLIB)	IMS.SDFSPLIB
FILE(ISPMLIB)	IMS.SDFSMLIB
FILE(ISPTLIB)	*user*.ISPTLIB, IMS.SDFSTLIB
FILE(ISPTABL)	*user*.ISPTLIB
FILE(SYSPROC)	IMS.SDFSEXEC

5.7.2 Using ALTLIB and LIBDEF

A second method is to use the TSO ALTLIB command and ISPF's LIBDEF service. The DFSSPSRT exec provides an example of starting the DFSSPOC program using ALTLIB and LIBDEF.

A SPOC startup REXX exec is supplied in IMS.SDFSEXEC(DFSSPSRT), which takes the IMS high level qualifier as a parameter and performs the necessary allocations. It can be invoked from an ISPF command line as follows:

```
TSO ex 'IMS810.SDFSEXEC(DFSSPSRT)' 'hlq(IMS810)'
```

Where, the data set name and the HLQ value uses your company's data set name prefix for the IMS distribution libraries. This will allocate the libraries and invoke the TSO SPOC application, and will place you on the TSO SPOC command entry panel.

The TSO data set allocations required for SPOC are also shown in Example 5-8.

Example 5-8 TSO allocations for SPOC

```
ISPLLIB     imspref.SDFSRESL
ISPPLIB     imspref.SDFSPLIB
ISPMLIB     imspref.SDFSMLIB
ISPTLIB     tsouserid.USERTLIB  Note-each SPOC user must have own USERTLIB
............imspref.SDFSTLIB
ISPTABL     tsouserid.USERTLIB  Note-each SPOC user must have own USERTLIB
SYSPROC     imspref.SDFSCLST
SYSEXEC     imspref.SDFSEXEC
```

5.7.3 Invoking IMS single point of control (SPOC)

SPOC is invoked (subsequent to the correct allocation of datasets) by issuing the TSO command DFSSPOC. Figure 5-2 shows the TSO SPOC command entry panel.

```
  File  Display  View  Options  Help
SPOC                    IMS Single Point of Control
Command ===> _
                                 Plex . _____  Route . _________  Wait . _____
Response for:

CSLM000I (C) Copyright IBM Corp. 2000. All rights reserved.

The user preferences have not been set. Please set them now using the
preferences panel from the Options list.
```

Figure 5-2 SPOC command entry panel

The initial panel contains the command input area for IMS commands up to 145 characters, the scrollable response area, and fields to override the target IMSplex name, the specific members to receive the routed command, and the time to wait for the command to complete.

These fields can be used to temporarily override the default values specified in the IMS SPOC preferences panel.

5.7.4 Setting SPOC preferences

Before utilizing the SPOC application, it is necessary to provide some default preferences. The preferences panel is used to set default values and to select some processing options.

Figure 5-3 shows the SPOC preference panel, and the preferences shown should be adequate to get you started exploring the features of the TSO SPOC application.

```
  Help
SPOC                IMS Single Point of Control Preferences

Command ===> _____________________________________________________________
Select your options and press the Enter key.


Default IMSplex . . . . PLEX1
Default routing . . . . *_____________________________________________________
                        ______________________________________________________

Wait interval . . . . . 0:30___
Waiting preference  . . 1_  1.  Wait for command to complete.
                            2.  Do not wait for command response.

Command shortcuts . . . 1_  1.  Use command shortcuts.
                            2.  Do not use command shortcuts.

Shortcut processing . . 1_  1.  Merge explicit and default parameters.
                            2.  Explicit parameters override defaults.

Initial view  . . . . . 1_  1.  SPOC command panel.
                            2.  SPOC status list.
```

Figure 5-3 SPOC preferences specification panel

To access the preferences dialog, select **Options —>1. Preferences**. The available options are:

- Default IMSplex

  Specify the name of the IMSplex that you want to use as the TSO SPOC default. Whenever the TSO SPOC application prompts you to enter an IMSplex name, the default IMSplex name that you enter here will be used unless you specify a different IMSplex name when prompted. You must specify a default IMSplex value the first time you use TSO SPOC application. Leave blank for all systems in the IMSplex.

- Default routing

  You may optionally specify which IMS systems or previously defined IMS groups, within the default IMSplex, will be the default IMS systems to process your commands. That is, all commands you enter will only affect these IMS systems, unless you override the preference by using the plex and route fields on the SPOC command entry panel. Each IMS system or group should be delimited by a space or comma. You can enter as many IMS names or group names that the space provided accommodates.

  Select **Options —> 2. Set IMS groups** to define groups of IMSplex members. Specify all of the IMSplex members that you would like to group under an arbitrary name that you provide. Your groups can also consist of other such groups. You can use the group name in the TSO SPOC routing fields.

- Wait interval

  You can optionally specify a wait interval to set a default time limit for all commands to process before the system times-out the command. That is, all your IMS commands must process within the time you specify here unless you override this value with a wait value on the TSO SPOC command entry panel. It is the OM timeout value. OM will stop waiting and return a 'timed out' status.

  If the IMS system does not process the command within the specified time, the system times-out the command and no response is returned. You know the IMS system is processing the command you entered because of the single point of control — the

Executing panel that appears until the command has been completed or until the wait interval has expired. Once the command process has finished, or has timed-out, you will be returned to the TSO SPOC command entry panel.

The default wait time, if no time is entered, is five minutes. All wait intervals are in the form of MMM:SS (M=minutes; S=seconds). If you only enter a number and no colon, the SPOC will interpret the number as seconds. For example, if you enter "120" as your wait interval value, the TSO SPOC will interpret this as 120 seconds, or two minutes.

- Waiting preference

  Optionally specify whether you want to wait for the command to complete and return a response or not. If you prefer not to wait for command completion, you can review the command status and output by selecting **Display —>3. Command status** from the action bar.

- Command shortcuts

  Shortcuts are abbreviated ways of entering commands. A command which is entered frequently can be abbreviated to just a few characters.

  You can enable or disable whether the TSO SPOC is to process shortcuts. TSO SPOC will not use shortcuts if this selection is not specified.

- Shortcut processing

  You can control whether to replace shortcut parameters or to merge the shortcut parameters. If the short cut is defined as follows:

  – Cmd & resource: QRY TRAN
  – Additional parameters: CLASS(1) SHOW(CLASS,QCNT)

  In the command line, you type the following:

  ```
  QRY TRAN  SHOW(PSB)
  ```

  With the merge option, the submitted command will be:

  ```
  QRY TRAN CLASS(1) SHOW(PSB,CLASS,QCNT)
  ```

  With the override option, the submitted command will be:

  ```
  QRY TRAN CLASS(1) SHOW(PSB)
  ```

- Initial view

  Optionally indicate whether you would like for the TSO SPOC command and response panel that allows you to enter commands and receive the output, or the SPOC status list that displays the execution status of all commands you submitted to appear upon startup. The command and response panel is the default if unspecified on the preference panel.

These values are saved in the ISPF profile dataset and are used the next time you use DFSSPOC.

5.8 Setting up a user written interface to Operations Manager

In addition to the TSO ISPF SPOC application, a REXX SPOC API has been provided. The REXX interface allows REXX programs to submit commands to OM and to retrieve the responses.

5.8.1 REXX SPOC functions

The REXX programming language is frequently used to implement automation software. Programs written in REXX can run under TSO, Netview, or some other environment. The

REXX programs use SCI to communicate with OM, and they may or may not be run on the same OS/390 as OM. Command responses are saved to a REXX stem variable.

5.8.2 REXX SPOC file allocation

In order to use the REXX interface, the executables must be available to the TSO session. This is done either by adding the SDFSRESL library to the TSO JCL in the STEPLIB DD statement or by using the TSOLIB command to add the SDFSRESL library to the TSO search order.

Table 5-2 shows the DD names and the associated files that need to be allocated via ALTLIB functions.

Table 5-2 SPOC REXX interface file allocation associations

Usage	Data set
TSOLIB command	IMS.SDFSRESL
FILE (SYSPROC)	user.written.execs

5.8.3 The REXX SPOC environment

The REXX SPOC components are:

- CSLULXSB program
- IMSSPOC subcommands
- CSLULGTS REXX function

5.8.4 CSLULXSB TSO command program

The REXX host command environment is setup by the CSLULXSB program. The purpose of this program is to:

- Establish the IMSSPOC host command environment.
- Establish the CSLULGTS function.
- Provides REXX variables for return code and reason code processing within the REXX program.

Since this is an OS/390 program, the "Address LINK CSLULXSB" instruction is used to set up the environment. Example 5-9 shows the call to CSLULXSB to establish the REXX SPOC environment.

Example 5-9 CSLULXSB program call

```
Address LINK 'CSLULXSB'
```

5.8.5 IMSSPOC environment

Once CSLULXSB has been successfully processed, the IMSSPOC environment is available to the REXX program. Host commands are typically quoted strings and are passed directly to the host command processor.

Commands IMS, ROUTE, WAIT, CART, and END are supported and perform specific local IMSSPOC functions. Any other passed string is assumed to be a command and is passed to SCI. Example 5-10 shows the use of the IMSSPOC environment.

Example 5-10 IMSSPOC environment call

```
Address IMSSPOC
 "IMS   plex1"    /* set the IMSplex name                  */
 "ROUTE im1a"     /* set explicit route for commands       */
 "CART  mytoken"  /* define the command and response token */
 "WAIT  0:30"     /* set the OM timeout interval           */
 "QRY IMSPLEX SHOW(ALL)"
```

The subcommands processed by the IMSSPOC command are:

- IMS

  Sets the name of the target IMSplex to plex1.

- ROUTE

  Explicitly routes this command to IMSplex member IM1A. This subcommand is optional, and if not specified, the command will be processed for all members of the IMSplex.

- CART

  Defines the command and response token to be used for this command to be mytoken. CART is a 16-byte field containing a command and response token to be associated with the message. The command and response token is a value used to associate a command and its command response.

- WAIT

  Sets the OM time-out interval to 30 seconds to wait for a command response. If the time is reached, OM will return with a 'timed out' return code rather than command response information. This subcommand is optional, and if not specified, the default time-out interval of five minutes will be used.

- END

  Cleans up control blocks.

5.8.6 CSLULGTS() REXX external function

This REXX external function is used by the REXX interface to get the response from OM. The XML tagged response returned by OM is parsed, and each individual line is saved in a REXX stem variable. The name of the stem variable is specified by the user as the first parameter to the CSLULGTS function call.

CSLULGTS has three parameters:

- stem name

  It is a set of variables in REXX. It is an array. The convention is to set the number of entries in the zero-eth member (the items are numbered beginning from zero). For example, if the stem name is 'resp.', resp.0 has the number of entries in the array.

- cartid

  Use the same cartid as was previously used in the CART subcommand.

- wait time

  It is the longest time to wait for the command response. The process is asynchronous, so your program can do something else while the command is executing.

Since there could be more than one active command response, the response for a given command invocation is correlated using the Command and Response Token value (CART) which is specified in both the IMSSPOC, and the CSLULGTS function call to retrieve the

response. Example 5-11 shows the use of the CSLULGTS function to retrieve the command response.

Example 5-11 CSLULGTS call

```
results = cslulgts('resp.','mytoken',"0:30")
```

The parameters passed to the CSLULGTS function are:

- resp.

  The stem variable to receive the command response.
- mytoken

  Name of the command and response token used to correlate responses with commands
- **0:30**

  Defines the CSLULGTS functions time-out value to 30 seconds. Note that the time-out value specified here is how long the REXX program should wait for OM, not how long OM should wait for IMS. This should be a longer time than that specified on the WAIT subcommand.

5.9 REXX SPOC API return and reason codes

Each of the IMSSPOC host commands and the CSLULGTS function set return code and reason code values. The values are provided in REXX variables:

- imsrc
- imsreason

The values of the variables are character representations of hex values. For example, the imsrc value is c'08000008X' when a parameter is not correct. The character 'x' is at the end of the string so REXX will treat it as a character data type.

5.9.1 API return codes

Table 5-3 shows the REXX SPOC API return codes and their meanings.

Table 5-3 REXX SPOC API return codes

Return code	Meaning
"00000000X"	Request completed successfully
"08000004X"	Warning
"08000008X"	Parameter error
"08000010X"	Environment error
"08000014X"	System error

5.9.2 API warning reason codes

Table 5-4 contains the REXX SPOC API warning reason codes and their meanings.

Table 5-4 REXX SPOC API warning reason codes

Reason code	Meaning
"00001000X"	Command still executing

5.9.3 API parameter error reason codes

Table 5-5 contains the REXX SPOC API parameter error reason codes and their meanings.

Table 5-5 REXX SPOC API parameter error reason codes

Reason code	Meaning
"00002000X"	Missing or invalid wait value
"00002008X"	Missing or invalid IMSplex value
"00002012X"	Missing or invalid STEM name
"00002016X"	Missing or invalid token name
"00002020X"	Too many parameters
"00002024X"	Request token not found
"00002028X"	Missing or invalid CART value

5.9.4 API system error reason codes

Table 5-6 contains the REXX SPOC API system error reason codes and their meanings.

Table 5-6 REXX SPOC API system error reason codes

Reason Code	Meaning
"00004000X"	Getmain failure

5.10 Automatic RECON loss notification (ARLN)

The automatic RECON loss notification function allows all instances of DBRC to automatically deallocate the discarded RECON immediately after a RECON reconfiguration occurs.

To support ARLN, DBRC uses SCI. SCI is the only CSL component required. You do not need to start RM or OM if you are using CSL *only* for the function of automatic RECON loss notification. Therefore, SCI must be available on each OS/390 image. OM and RM are only required when an IMS control region is started with the CSLG= parameter.

ARLN is optional. It requires DBRC registration with SCI. SCI is used to communicate between the DBRCs. The communication is used to inform other DBRCs that one has done a reconfiguration of the RECONs.

5.10.1 Using DBRC with SCI

Figure 5-4 is an illustration of the use of SCI by DBRC. Each DBRC instance in an IMSplex shares the same set of RECONs. The IMSplex name is stored in the RECONs. Each DBRC instance registers with SCI. It uses the IMSplex name for the registration. SCI is used for communication between the DBRCs in an IMSplex. SCI uses XCF for communication between different LPARs. This illustration shows the IMS control regions using SCI. This is not required for ARLN.

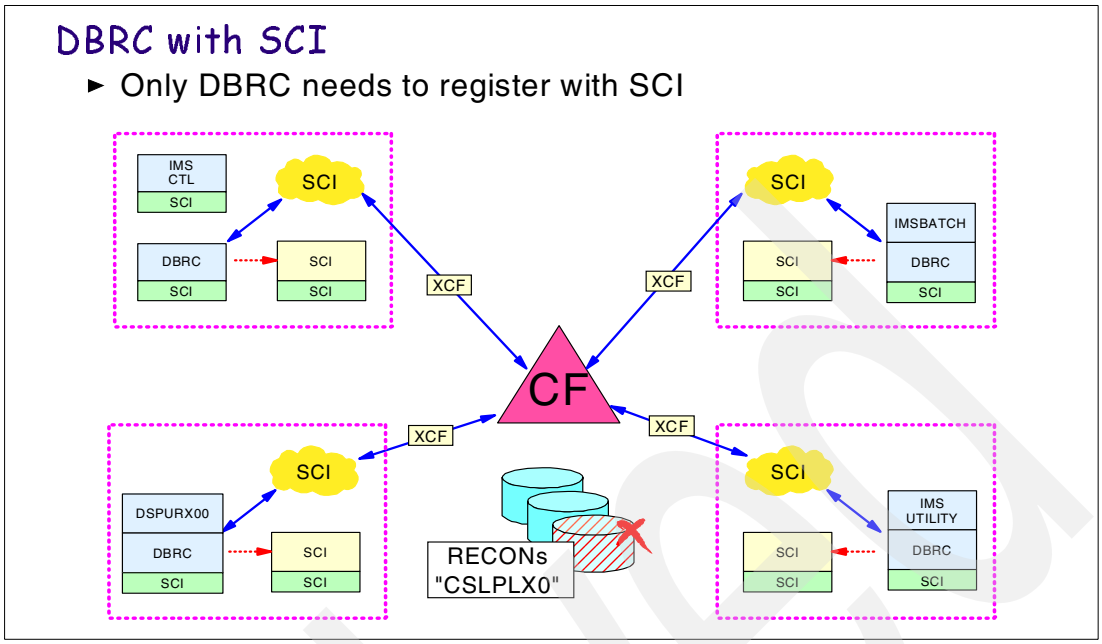

Figure 5-4 DBRC with SCI

The benefit of ARLN is the elimination of the wait for the other DBRCs to do their reconfiguration. This means that a new spare may be created much sooner.

When DBRC registers with SCI, ARLN is available. Registration may be enabled by coding the IMSPLEX name as an execution parameter for DBRC, or by writing an exit (DSPSCIX0) which can be used by all programs and which can invoke registration with SCI, eliminating the need to change all the JCL.

The IMSplex name is stored in the RECON header, so if a DBRC tries to register with the wrong set of RECONs, it will fail.

5.11 Two choices to enable ARLN

There are two ways to indicate the registration of a DBRC instance with SCI:

- By specifying an IMSplex name as an execution parameter within the EXEC statement
- By customizing the new DBRC SCI registration exit (DSPSCIX0)

5.11.1 IMSPLEX execution parameter

The IMSplex name can be set and referred by the execution parameter “IMSPLEX=”. It is the 24th positional parameter for batch regions (PGM=DFSRRC00). Additionally, the IMS Version 8 PROCLIB procedures used for the DBRC address space also provide this new keyword. To use the IMSPLEX parameter in this way requires consistent changes in your JCL for all DBRC instances for a certain set of RECONs.

5.11.2 DBRC SCI registration exit (DSPSCIX0)

The SCI registration exit routine (DSPSCIX0) is called by DBRC before registering with the SCI. DSPSCIX0 supplies the IMSplex name needed for SCI registration. If the exit is not

used, DBRC will behave as if the sample version of the exit was being used. DSPSCIX0 has the ability to set, modify, accept or reject IMSPLEX name for DBRC registration to SCI.

DSPSCIX0 must be found in an authorized library or in LINKLST. If DSPSCIX0 is found in a concatenated STEPLIB or JOBLIB, only the data set containing DSPSCIX0 must be authorized. If DSPSCIX0 is found in LINKLST, no authorization check is performed.

We recommend that you use your own customized user exit rather than the execution parameter if you feel comfortable coding this exit routine. The exit routine must be named DSPSCIX0 and may be used to decide if a DBRC instance should register with SCI and which IMSplex name it should use.

The data set name of one of the RECON data sets (RECON1, RECON2 or RECON3) is passed to the exit routine. If an “IMSPLEX=” execution parameter is provided, this value (one to five characters) is passed to the exit routine as well.

The exit passes back a return code to indicate the intended registration action to take and the IMSplex name to use, if the decision was made to register this DBRC instance.

DSPSCIX0 return codes

The return codes are listed in Table 5-7.

Table 5-7 Return codes of the DSPSCIX0 exit

Return code	Meaning and comment
‘0000’	IMSplex name is used to register with SCI (ARLN invoked)
‘0004’	No SCI registration; RECON access fails if the RECON contains an IMSplex name. Use this to avoid (accidental) registrations or if you do not want to use ARLN. This would also prevent any job step with any IMSplex name specified from setting the IMSplex name in the RECONs, if it is not yet set.
‘0008’	No SCI registration; any IMSplex name found in RECON is ignored and the RECON access is allowed. Use this with caution! Maybe you should provide a special copy of your exit routine for using this option only in emergency cases. Access to this exit routine should be restricted to authorized personnel only!
‘0012’	RECON access denied; SCI registration fails or could not be attempted for severe reasons, for example, because of an exit error. This generally results in an IMS user abend 2480.

5.12 Implementing ARLN

The use of automatic RECON loss notification feature is optional and is automatically enabled if:

- The SCI address space is up and running on the same LPAR.
- The DBRC instance registers with SCI as a member SCI’s IMSplex.

ARLN requires that an IMSplex name be specified and that SCI is active. The IMSplex name may be specified by the DBRC SCI exit routine. Since the exit routine could be used with DBRC instances using different RECONs, the exit routine should be able to handle different RECONs.

The sample exit routine is easily modified to do this. Table entries should be created for each set of RECONs.

- For sets using ARLN, the entries should include the IMSplex name and an indicator that RC=00 should be returned.
- For sets not using ARLN, the entries should include an indicator that RC=04 should be returned. RC=04 allows the RECONs to be used without SCI registration.

Alternatively, the IMSPLEX name may be specified through the IMSPLEX= execution parameter. This requires consistent changes in the JCL for all DBRC instances using a set of RECONs. This is likely to be a difficult process to implement.

The second step in the process is the implementation of SCI. An SCI address space must be started on each LPAR where a DBRC instance requiring ARLN is to be run.

5.12.1 Getting started with ARLN

The first DBRC instance that registers with SCI providing the IMSplex name, saves this name in the RECON and invokes the ARLN. SCI registration is required by DBRC for ARLN. SCI knows the names of all the DBRC address spaces which have registered and can be used by the first DBRC to notify all the others through SCI. A message can be sent to all type DBRC address spaces.

However, the rest of the CSL address spaces are NOT required if this is the only function desired. That is, ARLN can be used without all the other capabilities offered by RM and OM.

Keep in mind, once you have invoked ARLN by specifying this IMSplex name, all following IMS Version 8 DBRC instances using the same RECONs must specify the same IMSplex name, otherwise they will fail, issuing the following message:

```
DSP1136A RECON ACCESS DENIED, IMSPLEX NAME nnnnn NOT VALID
```

We are referring to any interested party accessing the RECON as a DBRC instance. This includes all IMS online and batch regions, most IMS utility programs, and the DBRC utility (DSPURX00), and they all need to specify the same IMSplex name. The RECON loss notification as well as the initialization and termination of any DBRC instance will be propagated to all other running DBRC instances as long as they are registered with SCI.

Note: DBRC can use SCI even if its IMS control region is not using SCI. When both DBRC and IMS control region register with SCI, they must specify the same IMSplex name (they cannot join different IMSplexes).

The SCI itself is assigned to the IMSplex name in the CSLSIxxx PROCLIB member by the IMSPLEX parameter, as follows:

```
IMSPLEX(NAME=PLEX1),
```

For the description of CSLSIxxx member, refer to “SCI initialization PROCLIB member CSLSIxxx” on page 65.

5.12.2 Specifying IMSPLEX parameter in batch and utility regions

For batch and utility regions, including DLI, DBB, ULU, UDR, and RDI, a new positional parameter has been added in IMS Version 8. It is the 24th parameter for batch regions. This parameter has also been included in the IMS supplied procedures that invoke programs using DBRC.

The IMSBATCH procedure is used for BMPs. Normally, BMPs do not invoke DBRC directly. The exception is the use of a BMP region for online image copy utility. This is the only use of the IMSBATCH procedure that could use the IMSPLEX parameter.

The IMSPLEX= parameter has been added to all utilities that invoke DBRC. Some of them are listed on this page. For these utilities, the parameter is not positional. It may be specified anywhere in the PARM parameter. It must be between quotation marks as shown below:

```
PARM='...,IMSPLEX=imsplex name,...'
```

5.13 Changing and removing the IMSplex name in RECON

For some reasons, for example during test period, you may want to change the IMSplex name or to stop using ARLN and remove the IMSplex name from RECON. The IMSplex name is saved in the RECON and it can be changed by using the following command:

```
CHANGE.RECON IMSPLEX(IMSplex name)
```

If you want to stop using ARLN, the IMSplex name can be removed from RECON by using the following command:

```
CHANGE.RECON NOPLEX
```

To use the commands mentioned above, the DBRC instance itself, which is a DBRC command utility execution, you still need to provide the valid old IMSplex name even if you want to go back to using option NOPLEX (otherwise RECON access is denied). But again, be aware that any DBRC instances registering with SCI following your changes need to be re-adjusted for the new IMSplex name or, in case of NOPLEX option, the next one starting with any IMSplex name is the first one which would save the given IMSplex name (by execution parameter or modified user exit) into the RECON.

The CHANGE command cannot be used for the initial setting of the IMSplex name into the RECON. The initial value of the IMSplex name into the RECON will be done by the first DBRC instance providing the IMSplex name either as the execution parameter or using the user exit. The sample exit tries to find a table entry associating the RECON data set name with an IMSplex name.

The CHANGE command is not supported for /RMCHANGE usage, it only may be invoked with DBRC command utility execution. This CHANGE command cannot be used if any DBRC instance using this RECON is active, except for any DBRC instance which was started before ARLN was activated. Any subsequent commands in the DBRC command utility (DSPURX00) job step will fail. We recommend restricting access to this command to authorized personnel.

5.14 DSPSCIX0 details

Since the exit routine could be used with DBRC instances using different RECONs, the exit routine should be able to handle different RECONs, maybe all of your RECON sets. Our sample exit routine can be easily modified to do this. There is a table (in PLEXTABL DSECT) in which you can create table entries for each set of RECONs. Some sample entries will give you an example. For those sets using ARLN, the entries should include the IMSplex name and an indicator that RC=00 will be returned. You should override any passed IMSplex execution parameter as well.

You can code entries, including an indicator for a RC=04, to be returned for other sets of RECONs not intended for use with ARLN. As mentioned in previous items, this allows the RECONs to be accessed without SCI registration of the DBRC instance.

Using SMP/E for your own common version of this exit routine, you can provide a USERMOD to apply to all of your zones and install it into the associated SDFSRESL libraries to be implemented in all of your environments.

5.14.1 The IMS provided default exit

If there is no DBRC SCI exit provided, IMS behaves as if the IBM supplied default exit were used. You will find a sample DSPSCIX0 exit in the ADFSSMPL distribution library. The supplied default exit routine works as described here:

- If IMSPLEX= parameter is specified:
  - It returns the specified IMSplex name with RC=00.
- If IMSPLEX= parameter is not specified:
  - It uses lookup table to match the passed RECON data set name with an entry containing the corresponding IMSplex name (the table is empty as supplied, you need to modify this).
  - It returns IMSplex name from the table entry with RC=00.
- If IMSPLEX= parameter is not specified and no match found in the table:
  - Set RC=04, meaning no SCI registration will be done. No IMSplex name will be saved in the RECON and RECON access fails if the RECON contains an IMSplex name. If the RECONs contain an IMSplex name, RECON access is denied.

Note: The DSPSCIX0 exit must be in an authorized library. If the library is concatenated, only the data set containing the exit needs to be authorized. DBRC performs a specific check on the dataset the module is loaded from, to determine that it is contained in the Authorized Program Facility (APF) list.

The SCI is a prerequisite for automatic RECON loss notification. That means ARLN is only available for IMS Version 8 systems running in an IMSplex environment. All IMS Version 6 and IMS Version 7 systems using DBRC are allowed to share the RECONs even though they are not able to register with SCI and they cannot support ARLN.

5.15 Recommendations for ARLN

A DBRC SCI exit routine should be used to control ARLN usage and the specification of IMSplex names. Without this exit routine, the coordination of IMSplex names across many DBRC instances would most likely be cumbersome.

The use of the CHANGE.RECON IMSPLEX and CHANGE.RECON NOPLEX command should be restricted with the new DBRC command authorization support. This could prevent unintended changes in IMSplex names.

You may want to have a special copy of DSPSCIX0 available to allow RECON access even when IMSplex names do not match for emergency use. A return code of 8 allows this.

- RC=8 allows RECON access even with IMSPLEX name mismatch
- RC=8 does not register with SCI

This copy of the exit routine should be restricted to special use.

5.16 Language environment (LE)

Language Environment (LE) is the replacement product of older language-specific run-time libraries and is now a base element of the z/OS, OS/390, VM/ESA®, and VSE/ESA™ operating systems. See Figure 5-5.

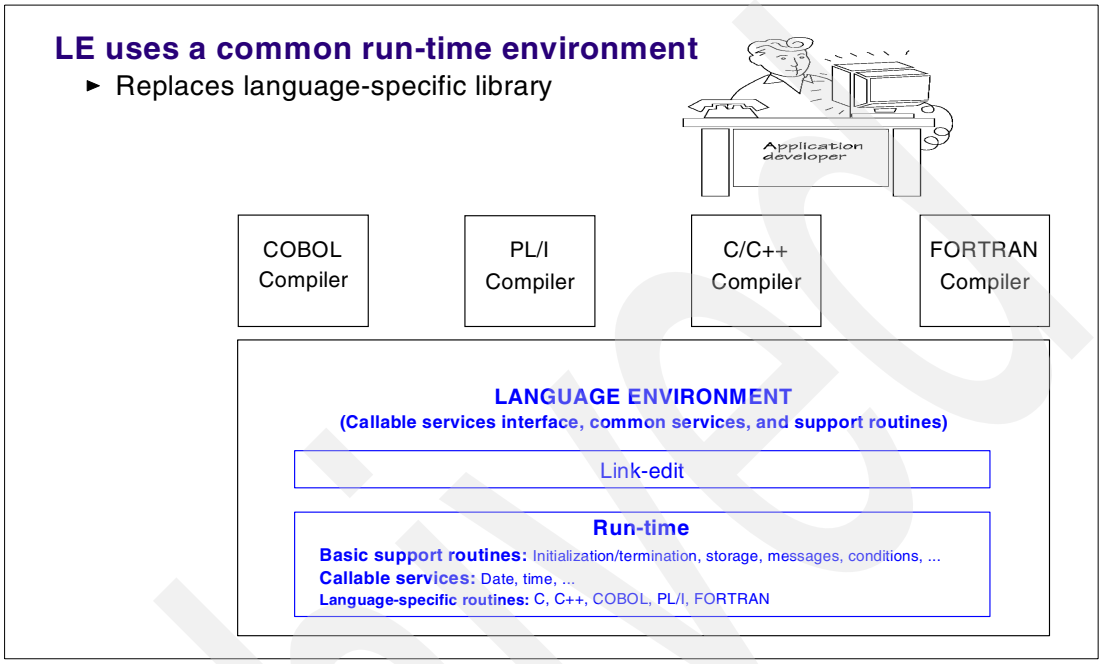

Figure 5-5 LE overview

5.16.1 Defining LE run-time options

LE run-time options are defined in run-time parameter modules that are created and installed in appropriate libraries. LE provides three levels of option specification. See also Figure 5-6.

- CEEDOPT

  Default options module provides installation level defaults. Each run-time option in CEEDOPT must be designated as overrideable or non-overrideable. Options designated as non-overrideable cannot be overwritten in CEEROPT nor CEEUOPT.

- CEEROPT

  Region options module provides defaults that apply to a dependent region. The options specified as overrideable in CEEDOPT may be changed or overwritten in CEEROPT. You can specify one set of options for all programs that run in a dependent region. You can segregate transactions by class so only specific transactions run in a dependent region, using specific options. This requires use of Library Routine Retention (LRR) and DFSINTxx member of IMS.PROCLIB (where xx is a suffix specified by the PREINIT keyword) that includes the name CEELRRIN.

- CEEUOPT

  User options module provides user options at the program level. Run-time options specified as overrideable in CEEDOPT and CEEROPT may be changed or overwritten in CEEUOPT. This module is assembled and linked with the program.

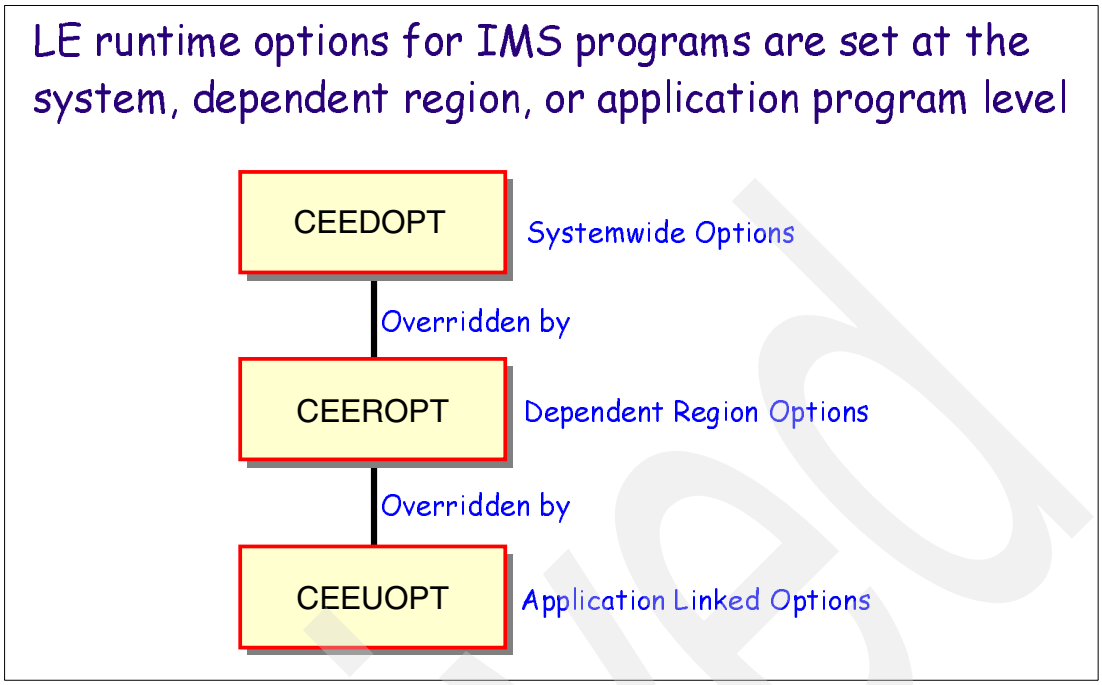

Figure 5-6 LE runtime IMS options

5.16.2 Changing LE run-time options

There are occasions when the LE run-time options may need to be changed. Possibly to collect problem diagnostic information by producing a dump, collecting trace data, or to enable a debug tool. Another reason may be to change storage options for an application. Prior to IMS Version 8, you may need to:

- Recompile and relink the application with the new or changed run time options module.
- Stop and restart the dependent region with the new or changed run time options module.
- Recompile and relink LE modules used to supply the run-time options.

Dynamic run-time option support eliminates the need to change, recompile or reassemble, and relink-edit modules used to supply run-time options for an IMS application, or an IMS dependent region.

5.17 Dynamic LE run-time option support

The Language Environment (LE) enhancement in IMS Version 8 provides a dynamic method of specifying LE run-time options for a transaction, logical terminal (LTERM), user ID, and/or program.

A new IMS startup parameter, LEOPT=, determines whether or not the IMS system allows LE parameter overrides. It can be specified in DFSCGxxx PROCLIB member. UPD LE SET(LEOPT(YES|NO) can also be used to enable or disable the dynamic run-time option support.

This new capability requires the TSO single point of control (SPOC) to be able to enter the command and Operations Manager (OM) to route UPD LE command to IMS. MPP, BMP, IFP and Java dependent regions are supported.

Batch regions cannot take advantage of LE dynamic run-time options, but can still utilize CEEROPT. Dynamic run time option support does not apply to Open Database Access (ODBA) applications.

5.17.1 DFSCGxxx LEOPT parameter

The DFSCGxxx PROCLIB member is used to specify parameters related to the Common Service Layer, Operations Manager, and the Resource Manager which are common to all IMS subsystems that are in the IMSplex. The suffix is specified on the CSLG= parameter.

A new keyword in the DFSCGxxx member, LEOPT=Y | N, indicates whether or not IMS applications running on this system can dynamically override LE run-time parameters.

- LEOPT=Y

  Indicates IMS should allow override parameters, if they exist for the application that is executing. This enables the DL/I INQY call to retrieve the address of the run-time option overrides.

- LEOPT=N

  Indicates IMS should not allow any overrides. The DL/I INQY call will not return the address of the run-time option overrides. N is the default.

Regardless of the LEOPT= specification, the **UPDATE LE** and **DELETE LE** commands perform updates to the LE run time options; however the run-time option overrides (updates) are not used unless LEOPT=Y is also used.

New IMSplex commands allow the customer to specify LE run-time options with 'filters'. A filter is a keyword (such as, TRAN or USERID) on the **UPDATE LE, DELETE LE**, or **QUERY LE** commands. RACF or an equivalent product may be used to secure the new IMSplex commands. Authorization checking is performed on the command verb and keyword level.

An IMS going through initialization retrieves existing LE run-time options from another IMS.

For example to start Debug Tool for a particular transaction code but limit the scope of Debug Tool invocation to a particular individual (user ID). IMSplex commands require use of the Operations Manager address space and the single point of control (SPOC) application (or equivalent automated operator).

5.18 Migration considerations for LE dynamic overrides

The new IMSplex commands (UPDATE LE, DELETE LE, and QUERY LE) are supported through the use of the Operations Manager (OM) address space. Commands are entered from the SPOC. The commands are routed to all IMS systems in the IMSplex. The dynamic parameter overrides supplied by the UPDATE LE and DELETE LE commands are only allowed when dynamic LE overrides have been enabled, which means that LEOPT=Y has been specified in the DFSCGxxx member of IMS.PROCLIB or the overrides are enabled with UPD LE SET(LEOPT(YES)) command.

RACF or an equivalent product may be used to secure the new IMSplex commands. Authorization checking is performed on the command verb and keyword level.

IMS does not validate the resource names/filters used in the commands.

An IMS going through initialization retrieves existing LE run-time options from another IMS.

5.19 DFSBXITA IMS exit

Additionally, a new IMS exit, DFSBXITA, provides an IMS specific version of the assembler exit CEEBXITA optimized for this environment. DFSBXITA is used to provide support for dynamic overrides to LE run-time options for a specific application or the entire IMS environment.

The new IMSplex commands UPDATE LE and DELETE LE are then used to specify the dynamic run-time option overrides. The exit issues the DL/I INQY LERUNOPT call to retrieve the run-time option overrides that have been set by the UPDATE LE and/or DELETE LE commands.

5.19.1 DFSBXITA logic

One of the first things that DFSBXITA is coded to check is whether or not it was invoked from an application running in an IMS environment. The presence (or absence) of an ECP off the TCB first save area determines whether or not the exit was invoked from an IMS environment. See Example 5-12.

Example 5-12 Determining if the exit is invoked in an IMS environment

```
*****************************************************
Check to see if the exit is running in an         *
*  IMS environment.                                 *
*****************************************************
L     R15,16               A(CVT)
L     R15,0(,R15)          A(NEW/OLD TCB)
L     R15,4(,R15)          A(CURRENT TCB)
L     R15,TCBFSA-TCB(,R15) A(1ST TCB SAVE AREA)
L     R8,24(,R15)          POSSIBLE DFSECP ADDRESS
SPACE 1
USING DFSECP,R8            R8: BASE ECP
CLC   ECPID,=C'ECP '       ECPID IN BLOCK?
BNE   RETURN               NO, NOT AN IMS DEP RGN
SPACE 1
****************************************************
```

If CEEBXITA was invoked from an IMS environment, the code added to CEEBXITA (by DFSBXITA) is executed. If CEEBXITA was NOT invoked from an IMS environment, the code added to CEEBXITA (by DFSBXITA) is bypassed.

5.19.2 CEEBXITA as an installation-wide exit

If your installation *is not* currently using the CEEBXITA exit routine as an installation-wide exit routine and you opt to do so in an IMS Version 8 environment, you need to do the following to provide dynamic run-time option support for the IMS environment:

- Add the DFSBXITA source code to CEEBXITA.
- Assemble and link-edit CEEBXITA.
  - When CEEBXITA is invoked from an IMS environment, the logic added to CEEBXITA by DFSBXITA is executed prior to returning control to LE.
  - When CEEBXITA is invoked from non-IMS environments, the logic added to CEEBXITA by DFSBXITA is bypassed. In customizing the logic in CEEBXITA, you will need to determine where to branch to in CEEBXITA prior to returning control to LE.

If your installation *is* currently using CEEBXITA as an installation-wide exit routine, you will need to decide where (in CEEBXITA) to add the logic from DFSBXITA to provide the dynamic run-time option support for the IMS environment. To enable the support you will need to:

- Determine where the DFSBXITA logic should be placed in CEEBXITA.
- Add the DFSBXITA source code to CEEBXITA.
- Assemble and link-edit CEEBXITA.
  - When CEEBXITA is invoked from an IMS environment, the logic added to CEEBXITA by DFSBXITA is executed prior to returning control to LE.
  - When CEEBXITA is invoked from non-IMS environments, the logic added to CEEBXITA by DFSBXITA is bypassed. In customizing the logic in CEEBXITA, you will need to determine where to branch to in CEEBXITA prior to returning control to LE.

Figure 5-7 shows these installation options.

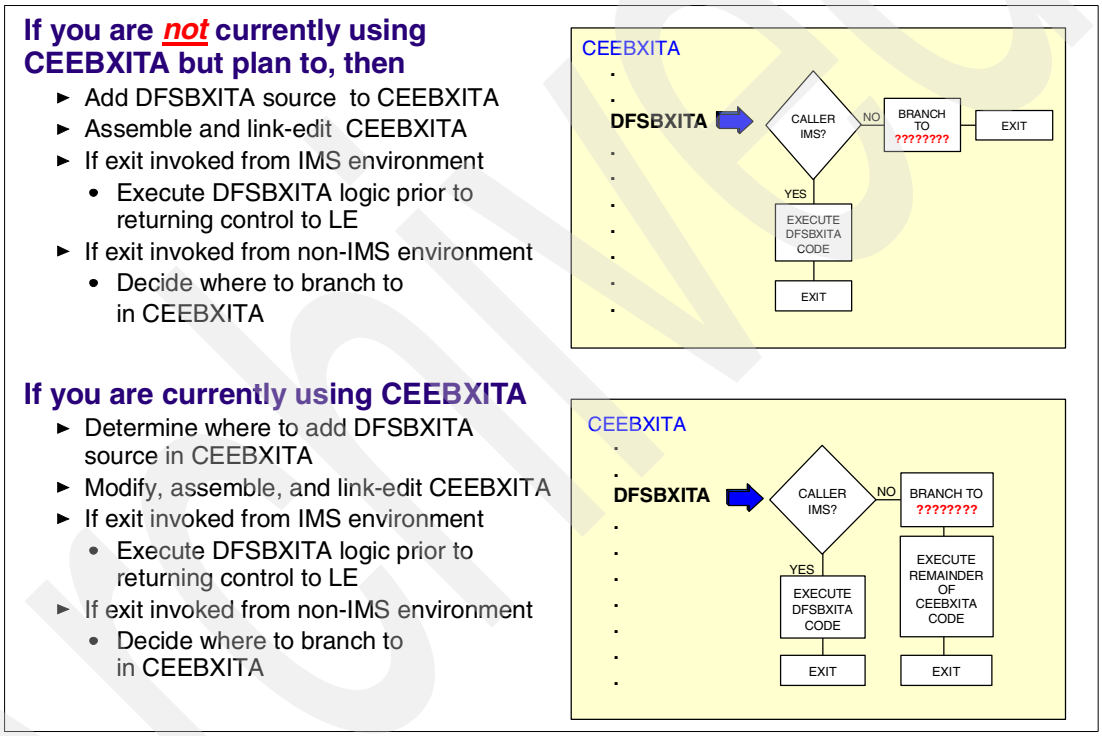

Figure 5-7 CEEBXITA installation

5.19.3 LE option recommendations for IMS

Table 5-8 provides a list of LE run-time options for IMS environments and shows the defaults and recommended values.

Table 5-8 LE options, defaults and recommended settings

Option	Default	Recommended
ABPERC	None	None
ABTERMENC	ABEND	ABEND
AIXBLD	NOAIXBLD	NOAIXBLD
ALL31	ON	ON

Option	Default	Recommended
ANYHEAP	16K,8K,ANY,FREE	16K, 8K, ANY FREE (C, FORTRAN, Multi, PL/I) 48K,8K,ANY,FREE (Fortran)
ARGPARSE	ARGPARSE	ARGPARSE
AUTOTASK	NOAUTOTASK	NOAUTOTASK
BELOWHEAP	8K,4K,FREE	8K,4K,FREE
CBLOPTS	ON	ON
CBLPSHPOP	ON	N/A
CBLQDA	OFF	OFF
CHECK	ON	ON
COUNTRY	US	User defined
DEBUG	DEBUG(OFF)	DEBUG(OFF)
DEPTHCONDLMT	10	0
ENV	No default	User defined
ENVAR	No default	User defined
ERRCOUNT	0	0
ERRUNIT	6	6
EXECOPS	EXECOPS	EXECOPS
FILEHIST	FILEHIST	FILEHIST
FILETAG	NOAUTOCVT,NOAUTOTAG	NOAUTOCVT,NOAUTOTAG
FLOW	NOFLOW	NOFLOW
HEAP	32K, 32K, ANY, KEEP, 8K, 4K	32K, 32K, ANY, KEEP, 8K, 4K (C, COBOL, Multi, PL/I) 4K, 4K, ANY, KEEP, 8K, 4K (FORTRAN)
HEAPCHK	OFF, 1, 0, 0	OFF, 1, 0, 0
HEAPPOOLS	OFF, 8, 10, 32, 10, 128, 10, 256, 10, 1024, 10, 2048, 10	User defined
INFOMSGFILTER	OFF	OFF
INQPCOPN	INQPCOPN	INQPCOPN
INTERRUPT	OFF	OFF
LIBRARY	SYSCEE	SYSCEE
LIBSTACK	4K,4K,FREE	4K,4K,FREE
MSGFILE	SYSOUT, FBA, 121, 0, NOENQ	DD name
MSGQ	15	15
NATLANG	ENU	ENU

Option	Default	Recommended
NONIPTSTACK	Replaced by THREADSTACK	
OCSTATUS	OCSTATUS	OCSTATUS
PC	NOPC	NOPC
PLIST	HOST	HOST
PLISTASKCOUNT	20	20
POSIX	OFF	OFF
PROFILE	OFF,' '	OFF,' '
PRTUNIT	6	6
PUNUNIT	7	7
RDRUNIT	5	5
RECPAD	RECPAD(OFF)	RECPAD(OFF)
REDIR	REDIR	REDIR
RPTOPTS	OFF	OFF
RPTSTG	OFF	OFF
RTEREUS	RTEREUS(OFF)	RTEREUS(OFF)
RTLS	OFF	OFF
SIMVRD	SIMVRD(OFF)	SIMVRD(OFF)
STACK	128K, 128K, ANY, KEEP, 512K, 128K	128K, 128K, ANY, KEEP, 512K, 128K (C, FORTRAN, Multi, PL/I) 64K, 64K, ANY, KEEP (COBOL)
STORAGE	NONE,NONE,NONE,0K	NONE,NONE,NONE,0K
TERMTHDACT	TRACE, , 96	TRACE, , 96 (C, FORTRAN, Multi, PL/I) UATRACE, , 96 (COBOL)
TEST	NOTEST (ALL, *, PROMPT, INSPREF)	NOTEST (ALL, *, PROMPT, INSPREF)
THREADHEAP	4K, 4K, ANY, KEEP	4K, 4K, ANY, KEEP
TRACE	OFF, 4K, DUMP, LE=0	OFF, 4K, DUMP, LE=0
TRAP	ON,SPIE	ON,SPIE
UPSI	00000000	00000000
USRHDLR	NOUSRHDLR	NOUSRHDLR
VCTRSAVE	OFF	OFF
VERSION	' '	' '
XPLINK	OFF	OFF
XUFLOW	AUTO	AUTO

6

Implementing VTAM Generic Resources and RNR

This chapter describes the tasks necessary to implement both VTAM Generic Resources and Rapid Network Reconnect (RNR) facilities.

6.1 VTAM Generic Resource implementation

VTAM Generic Resources can be used by IMS, once this feature has been enabled within VTAM. In this chapter we briefly discuss what needs to be in place in VTAM for VTAM Generic Resources to be enabled.

6.2 VTAM components of VTAM Generic Resources

For VTAM Generic Resources to be enabled in VTAM, the following should be in place:

- The VTAM start option parameter STRGR settings
- Coupling facility structure definitions
- VTAM Generic Resources resolution exit

Generic resources requires that IMS is executing on a node that is defined either as an APPN network node or APPN end node.

6.2.1 VTAM start option parameter

This name identifies the Coupling Facility structure to VTAM and can be specified in VTAM's STRGR start option parameter. The name of this structure defaults to ISTGENERIC.

6.2.2 Coupling Facility structure

This structure and its attributes need to be defined in the active Coupling Facility Resource Management (CFRM) policy for the sysplex.

A VTAM Generic Resource structure must be defined. Generic resources are supported by VTAM when a generic resources structure is defined in the CFRM policy. The default name for this structure is ISTGENERIC. Users of VTAM 4.3 or later releases may specify a different name. This is done in the STRGR VTAM start option parameter. The name must begin with `IST`. One structure is used by VTAM to support all generic resource groups. All of the VTAMs sharing generic resource information must specify the same structure name.

Information about the generic resource group and individual terminals logged on through the group, is stored in this structure. In this structure we find the generic resource group name and the logon APPLID of each member application (with a session count). An additional table contains a list of affinities between session ID's (node names) and linked APPLIDs. A bit flag, in the structure, indicates whether IMS or VTAM has control over affinity management.

6.2.3 VTAM Generic Resources resolution exit

Installations may code a VTAM Generic Resource Resolution Exit (ISTEXCGR). This exit has the capability to override the node selected by VTAM Generic Resources for a session establishment request.

This exit needs to be in a load library accessible by VTAM and, if allowed to follow its default processing algorithm, allows Workload Manager (WLM) to determine session destination.

6.3 IMS definitions for VTAM Generic Resources

The components of VTAM Generic Resources within IMS consist of three parameters and two user exits. The exit usage is dependent on whether significant statuses (for terminals) need to be reset.

- IMS start-up parameter
- Affinity management parameter
- IMS ESTAE processing parameter
- Logoff/signoff user exits

Note that the logoff/signoff exits are not specific to VTAM Generic Resources. They are standard IMS exits that can be used outside of a VTAM Generic Resources environment.

6.3.1 IMS start-up parameter

IMS needs to be connected to a generic resource group (GRG). The first step in implementing VTAM Generic Resources is to create an IMS GRG. The GRG defines to VTAM all the IMSs available to participate in a session balancing group. The GRG has a generic resource name (GRSNAME) and this is how the group of IMS subsystems will be known to both VTAM and the end users.

In the IMS procedure, a new parameter, GRSNAME, specifies which GRG this IMS will be a member of. Each IMS subsystem still retains its own unique VTAM APPLID, specified on the IMS COMM macro. Once a GRSNAME is specified at startup, each IMS subsystem joins the group and becomes eligible to participate in session balancing.

6.3.2 Affinity management parameter

Either IMS or VTAM can perform affinity management. The GRAFFIN parameter specifies whether IMS or VTAM should manage affinities in case of session failure.

6.3.3 IMS ESTAE processing parameter

If GRAFFIN=IMS was chosen (or defaulted to), then this parameter (in member DFSDCxxx) determines how IMS would handle affinities during abend processing. Y indicates that IMS should continue the existing ESTAE logic to delete affinity for all nodes where no status remains before closing the ACF/VTAM ACB.

N indicates that IMS should close the ACF/VTAM ACB immediately to expedite IMS termination and leave affinity for all nodes set.

6.3.4 Logoff and signoff user exits

These exits allow the removal of a significant status from either a static or an ETO terminal.

6.4 Tasks related to generic resource implementation

Before activating VTAM Generic Resources for IMS, generic resources should be enabled within VTAM. Your VTAM system programmer needs to ensure that VTAM knows which Coupling Facility structure to access, and that APPN is the network protocol being used. Certain tasks need to be performed, and sizing calculations need to be made. The following tasks are the responsibility of the VTAM systems programmer:

- Ensure that the XCF functions exist in the sysplex.
- Ensure that the APPN protocol is available.
- Determine the Coupling Facility structure (default name ISTGENERIC) attributes:
  - Calculate the size of the Coupling Facility structure:

    Roughly 300 bytes per session is required in the structure. The requirements for IMS need to be added to those of any other users of VTAM Generic Resources.
  - Define the generic resource structure (ISTGENERIC) in the active CFRM policy and ensure that it can be rebuilt.
- Ensure that the parameter STRGR is specified with a structure name (or allowed to default to ISTGENERIC) in the VTAM start options.
- Change, assemble, link-edit, and test the VTAM Generic Resources resource resolution exit.

6.5 Implementing VTAM Generic Resources for IMS

Here is a list of items to consideration when implementing VTAM Generic Resources for IMS use.

The generic resource group name must be specified to the IMS systems. IMS uses VTAM Generic Resources when the GRSNAME IMS execution parameter is specified. All IMS systems in a generic resource group must specify the same name.

Recommendations:

- Use the DFSPBxxx member to specify the GRSNAME. When using the /START VGRS command, an incorrectly typed entry would require IMS to be shut down and restarted.
- If PQ18590 is applied, the GRAFFIN parameter may be specified. This determines whether IMS or VTAM will manage generic resource affinities. GRAFFIN=IMS is the default. LU 6.1 (ISC) affinities are always managed by IMS.

- OS/390 VTAM preparation must be done.

  Ensure that the tasks discussed in 6.4, “Tasks related to generic resource implementation” on page 133, have been implemented.
- Update the DFSPBxxx member in the IMS PROCLIB.

  Specify DC=xxx to tell IMS what the 3-character suffix for the DFSDCxxx member would be. Add the GRSNAME=aaaaaaaa parameter (if you decided to have the GRSNAME as a hard coded parameter) to specify the resource group name.
- Update DFSDCxxx member in the IMS PROCLIB.

  If shared queues has not been implemented yet, this member may not yet exist, and should be created first. Add the GRAFFIN and GRESTAE parameters to indicate to IMS your preference for affinity management and ESTAE processing.
- Add or modify the logoff and/or signoff user exits.

  These exits should be included if the significant status of a static or an ETO terminal needs to be reset at session termination time.
- Start IMS.

A warm start of IMS is sufficient to activate the VTAM Generic Resources related parameters within IMS. An IMS start command, to specify the GRSNAME, may need to be done if the GRSNAME parameter was not added to the DFSPBxxx IMS PROCLIB member.

Recommendation: Cold start IMS (after a normal shutdown was done) to ensure cleanup of all significant statuses.

6.6 How to implement Rapid Network Reconnect

Here are the steps required to ensure that RNR is set up.

- Update IMS PROCLIB member DFSPBxxx.

  Specify DC=xxx to tell IMS which DFSDCxxx member to select.
- Update IMS PROCLIB member DFSDCxxx.

Two new IMS parameters are provided: RNR and PSTIMER. They are both specified in the DFSDCxxx member of PROCLIB. RNR must be defined for RNR support to be activated. Valid values for RNR include NRNR (Default) and ARNR. PSTIMER values are in seconds, with 0-84600 being allowed. 3600 is the default.

- RNR=NRNR

  NRNR means no reconnect. The terminal session will be terminated during IMS restart. The sessions can be re-established using standard procedures.
- SNPS + RNR=ARNR

  ARNR is automatic session reconnect. If IMS abends, and the RNR=ARNR option is set, VTAM closes the ACB on IMS, but retains the LU-LU sessions. It saves the allocated resources and control blocks in a data space and effectively shields the network of the IMS failure.

  VTAM stores the incoming data, so the network views IMS as active, but not currently responding. When IMS restarts, and the /START DC command has processed, VTAM automatically reconnects the session to a point equivalent to a logon without user sign-on data being made available. This will result in a sign-on message (DFS3649) being required, as appropriate for the terminal type and definition.

  How long VTAM waits is dependent on the PSTIMER option. If the PSTIMER is defined as 0, then VTAM sessions will remain pending recovery for an indefinite period.
- MNPS + RNR=ARNR

  With VTAM and IMS set up in MNPS mode, the session data is stored in the MNPS structure in the Coupling Facility. When an IMS fails, it can be restarted on another system in the sysplex, and the sessions can be connected automatically to it again without user sign-on data being made available.

6.6.1 Rapid Network Reconnect special notes

RNR is not supported in an XRF environment. A warning message will be issued and IMS operation will continue with RNR disabled. RNR does not apply to APPC. The support for APPC persistent sessions is provided by APPC OS/390. The PSTIMER value for these sessions will be defined in the APPCPMxx member of SYS1.PARMLIB.

Part 2

IMSplex operations

Part 2 outlines the operational considerations of an IMSplex.

7

Operations in a data sharing environment

IMS sysplex data sharing requires the use of the Coupling Facility and places more emphasis on DBRC, the RECONs, and IRLM as integral components. In this chapter we introduce you to new commands, features, and messages. We also revisit some existing commands and features, and highlight areas of particular importance.

7.1 Support procedures

Operating and supporting an IMS Parallel Sysplex data sharing environment will require changes to a number of procedures currently in place and new procedures to manage the new facilities.

The following procedures are used in support of the IMS environment:

- Change control

  If the IMS systems are cloned, changes to one system should also be reflected in the others, except for split applications or databases.

- IMS maintenance

  A process for applying maintenance to one or more of multiple, cloned IMS systems must be established.

- IMS system definition

  If the IMS systems are cloned, changes to one system should also be reflected in the other(s).

  – MFS format generation
  – DBD, PSB, and ACB generation
  – Security generation
  – Online change

  The *staging* phase of online change must be reviewed and updated to reflect changes made to accommodate multiple cloned IMS systems.

- Database reorganization

  Database reorganization procedures will require the database to be made unavailable to all IMS systems.

- Application maintenance

  Application code changes, and updates to application program libraries, must be coordinated across multiple IMS systems.

- Coupling Facility structure size changes

  A procedure should be developed for changing Coupling Facility structure sizes.

We examine a few of the above operational processes that must be modified to function in a data sharing environment.

7.2 Online change procedures

Online change for each IMS online system is executed independently. For pre IMS Version 8 cloned systems, procedures will have to be developed to coordinate online changes on all of the systems. IMS Version 8 introduces coordinated global online change.

If the systems share the data set (ACBLIB, MATRIX, MODBLKS, or FORMAT) which is being switched, the coordination is simpler. In this case, the library is copied from the staging library to the inactive library by the online change utility (DFSUOCU0). The MODSTAT data set of any of the sharing subsystems can be used. Of course, they all must indicate the same DDNAME for the active library. The /MODIFY PREPARE and /MODIFY COMMIT commands are issued independently for the online systems.

If each system has a unique version of the data set (ACBLIB, MATRIX, MODBLKS, or FORMAT) which is being switched, the online change utility must be run for each system. In this case, it is advisable for each invocation of the utility to refer to the MODSTAT data set associated with the system for which the copy is being done. The /MODIFY PREPARE and /MODIFY COMMIT commands are issued independently for the online systems.

The following procedure can be used to perform online change when each system has a unique version of the data sets related to online change:

- Use an automated operator application to stop resources you want to change or delete, and also issue the /MODIFY PREPARE or /MODIFY ABORT commands.
- Define the active, inactive, and staging libraries.
- Share the active and inactive libraries.
- Define each IMS with its own MODSTAT data set, using the same active and inactive ACBLIB, FMTLIB, MATRIX, and MODBLKS definitions.
- Start each IMS subsystem.
- To perform the online changes:
  - Perform system generations, using the staging libraries for the changes.
  - Run the online change utility.
  - Issue a /MODIFY PREPARE command. If the command fails on any IMS subsystem in the sysplex, the operator or automated operator application program must issue the /MODIFY ABORT command on all IMS subsystems for which /MODIFY PREPARE was successful.
  - Resolve any work in progress.
  - Issue the /MODIFY COMMIT command.
  - Start resources that were changed or added.

In either case, it is important for the change to be made in each of the cloned systems or for all to revert to their original library. For example, all should be using ACBLIBA or all should be using ACBLIBB when the procedures are completed.

Enqueues are used to protect libraries used with online change. The online systems always have shared enqueues on the active libraries. The online change utility holds exclusive enqueues on both the staging and target libraries. For example, if online system IMS1 is using ACBLIBA as its active library, it will hold a shared enqueue on ACBLIBA. If the online change utility is run using IMS1's MODSTAT, it will select ACBLIBB as its target. The utility will request exclusive enqueues on the staging library, ACBLIB, and on the target library, ACBLIBB. Obviously, the online change utility cannot acquire the exclusive enqueue on ACBLIBB if another online system is using it for its active library. The enqueues would conflict.

7.2.1 Overview of IMS Version 8 global online change

The processing of global online change is similar to that of local online change. The resource preparation and staging is the same as in the past with local online change. For IMS Version 8, modifications have been made to coordinate the change across multiple IMS systems.

The online change status (OLCSTAT) data set is shared by all of the IMS systems. Its function is similar to that of the MODSTAT data set with local online change. OLCSTAT contains the DDNAMEs of the active libraries. An active library is either an “A” or “B” library.

With global online change, typically all of the IMS systems share the same libraries. They each use the same DDNAME. For example, if one IMS is using its IMSACBA DD for ACBLIB, all other IMSs will also using this DD for ACBLIB.

Online change is started by invoking the INIT OLC PHASE(PREPARE) command. This is similar to the /MODIFY PREPARE command used with local online change. It also coordinates the prepare processing across all of the IMSs.

After the prepare has completed, the actual changes are invoked with the INIT OLC PHASE(COMMIT) command. This is similar to the /MODIFY COMMIT command used with local online change.

An online change may be aborted by issuing a TERMINATE OLC command. It coordinates the online change abort phase across all the IMSs in the IMSplex. TERMINATE OLC is similar to the /MODIFY ABORT command used with local online change.

A QUERY MEMBER command may be used to show online change status of IMSs. It reports the current status of each IMS participating in the online change.

Refer to 5.1, “Global online change” on page 100 for the implementation, and 12.1, “Global online change” on page 252 for the operational information, and *IMS Version 8 Implementation Guide*, SG24-6594, for detailed information on this feature.

7.3 Job and region scheduling procedures

With multiple systems available, job scheduling procedures and JOB classes should be reviewed. An installation might want to restrict BMPs to a particular system for performance or capacity reasons. This can be done with JOB classes. If an installation wants to run BMPs on multiple IMS systems, it must ensure that the *IMSID* parameters for the BMPs are correct.

7.3.1 Submitting BMPs on any IMS in the Parallel Sysplex

The IMS Version 6 APAR PQ21039 simplifies the specification for users whose BMPs can execute under multiple control regions in the IMSplex. With this enhancement, no changes to BMP JCL are required even when different control regions are used. Each control region in an IMSplex may specify the same value for IMS group in the DFSPBxxx. The new parameter is IMSGROUP and the value specified for it can be used in the dependent region JCL as the value for IMSID parameter.

The following tables illustrate a use of the IMSGROUP parameter.

Table 7-1 Use of IMSID before cloning

Control region IMSID	Dependent region IMSID
IMSID=IMSP	IMSID=IMSP

Table 7-2 Use of IMSID and IMSGROUP with cloning

Control region IMSID and IMSGROUP	Dependent region IMSID
IMSID=IMP1, IMSGROUP=IMSP	IMSID=IMSP
IMSID=IMP2,IMSGROUP=IMSP	IMSID=IMSP

It is important to note that dependent region JCL does not have to change. Only the control region execution parameter is changed to exploit the function supplied by IMSGROUP.

This function applies to MPP and IFP regions as well as BMPs.

7.3.2 /START REGION command enhancements

Originally introduced in IMS Version 6, enhancements to the /START REGION command allow multiple IMS subsystems to share the same member in IMSRDRs IEFRDER library for MSG, BMP, and IFP regions of the same type. The IMSID is passed to the dependent region as part of the /START REGION processing. Also, there is an option to pass the JOBNAME, which is useful when multiple IMSs execute on the same OS/390.

The following formats of the command are supported:

```
/START REGION membername LOCAL
/START REGION membername JOBNAME jobname LOCAL
```

In addition, membername is still optional, with the default as IMSMSG.

- If LOCAL is specified; the IMSID found from the SSCDIMID field replaces IMSID on the executed procedure even if the IMSID is specified in the JCL.
- If JOBNAME jobname is specified:
  - It becomes the jobname of the started region identified in membername.
  - The parameter LOCAL is the default if the JOBNAME parameter is specified.
- If neither the LOCAL nor the JOBNAME parameter is specified, then the /START REGION command is processed as in releases prior to IMS Version 6.

Figure 7-1 illustrates how the /START REGION membername JOBNAME jobname command functions.

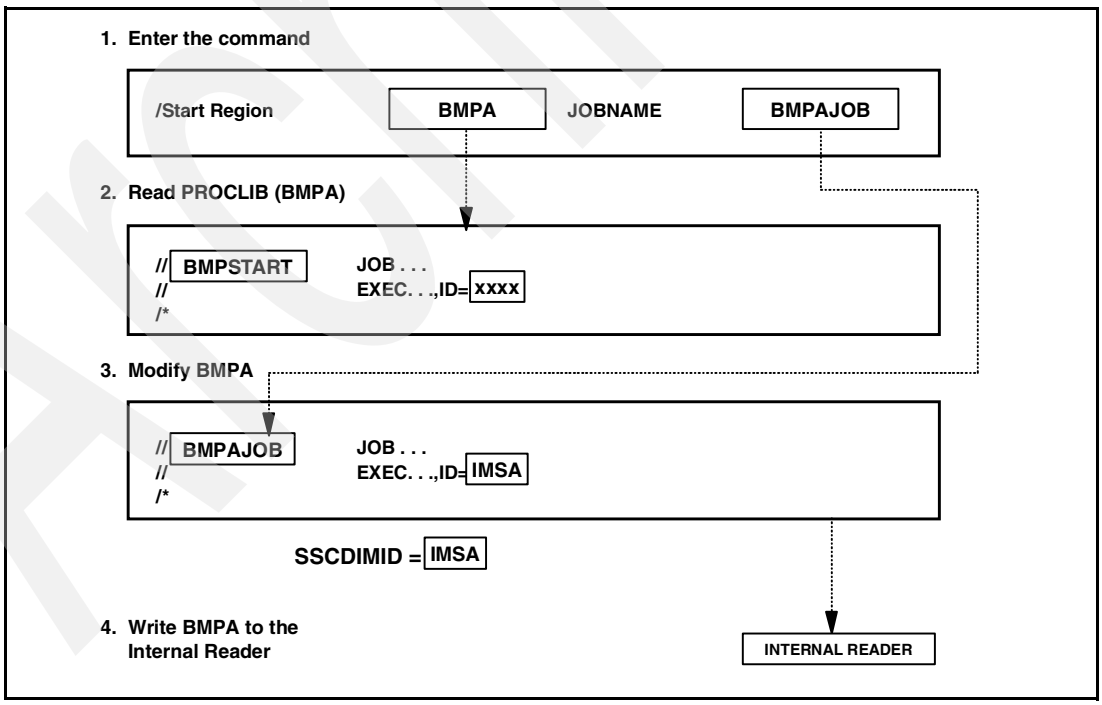

Figure 7-1 /START REGION command processing

7.4 Command recognition character support for TM and DCCTL

Command recognition character (CRC) support has been enhanced over the years to improve IMS Parallel Sysplex operation:

- As of IMS Version 6, CMDCHAR in the IMSCTRL stage one macro and the CRC= parameters in IMS startup procedure allow CRC support. For example, if the CRC= value is #, the following display command can be entered from an OS/390 console:

  ```
  #DISPLAY ACTIVE
  ```

- Use of IMSID in commands on OS/390 console

  For example, to direct the command to the IMS system with subsystem name = IM1A from the OS/390 console, the following command can be used:

  ```
  IM1ADISPLAY DATABASE ALL
  ```

- Use of the same CRC by multiple IMSs in the same OS/390 image

  With IMS Version 6, both IMS TM and DBCTL can enter commands by using the same CRC. When multiple subsystems on the same OS/390 image have the same CRC, any command entered using this CRC will be processed by all those subsystems.

- Support for IMS TM and DBCTL commands from MCS and E-MCS consoles

  IMS Version 6 allows both IMS TM and DBCTL to enter commands by using a CRC or the subsystem name. Responses to commands entered from MCS and E-MCS consoles are returned to the originating console. This includes both synchronous and asynchronous responses.

These facilities allow commands for multiple IMS systems on different OS/390 images in a Parallel Sysplex to be entered from one console. The OS/390 *ROUTE* command can be used to send IMS commands to any OS/390 system in a Parallel Sysplex.

Figure 7-2 shows different variations how CRC can be used in data sharing environment.

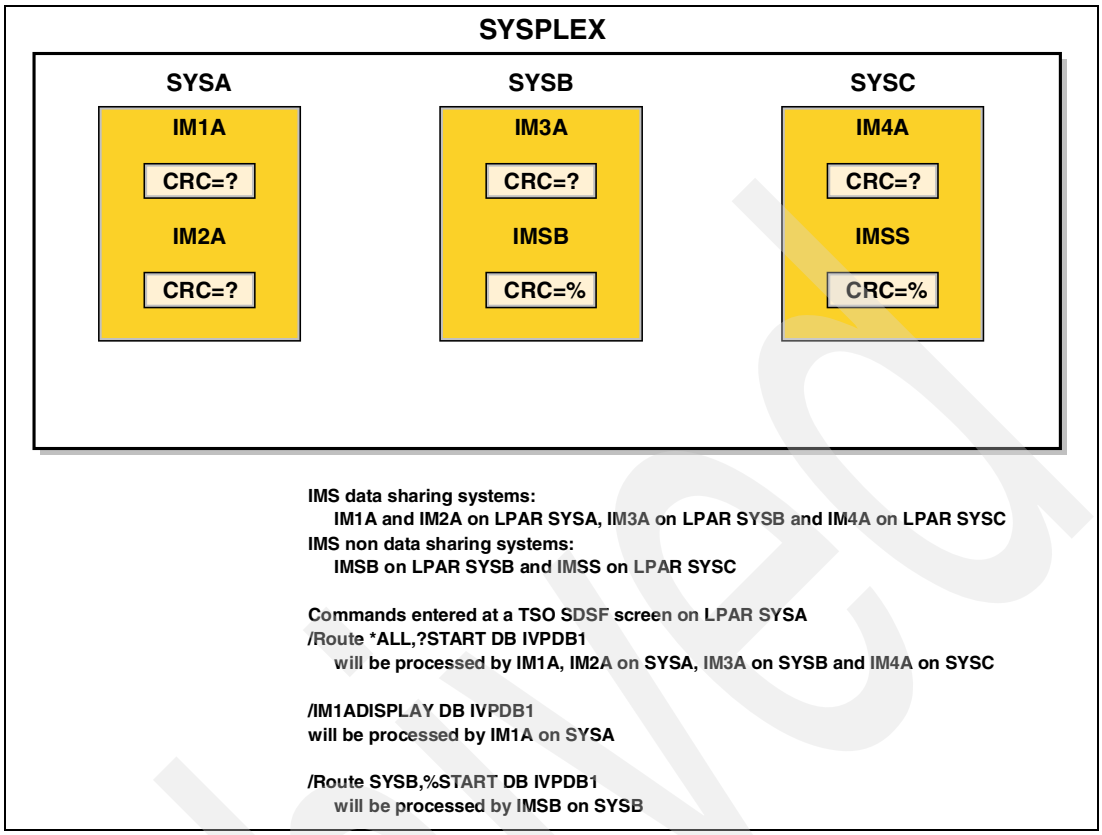

Figure 7-2 Sysplex IMS command processing

7.5 IMS global scope commands

The GLOBAL parameter can be used on the following IMS commands to indicate that the command applies to all subsystems sharing a database:

```
/START DATABASE ddddddd GLOBAL
/START AREA aaaaaaaa GLOBAL
/DBRECOVERY DATABASE ddddddd GLOBAL
/DBRECOVERY AREA aaaaaaaa GLOBAL
/DBDUMP DATABASE ddddddd GLOBAL
/DBDUMP AREA aaaaaaaa GLOBAL
/STOP DATABASE ddddddd GLOBAL
/STOP AREA aaaaaaaa GLOBAL
/STOP ADS
```

7.5.1 Alternative to global commands

Many installations might find the use of the *ROUTE *ALL* command from an OS/390 console to be more convenient than the use of GLOBAL commands. It has the following advantages.

- All responses from all IMS systems are sent to the originating console.
- All commands are supported. The GLOBAL parameter is supported only on /START, /STOP, /DBR, and /DBD commands for databases and AREAs. It is not supported for other commands, such as /DISPLAY DB.
- DBRC flags are not affected. The GLOBAL parameter also requires the NOPFA parameter to avoid setting or resetting the *prevent further authorization* or *read only* flags in DBRC.

7.5.2 IMSplex command support

Commands may be entered to IMS either through traditional channels, such as the MTO, the WTOR reply, automated operator programs using the CMD or ICMD call, an E-MCS console, and now in IMS Version 8, from a single point of control (SPOC). There are several new commands available when running in the IMS Version 8 Common Service Layer (CSL) environment. These are generally referred to as *IMSplex commands* or *enhanced commands* and can only be entered through the IMS Version 8 Operations Manager (OM) interface. Those IMS commands which traditionally have started with the slash (/), are referred to as *classic commands*.

Refer to 10.1, “CSL operations” on page 222 for more information on this new IMSplex command support that sysplex environments can utilize.

7.6 Commands associated with Coupling Facility operation

In this paragraph we describe OS/390 commands that you can use to display and alter the status of various Coupling Facility objectives.

7.6.1 Display commands associated with Coupling facility and structures

The following OS/390 commands are useful in monitoring structure status:

D XCF,STR

The XCF display command, DISPLAY XCF is used to show how the Coupling Facility and the structure resources are used in the sysplex. You can obtain summary information for each CF or detailed information by individual structure within the CF. Example 7-1 shows an example of output from the DISPLAY XCF STRUCTURE command. Only those IMS related structures in the sysplex are shown in this example.

Example 7-1 Output of D XCF,STR command

```
/D XCF,STR

IXC359I  18.59.14  DISPLAY XCF 557
STRNAME            ALLOCATION TIME   STATUS

IMOA_AREADI01A   11/06/2002 18:20:37 ALLOCATED
IMOA_AREADI01B   11/06/2002 18:20:38 ALLOCATED
IMOA_AREAIT01A   11/13/2002 18:11:51 ALLOCATED
IMOA_AREAIT01B   11/13/2002 18:11:52 ALLOCATED
IMOA_AREAWH01A   11/06/2002 18:20:42 ALLOCATED
IMOA_AREAWH01B   11/06/2002 18:20:43 ALLOCATED
IMOA_EMHO             --       --    NOT ALLOCATED
IMOA_EMHP        10/19/2002 07:56:01 ALLOCATED
IMOA_IRLM        08/23/2002 18:28:31 ALLOCATED
IMOA_LOGE        11/06/2002 18:19:08 ALLOCATED
IMOA_LOGM        11/06/2002 18:19:07 ALLOCATED
IMOA_MSGO        08/23/2002 18:27:23 ALLOCATED
IMOA_MSGP        08/23/2002 18:27:06 ALLOCATED
IMOA_OSAM        11/06/2002 18:20:25 ALLOCATED
IMOA_RSRC        08/23/2002 18:27:00 ALLOCATED
IMOA_VSAM        11/06/2002 18:20:26 ALLOCATED
```

D XCF,STR,STRNM=ssssssss

By specifying the STRNM=structure_name in D XCF command, you get detailed information for that specific structure. Example 7-2 shows structure policy, allocation and connection information for the IM0A_IRLM IRLM lock structure. The asterisk (*) is allowed as a wildcard for the structure name, so the command D XCF,STR,STRNM=IM0A_* would display information for all the structures whose name begins with IM0A_.

Example 7-2 Output of D XCF,STR,STRNM=IM0A_IRLM

```
/D XCF,STR,STRNM=IM0A_IRLM

IXC360I  19.28.14  DISPLAY XCF 730
STRNAME: IM0A_IRLM
 STATUS: ALLOCATED
  POLICY SIZE    : 32768 K
  POLICY INITSIZE: N/A
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : DISABLED
  PREFERENCE LIST: CF05     CF06
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY

 ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 08/23/2002 18:28:31
  CFNAME         : CF05
  COUPLING FACILITY: 009672.IBM.02.000000050822
                     PARTITION: F   CPCID: 00
  ACTUAL SIZE    : 32768 K
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: B81F9AE1 377BE880
  LOGICAL  VERSION: B81F9AE1 377BE880
  SYSTEM-MANAGED PROCESS LEVEL: 8
  XCF GRPNAME    : IXCL0083
  DISPOSITION    : KEEP
  ACCESS TIME    : 0
  MAX CONNECTIONS: 7
  # CONNECTIONS  : 3

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 IM0AIRLM$IR0A001 02 00020012 SC53     IM1AIRLM 009A ACTIVE
 IM0AIRLM$IR0A002 03 0003000C SC47     IM2AIRLM 0097 ACTIVE
 IM0AIRLM$IR0A003 01 00010022 SC54     IM3AIRLM 0054 ACTIVE
```

D XCF,POL

Use the DISPLAY POLICY command to display the name of the current policy (CFRM093) loaded from the CFRM data sets into the CF, as shown in Example 7-3.

Example 7-3 Output of a D XCF,POL command

```
D XCF,POL

IXC364I  19.46.58  DISPLAY XCF 829
TYPE: ARM
POLICY NOT STARTED
TYPE: CFRM
```

```
     POLNAME:       CFRM093
     STARTED:       11/04/2002 15:43:41
     LAST UPDATED:  11/04/2002 15:43:21
TYPE: SFM
     POLNAME:       SFM02
     STARTED:       11/08/2000 14:42:23
     LAST UPDATED:  01/02/1996 09:59:58
```

D XCF,CF

D XCF,CF command displays the type and module number information regarding the Coupling Facilities within this sysplex.

Example 7-4 Output of D XCF,CF command

```
D XCF,CF

IXC361I  19.51.00  DISPLAY XCF 851
  CFNAME     COUPLING FACILITY
  CF05       009672.IBM.02.000000050822
             PARTITION: F   CPCID: 00
  CF06       002064.IBM.02.000000010ECB
             PARTITION: F   CPCID: 00
```

D CF

D CF command displays a great deal of information regarding all the CFs in the syplex. We only listed the compete display for CF06, although CF05 information was created as well, from this command.

Example 7-5 Output of the D CF command

```
D CF

IXL150I  19.53.44  DISPLAY CF 872
COUPLING FACILITY 002064.IBM.02.000000010ECB
                  PARTITION: F  CPCID: 00
                  CONTROL UNIT ID: FFFB
NAMED CF06
COUPLING FACILITY SPACE UTILIZATION
 ALLOCATED SPACE                 DUMP SPACE UTILIZATION
  STRUCTURES:     328960 K        STRUCTURE DUMP TABLES:        0 K
  DUMP SPACE:       2048 K                  TABLE COUNT:        0
 FREE SPACE:      654336 K       FREE DUMP SPACE:            2048 K
TOTAL SPACE:      985344 K      TOTAL DUMP SPACE:            2048 K
                                MAX REQUESTED DUMP SPACE:        0 K
   VOLATILE:         YES         STORAGE INCREMENT SIZE:      256 K
    CFLEVEL:          12
    CFCC RELEASE 12.00, SERVICE LEVEL 04.15
    BUILT ON 09/06/2002 AT 16:03:00

COUPLING FACILITY SPACE CONFIGURATION
                          IN USE          FREE         TOTAL
CONTROL SPACE:          331008 K      654336 K      985344 K
NON-CONTROL SPACE:           0 K           0 K           0 K

SENDER PATH       PHYSICAL            LOGICAL       CHANNEL TYPE
        FA         ONLINE              ONLINE        ICP
        FB         ONLINE              ONLINE        ICP
COUPLING FACILITY DEVICE     SUBCHANNEL     STATUS
                   FFEA         211B        OPERATIONAL/IN USE
```

```
                   FFEB         211C        OPERATIONAL/IN USE
                   ...........
COUPLING FACILITY 009672.IBM.02.000000050822
                  PARTITION: F  CPCID: 00
                  CONTROL UNIT ID: FFFE
NAMED CF05
COUPLING FACILITY SPACE UTILIZATION
................
```

D CF,CFNM=cccccc

With CFNM=coupling_facility_name specified, the D CF command displays the same information as the D CF command but for specific CFs.

7.6.2 SETXCF commands

The following OS/390 commands control the status of structures in CFs:

- SETXCF START,policy,type=cfrm,polname=pppppppp

  Starts a CFRM policy.
- SETXCF FORCE,con,strnm=ssssssss,conname=all

  Breaks any failed persistent connections to structure.
- SETXCF force,str,strnm=ssssssss

  Deallocates the structure.
- SETXCF start,rebuild,strnm=ssssssss,location=normal

  Rebuilds the structure into the first Coupling Facility in the preference list.
- SETXCF start,rebuild,strnm=ssssssss,location=other

  Rebuilds the structure into the other Coupling Facility specified in the preference list.

The Coupling Facility can be either a dedicated processor (S390 9674 CMOS unit) or an LPAR in any of the sysplex enabled S390 processors. The commands can be entered from any master support console in the sysplex.

Use of the SETXCF Command

Coupling Facility structures are central to data sharing in the sysplex. In most cases, the IRLMs or the IMSs in the sysplex automatically rebuild the structures. There are circumstances, however, when the rebuilding requires operator intervention.

The SETXCF command is used to start rebuilding the Coupling Facility structures either in the same, or another CF. Example 7-6 shows the command and response for rebuilding a Coupling Facility structure.

Example 7-6 Output of the rebuild of a OSAM structure command

```
SETXCF START,REBUILD,STRNAME=IMOA_OSAM,LOCATION=OTHER

IXC521I REBUILD FOR STRUCTURE IMOA_OSAM
HAS BEEN STARTED
IXC367I THE SETXCF START REBUILD REQUEST FOR STRUCTURE
IMOA_OSAM WAS ACCEPTED.
IXC526I STRUCTURE IMOA_OSAM IS REBUILDING FROM
COUPLING FACILITY CF06 TO COUPLING FACILITY CF05.
REBUILD START REASON: OPERATOR INITIATED
```

```
INFO108: 00000046 00000000.
IXC521I REBUILD FOR STRUCTURE IMOA_OSAM
HAS BEEN COMPLETED
```

The rebuild processing only took a few seconds in our test environment.

Specific commands used to deallocate structures

If there are no connections to an IRLM structure, it may be deallocated with the following command:

```
SET FORCE,STRUCTURE,STRNAME=IMOA_IRLM
```

In order to break failed persistent connections to a structure, you can issue the following command:

```
SETXCF FORCE,CON,STRNM=IMOA_IRLM,CONNAME=ALL
```

7.6.3 Manipulating the Coupling Facility links

The CONFIG command is used to change the online or offline status of channel paths connected to the CF.

Example 7-7 shows the command which disconnects channel path 00 from the CF. Note that if the command causes the last link to the CF to be removed, it requires confirmation from the operator.

Example 7-7 Output of the D XCF,CF and the CONFIG CHP(00), OFFLINE,FORCE commands

```
D XCF,CF

IXC361I  20.36.07  DISPLAY XCF 152
  CFNAME     COUPLING FACILITY
  CF05       009672.IBM.02.000000050822
             PARTITION: F   CPCID: 00
  CF06       002064.IBM.02.000000010ECB
             PARTITION: F   CPCID: 00
CONFIG CHP(00),OFFLINE,FORCE
IXL126I CONFIG WILL FORCE OFFLINE LAST CHP(00) TO Coupling Facility
 CF05
 65 IXL127A REPLY CANCEL OR CONTINUE
 R 65,CONTINUE
 IEE600I REPLY TO 65 IS;CONTINUE
IEE503I CHP(0),OFFLINE
IEE712I CONFIG   PROCESSING COMPLETE
```

If you now do a status display for the CF, you can see that the channel paths are physically not operational.

7.7 Commands used by IRLM in support of data sharing

You communicate with IRLM by using the standard OS/390 modify command with various options. Remember that DB2 also uses an IRLM, so any command and displayed information applies equally to both IMS and DB2. In the data sharing environment, DB2 cannot share an IRLM with IMS, but multiple IMSs on the same OS/390 image can use the same IRLM.

The OS/390 commands used for an IRLM operation are documented in the *IMS Version 8 Command Reference*, SC27-1291.

7.7.1 MODIFY IRLM STATUS command options

The MODIFY IRLM STATUS command without other keywords displays status, work units in progress, and detailed lock information for each subsystem identified to the IRLM, as shown in Example 7-8.

Example 7-8 Output of F IM1AIRLM,STATUS command

```
F IM1AIRLM,STATUS

DXR101I IROA001 STATUS SCOPE=NoDISC
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS       UNITS      HELD    WAITING    RET_LKS
        IM1A     UP               2         9          0          0
    DXR101I End of display
```

The ALLD option shows all the subsystems connected to all the IRLMs in the data sharing group that the IRLM belongs to. The RET_LKS field is very important. It shows how many database records are retained by a failing IRLM and are therefore unavailable to any other IMS subsystem. A sample is shown in Example 7-9.

Example 7-9 Output of the F IM1AIRLM,STATUS,ALLD command

```
F IM1AIRLM,STATUS,ALLD

DXR102I IROA001 STATUS
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS    RET_LKS   IRLMID  IRLM_NAME   IRLM_LEVL
        IM1A     UP              0    001      IROA          009
        IM2A     UP              0    002      IROA          009
        IM3A     UP              0    003      IROA          009
    DXR102I End of display
```

The ALLI option shows the names and status of all IRLMs in the data sharing group that the IRLM belongs to. An example is shown in Example 7-10.

Example 7-10 Output of the F IM1AIRLM,STATUS,ALLI command

```
F IM1AIRLM,STATUS,ALLI

DXR103I IROA001 STATUS
        IRLMS PARTICIPATING IN DATA SHARING GROUP FUNCTION LEVEL=021
        IRLM_NAME IRLMID STATUS LEVEL SERVICE MIN_LEVEL MIN_SERVICE
          IROA     002     UP    021  PQ48823    012      PN90337
          IROA     003     UP    022  PQ52360    012      PN90337
          IROA     001     UP    022  PQ52360    012      PN90337
    DXR103I End of display
```

The STOR option shows the current and maximum specified allocation for CSA and ECSA storage. Example 7-11 show the output of this command.

Example 7-11 Output of the F IM1AIRLM,STATUS,SOTR command

```
F IM1AIRLM,STATUS,STOR

DXR100I IROA001 STOR STATS
PC: NO   MAXCSA:     6M  LTE:     8M RLE:   96795  RLEUSE:       1
CSA USE: ACNT:   132K  AHWM:   132K  CUR:  2648K  HWM:  2648K
        ABOVE 16M:    46   2643K     BELOW 16M:     3      5K
CLASS   TYPE  SEGS     MEM   TYPE  SEGS     MEM   TYPE  SEGS     MEM
```

```
ACCNT   T-1     1     64K     T-2     1     64K    T-3     1      4K
PROC    WRK     5     30K     SRB     1      1K    OTH     1      1K
MISC    VAR    37   2453K     N-V     6     22K    FIX     1     24K
    DXR100I End of display
```

Options MAINT and TRACE can be useful in diagnosing problems. MAINT provides the information of the maintenance levels for IRLM modules and TRACE provides information about IRLM subcomponent trace types.

7.7.2 Other MODIFY command options

Being an OS/390 command, MODIFY has many options that do not apply to IRLM. We list only those options that apply to IMS data sharing:

- F IM1AIRLM,ABEND

  Causes IM1AIRLM to terminate abnormally. IRLM informs all subsystems linked to it, through their status exits, that it is about to terminate.

- F IM1A,RECONNECT

  Causes IM1A to reconnect to the IRLM specified in the IRLMNM parameter in the IMS control region JCL. This is necessary after an IRLM is restarted following an abnormal termination when IM1A is not taken down.

- F IM1AIRLM,PURGE,IM1A

  Causes IM1AIRLM to release any retained locks it holds for IM1A. This command must be used with utmost care in these situations:

  – The RECON reflects that database backout was done, but IRLM was not up at the time of the backout.

  – A conscious decision is made not to recover, or to defer recovery, but the data is required to be available to other IMSs.

  The PURGEALL option of this command is even more hazardous. It allows you to release all retained locks of all IMSs held by a specific IRLM.

Messages and codes

All IRLM related console messages are prefixed by DXR and they all contain the name of the issuing IRLM system. Where the identity of a connected IRLM is required, the IRLM name is constructed from the IRLMNM value concatenated to the IRLMID value that is specified in that IRLMs startup procedure. IRLM terminates itself under certain circumstances and issues user abend codes when this happens. The IRLM user abend codes start in the 2000 code range.

7.7.3 IRLM Version 2 Release1 tracing

The command, F irlmproc,START trace is no longer valid for IRLM Version 2 Release 1. You must use the OS/390 TRACE CT command to start, stop or modify the IRLM diagnostic trace in the sysplex data sharing environment. This command can be entered only from the master console. The command requires an appropriate level of OS/390 authority, as described in *OS/390 V1R3.0 OS/390 System Commands*, GC28-1781.

The format of the command is:

```
TRACE CT, ON|OFF, COMP=irlmssnm, SUB=(yyy)
```

Where, 'irlmssnm' is the IRLM subsystem name.

The SUB= parameter identifies a specific component IRLM communicates with, which is to have the interface flow traced. For example, IRLM interactions with the systems lock manager, the cross Coupling Facility, and the database management subsystems can be traced. If the SUB= parameter is not used, the TRACE command operates on all subtraces. OS/390 issues a WTOR message and requires a response to indicate an end to the TRACE command, because a variety of options may be passed to CTRACE. IRLM does not use these options because it controls its own tracing activity internally. Exception tracing is activated at IRLM initialization and remains active throughout the IRLM execution.

IRLM Version 2.1 can use the OS/390 external writer facility that enables you to store the trace buffers on data sets. This support is provided through an additional TRACE CT command issued before the trace is switched on and after it is stopped. The WTRSTART and WTRSTOP keywords of the TRACE CT command are used for this purpose.

Example 7-12 presents an example sequence of commands to start and stop an IRLM INT trace for the IRLM with subsystem name IRLA. The trace data is written to an external writer data set identified in procedure CTWTER:

Example 7-12 Starting and stopping the IRLM V2.1 CT trace

```
TRACE CT,WTRSTART=CTWTER
TRACE CT,ON,COMP=IRLA,SUB=(INT)

(OS/390 asks for a reply.)

R xx,WTR=CTWTER,END
TRACE CT,OFF,COMP=IRLA,SUB=(INT)

(Wait to ensure that the trace buffers are externalized)
```

Because this command uses OS/390 component trace services, make sure that the IRLM start and stop load module, DXRRL183, is in the OS/390 link list.

IRLM V2.1 APAR PN90337 includes a serviceability enhancement that uses a new parameter TRACE in the IRLM startup procedure. This parameter can be used to turn on IRLM wrap-around traces at IRLM startup. These are the same traces that can be turned on or off by the TRACE CT command. This allows your installation to establish the default state of these traces and removes the requirement that the operator remember to turn them on each time an instance of IRLM is started.

Formatting the IRLM trace

Both the IRLM trace buffers and the external writer data sets can be formatted using IPCS. The IRLM trace formatting load module DXRRLFTB, and the buffer-find routine load module DXRRL186 must be available to IPCS.

Sample IRLM trace entries

The sample IRLM Version 2.1 trace in Figure 7-3 shows the activities IRLM performs to join a data sharing group. IRLM must become a fully updated and active member before the first IDENTIFY is complete so it can immediately do global locking.

```
IRLM NAME= IRLA ANALYSIS BEGINS: TRACE TYPE = INT

CTRC 032A8060 DXRRL1H1-01: START IDENT REQ
                    IMS first identifies to IRLM. IMS suspended until identify completes
CTRC 032A8162 DXRRL710-01: QUERY STRUCTURE
                    IRLM queries XES(OS/390) to get the lock table size
CTRC 032A81AC DXRRL710-02: QUERY RETURN
                    OS/390 returns from the QUERY request
CTRC 032A820E DXRRL710-02: JOIN GROUP
                    IRLM joins the XCF group (name for group is in IRLM start proc)
CTRC 032A8240 DXRRL710-03: JOIN RETURN
                    OS/390 returns from the JOIN request
CTRC 032A828A DXRRL710-04: CONNECT TO GROUP
                    IRLM connects to the lock structure via the SLM
CTRC 032A82BC DXRRL752-01: EXIST CONNECTION
                    EVENT EXIT driven for another IRLM already a group member
CTRC 032A83F6 DXRRL751-14: EVENT EXIT RETURN
                    IRLM returns to the SLM from the EVENT exit
CTRC 032A8530 DXRRL752-02: DUMMY EVENT
                    Dummy event tells IRLM about all other active and failed persistent
                    connections. Receive an EXISTING EVENT for all existing
                    connections when IRLM connects to the lock structure
CTRC 032A866A DXRRL751-14: EVENT EXIT RETURN
                    IRLM returns to the SLM from the EVENT exit
CTRC 032A8A18 DXRRL710-05: CONNECT RETURN
                    OS/390 returns from the CONNECT request
CTRC 032A8A62 DXRRL2R1-01: IXLRT READBYCONN
                    IRLM reads all potential retained locks for the existing event in the
                    event exit as part of connecting to the lock structure
CTRC 032A8AF6 DXRRL210 IDENT REQUEST COMPLETE
                    IRLM finally completes the IDENTIFY request and resumes IMS
```

Figure 7-3 Trace of IRLM joining the data sharing group

IRLM and XRF

The process of an XRF takeover is simplified with IRLM Version 2.1. The messages that IRLM issues during and after a takeover have changed. No IRLM-specific operator interventions are required for an XRF takeover.

7.8 DBRC and IMS data sharing operations

The RECON data sets contain a wealth of information about the various IMS subsystems in the data sharing group and about the databases being shared. All this information is available to you through DBRC list commands. You can also trigger a variety of jobs using DBRC commands, if you have set up your environment accordingly.

Table 7-3 presents a summary of some of the DBRC commands associated with data sharing environments.

Table 7-3 List of useful DBRC commands in data sharing environment

Command	Explanation
/RMCHANGE	Changes or modifies information in the RECON
/RMDELTE	Deletes information from the RECON
/RMINIT	Creates records in the RECON
/RMLIST	Lists information from the RECON
/RMNOTIFY	Adds information to the RECON
/RMGENJCL	Generates JCL for change accumulation, log archive, log recovery, database recovery, database image copy, and predefined user jobs.

These commands are documented in the *IMS Version 8 DBRC Guide and Reference*, SC27-1295.

Note that the output from a DBRC command is limited to 4 KB, except for the LIST command, which has a limit of 8 KB. The output goes to a terminal buffer, from where you can page through it using the PA1 key. Any output that exceeds these limits is truncated. You can circumvent these restrictions by issuing the corresponding commands in batch through the DBRC utility (DSPURX00).

7.9 Automated operations

The cloning of an IMS system into multiple IMS systems will probably have an impact on an installation's existing automation. The degree of impact will depend upon the nature of the automation in place. If the automation function is *local* in nature (does not deal with shared or common resources), only minor changes would be required. However, if the automation function is *global* in nature (deals with shared or common resources), major changes could be required.

A common example of local automation would be to restart a transaction and program following an application program abend (message DFS554). Since an application program abend is an event that is localized to a particular IMS system, this type of automation should require little, if any, change.

A common example of a global automation would be to deallocate (/DBR) a set of databases from all of the IMS systems. In this case, the automation would need to be changed, since the objective is to /DBR the databases from all of the IMS systems within the IMSplex.

7.10 Shared VSO operation

Shared DEDB VSO AREAs in the CFs present differences in their operational control as compared to normal Fast Path DEDBs residing completely on DASD.

7.10.1 Connecting to the SVSO structures

Connection is deferred until AREA open when IMS interrogates DBRC for VSO options and structure names. When private buffer pools are required, the current private buffer pools are searched for a suitable pool for this area:

- For an AREA specific pool defined in DFSVSMxx
- For a database specific pool defined in DFSVSMxx
- For a pool defined with the correct CI size and LKASID option

If no suitable private pool definition is available, a default pool is created.

The structure is built in the CF if this is the first OPEN for this area, and the AREA is connected to the structure(s).

OPEN will fail if:

- There is not a suitable CRFM policy with enough space allocation for the structure.
- If PRELOAD is set and the structure is too small.
- If NOPREL is specified with two structures with different sizes.

If an OPEN fails and AREA use is necessary, change the AREA to NOVSO in DBRC and /START the area. Data access will be from DASD. The CFRM policy can be modified, then started, and finally a /START AREA can be issued.

7.10.2 Preloading the data into the structure

PRELOAD processing then occurs, one CI at a time because an IMS may have caused some earlier loading of data.

- For each CI, a conditional lock is obtained and the CI is read from DASD into a private buffer.
- A read from the CF facility is attempted and, if successful, the CI is already loaded and does not need to be written by this IMS.
- If the CI read fails, the CI is written to the CF and the lock is released. If the lock is not obtained, it is because some application already owns the CI and will write it to the CF so preload processing will skip that CI.

7.10.3 Reading from and writing to the CF structure

The data requests are supported differently depending on the LKASID option:

- The CI lock is always obtained first.
- NOLKASID

  For NOLKASID pools, a buffer on the available queue is obtained and moved to the requestor queue.

  The CI is read from the CF or read from DASD and written to the CF if not currently there. The CI interest is registered with the CF when the CI data is obtained and the local cache vector is updated. If a buffer is not available, the application waits for a buffer to become free.
- LKASID

  For LKASID pools, the look-aside queue is searched for the CI. If found but invalid, the buffer is moved to the requestor queue.

If the look-aside buffer is not valid, the buffer is placed on the requestor queue and reused. The original CI is de-registered as interest is lost. If there is no available buffer, the oldest buffer on the look-aside queue is selected and moved to the requestor queue. If there is no buffer on the look-aside queue, the task waits for a buffer to become available before moving it to the requestor queue.

- The data is read from the CF, or from DASD if it is not in the CF, and passed to the Fast Path action modules. The valid buffer is registered and the local cache vector is updated.

Writing the updated CIs to the CF occurs after synchronization processing in the conventional way:

- Updates are logically logged in X'5950' records, with a VSO flag, and committed by the X'5937' record.
- When physical logging of the X'5937' occurs, an othread is scheduled to handle the updated CIs.
- The CIs are written to the CF and the data item entry in the structure is marked changed and the CI locks are released.
- A X'5612' log record is written when all CIs have been written to the CF.
- The buffer is returned to the available queue or the look-aside queue as appropriate. Any buffer in a sharing IMS which has a copy of an updated buffer is invalidated as part of the CF write operation.

7.10.4 Cast-out to DASD

Cast-out processing is performed by each sharing IMS at its system checkpoint and at AREA close:

- A X'5910' log record is written at the start of castout processing.
- IMS issues a READ_DIRNFO to request a list of changed CIs in the CF; this is kept in the structure directory.
- A CASTOUT_DATA request is issued for each changed CI in the list. This causes a cast-out lock to be obtained on behalf of the IMSs to prevent multiple IMSs from trying to cast out the same CI. The CI is still available for read and update.
- The CIs are read from the CF into DREF buffer space in approximately 100 KB chunks.
- The directory entries are marked unchanged, the CIs are written to DASD, and an UNLOCK_CASTOUT request is issued to release the lock on the CIs.
- This is repeated until all changed CIs have been cast out. A X'5912' log record is written at the end of castout processing.

7.10.5 New and changed VSO commands

There are changes to the following commands in support of SVSO.

- /START AREA aaaaaaaa
- /DISPLAY POOL fpdb
- /DISPLAY FPVIRTUAL
- /VUNLOAD AREA aaaaaaaa

START AREA command

The /START command (/START AREA areaname) reinstates the VSO option globally, but does not start stopped areas in other systems. VSO load is serialized through the exclusive AREA lock. Updates can continue during the load process.

Changing the access intent of a database with the /START DB ACCESS= command causes any VSO AREAs of the database to be removed from virtual storage. Because the access intent of a DEDB cannot be changed while any of its AREAs are authorized to an IMS subsystem, IMS closes any open AREAs before processing the command. If a VSO AREA is closed as a result of the command, it is also removed from the data space.

The /START AREA command must be used to reactivate the VSO options, PREOPEN, and PRELOAD options for the area. If an area is opened as a result of an access request rather than by the /START AREA command, it is opened as a non-VSO area. Non-VSO AREAs with the PREOPEN option are closed as a result of the /START DB ACCESS=command. These AREAs are reopened either at the next access request for the area or by the /START AREA command. The GLOBAL and ACCESS keywords are mutually exclusive. The /START command is rejected if both keywords are specified.

DISPLAY POOL FPDB command

Example 7-13 shows a typical /DISPLAY POOL FPDB (display Fast Path database pools) command. In this example; for POOL0210, the buffers allocated for the primary pool are 500 (specified in the VSPEC member) and the buffers allocated for the secondary pool are 100.

The HITS column is the percentage of times (60% in this case) a buffer was found in the LKASID pool since the last system checkpoint. The VALID column is the percentage of hits that were valid. Therefore a buffer was found in the pool 60% of the time, and of that 60%, 75% of the buffers found had valid data, that is, 45% of the HITS had valid data.

Example 7-13 Output of a /DISPLAY POOL FPDB command

```
FPDB BUFFER POOL:
  AVAIL =    50  WRITING =     0  PGMUSE =     0  UNFIXED =   550
POOLNAME CISIZE PBUF  SBUF   MAX CURRENT LK HITS VALID
POOL0210 04096--00500-00100-00500-00200--Y---060%-075%-----------
DISTPOOL 01024  00100 00025 00300 00100  Y   000%  NA
WAREPOOL 00512  00020 00005 00030 00020  Y   000%  NA
 *2002317/184426*
```

/DISPLAY FPVIRTUAL command

/DISPLAY FPVIRTUAL displays the AREAs loaded into an OS/390 data space or CF. This is those AREAs that have been specified with the Virtual Storage Option (VSO) are displayed with this command.

/VUNLOAD AREA command

The /VUNLOAD command (/VUNLOAD AREA areaname) removes an AREA from a data space (VSO SHARELVL=0 or 1) or from a CF structure (VSO SHARELVL=2 or 3). The command is persistent across IMS restarts and can be reset only by a /START AREA command. With the CF, the command is executed on all sharing subsystems and is synchronized across subsystems by taking a lock on CI 0 to prevent updates during the unload process.

7.10.6 System managed rebuild and automatic alter support for SVSO

Via APAR PQ50661 shared VSO has been enhanced to support system-managed rebuild and automatic alter functions that OS/390 offers. These functions are only supported on processors and CFs that support these facilities.

System-managed rebuild

This allows users to migrate a VSO structure from one CF to another without taking the VSO structure offline for planned reconfigurations. It does not support rebuild for loss of connectivity, structure of CF failure cases. This support enhances CF system management by providing a means of using a single operator command to migrate all structures (VSO and non-VSO) from a CF to another.

With system-managed rebuild, a user does not need to take a VSO structure offline via a /VUNLOAD command before CF maintenance can be applied. The user simply migrates all structures in a CF via a single operator command.

OS/390 Version 2 Release 8 and CF Level 9 are the minimum software and hardware requirements for this feature.

The rebuild process is as follows:

1. Define two or more CF in the preference list, PREFLIST( ), for a VSO structure.
2. Initiate a system-managed rebuild for the target VSO structure via either one of the following commands:

```
SETXCF START,REBUILD,STRNAME=strname
SETXCF START,REBUILD,CFNAME=cfname
```

If using the STRNAME=strname parameter, then only the structure identified by the strname is moved from one CF to another. If using the CFNAME=cfname parameter, then all the structures from that CF are moved to another CF.

Automatic alter

This allows OS/390 to dynamically expand or contract a structure based on its actual CF storage usage. OS/390 attempts to dynamically expand or contract only if it needs CF storage. This function is internally enabled, but it is strictly under a user's control. It does not take effect unless users explicitly specify ALLOWAUTOALT(YES) for a structure in CFRM policy.

This support may be beneficial for PRELOAD VSO structures. When there is unused storage for a PRELOAD VSO structure, the over-configured storage is wasted, and it is not reclaimed. Users should be cautious of requesting the auto alter function for non-preload VSO structures as they are loaded on demand.

OS/390 Version 2 Release 10 and CF Level 9 are the minimum software and hardware requirements for this feature.

The setup process for automatic alter for VSO structures is as follows:

1. Using the CFRM Policy Update utility (IXCMIAPU) for CFRM parameters under the STRUCTURE keyword, specify ALLOWAUTOALT(YES).
2. Set the MINSIZE (minsize) value:
   - For OS/390 Version 2 Release 10 and z/OS Version 1 Release 1: the MINSIZE default is 50% of INITSIZE or 50% of SIZE if INITSIZE is not specified.
   - For z/OS Version1 Release 2: the MINSIZE default is 75% of INITSIZE or 75% of SIZE if INITSIZE is not specified.
3. Set the FULLTHRESHOLD(value). The default is 80% of INITSIZE or 80% of SIZE if INITSIZE is not specified. If FULLTHRESHOLD(0) is specified, the automatic alter will not be active even with ALLOWAUTOALT(YES) specified.

7.11 Shared SDEP operation

This feature introduces a few operational differences:

- Pre-allocation is now at first ISRT time rather than at open. With this design, CIs are not wasted if the data is only read.
- A review of small AREAs containing SDEPs should be completed to determine if more space should be allocated, since the IMSID in the CI prefix and the STCK time stamp field in each segment will take up more space in the portion of the area assigned to SDEPs.
- The DEDB SDEP scan and delete utilities may take longer to run with IMS Version 6 than with Version 5 because the merging of the SDEP segments is left until the utility execution time rather than during the online update period.
- New parameters have been added to the DEDB SDEP scan and delete utilities that should be reviewed.
- The NOSORT Sysin DD utility control statement for the DEDB scan utility must be specifically set if required since it is not the default. You may want to set the NOSORT statement for non-shared AREAs or if applications invoked after the scan utility sort the data.
- When an area is closed and reopened, the pre-allocated but unused CIs are lost and have to be made available by use of the DELETE utility. keyword on a SDEP utility run to make all of the sharing partners quit using their allocated SDEP CI RBAs immediately before the utility begins reading SDEP CIs.

  QUITCI is not enforced when using the V5COMP control statement but it is suggested that you observe it. Otherwise partners holding CIs will place new segments in them and the user will have to read from the logical beginning point in the SDEP portion of the DEDB area to pick up the new work. For the user that wants to cut down on how many control intervals are read and processed, QUITCI and V5COMP offer a way to do this.
- Emergency restart does not recover an area's allocated next CI set that was lost when an IMS system failed. This preallocated CI that never became a current CI will have a dummy aborted first segment placed in the CI and the creation of a type X'5953' log record will make the CI prefix and its first segment recoverable. But these CIs are not reused until they are made available by the delete utility.

Utilities that access and interpret the data associated with the SDEPs must be modified to understand the new SDEP segment and CI formats.

7.12 FDBR operation

There are four major phases of operation for FDR: initialization and synchronization, tracking and surveillance, the recovery itself, and post recovery operations.

7.12.1 FDBR initialization and synchronization

The first phase of FDBR is initialization:

- IMS is started with a FDRMBR=xx specified:
  - The DFSFDRxx member enables this IMS to interface with FDBR.
  - IMS joins the special XCF group for monitoring purposes. The following messages should be observed:

```
DFS4190I IMS SYSTEM IS FDR CAPABLE
DFS3871I JOINAVM WAS SUCCESSFUL
```

- DFS3871I indicates that the OS/390 availability manager was connected to IMS via the JOINAVM call.

- IRLM is started (if not already available) for FDBR:
  - IRLM joins the data sharing group of the active IMS.
  - FDBR connects to this IRLM using the FDBR plus the IMSID naming convention.

  IRLM status can be checked using the command:

  ```
  F irlmname,STATUS,ALLD
  ```

- FDBR is involved by the OS/390 START command on the system it will reside on.
- DBRC and DLISAS modules, necessary control blocks, and pools are loaded and allocated into the FDBR address space. This is the synchronization phase.
  - FDBR attempts to identify itself to IRLM and if the IRLM is not active, a U0574 abend occurs. Also, FDBR must verify that the active IMS is alive in the data sharing group. If so, it joins the XCF group with the active IMS for monitoring. If the active IMS does not join the XCF group, FDBR ends abnormally with a U0574 abend.
  - When the FDBR processes all checkpoint records obtained, this message is issued:

    ```
    DFS4161I FDR FOR imsid TRACKING STARTED
    ```

    Now synchronization is complete. If there is a failure to locate checkpoint log records, a U0574 abend occurs.

7.12.2 FDBR tracking

The DBRC region, once established, tracks an IMS subsystem in the same XCF group.

- Each second, FDBR reads the IMS online log data set (OLDS) until it reaches logical end-of-file (EOF), ignoring unnecessary log records such as those associated with IMS TM activity, MSDBs, and DEDBs specified as SHARELVL = 0 or 1 with NORECOV set.
- The DB related control blocks and buffers are updated as required in a manner similar to the emergency restart process. The online change process is supported by FDBR tracking.
- There are several commands that can be used to monitor and modify FDBR activity and status:

  ```
  /DISPLAY FDR
  /CHANGE FDR TIMEOUT nnn
  F fdbrname, STATUS | RECOVERY | TERM | STOP | DUMP
  ```

7.12.3 FDBR surveillance

FDBR can detect an IMS or system failure in several ways:

- If IMS abends, it writes a x'0607' log record. When FDBR reads this record, it invokes its recovery actions and sends the following message:

  ```
  DFS4166I FDBR FOR imsid DB RECOVERY PROCESS STARTED,REASON = IMS FAILURE
  ```

- FDBR also has a log time-out function. If IMS does not write any log records for five seconds, FDBR writes a warning message. However, the lack of a log record will not cause FDBR to invoke its recovery processing. The warning message is:

  ```
  DFS4164W FDBR FOR imsid DETECTED TIMEOUT DURING LOG AND XCF SURVEILLANCE
  ```

- XCF status monitoring

  When FDBR is used, IMS and its tracking FDBR system join an XCF group. XCF status monitoring is used to inform FDBR if the IMS system has failed. XCF monitors both the status of the IMS system and the status of the OS/390 system in which it is executing.

  If the IMS system fails to update its status field within the user defined time-out interval, FDBR is notified by its XCF group user routine. In this case, FDBR writes a warning message:

  ```
  DFS4165W FDBR FOR imsid XCF DETECTED TIMEOUT ON ACTIVE IMS SYSTEM, REASON = IMS SURV.
  DIAGINFO = xxxx
  ```

  As with log time-outs, FDBR will not invoke recovery for an XCF detected time-out. If the IMS system's address space terminates, FDBR is notified by its XCF Group user routine. In this case, FDBR writes a message and invokes its recovery processing. The message is:

  ```
  DFS4166I FDBR FOR imsid DB RECOVERY PROCESS STARTED, REASON = XCF NOTIFICATION
  ```

  Note that this only occurs when the address space terminates. It is not invoked for merely an abend. If IMS is a started task, an abend will cause address space termination. If IMS is a job, an abend will not cause address space termination. So, XCF status monitoring of the IMS system will cause automatic invocation of recovery processing for IMS started tasks, but not for jobs.

- If the OS/390 system fails to update its sysplex couple data set within the user defined time-out interval, FDBR is notified by its XCF Group User routine. FDBR writes a warning message. The message is:

  ```
  DFS4165W FDBR FOR imsid XCF DETECTED TIMEOUT ON ACTIVE IMS SYSTEM, REASON = SYSTEM,
  DIAGINFO = xxxx
  ```

  This user defined time-out interval is specified by the TIMEOUT parameter in the IMS DFSFDBRxx member. This time-out does not cause FDBR to invoke its recovery processing.

- A failure of the OS/390 system to update its sysplex couple data set might result in system isolation. If system isolation occurs for the OS/390 on which IMS is running, FDBR is notified by its XCF Group user routine. This causes FDBR to write a message and invoke its recovery processing. The message is:

  ```
  DFS4166I FDBR FOR imsid DB RECOVERY PROCESS STARTED, REASON = XCF NOTIFICATION INVOKING
  RECOVERY
  ```

  If FDBR fails while it is tracking IMS, it can be restarted and resume tracking. The restart can be done by the operator or by the Automatic Restart Manager (ARM). ARM support for FDBR is explained below.

7.12.4 FDBR recovery processing

There are two parts to FDBR recovery. First, FDBR must complete reading all of the log records to prepare for the full function backouts and DEDB redos. Second, it must do the backouts and redos. Completion of the log reads can be done as soon as FDBR is aware that IMS is no longer running. This is signaled by one of three methods:

- FDBR reads an IMS abend log record (type x'0603').
- XCF invokes the FDBR Group User Routine informing FDBR of the termination of OS/390 or the IMS address space.
- An operator command is issued requesting an FDBR recovery. This command is:

  ```
  F FDBRPROC,RECOVER
  ```

FDBR cannot begin the backouts and redos before it is sure that all I/O which the failed IMS has started is complete. OS/390 has a function that indicates that this I/O is complete. It is called I/O prevention. When I/O prevention is complete for an address space, OS/390 has ensured that no more I/Os will be done by it. Of course, if OS/390 fails, it cannot invoke I/O prevention. In this case, the system isolation function of Parallel Sysplex ensures that no more I/Os are being done.

Finally, the termination of an address space indicates that all I/Os are complete. XCF monitoring will make FDBR aware of the system isolation of an OS/390 or address space termination. In other cases, the operator must make FDBR aware of the complete of all I/Os. When the operator issues the “F fdbrproc,RECOVER” command, FDBR assumes that I/Os by the failed IMS have completed.

7.12.5 Restarting IMS after FDBR completion

IMS must be restarted after FDBR completes its recovery processes. Restart is required to resolve in-doubt threads with CICS® and DB2, recover message queues, change the DBRC subsystem record from failed to normal, release DBRC authorizations, and recover MSDBs. This recovery should be done as soon as possible, but cannot be done before FDBR completes.

When FDBR completes its recovery processing, it issues the following message:

```
DFS4168I FDBR FOR imsid DATABASE RECOVERY COMPLETED
```

Automation can use this message to indicate that FDBR's recovery is complete and that the associated IMS should be emergency restarted.

If FDBR fails while it is in database recovery phase, it cannot be restarted. FDBR restart requires that its associated IMS be active and that IMS system should be emergency restarted. This will complete the database recovery processes.

7.12.6 DBRC authorizations with FDBR

FDBR does not change the DBRC subsystem record from failed to normal. It also does not release database authorizations. When FDBR completes, it has released locks held by the failed IMS, but this IMS still holds database authorizations. Other IMS subsystems cannot obtain new authorizations for databases which are authorized to the failed IMS. Authorization protocols prevent this by checking that all authorized subsystems are known to the IRLMs before granting new authorizations for a database.

This implies that from the time FDBR completes its recovery until the IMS subsystem is emergency restarted, new database authorizations will not succeed for any databases authorized to the failed IMS. Restarting the failed IMS quickly is highly recommended.

7.12.7 General FDBR operational recommendations

The following actions are highly recommended to make FDBR most effective. If they are not followed, some FDBR recoveries might require operator actions.

- Execute IMS as a started task, not a job.

  If IMS is executed as a job, abends require operator action on the FDBR system to complete recovery processing. The operator must manually inform FDBR of the completion of I/O prevention.

- Specify `ISOLATETIME` in the sysplex failure management (SFM) policy.

This causes system isolation to be invoked for a failed OS/390 system. A specification of PROMPT, instead of ISOLATETIME, requires operator actions when XCF status monitoring recognizes the time out of an OS/390 system.

- Use a name that associates an FDBR with the IMS it tracks.

  Commands for FDBR are issued by using an OS/390 modify command and specifying the FDBR procname. This procname should be easily associated with the IMS system which the FDBR system tracks.

  Similarly, the XCF group name used for FDBR should also include the IMSID. The default group name is FDBR followed by the IMSID. It is reasonable to make the FDBR procedure name the same as the XCF group name.

8

Data sharing backup, recovery, and restart

The block level data sharing environment introduces significant changes in the way in which data integrity is maintained. IMS, DLI separate address space (DLI/SAS), OS/390 services, Coupling Facilities (CFs), IRLM, and DBRC now become integral components that manage shared resources. These facilities all provide the mechanism to manage locks, data concurrency, system and subsystem status, event monitoring and communication.

For example, to effect a typical VSAM database retrieval and update in a block level data sharing environment:

- The subject IMS subsystem must confer with DBRC to determine whether this particular IMS is authorized to update this database record.
- IMS subsystem will issue a database record lock request to IRLM on the subject database record. A lock entry in the IRLM lock table will be created.
- DLI/SAS, determines whether the database record is already in a buffer. If it is, buffer validity must be determined by interfacing with OS/390. If the local buffer has been invalidated, DLI/SAS retrieves the record from the database, and puts it into a local buffer. In the buffer invalidate cache structure, interest in the CI will be registered through the architected OS/390 XES interface.
- IRLM must obtain an update or exclusive lock for the management of this VSAM CI of buffered data. A record list entry in the IRLM lock structure is created.
- For DL/I databases, IMS logs the before image of the data.
- The data will be updated in the local buffer and IMS logs the changed data event.
- At syncpoint/checkpoint time, the changed buffer will be written back to disk by DLI/SAS and IMS writes a commit record to the log.
- OS/390 services and hardware invalidates the buffer in all IMSs in the sysplex that registered an interest in the CI with the CF.
- IRLM is called to remove the database record and CI lock on the updated buffer from the lock structure within the CF.

This increased involvement in several major processing components intensifies the need for those responsible for data and system integrity, and high availability in BLDS environments to understand backup, recovery, and return to service for both data and system resources.

8.1 Database backup and recovery responsibilities

Data retained in IMS databases is secured in a variety of ways. You might use the standard IMS image copy utilities, or IBM or non-IBM tools, or have developed your own utilities. If you are using backup and recovery facilities that do not support DBRC, you now become responsible for ensuring that:

- The database is copied only when it is in a valid, current state. This goes considerably beyond just issuing a /DBR or /DBD command to take the database offline.
- All database components are at a consistent level. This can be tricky if using secondary index or logical relationships or any multiple data set database.
- Recovery is done correctly. For example, selecting the appropriate change accumulation volumes for database recovery is complex in a large BLDS environment with many sharing and updating IMS subsystems.

Those of you who intend to implement data sharing must reassess your backup and recovery strategies to ensure that they are still adequate. In many cases the volumes of recovery media created today by online production environments demand supported automated sets of tools and procedures.

8.2 Database backup

IMS provides these backup utilities:

- Database image copy (DSFUDMP0)
- Concurrent image copy (DFSUDMP0 with the CIC option)
- Online database image copy (DFSUICP0)
- HSSP image copy
- Database Image Copy 2 (DFSUDMT0)

Some considerations specifically apply to these facilities in block level data sharing environments.

8.2.1 Database image copy (DFSUDMP0)

With the batch image copy utility (DFSUDMP0) you get a valid, complete copy of your database. The database being copied must not be updated from anywhere in the data sharing group for the duration of the image copy job. Follow these steps:

1. Issue the /DBRECOVERY DB *DBNAME* GLOBAL NOFEOV command. You may use the /DBDUMP command instead for non Fast Path databases if you require read access during the image copy process.

   The /DBRECOVERY command for Fast Path Virtual Storage Option (VSO) AREAs in an OS/390 data space removes the AREAs from the data space and forces updates to be written back to DASD. The command is propagated to each IMS system in the sysplex, while DBRC is updated not to authorize access to the database. This processing is unique to non shared VSO DEDB AREAs.

2. Verify that the command executed successfully. The command is propagated to each IMS system in the sysplex, while DBRC is updated not to authorize access to the database.

3. Run the image copy job.
4. Issue the /START DB DBNAME GLOBAL command. For Fast Path databases, you must start the AREAs individually by using the /START AREA command. For virtual storage option (VSO) AREAs that have been defined with the PREOPEN option, /START AREA causes the AREAs to be pre-opened. If the VSO area is defined with the PRELOAD option, /START AREA causes the AREA to be opened and loaded into the OS/390 data space. A scripted file containing these commands could be invoked by the time-controlled option (TCO) facility.

This image copy can be used to back up HDAM, HIDAM, HISAM, SHISAM, secondary index, and DEDB database data sets.

8.2.2 Concurrent image copy

The Concurrent image copy (CIC) is an option of DFSUDMP0 that allows you to image copy a database data set without any of the sharing subsystems having to release access to the target database.

The backup that is created is a *fuzzy* image copy. Fuzzy means that there might have been concurrent updates during the image copy process. DBRC determines all of the logs required for recovery by establishing a safe point in time, before the creation of the backup, from which it can guarantee full integrity. A safe point is considered to be the last completed IMS system checkpoint before the start of the image copy. DBRC potentially goes back two IMS checkpoints, which implies that a log tape that was closed before the start of the concurrent image copy (CIC) can be legally included in the recovery JCL.

The main drawback to the CIC option is that it supports OSAM and VSAM ESDSs only, and therefore excludes HIDAM index and HISAM databases, which are stored in VSAM KSDSs. This restriction is implemented to avoid a concurrent CI/CA split which could move a logical record into a CI that is never copied by the utility.

8.2.3 Online image copy (DFSUICP0)

Online image copy (OLIC) also creates a fuzzy image copy. It runs as a BMP procedure with jobname DFSUICP0 and does not support Fast Path databases.

OLIC becomes somewhat restrictive in the Parallel Sysplex IMS data sharing. It requires that the IMS subsystem servicing the DFSUICP0 BMP be the only update capable IMS in the sysplex. It enforces this through DBRC authorization protocols. The database to be image copied must be restricted to *read only* status in all other sharing subsystems before DFSUICP0 can run.

8.2.4 HSSP image copy

If you use the image copy option of HSSP, IMS creates image copies of DEDB AREAs for you. The image copy contains the *after images* of the HSSP PCB, and is a fuzzy copy. IMS logs all other PCB database changes as usual. During database recovery, you must apply any concurrent updates since the start of the image copy process. A fuzzy image copy can be created if a non-HSSP region updated the same DEDB area during the image copy process.

IMS treats HSSP image copies like concurrent image copies, so you can use the database recovery utility (DFSURDB0) without telling it that the image copy is an HSSP image copy.

8.2.5 Database Image Copy 2 (DFSUDMT0)

SMS concurrent copy has been used by IMS customers before, but starting with IMS Version 6, IMS provides an integrated DFSMS and IMS DBRC supporting solution.

The concurrent copy function of DFSMS is a hardware and software solution that allows you to back up a database (or any collection of data) at a point in time and with minimum down time for the data. Update processing must be stopped only long enough for DFSMS to initialize a concurrent copy session, which is a very small fraction of the time that the complete backup would take.

DFSMSdss™ is the component of DFSMS that provides the software support for the D/T 3990 Concurrent Copy feature. For more information on DFSMS, see *DFSMS/MVS® V1R5 DFSMSdss Storage Administration Referenc*e, SC26-4929 and *DFSMS/MVS V1R5 DFSMSdss Storage Administration Guid*e, SC26-4930.

The Image Copy 2 utility, DFSUDMT0 invokes DFSMSdss with a DUMP command specifying the CONCURRENT option. The invocation is through the DFSMSdss cross-memory application program interface to a separate address space. The image copy data sets produced by DFSUDMT0 are in DFSMS dump format. The DFSMS dump format is different from the format of other IMS IC data sets and may not lend itself to certain types of post processing unless the tools in question support this IC format for input.

There are benefits to the use of this facility:

- The time the database is unavailable for updates when a clean consistent copy is required is reduced because:
  - The process operates as a logical and physical copy, and updating may resume after the logical copy phase completes.
  - The utility can copy a database that is either stopped or active. If the database is stopped, it can be restarted after the logical copy is complete, and database updating can continue.
  - The output or physical image copy does not include concurrent updates.
- DBRC support is introduced. DBRC is notified by the utility when the physical copy is completed. A NOTIFY.UIC is not required.
- Additional copies (COPY3 and COPY4) can be used for remote site disaster recovery purposes. Only COPY1 and COPY2 are registered with DBRC.
- A KSDS can be copied during concurrent update processing from online and batch applications to create a fuzzy image copy. The KSDS have to be SMS managed to be targets of this processing and in the AMS *DEFINE* or *ALTER* statements *BWO(TYPEIMS)* must be specified for *fuzzy* copies of KSDS.

There are two image copy types available to IC2 users:

- SMSNOCIC, which is consistent (clean). The database can be stopped briefly to produce a consistent or clean image copy. For the clean version, IMS database or AREA authorization processing prevents IMS applications from updating the database or AREA while the concurrent copy session is being initialized.
- SMSCIC which is the fuzzy version. For IMSs implementation of DFSMS concurrent copy support, a database can remain available for update processing for the entire duration of DFSMS processing in which case this fuzzy image copy is produced.

This image copy can be used to back HDAM, HIDAM, HISAM, SHISAM, secondary index, and DEDB database data sets. The database recovery utility (DFSURDB0) accepts

SMSNOCIC and SMSCIC image copies as input and DFSMS is invoked to restore the data set from an SMS image copy.

Image Copy 2 functions have been enhanced in IMS Version 7 and 8:

- In IMS Version 7, compression can be used for image copies. Via an user control card selections or the (COMPRESS) keyword added to the GENJCL.IC command, the compress option will be selected on the DFSMSdss DUMP command. DFSMSdss RESTORE invoked by the IMS database recovery utility expands the compressed data. The trade-offs are that image copies require less space but more processing is required to produce the compression.
- With IMS Version 8 several facilities have been added. Refer to *IMS Version 8 Implementation Guide: A Technical Overview of the New Features,* SG24-6594, for details on these features.
  - Multiple database data sets to be copied in one execution of the utility. Multiple control statements can be specified, one per database data set to be copied in a single execution. All of the DBDSs specified on the control statements are passed to DFSMSdss on a single invocation. This allows DFSMSdss to start multiple dump processes in parallel, and logical completion can be achieved for all of the DBDSs in a very brief period of time.

    Note: Without DFSMSdss APAR OW54614, DFSMSdss would serialize multiple dump or restore tasks that are using the same DASD volume, even if the PARALLEL keyword was specified. APAR OW54614 allows multiple dump tasks to execute in parallel while dumping to the same DASD volume. The serialization still applies for multiple dump tasks dumping to the same tape volume.

  - A group name can be specified to represent the collection of DBDSs that are to be copied in a single execution. The group name is specified on the group name control statement. The group name should not be confused with the DBRC DB or DBDS groups defined in the RECON. The group name exists only for the execution of the Image Copy 2 utility. However, using a DB or DBDS group name on the group name statement for the utility can simplify operations.
  - DFSMSdss provides the capability to dump up to 255 data sets into one output data set. The multiple dumps are written one after another in the output data set, each preceded by DFSMSdss dump header records. By exploiting this capability, Image Copy 2 provides an option to concatenate the image copies that are created by a single execution of the utility.
  - There are changed and new messages in IMS Version 8 for image copy completion and failure notification. This provides more information that you have of the operational status of the utility and options to automate responses when these messages are issued. There are a few migration considerations for IMS Version 7 users that automate on these messages and upgrade to IMS Version 8:
  - The logical copy complete message DFS3121A either can or cannot have the data set name included based on the options chosen.
  - The physical copy complete message DFS3141A and DFS3141I listing the database or AREA did not exist in IMS Version 7.
  - The DFSMSdss OPTIMIZE option is support. DFSMSDSS provides four levels of optimization to control the number of DASD tracks transferred in one I/O operation.

    OPTimize(1) 1 track (IC2 default for fuzzy ICs)
    OPT(2) 2 tracks
    OPT(3) 5 tracks

OPT(4) 1 cylinder (IC2 default for clean ICs)

You can specify the optimization level 1|2|3|4 on the GROUP or DBDS control statement at column 61.

> **Note:** If you are taking advantage of the capability to have the Image Copy 2 utility specify SET PATCH commands to customize DFSMSdss processing, be aware that the zap now has to be applied in module DFSUDMT2 instead of DFSUDMT0. (IMS Version 8 APARs PQ63048 and PQ67001 contain information about the support for SET PATCH commands.)

– The enhancements for Image Copy 2 are also available using GENJCL.IC commands and are supported by new and changed skeletal JCL members.

Restrictions associated with DFSMS Concurrent Copy and IMS

The following restrictions apply to usage of DFSMS Concurrent Copy with IMS:

- The database and AREA data sets to be copied must reside on hardware that supports the concurrent copy function (such as, a 3990 Storage Control Model 3, extended function with licensed internal code, or an equivalent device).
- The Database Image Copy 2 utility (DFSUDMT0) must be executed using region control (DFSRRC00). Message DFS3149A is issued and DFSUDMT0 will terminate if executed as a stand alone program.
- HSAM, GSAM or MSDB databases cannot be copied with this utility.
- The database data sets or AREAs must be free of errors (EQEs or EEQEs) to be processed by the utility. If multiple AREA data sets (MADS) exist for an AREA, at least one AREA must be free of errors.
- The image copy will be aborted if:
  - A single copy has been selected and COPY1 has I/O errors.
  - A dual copy has been selected and both copies have I/O errors.

  If either of the above situations should occur, the image copy is aborted even if COPY3 and COPY4 are free of errors.
- The databases and AREAs must be registered with DBRC and have a SHARELVL greater than zero.
- Databases that are specified as nonrecoverable must be stopped for the Database Image Copy 2 utility.
- VSAM data sets must be cataloged in an integrated catalog facility (ICF). This is a DFSMSdss requirement which the IMS image copy and recovery utilities do not attempt to enforce. One of the following must be satisfied:
  - SMS-managed
  - Cataloged using an alias (That is, the high-level qualifier(s) of the data set name is an alias for the catalog)
  - Cataloged in the master catalog
- To take a fuzzy image copy of a KSDS, the KSDS must be SMS-managed and the DEFINE CLUSTER attribute, BWO(TYPEIMS); where, BWO stands for *back up while open* must be specified.

8.2.6 Restrictions for concurrent updates with different image copy types

Table 8-1 summarizes the restrictions for concurrent updates when using each of the image copy types.

Table 8-1 Image Copy with concurrent updates

Image Copy type	Restrictions
Image copy (DFSUDMP0) without CIC	Concurrent updates are not allowed.
Image copy (DFSUDMP0) with CIC	Concurrent updates of KSDS not allowed.
Online image copy (DFSUICP0)	Concurrent updates by other subsystems not allowed.
HSSP image copy	Must initialize the image copy data set with a DBRC INIT.IC command. Database to image copy must be registered to DBRC.
Image Copy 2 (DFSUDMT0)	None.

8.3 Log archive

In a cloned data sharing environment, there are multiple sets of logs that are created by IMS subsystems and these separate log streams. The log archive function executes as a batch job and it may run on any OS/390 system, even one that the archived OLDS was not created on.

The automatic use of the GENJCL.ARCHIVE command is appropriate for data sharing. The class used for an archive job is specified in the JOBJCL member of the DBRC skeletal JCL. If all archive jobs use the same class, they could all use the same skeletal JCL. The number of initiators servicing this class will probably have to be increased as the volume of logging grows.

The SSID (subsystem name) is available for the GENJCL process; and different systems may create output data sets which include their own SSID as a qualification in the created archived data set.

8.4 Change accumulation

When more than one IMS subsystem updates databases concurrently, the log records required for recovery must be merged. You can use the merge function of change accumulation (DFSUCUM0) to accomplish this, running DFSUCUM0 as often as your workload requires. You thereby accumulate change records on a timely basis and keep the database recovery log input to a reasonable number of volumes.

It is not necessary to take IMS down, switch logs or stop any databases. If you run the change accumulation (CA) utility with an incomplete log subset, you generally specify the CA output data set as input to your next database change accumulation run.

Use the DBRC LIST.CAGRP command to determine whether the log subset of the CA group is complete or not. CA combines all database change records created across the data sharing group into a single data set. The log data sets must be available at the OS/390 site that controls the change accumulation job. If you intend to implement your sysplex across physically separate sites and write your recovery logs to tape, you may have to consider writing them to disk or to use your OLDS as input to DFSUCUM0.

You have the option of a shared master catalog in the sysplex, but you may not want to record your log data sets in there. If you do not, consider the use of the NOCATDS feature, which causes DBRC to select the required data sets in the RECON by volume serial number rather than by catalog name.

In IMS Version 7, the CA utility was enhanced so that there is a reduction in spill records. Spill records are produced in a data sharing environment, because when the CA is run, not all of the log data is available. It could be OLDS not yet archived on all the active IMS systems and archiving in a sysplex environment cannot be coordinated to have the archives completed at any one time.

IMS Version 7 tends to produce fewer spill records than IMS Version 6. This is because IMS Version 6 uses the DSSN value in the log records to determine which records to include. The utility does not accumulate any records if any log records have a lower DSSN value. Therefore the size of the CA data set is smaller and this process will take less time.

8.5 Database recovery

The operational aspects of recovering a database involve some changes in IMS data sharing environments. Database recovery is accomplished through one, or a combination of these facilities:

- Forward recovery
- Time stamp recovery
- Dynamic backout
- Batch backout
- Remote site recovery

8.5.1 Forward recovery

You do forward recovery in data sharing environments by supplying the most recent backup data set to the database recovery utility (DFSURDB0) together with the latest change accumulation data set.

It is no longer possible to supply input log data sets to DFSURDB0 if more than one IMS data sharing subsystem updated the database since the last image copy. DFSURDB0 expects all database changes up to the recovery point to have been merged into a complete log subset. A GENJCL.RECOV command or execution of the database recovery utility using DBRC fails if a complete log subset is not available. The failure is accompanied by a message indicating that you should run change accumulation.

When recovery is required, follow these steps:

1. Issue the /DBRECOVERY DB *dbname* GLOBAL command to deallocate the target databases from all IMSs in the data sharing group and flag them as unauthorized in DBRC. This also forces a log switch on each IMS. An OS/390 ROUTE command can also be used to send the /DBR command to all IMS subsystems in the data sharing group.
2. Issue a GENJCL.CA command for the change accumulation group.
3. Run change accumulation.
4. Issue a GENJCL.RECOV command for the database to be recovered.
5. Run database recovery.
6. Issue the /START DB *dbname* GLOBAL command. For Fast Path databases, start the AREAs individually using the /START AREA *area name* GLOBAL command.

8.5.2 Time stamp recovery considerations

A time stamp recovery is a database recovery action that recovers an IMS database to a prior state or time stamp, rather than its most current state. Hence, the name, time stamp recovery.

DBRC dictates that the database must not have been in use by any IMS subsystem at the desired prior state or time stamp. In other words, an IMS recovery point must have been established prior to the need for the IMS recovery point. An IMS recovery point is normally established by:

- Issuing /DBR or /DBD commands (without the NOFEOV parameter) for the database or databases against all of the IMS sharing subsystems.
- Waiting for all of the commands to successfully complete.
- Issuing /STA commands for the database or databases against all of the IMS sharing subsystems.

It is imperative that the /DBR or /DBD commands complete on all of the IMS subsystems prior to the issuance of /STA commands on any of the IMS subsystems.

The establishment of IMS recovery points will result in a loss of database availability (either complete loss of availability in the case of /DBR commands or loss of update capability in the case of /DBD commands) while the recovery point is being established. In addition, the establishment of IMS recovery points requires the issuance of multiple, coordinated commands across multiple IMS subsystems.

We discuss the use of the IMS Data Management Tool, IMS Online Recovery Service (program number 5655-E50) in 8.6, “IMS Online Recovery Service (ORS) support” on page 174. It revolutionizes the manner in which you can control recovery of your IMS data.

8.5.3 Dynamic backout

Dynamic backout is the most used database recovery mechanism, responsible for backing out changes to databases by inflight transactions or jobs at the time of application failure or IMS emergency restart. Although the changes related to IMS data sharing environments are transparent to customer application programs, they have quite an impact in the event of a system failure.

When a database resource is to be updated, the IRLM requests nonvolatile storage in the CF IRLM lock structure record list to be updated to represent a *MODIFY* lock. In the event of a failure of either the database management system or the IRLM holding an interest in the resource, the MODIFY locks are converted to *RETAINED* locks. This causes updated resources to remain locked until dynamic backout is done during the recovery of the failing component through /ERE or FDBR processing.

The implication is that other subsystems in the data sharing group may be adversely affected until the failing subsystem is restarted. Any application call against these resources causes a LOCK REJECT condition. If applications are coded to cope with these conditions by issuing the STATUS GROUPA call, they receive a BA or BB status code and take appropriate action. If not, they are ended abnormally with a U3303 abend message and the associated transactions are to be USTOPPED.

8.5.4 Batch backout

The Batch Backout utility has additional restrictions in BLDS environments:

- IRLM is always required.

- The utility can back out to the last successful checkpoint only.

The CHKPT statement is valid only if the input log is from a batch job that did not use IRLM. Because IRLM is required when the data is shared, the CHKPT statement is ignored.

8.5.5 Remote Site Recovery

Remote site recovery (RSR) requires an isolated log sender at the primary site and a tracker IMS at a remote site to create a near-real-time disaster backup facility.

In IMS data sharing environments, all IMS subsystems use the same tracker IMS, so only one isolated log sender is required. Even though the isolated log sender is associated with a particular IMS subsystem, it fills the gaps for the whole group of sharers. It therefore must have access to the shared RECON data sets and to the archive logs of all the IMS subsystems in the data sharing group.

IMS maintains database integrity for shared, covered databases at the tracking site. The log router merges database update records in creation-time sequence. If log records are missing (that is, a gap exists) from one or more active subsystems for any shared database, the log router continues presenting records to the database tracking components, up to the beginning of the gap in the log. The log router stops all routing of the log data at the gap point, until the gap is filled. The tracking system continues accepting log data from the active site. After the gap in the log has been filled, routing resumes.

8.6 IMS Online Recovery Service (ORS) support

IMS Online Recovery Service (ORS) is a product which provides a database recovery process in an IMS online system. IMS dynamically allocates all of the required data sets to perform the recovery. ORS utilizes IMS commands as opposed to DBRC commands and generated batch recovery job streams. The IMS online system uses a dependent address space to restore the image copy and change accumulation files onto the database(s). Once these have been processed, IMS ORS reads log data sets and passes a recovery stream to the IMS control region address space to update the databases.

ORS support has been enhanced in IMS Version 8, and the enhancements made available to IMS Version 7 through the service process. IMS ORS will accept image copy data sets created by all releases of the IMS Database Image Copy 2 utility. ORS is now able to:

- Process IC2 SAMEDS image copies
- Utilize compressed image copies produced by IBM IMS Image Copy Extensions product
- Use virtual tape caching
- Send messages to IMS MTO (requires IMS Version 8)

To learn more about IMS Online Recovery Service, please see the IBM Redbook *A DBA's View of IMS Online Recovery Service*, SG24-6112.

8.7 Failure and recovery scenarios

In this section, we discuss the failure and recovery scenarios that pertain to data sharing systems components. We covered the recovery of IMS databases, separately, in 8.5, "Database recovery" on page 172.

Your installation should have a plan for what is to happen when any component of the Parallel Sysplex fails. For example, if a processor fails, there should be procedures that

determine what systems, such as IMS and CICS, should be available on other processors to continue the operation of core business computing applications. This has an effect during the planning and configuration stages of your migration, since naming conventions and backup capacity allocations must be kept in mind for the movement of workload in the case of failures.

8.8 Testing the data sharing sysplex

Determining what is needed for a valid test environment is an element of your configuration and availability requirements. The trade-off of equipment and resources against improved availability needs to be evaluated carefully. The greater the availability requirement, the more robust the test environment and plan must be, to ensure that errors are not introduced when moving or cloning systems into production environments.

For example, to functionally test transaction routing, the following is needed:

- At least two systems with the required software installed
- A CF or ICMF
- CF channels
- A transaction routing facility
- Shared databases for the applications to access
- A network driver such as a teleprocessing network simulator to simulate high transaction flows

It is this type of resource mapping and allocation that is required to effectively test function and work load stress management capacities.

8.9 Major data sharing component recovery restart processes

We now examine how the major components associated with IMS data sharing manage failure, recovery, and restart.

8.9.1 Failures of the CF component and its structures

There are various failure scenarios specific to the CF. In this section we examine how the major CF subcomponents function when failures occurs.

There are three major types of Coupling Facility failures:

- A failure of the CF containing the IRLM lock structure.
- A failure of the CF containing the OSAM or VSAM XI cache structures.
- A failure of the CF containing the DEDB SVSO cache structure.

8.9.2 Use of the event notification facility

Before we discuss the other data sharing components, it is important to understand how IMS is altered when Coupling Facility component status changes. The OS/390 event notification facility (ENF) event code 35 is used to inform interested subsystems such as IMS of changes in the state of CF resources.

When there is a change in the state of CF resources, the OS/390 system sends out an ENF event code 35. The IMS subsystem listens for the ENF event code 35 and the subsystem processes the particular function code for the event being signaled.

8.9.3 Failures with individual Coupling Facility structure components

The CF contains four structures that can be used by IMS for data sharing: OSAM, VSAM, DEDB VSO AREA cache structures, and a lock structure used by IRLM.

Failure of a CF with an IRLM lock structure

Structure failures can occur if the CF fails during operations or if structure storage is corrupted. When a loss of an IRLM structure or the CF that contains the IRLM structure occurs:

- IMS batch (DLI and DBB) data sharing jobs end abnormally with a U3303 abend code on the system with the loss of connectivity. Backout is required for updaters and all the batch data sharing jobs must be restarted later.
- Although the online system continues operating, data sharing quiesces and transactions making lock requests are suspended until the lock structure is rebuilt. Each IRLM participating in the data sharing group is active in the automatic rebuild of the IRLM lock structure in the alternate CF.
- When the rebuilding is complete, transactions that were suspended have their lock requests processed.

  To invoke automated recovery, a second CF is required and the CFRM policy must specify an alternate CF in the preference list.

  REBUILD uses the preference list to select the CF to automatically rebuild to. In the following example STRUCTURE statement, CF CF2 is the target of the rebuild of the structure IM0A_IRLM, which originally resides on CF CF1:

  ```
  STRUCTURE NAME(IM0A_IRLM) SIZE(4096) PREFLIST(CF1,CF2)
  ```

  The target CF structure is repopulated with active locks from the IRLMs.

  Figure 8-1 presents the state of the IRLM lock structure and IRLM local lock tables in the IMS subsystems in the data sharing group before the failure of the IRLM lock structure.

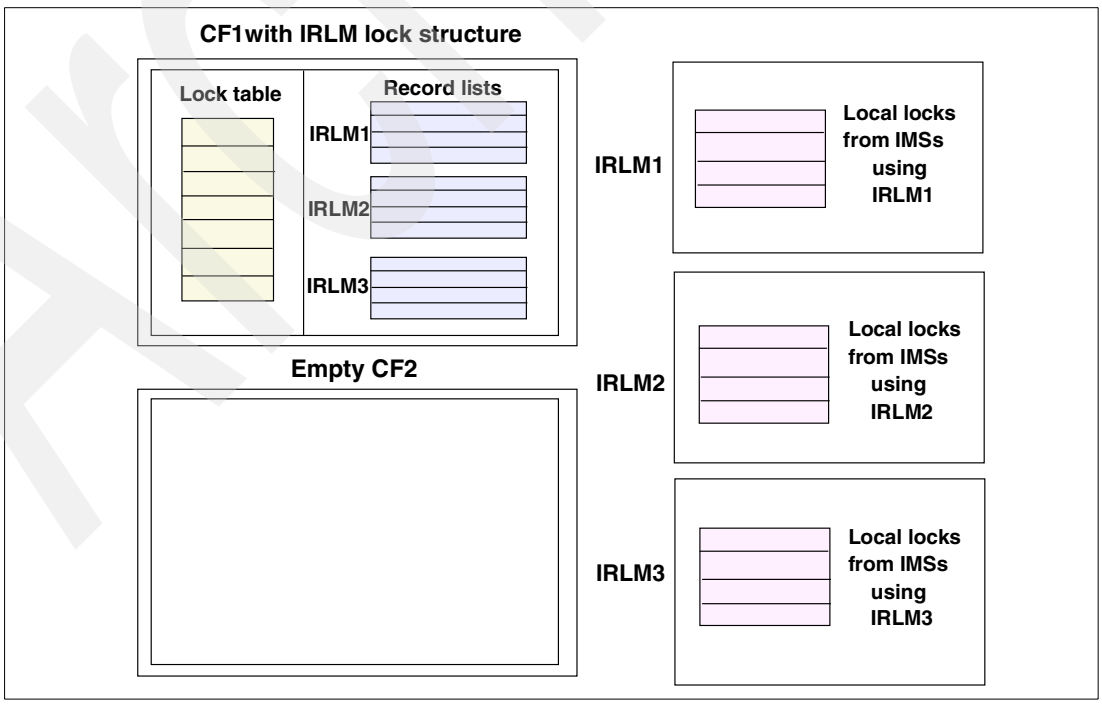

Figure 8-1 Lock structure and local IRLM lock tables before a CF failure

- All the IRLMs must connect to the newly rebuilt lock structure before they can determine whether the rebuild was successful.

Figure 8-2 presents the state of the IRLM lock structure and IRLM local lock tables in the IMS subsystems in the data sharing group after the failure of the IRLM lock structure.

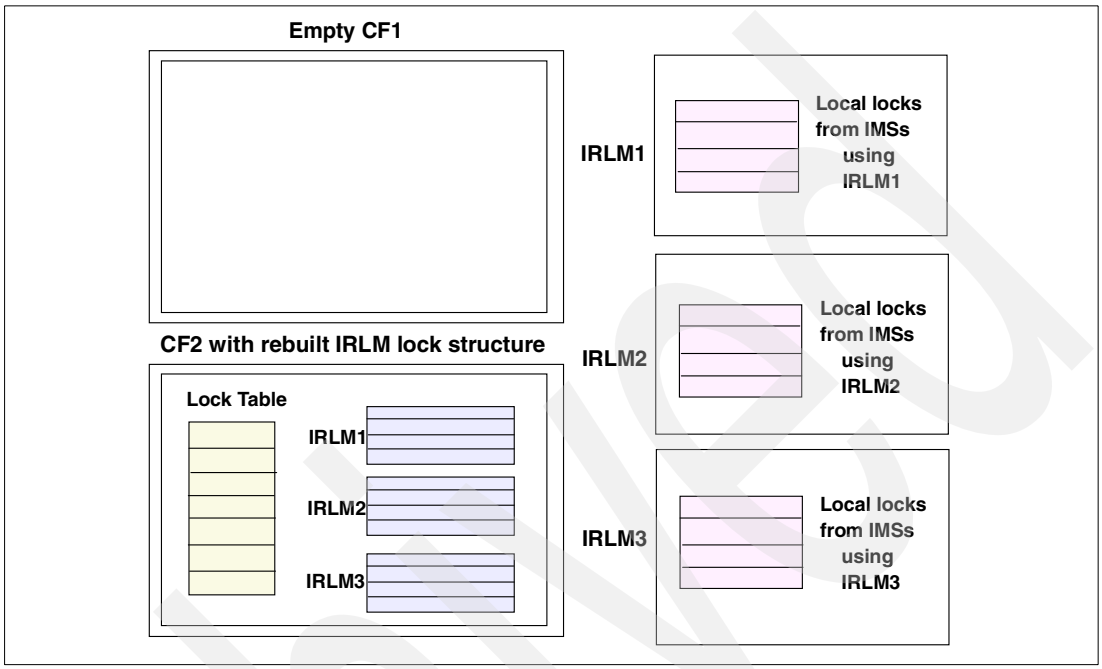

Figure 8-2 Lock structure and local IRLM lock tables after a CF failure

- If the automatic rebuild fails, it is probably because:
  - No other available CF was in the preference list for the lock structure.
  - The CF in the preference list has insufficient storage to rebuild the lock structure.

  If the automatic rebuild fails, this is treated as though the IRLM abended abnormally for the IMS subsystems, with batch (DLI and DBB) jobs failing with U3303 abend codes and the IMS TM and DBCTL quiescing. Manual recovery is then required. This comprises of:
  - A repair of the CF. The exact nature of this repair activity varies. If the CF is repaired, it must also be brought online to the processor by the OS/390 VARY command.
  - A policy may have to be redefined before a rebuild occurs to specify a different CF in the preference list, with enough allocated space.

Here is an example of the SETXCF command for REBUILD:

```
SETXCF START,REBUILD,STRNAME=IMOA_IRLM
```

Failure of a CF with an OSAM or VSAM cross invalidate structure

When failures to the CF OSAM and VSAM cross invalidate (XI) cache structures occur:

- All local OSAM buffers are invalidated if the CF contains an OSAM XI structure. OSAM buffers contain blocks from SHARELVL=1, 2, and 3 databases. This means that buffers cannot be used and any future request for blocks requires a read from DASD. SHARELVL=1 databases are affected, even though they do not participate in data sharing.
- All local VSAM buffers are invalidated if the CF contains a VSAM XI structure. The process for the VSAM buffer set is the same as for the OSAM buffers.

- The online subsystems continue, but some transactions may receive an error return code from an operation to the OSAM or VSAM structure. This results in an abend U3303 code or return of status codes BA or BB, depending on how the applications have prepared for a situation such as this.
- All transactions that would initiate an operation that will result in an access to the structure are placed into a wait. When the structure is rebuilt, they are taken out of this wait state.
- IMS batch (DLI and DBB) data sharing jobs abend, and backout is required for updates. All of the batch data sharing jobs must be restarted later.
- Automatic rebuild of the structures is attempted and if the rebuild is successful, all IMS subsystems reconnect to the structures. All IMS subsystems participate in the rebuild process since the architecture requires this for all connectors during a rebuild. The contents of the buffer pools are not used in this rebuild: the OSAM and VSAM structures are rebuilt but empty.

  No operator involvement is necessary and the time required for structure rebuild is measured in seconds, rather than minutes.
- If the automatic rebuild fails, it is probably because:
  - There was no other available CF in the preference list for the structure.
  - The CF in the preference list had insufficient storage to rebuild the structure.
- If a rebuild of the OSAM or VSAM structure fails, IMS quiesces data sharing. The following defines an IMS subsystem quiesce state:
  - IMS issues message: DFS3384I DATA SHARING STOPPED.
  - All SHARELVL=1,2, and 3 full function databases are stopped. This includes both OSAM and VSAM databases, even if only one of these structures is lost.
  - Calls to databases fail with abend code U3303 or the return of status codes BA or BB, depending on how the particular applications have been coded.
  - IMS TM places failed transactions on the suspend queues.
  - IMS TM can USTOP transactions.
  - DBCTL PSBs may be stopped.

  This quiesce condition is not the same as that produced by a /STOP DATABASE command. A /START DATABASE command, for example, will not change this condition.
- After resolving the problems associated with the automatic rebuild failure, issue these commands:

  ```
  SETXCF START,REBUILD,STRNAME=IMOA_OSAM or
  SETXCF START,REBUILD,STRNAME=IMOA_VSAM
  ```
- Following this manual rebuild, IMS TM or DBCTL receives notification of the rebuild via ENF code 35.
- IMS then reconnects to the structure and, when IMS starts databases, transactions are released from the suspend queues.

All buffers in subsystem pools are invalidated and we lose all data in the cache structure. Integrity of the data in the databases is not compromised, since this is a store-through cache structure. Everything that was in the structure has already been written to DASD.

Failure of a CF with a DEDB SVSO structure

Shared VSO DEDBs can reside in one or two duplicated structures, and these configurations have a marked effect on processing if a SVSO structure failure occurs.

Procedure for a failure of one of two structures

If only one of two structures for an AREA fails, it is not necessary to recover immediately. Processing continues with the remaining structure.

Based on your IMS version and maintenance levels there are two ways that the failed structure can be reinstituted:

- IMS Version 6 and Version 7 without APAR PQ50661 could use this method. Discontinue using the current structure by closing the AREA in question or issuing a /VUN, /STO, or /DBR command. The next time the AREA is opened, connections to the two structures will be attempted based on what is in the CFRM policy. This will cause the structures to be built.
- IMS Version 7 with APAR PQ50661 or IMS Version 8 can use the system-managed rebuild that OS/390 offers. This allows users to migrate a VSO structure from one CF to another without taking the VSO structure offline without a /VUNLOAD command.

There are requirements for the use of this feature:

- Hardware and software:
  - OS/390 Version 2 Release 8
  - CF Level 9
  - The connector (IMS) must allow system managed processes
- The CFRM CDS must be initialized with SMREBLD specified.
- The CFRM must define two or Level 9 CFs in the PREFLIST for the VSO structure.

Procedure for failure of the single remaining or both structures

If the only remaining structure for an AREA fails, the AREA must be recovered. Standard AREA recovery procedures apply. DBRC's GENJCL.CA and GENJCL.RECOV can be used to generate the appropriate recovery jobs. After the recovery of the AREA, it can be restarted on the sharing IMS subsystems. This will cause the structure or structures to be rebuilt. It might be necessary to change the CFRM policy so that the structures can be built on available CFs.

8.9.4 DEDB VSO structure read and write errors

For a DEDB implemented with either the local or shared variants of Virtual Storage Option (VSO), the writes to DASD occur periodically rather than as part of an output thread initiated by an application sync point. I/O error toleration handles any I/O errors in exactly the same way as if they had occurred in normal output thread processing.

- Read errors from the CF structure:

  If only one VSO cache structure is defined, the application is presented with an AO status code and message:

  ```
  DFS2830I CF read error, RBA rrrrrrr, AREA AREAname, STR structurename RC=xxxx RSN=yyyy.
  ```

  If two structures are defined for the DEDB VSO AREA, the read request is reissued to the second structure.
- When one read error occurs with a single structure:

  The AREA is stopped and the structure becomes failed persistent.
- When one read error occurs with dual structures:

  The first read error causes a disconnect from the offending structure and a continuation using the second structure. When both structures become unavailable for use, then the

AREA is stopped on that local system. If the CI is not in the CF, it is read from DASD and written to the CF.

- Write errors to a single CF structure:

  IMS Version 7 exploits the I/O toleration (IOT) function that was previously used only by full function databases, to improve the availability of data in a data entry database (DEDB). IOT is used to store data from CIs that experience write I/O errors in IOT buffers which are kept in storage and maintained across IMS restarts until the AREA concerned is recovered.

  When a write error occurs:

  – An EEQE is created in the RECON.
  – IOT buffer is built in ECSA.
  – The AREA is flagged as *recovery needed* in the RECON until the EEQE is resolved.
  – The data is stored in that buffer.
  – On this particular IMS subsystem, future access will be from this IOT buffer.
  – Data sharing IMS partners will be notified by the IOT component that a given CI is read-inhibited. Any copy of the data in CF cache will be deleted when the write error occurs and an IOT buffer is built.
  – Access originating from any other IMS receives an AO status.

  Rewrite is attempted:

  – At each IMS checkpoint.
  – Whenever a /STO, /DBR or /VUN is issued against this AREA.
  – If the rewrite is successful, the EEQE and IOT Buffer are cleared.
  – If the rewrite is not successful, users will need to eventually build an error-free ADS to make the data available to all data sharing IMSs. This can be done by a MADS Create function on the IMS system with the IOT buffer.

  There is a limit of 100 EEQEs before the AREA is stopped.

- Write errors in a dual CF structures:

  If multiple structures are defined, and a write request to one of the structures fails, the entry is deleted from the structure by this local IMS or one of the sharing partners. The write is then done to one of the other structures. If one of the writes is successful, then it is considered to be a completed write. If the write fails on both structures, IMS will process as if the write error occurred in a single MADS environment using EEQEs and IOTs.

- When one write error occurs with a single structure:

  The AREA is unloaded from the CF using castout processing and the AREA is stopped on the local IMS. Only changed CIs are written and if the CI with the read error has been modified, an EQE is created. Processing continues from DASD.

Emergency restart and SVSO AREAs

A restarted IMS must perform REDO processing for committed updates. For shared VSO AREAs, the committed updates may not have been written to the CF structures. Written CIs are shown by X’5612’ records on the log and will have been cast out by other IMSs.

The restart process looks for X’5950’ change records followed by X’5937’ commit records. Any of these not followed by a X’5612’ need REDO handling. Retained locks are released after the CIs have been updated. The initial system checkpoint after emergency restart casts out CIs for open AREAs.

8.9.5 Failures of the Coupling Facility link component

Link failures that cause a loss of connectivity can occur in multiple ways:

- A patch panel connection could be incorrectly setup or reinstalled.
- A plug could be removed by mistake.
- The link could be severed because of maintenance accidents.

If a CF link fails, a connection is automatically made using an available link. When the last available link fails, the IMS subsystem that lost connectivity is impacted.

Sysplex failure management (SFM) can be used to cause some connection failures to be treated as if they were structure failures. In these cases, an automatic rebuild on another CF is attempted.

A sysplex failure management (SFM) policy, with CONNFAIL(YES) specified (or allowed as a default) is required if you want to use the REBUILDPERCENT definition of the CFRM policy. The default of REBUILDPERCENT is a value of 100, meaning a rebuild will only be done if all connectors lose access to the structure (the whole CF fails). Rebuild is attempted only if the percentage of the total weights of the systems whose connections have lost connectivity to the CF, divided by the total weight of all systems with connectivity to the CF, is equal to or exceeds the specified REBUILDPERCENT value.

Recommendation: We recommend specifying a value of 1 for REBUILDPERCENT so that the rebuild attempt will always occur. Otherwise, the user of a system losing access to the structure is likely to drop out of the data sharing group.

When the links to the lock structure fail

SFM might cause a CF connection failure to be treated as a structure failure. If it does not, the following applies.

If the link was between an IRLM and the lock structure, the IRLM survives; however, the effects on the IMS subsystem using that IRLM are the same as if that IRLM abended:

- Batch data sharing jobs abend. After the reconnect, these jobs must be backed out and restarted.
- The affected IRLM disconnects from the data sharing group.
- Both IMS TM and DBCTL systems that use the failed IRLM quiesce.
- Transactions accessing the shared databases end abnormally with a U3303 abend.
- A display of the lock structure by issuing the following command will show the IRLM is in a FAILED-PERSISTENT state since the DISPOSITION of an IRLM lock table is KEEP:

  ```
  D XCF,STRUCTURE,STRNAME=IM0A_IRLM
  ```

  After the link failure, a display of the IRLM status by issuing the following command will show IMS subsystems in SFAIL status with a number of retained locks if update activity was processing when the CF link to the IRLM structure failed:

  ```
  F IM1AIRLM,STATUS,ALLD
  ```

- IMS TM or DBCTL listens for a CF link-status change and, if the link is repaired or rebuilt, OS/390 notifies the subsystems with a ENF event code 35. At that time:
  - IMS reconnects to the IRLM.
  - The IRLM reconnects to the structure and joins the data sharing group.
  - IRLM takes IMS out of its quiesced state.

- After reconnecting by IMS to the IRLM, each DEDB AREA must be started. This is not required for full function databases.

When the links to the OSAM and VSAM structures fail

SFM might cause a CF connection failure to be treated as a structure failure. If it does not, the following applies.

If the failing link is to a CF structure (OSAM or VSAM), then IMS invalidates all the buffers in this system, notifies other data sharing partners through IRLM, and quiesces data sharing. This is similar to what occurs with a CF failure without a successful rebuild.

The reason that IRLM is used to transport the notifications to data sharing subsystems is that buffer invalidations can occur because of database writes when the U3303 abends cause backout to be performed. If that happens, buffer invalidations are required for the other data sharing systems that can still access the structures. Since this system no longer has access to the buffer invalidation structures, the invalidations are distributed through the IRLMs, using the notify process. If the CF connection is repaired, or if the structure is rebuilt, the connection of IMS to the IRLM will be automatic.

When the link is established and IMS receives the ENF notification, IMS connects to the structure. It does not rebuild it. Instead:

- When IMS reconnects to the structure and starts databases, IMS TM will release affected transactions from the suspend queue. Batch data sharing jobs must be backed out and restarted.
- If rebuild fails, SHARELVL 1,2,3 OSAM and VSAM databases are stopped even if only one structure is lost; data sharing is quiesced with the message:

  ```
  DFS3384I DATA SHARING STOPPED.
  ```

When the links to the SVSO cache structure fail

When the connectivity to a Fast Path DEDB SVSO cache structure is lost, all connected IMSs are notified by XES:

- If multiple structures have been defined for the VSO DEDB and the second structure access remains available, then IMS continues processing with the remaining structure.
- If there is not another structure, the AREA is stopped in each IMS which loses connectivity. Their connections become failed-persistent connections. Sharing and updating of the structure is continued only by the IMSs that continue to have connectivity. When the links fail, all the applications that have in-flight updates will terminate so no retained locks will exist. DBRC is not made aware of this situation.
- At the time of the connectivity failure, an EEQE is created for CI 0 (RBA=00000000). This EEQE will prevent an image copy from being taken before a complete castout operation can take place. The EEQE will be deleted after a complete castout occurs.
- Starting the AREA will attempt a reconnect. If it is successful, sharing will be re-established for the subsystem. Also with IMS Version 8 and IMS Version 7 with APAR PQ50661 system-managed rebuild is supported for Shared VSO DEDBs.

8.9.6 Failure of the OS/390 component

When an OS/390 system fails, so does the IRLM on that image, and other IRLMs are notified by XCF to read and retain the modified locks associated with the failed IRLM.

After the OS/390 system is restarted the following must occur:

- IRLM has to be restarted before IMS, either on the same OS/390, or on another. IMS connects to the IRLM without intervention.
- IMS emergency restart or FDBR will free the retained locks during dynamic backout processing. The /ERESTART command can be executed on any OS/390 system in the complex and any IRLM in the data sharing group can be used.
- Emergency restart CICS (if used).

  This is needed to resolve any in-doubts between CICS and DBCTL. ARM can be used.
- Restart DB2 (if used).

  ARM can be used for DB2 V4 or later releases. This is needed to resolve any in-doubts between IMS TM and DB2.
- Restart BMPs.

  If `LAST` is specified for CKPTID, the same IMS system must be used. If the restart records are read from the IMS log, the same IMS system must be used.

  If the restart is done by application logic using data in a shared IMS database, the restart can be done on any IMS system in the data sharing group. If the restart records are read using an IMSLOGR DD statement, the restart can be done on any IMS system in the data sharing group.
- Batch (DLI and DBB) data sharing jobs must receive attention. Back out update jobs and any read jobs which hold modify locks. Any utilities must have a CFNAMES statement in its DFSVSAMP data set. Until component restarts, and batch and dynamic backouts are completed, there is the possibility that potential lock rejects can occur.

8.9.7 Failure of the IMS subsystem component

Whether the IMS subsystem component that fails is IMS TM, or DBCTL, or batch (DBB or DLI) procedures for failure management, restart and recovery are required.

A failure of IMS to connect to the CF structure

During initialization, IMS is unable to connect to the OSAM or VSAM cross invalidate structures, CF initialization fails. Data sharing also stops.

The following messages are displayed for an OSAM XI structure connection failure:

```
DFS3381E - CF INITIALIZATION FAILED, DATA=10 00000008 00000000 08D08000
DFS3384I - DATA SHARING STOPPED. IM1A (where IM1A is the IMSID)
```

If the OSAM XI connection fails, VSAM XI connection is not attempted.

The following messages are displayed for a VSAM XI structure connection failure:

```
DFS3381E - CF INITIALIZATION FAILED, DATA=40 00000008 00000000 08D08000.
DFS3384I - DATA SHARING STOPPED. IM1A
```

IMS continues with initialization and, when the CF resource becomes available, the connection is retried.

Online IMS subsystem abends

When an IMS subsystem ends abnormally, a display of the VSAM or OSAM XI structures shows how that particular IMS was connected. To bring up the structure display, use the following command:

```
D XCF,STRUCTURE,STRNAME=IMOA_VSAM
```

Also the IRLM lock structure can be displayed by issuing the command:

```
D XCF,STRUCTURE,STRNAME=IMOA_IRLM
```

A display of the IRLM associated with the failed IMS system can be displayed via these commands:

```
F IM1AIRLM,STATUS
F IM1AIRLM,STATUS,ALLI
F IM1AIRLM,STATUS,ALLD
```

Starting IMS in a data sharing environment after failures

Several items are worth noting when starting an IMS in a data sharing group environment:

- Automatic restart management can be used to restart IMS TM, DBCTL, DCCTL and XRF. ARM can be used for the emergency restart of CICS to resolve any items in doubt between CICS and DBCTL. ARM can also be used by the restart of DB2 to resolve any in-doubt conditions between DB2 and IMS.
- The IRLM component must be started before IMS, but once the IRLM is started, IMS connects to the IRLM without intervention. If IRLM has not completed its initialization when the IMS subsystem tries to connect to it, IMS issues a DFS039A message that says the IRLM is not active, and asks the operator to reply to the message: retry, cancel, or dump.
- Lock rejects may cause other subsystems to be affected until recovery is completed for the failing IMS subsystem holding locks. FDBR can be used to release retained locks as soon as possible.

Cold start after an IMS subsystem abend

If /ERESTART fails, batch backout is required to release retained locks for active units of work (UOW). A UOW is either an IMS TM transaction, a DBCTL thread, or a BMP sync interval. Batch backout is followed by a cold start of the failing IMS using /ERESTART COLDSYS. This releases modify locks and releases DBRC authorizations. Batch backout must be used for batch jobs if they are not using dynamic backout or if dynamic backout fails. The following command may be required:

```
F irlmproc,PURGE,imsname
```

This is needed if:

- The /ERESTART is done on another unconnected IRLM.
- In-flight units of work have been backed out by batch backout.

If an F irlmproc,PURGE,imsname command is issued before a backout, locks are released and could cause data integrity problems, since the updates protected by those locks would not be backed out. The following DBRC commands may be required to release authorizations:

```
CHANGE.SUBSYS SSID(subsysname) STARTRCV
CHANGE.SUBSYS SSID(subsysname) ENDRECOV
```

If the DBRC CHANGE.SUBSYS commands are issued before backout is completed, authorizations will be released, causing data integrity problems.

8.9.8 Failure of the IRLM subsystem component

Figure 8-3 presents the state of the IRLM lock structure before an IRLM failure:

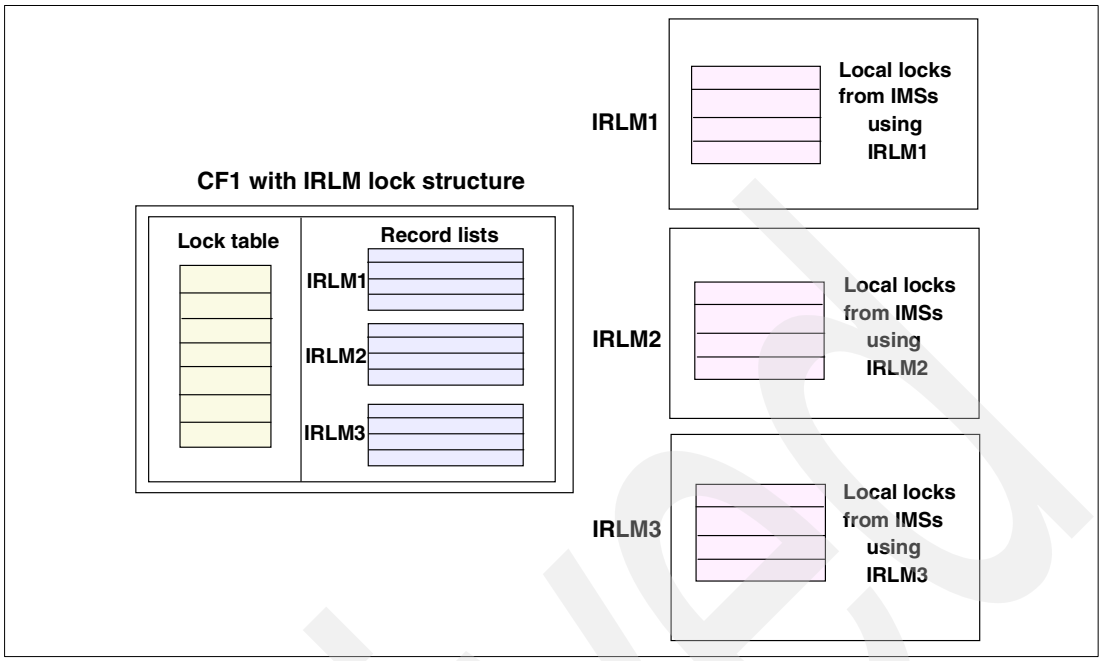

Figure 8-3 Locking environment before an IRLM failure

When an IRLM fails, several actions occur:

- Batch jobs in progress end abnormally with a U3303 abend while the associated IMS TM and DBCTL subsystems are quiesced.
- In-flight work is abended or backed out.
- No new PSBs (transactions) are scheduled by IMS.
- Other IRLMs are notified of the situation through XCF.
- Modify locks are read and retained by other IRLMs.
- Subsystems using the surviving IRLMs are unaffected except for requests of those retained locks. Such requests result in a U3303 abend or status BA or BB if the required logic is coded within applications.

Figure 8-4 shows the IRLM local table and CF lock structure after the failure of IRLM2.

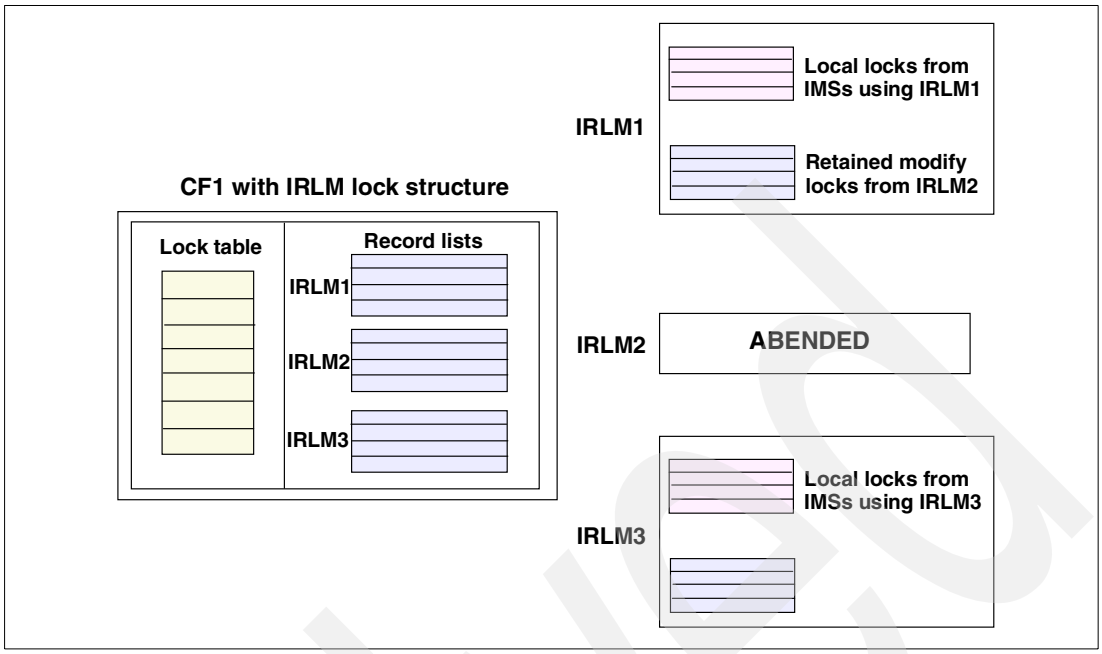

Figure 8-4 Locking environment after an IRLM failure

- The failed IRLM must be restarted.
- Batch backout is required to free retained locks associated with batch jobs. These batch job backouts can be done using any IRLM in the data sharing group. Similarly, the IMS TM or DBCTL subsystem may be terminated and restarted using any IRLM in the data sharing group; however, it should be faster to start the IRLM on the same OS/390 and reconnect to it.
- When the IRLM is reconnected to its IMS (which had been quiesced) through the use of the following command, the IMS subsystem quiesce state is reset:

  ```
  F IMSPROC,RECONNECT
  ```

  There is no automatic reconnection when the IRLM is restarted.

If Fast Path DEDB AREAs exist, each AREA must be started. This is not a full function database requirement.

For more detailed information, refer to 8.13.3, “Example IRLM failure and recovery scenario” on page 195.

8.9.9 CICS component failures

A CICS application owning region (AOR that fails must reconnect to the same IMS DBCTL) region after a restart. Restart does not need to be on the same OS/390 as the original. To resolve items in doubt, a CICS emergency restart is required.

8.10 IMS Extended Recovery Facility and data sharing

The IMS Extended Recovery Facility (XRF) feature is fully supported in a data sharing environment. XRF is used to provide protection against hardware platform, OS/390, and IMS failures to support controlled change over any of those resources and to provide high availability to the network of user terminals (each terminal belonging to one IMS).

The IMS XRF alternate is an IMS subsystem that connects to IRLM and as such is included in the 255-way subsystem limitation. If you are using IRLM Version 2.1 in a data sharing environment and the alternate subsystem is available, there are no required tasks to maintain continued data availability that belong to what is normally required for XRF operation.

During takeover, the alternate subsystem in an XRF complex backs out transactions that were in process when takeover was initiated.

For DEDBs on shared DASD, the alternate subsystem performs forward recovery. For DL/I databases on shared DASD, the alternate subsystem reads the database block to which the change log record refers, and then it backs out the change. If I/O toleration has completed, it writes the block back. If I/O toleration is still active, the alternate subsystem creates EEQEs for these blocks and places them in virtual buffers residing above the 16-megabyte line.

Transactions (running on the new active IMS) that access these blocks prior to the end of I/O toleration use these virtual buffers instead of the blocks on DASD. DBRC records the presence of the EEQEs and the IRLM propagates this information to all sharing subsystems. Therefore, until I/O toleration ends, these databases can continue to be shared, but the presence of the EEQE prevents a sharing subsystem from accessing these particular blocks.

8.11 IMS Remote Site Recovery and data sharing

The IMS Remote Site Recovery (RSR) feature is fully supported in data sharing environments. In a data sharing environment, IMS maintains database integrity for shared, covered databases at the tracking site. Active subsystems perform the necessary activities for support of tracking databases participating in block level data sharing.

The remote system is not part of a data sharing group so does not effect the 255-way limitation since it is a single system.

These are log merge requirements in a data sharing environment:

- The remote system receives log stream data from each subsystem.
- Recovery level tracking requires the use of the change accumulation feature.
- Database level tracking merges the database updates via the RSR log router, so change accumulation is not required.
- Time stamps in the log records are used to determine the update sequence in the shadow databases.

In data sharing environments where XRC transmits DB2 data to a remote site, RSR provides support for coordinating disaster recovery for IMS and DB2.

XRC is a separate hardware function that tracks DB2 logs. Likewise, RSR tracks IMS logs. RSR coordinates with XRC so that both IMS and DB2 logs are tracked and can be synchronized. To request XRC tracking, in the DFSRSRxx PROCLIB member, you specify the session ID of the XRC session that is shadowing the DB2 volumes. As a result, the RSR tracking subsystem processes and synchronizes with XRC processing. This support allows IMS and DB2 logs to be synchronized and ensures that IMS and DB2 databases are consistent if a disaster occurs.

8.12 General data sharing recovery recommendations

After any IMS TM, or DBCTL failure:

- Perform emergency restart as soon as possible to release retained locks and limit the number of lock rejects.
- Obtain and populate two CFs at least for your data sharing environment distributing the placement of OSAM and VSAM structures on one CF and the IRLM structure on the other.
- Define your CFRM policy with alternative CFs for structure rebuilds and install the necessary environment so recovery from CF failures can be automatic.
- Use FDBR.
- The OS/390 ARM feature is also recommended to improve data access and application availability.

After failures of batch jobs that update databases:

- Back out as soon as possible to release locks to limit the number of lock rejects.

8.13 Sample failure recovery scenarios

The following recovery scenarios are presented in detail:

- A Coupling Facility link failure to a CF containing a VSAM buffer invalidate structure.
- An IMS subsystem control region failure.
- An IRLM failure.
- Example of an operator controlled structure rebuild.

These are presented as a guide of the processing that occurs in a BLDS environment when major components fail.

8.13.1 A Coupling Facility link failure recovery scenario

Link failures that cause a loss of connectivity can occur in multiple ways:

- A patch panel connection could be incorrectly setup or reinstalled.
- A plug could be removed by mistake.
- The link could be severed because of maintenance accidents.
- The channel path being forced offline by mistake.

We will investigate, using the output of action and display commands, what to expect when the connection to a VSAM buffer invalidate cache structure fails. Figure 8-5 illustrates the configuration that was used in this example.

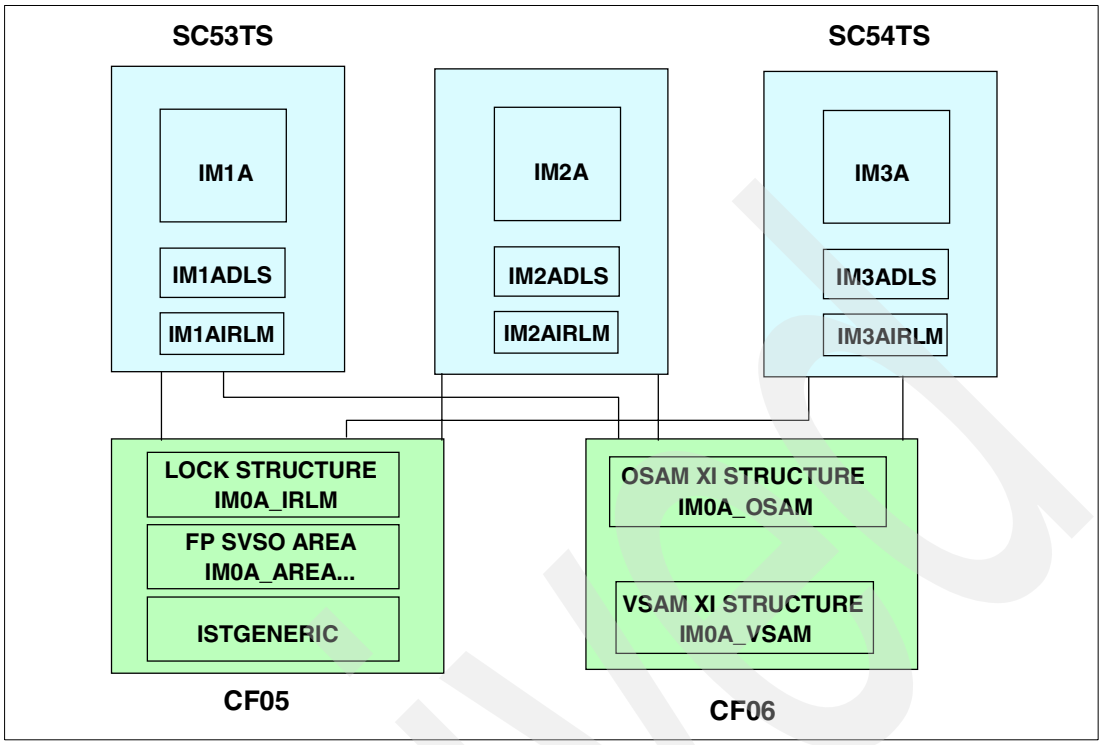

Figure 8-5 Sample configuration for our CF link failure discussion

In our example, the VSAM buffer invalidate structure IM0A_VSAM resides in CF CF06 with link connections to three IMS subsystems: IM1A, IM2A and IM3A. We will be simulating a link failure from IM1A to this VSAM buffer invalidate structure.

Example 8-1 displays the output of from the following command before the link to this structure is varied offline:

```
D XCF,STRUCTURE,STRNAME=IM0A_VSAM
```

Example 8-1 Output of D XCF,STRUCTURE,STRNAME=IM0A_VSAM

```
IXC360I  15.49.38  DISPLAY XCF 635
STRNAME: IM0A_VSAM
 STATUS: ALLOCATED
  POLICY SIZE    : 1536 K
  POLICY INITSIZE: N/A
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : DISABLED
  PREFERENCE LIST: CF06 CF05
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY

 ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 11/18/2002 00:29:47
  CFNAME         : CF06
  COUPLING FACILITY: 002064.IBM.02.000000010ECB
                     PARTITION: F   CPCID: 00
  ACTUAL SIZE    : 1536 K
  STORAGE INCREMENT SIZE: 256 K
```

```
  PHYSICAL VERSION: B88C0C06 A6C42361
  LOGICAL  VERSION: B88C0C06 A6C42361
  SYSTEM-MANAGED PROCESS LEVEL: 8
  DISPOSITION    : DELETE
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 3

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 IXCL00850001     01 00010073 SC53     IM1ADLS  00B2 ACTIVE
 IXCL00850002     02 00020041 SC47     IM2ADLS  03FD ACTIVE
 IXCL00850003     03 0003001A SC54     IM3ADLS  03F9 ACTIVE
```

The link from IM1A on SC53TS to CF CF06 (containing the VSAM and OSAM buffer invalidate (XI) structures) is forced offline, via the following command:

```
CF CHP(FA,FB),OFFLINE,FORCE
```

FA and FB are the channel paths from SC53TS to CF CF06. When this command completes IM1A drops out of data sharing and issues the following message:

```
DFS3384I DATA SHARING STOPPED
```

Transactions end abnormally with U3303s abends, as presented in Example 8-2. Of course if automatic rebuilds to CF05 were successful, processing continues.

Example 8-2 Forcing a link failure on the CF link to CF CF06

```
CF CHP(FA,FB),OFFLINE,FORCE

IXL126I CONFIG WILL FORCE OFFLINE LAST CHP(FB) TO CF CF06
IXL127A REPLY CANCEL OR CONTINUE
R 09,CONTINUE
IEE600I REPLY TO 09 IS;CONTINUE

IEE503I CHP(FA),OFFLINE
IEE503I CHP(FB),OFFLINE
IEE712I CONFIG   PROCESSING COMPLETE
IXC518I SYSTEM IM1A NOT USING 197
              CF ND02...
                                 PARTITION: 0    CPCID: FA
              REASON: CONNECTIVITY LOST.
              REASON FLAG: 13300002.
DFS554A MPP14    00004 MPP      HPC$M$00(1) HPCSTCL4 000,3303
                 02/326 11:25:12 IM1A
DFS554A MPP16    00006 MPP      HPC$M$00(1) HPCSTCL6 000,3303
                 02/326 11:25:12 IM1A
DFS554A MPP15    00005 MPP      HPC$M$00(1) HPCSTCL5 000,3303
                 02/326 11:25:12 IM1A
DFS3384I DATA SHARING STOPPED   IM1A
```

The following command will cause a rebuild of this VSAM XI structure on the other CF:

```
SETXCF START,REBUILD,STRNAME=IM0A_VSAM,LOCATION=OTHER
```

For more details, refer to 8.13.4, “Example of an operator initiated rebuild” on page 198.

8.13.2 IMS subsystem failure and recovery scenario

This is an example of a failing IMS system and its recovery, accompanied with useful commands and status information. In this situation let's assume that IM3A and with it, IM3ADLS, as shown in Figure 8-6, have failed for some reason.

In the case of an IMS abend or other failures that terminate it, modify locks associated with this IMSs database updates are retained in the IRLM lock structure record list. Other locks are released but requestors of retained locks end abnormally. An IMS /ERESTART or FDBR execution is required to free retained locks. This can be executed on any OS/390 image in the IMS data sharing group.

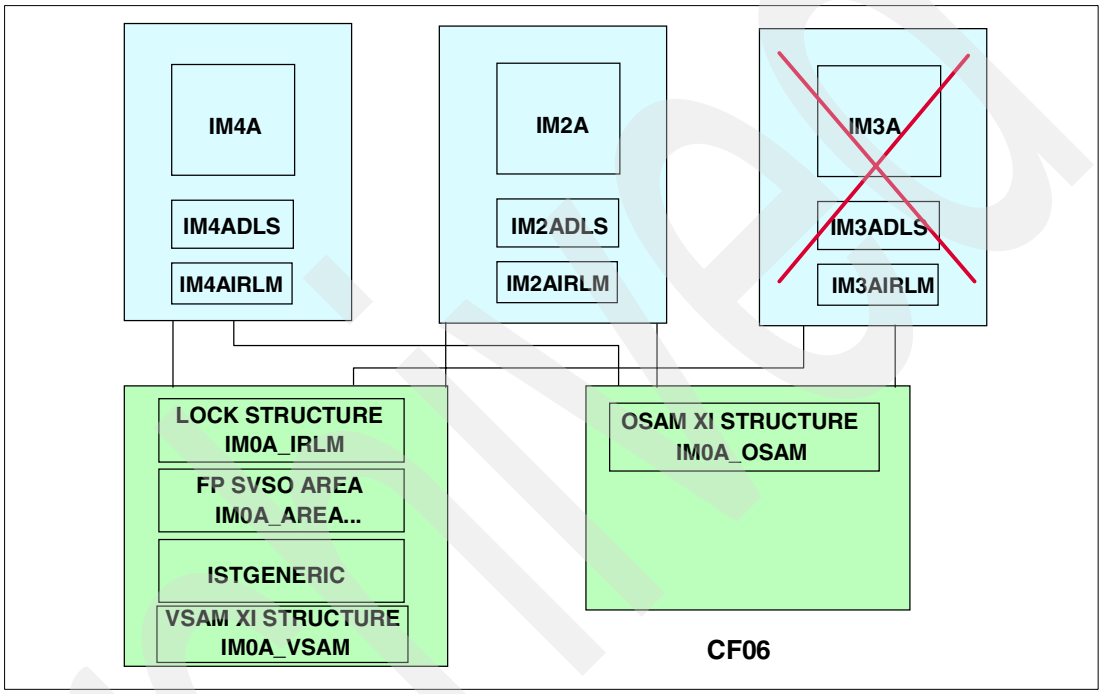

Figure 8-6 Configuration for the IMS subsystem failure scenario

Example 8-3 presents the status of a VSAM XI structure before the IM3A failure within SC54TS in response to the XCF display command:

```
D XCF,STRUCTURE,STRNAME=IM0A_VSAM
```

Example 8-3 Display of the VSAM structure

```
D XCF,STR,STRNAME=IM0A_VSAM
IXC360I  19.37.01  DISPLAY XCF 246
STRNAME: IM0A_VSAM
 STATUS: ALLOCATED
  POLICY SIZE    : 1536 K
  POLICY INITSIZE: N/A
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : DISABLED
  PREFERENCE LIST: CF06     CF05
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```

```
  ACTIVE STRUCTURE
----------------
  ALLOCATION TIME: 11/23/2002 09:22:20
  CFNAME         : CF05
  COUPLING FACILITY: 009672.IBM.02.000000050822
                     PARTITION: F   CPCID: 00
  ACTUAL SIZE    : 1280 K
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: B892CC62 CFCBE801
  LOGICAL  VERSION: B892CC62 CFCBE801
  SYSTEM-MANAGED PROCESS LEVEL: 8
  DISPOSITION    : DELETE
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
    # CONNECTIONS  : 3
 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ---------
 IXCL00850002     02 00020041 SC47     IM2ADLS  03FD ACTIVE
 IXCL00850003     03 0003001A SC54     IM3ADLS  03F9 ACTIVE
 IXCL00850004     04 00040014 SC67     IM4ADLS  00B8 ACTIVE
```

This display shows that IM3ADLS is connected to this VSAM structure. Example 8-4 presents the use of the OS/309 command:

```
CANCEL IM3ACTL
```

It causes the termination of this IMS subsystem. The detail of message output associated with this shutdown of IM3ACTL has been reduced for clarity:

Example 8-4 Cancel of IM3ACTL

```
C IM3ACTL
IEE301I IM3ACTL          CANCEL COMMAND ACCEPTED
DFS616I OLDS BUFFERS SUCCESSFULLY PURGED  IM3A
DFS629I IMS STM TCB ABEND - SYS 222           IM3A
DFS0617I RDS BUFFERS HAVE BEEN SUCCESSFULLY PURGED  IM3A
IEF404I IM3ADBRC - ENDED - TIME=20.57.41 - ASID=03F5 - SC54
DFS629I IMS DLI TCB ABEND - IMS 0150          IM3A
DFS629I IMS DYS TCB ABEND - IMS 4095-DLS      IM3A
.....
DFS603I IMS DLS CLEANUP ( EOT ) COMPLETE FOR JS IM3ADLS .IM3ADLS .
    ,RC=00
DFS629I IMS TCO TCB DUMP  - SYS 222           IM3A
DFS629I IMS DCC TCB ABEND - SYS 222           IM3A
IST804I CLOSE IN PROGRESS FOR IMS3A OPENED BY IM3ACTL
IST400I TERMINATION IN PROGRESS FOR APPLID IMS3A
IST805I VTAM CLOSE COMPLETE FOR IMS3A
DFS629I IMS XFP TCB ABEND - SYS 222           IM3A
IEF404I IM3ADLS - ENDED - TIME=20.57.43 - ASID=03FA - SC54
DFS629I IMS SQ2 TCB ABEND - IMS 4095          IM3A
.....
DFS4452I RESOURCE CLEANUP STARTING FOR IM3A IM4A
DFS4453I LTERM MSGQ MESSAGES UNLOCKED FOR IM3A IM4A
DFS4453I LTERM EMHQ MESSAGES UNLOCKED FOR IM3A IM4A
DFS4452I RESOURCE CLEANUP COMPLETE FOR IM3A IM4A
CSL3102I SCI CLEANUP HAS COMPLETED FOR MEMBER IM3A     IM3ASC
DFS629I IMS RCF TCB ABEND - SYS 222           IM3A
.....
AVM005A. I/O PREVENTION IS COMPLETE FOR
        SUBSYSTEM IM3A, FAILING ACTIVE ELEMENT OF
        RSE IM3A
```

```
IEC214I 614-04,IFG0202E,IM3ACTL,IM3ACTL,DFSOLP02,35C1,TOTIMM,
IMSPSA.IM3A.OLP02
DFS627I IMS RTM CLEANUP ( EOT ) COMPLETE FOR JS IM3ACTL .IM3ACTL
.IEFPROC ,RC=00
IEF450I IM3ACTL IM3ACTL - ABEND=S222 U0000 REASON=00000000 677
```

After the IMS failure, only IM2A and IM4A DLI/SAS tasks are connected to the VSAM structure as shown, as seen in Example 8-5.

Example 8-5 Display of VSAM structure after IM3A is cancelled.

```
D XCF,STR,STRNAME=IM0A_VSAM
IXC360I  21.10.07  DISPLAY XCF 714
STRNAME: IM0A_VSAM
 STATUS: ALLOCATED
.....
MAX CONNECTIONS: 32
 # CONNECTIONS  : 2

CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
---------------- -- -------- -------- -------- ---- ----------------
IXCL00850002     02 00020042 SC47     IM2ADLS  03FD ACTIVE
IXCL00850004     04 00040014 SC67     IM4ADLS  00B8 ACTIVE
```

Example 8-6 presents IM3A_IRLM as active in response to a XCF display command:

D XCF,STRUCTURE,STRNAME=IM0A_IRLM

Example 8-6 Display of the IRLM structure

```
D XCF,STR,STRNAME=IM0A_IRLM
IXC360I  21.12.04  DISPLAY XCF 721
STRNAME: IM0A_IRLM
 STATUS: ALLOCATED
# CONNECTIONS  : 3

CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
---------------- -- -------- -------- -------- ---- ----------------
IM0AIRLM$IR0A002 01 00010025 SC47     IM2AIRLM 00A0 ACTIVE
IM0AIRLM$IR0A003 03 0003000D SC54     IM3AIRLM 03F4 ACTIVE
IM0AIRLM$IR0A004 04 0004000A SC67     IM4AIRLM 00BA ACTIVE
```

Example 8-7 shows the response to command:

F IM3AIRLM,STATUS,ALLD

Example 8-7 Response to F IM3AIRLM,STATUS,ALLD command

```
DXR102I IR0A003 STATUS
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS    RET_LKS   IRLMID  IRLM_NAME   IRLM_LEVL
        IM2A     UP              0    002      IR0A          009
        IM3A     DOWN            1    003      IR0A          009
        IM4A     UP              0    004      IR0A          009
    DXR102I End of display
```

Note that IM2A status with respect to IRLM is ‘DOWN’

Example 8-8 shows the response to command:

F IM3AIRLM,STATUS,ALLI

Example 8-8 Response to F IM3AIRLM,STATUS,ALLI

```
F IM3AIRLM,STATUS,ALLI
DXR103I IROA003 STATUS
        IRLMS PARTICIPATING IN DATA SHARING GROUP FUNCTION LEVEL=021
        IRLM_NAME IRLMID STATUS LEVEL SERVICE MIN_LEVEL MIN_SERVICE
          IROA     004     UP    022  PQ52360    012      PN90337
          IROA     002     UP    021  PQ48823    012      PN90337
          IROA     003     UP    022  PQ52360    012      PN90337
    DXR103I End of display
```

As seen in Example 8-9, IM3ACTL is started with an /ERE. Processing is completed and the XCF connect request for the VSAM structure was successful.

Example 8-9 Start of IM3ACTL

```
S IM3ACTL
$HASP373 IM3ACTL  STARTED
IEF403I IM3ACTL - STARTED - TIME=21.17.52 - ASID=03FC - SC54
.....
DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB  MEMBER = DFSPB03Q  IMS
DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB  MEMBER = DFSVSM0S  IM3A
DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB  MEMBER = DFSFDR03  IM3A
DFS4190I IMS SYSTEM IS FDR CAPABLE IM3A
.....
IEF695I START IM3ADLS  WITH JOBNAME IM3ADLS  IS ASSIGNED TO USER STC
   , GROUP SYS1
$HASP373 IM3ADLS  STARTED
IEF403I IM3ADLS - STARTED - TIME=21.17.59 - ASID=03FB - SC54
$HASP373 IM3ADBRC STARTED
IEF403I IM3ADBRC - STARTED - TIME=21.17.59 - ASID=03F5 - SC54
DFS3613I - STM TCB INITIALIZATION COMPLETE  IM3A
.....
IXL014I IXLCONN REQUEST FOR STRUCTURE IM0A_OSAM 971
WAS SUCCESSFUL.  JOBNAME: IM3ADLS ASID: 03FB
CONNECTOR NAME: IXCLO0840001 CFNAME: CF06
DFS3382I DL/I CF INITIALIZATION COMPLETE   IM3A
IXL014I IXLCONN REQUEST FOR STRUCTURE IM0A_VSAM 972
WAS SUCCESSFUL.  JOBNAME: IM3ADLS ASID: 03FB
CONNECTOR NAME: IXCLO0850001 CFNAME: CF05
DFS228I - DLS REGION INITIALIZATION COMPLETE  IM3A
....
R 2741,/ERE.
IEE600I REPLY TO 2741 IS;/ERE.
DFS058I 21:26:47 ERESTART COMMAND IN PROGRESS   IM3A
....
2743 DFS996I *IMS READY*  IM3A
```

In Example 8-10 the XCF display of the VSAM structure shows that IM3ADLS is connected to the VSAM XI structure again.

Example 8-10 Display of the IM0A_VSAM structure after IM3A is restarted

```
D XCF,STRUCTURE,STRNAME=IM0A_VSAM
.....
CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
---------------- -- -------- -------- -------- ---- ----------------
IXCLO0850001     01 00010076 SC54     IM3ADLS  03FB ACTIVE
IXCLO0850002     02 00020042 SC47     IM2ADLS  03FD ACTIVE
IXCLO0850004     04 00040014 SC67     IM4ADLS  00B8 ACTIVE
```

8.13.3 Example IRLM failure and recovery scenario

The same configuration as in Figure 8-6 on page 191 will be used for this IRLM failure and recovery scenario.

The following scenario simulates a failure of IM3AIRLM on IMS IM3A after all IMSs are up and running transactions. The failing IRLM is then restarted and reconnects to the data sharing group. After that, an F IM3ACTL,RECONNECT command is issued and data sharing resumes.

Example 8-11 lists the output of the command:

```
D XCF,STRUCTURE,STRNAME=IM0A_OSAM
```

Example 8-11 Output of a display command for the OSAM structure before an IRLM failure

```
D XCF,STRUCTURE,STRNAME=IM0A_OSAM
IXC360I  21.43.48  DISPLAY XCF 257
STRNAME: IM0A_OSAM
.....
MAX CONNECTIONS: 32
 # CONNECTIONS  : 3

CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
---------------- -- -------- -------- -------- ---- ----------------
IXCL00840001     01 00010079 SC54     IM3ADLS  03FB ACTIVE
IXCL00840002     02 00020042 SC47     IM2ADLS  03FD ACTIVE
IXCL00840004     04 00040014 SC67     IM4ADLS  00B8 ACTIVE
```

Example 8-12 lists the output of command:

```
D XCF,STRUCTURE,STRNAME=IM0A_IRLM
```

Example 8-12 Display of the IRLM structure before the IM3AIRLM is terminated

```
D XCF,STRUCTURE,STRNAME=IM0A_IRLM
IXC360I  21.46.32  DISPLAY XCF 289
STRNAME: IM0A_IRLM
 STATUS: ALLOCATED
.....
MAX CONNECTIONS: 7
 # CONNECTIONS  : 3

CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
---------------- -- -------- -------- -------- ---- ----------------
IM0AIRLM$IR0A002 01 00010025 SC47     IM2AIRLM 00A0 ACTIVE
IM0AIRLM$IR0A003 03 0003000D SC54     IM3AIRLM 03F4 ACTIVE
IM0AIRLM$IR0A004 04 0004000A SC67     IM4AIRLM 00BA ACTIVE
```

Before IM3AIRLM is cancelled, three BMPs are run, requesting database resources which will create lock requests before they all go into WTOR forced waits.

Example 8-13 Status of IM3AIRLM before it is terminated

```
F IM3AIRLM,STATUS
DXR101I IR0A003 STATUS SCOPE=NoDISC
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS       UNITS      HELD    WAITING    RET_LKS
        IM3A     UP               3         9          0          0
    DXR101I End of display
```

The cancellation of IM3AIRLM is shown in Example 8-14. Some of the messages have been removed or placed in order to demonstrate the shutdown process more clearly.

Example 8-14 Termination of IM3AIRLM

```
C IM3AIRLM
DXR122E IROA003 ABEND UNDER IRLM TCB/SRB IN MODULE DXRRL020 ABEND
CODE=S222
IEE301I IM3AIRLM          CANCEL COMMAND ACCEPTED
DFS2011I IRLM FAILURE - IMS QUIESCING IM3A
.....
DFS3705I AREA=AREADI01 DD=DISTDD01 CLOSED IM3A
DFS2500I DATASET  DISTDD01 SUCCESSFULLY DEALLOCATED IM3A
DFS2823I AREA AREADI01 DISCONNECT FROM STR: IMOA_AREADI01A
SUCCESSFUL IM3A
DFS2574I AREA=AREADI01 STOPPED                          IM3A
.....
IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE IMOA_IRLM, 452
CONNECTOR IMOAIRLM$IROA003:
IXL031I CONNECTOR CLEANUP FOR LOCK STRUCTURE IMOA_IRLM, 453
CONNECTOR IMOAIRLM$IROA003, HAS COMPLETED.
.....
DXR137I IROA002 GROUP STATUS CHANGED. IROA 003 HAS BEEN DISCONNECTED
FROM THE DATA SHARING GROUP
DXR137I IROA004 GROUP STATUS CHANGED. IROA 003 HAS BEEN DISCONNECTED
FROM THE DATA SHARING GROUP
.....
IXL020I CLEANUP FOR LOCK STRUCTURE IMOA_IRLM, 483
CONNECTION ID 03, STARTED BY CONNECTOR IMOAIRLM$IROA004
IXL021I GLOBAL CLEANUP FOR LOCK STRUCTURE IMOA_IRLM, 484
CONNECTION ID 03, BY CONNECTOR IMOAIRLM$IROA004 HAS COMPLETED
IXL020I CLEANUP FOR LOCK STRUCTURE IMOA_IRLM, 656
CONNECTION ID 03, STARTED BY CONNECTOR IMOAIRLM$IROA002
IXL022I LOCAL CLEANUP FOR LOCK STRUCTURE IMOA_IRLM, 485
CONNECTION ID 03, BY CONNECTOR IMOAIRLM$IROA004 HAS COMPLETED.
IXL021I GLOBAL CLEANUP FOR LOCK STRUCTURE IMOA_IRLM, 657
CONNECTION ID 03, BY CONNECTOR IMOAIRLM$IROA002 HAS COMPLETED.
.....
DFS3304I IRLM LOCK REQUEST REJECTED. PSB=PSBBA    DBD=CUSTDB
JOBNAME=DEDLOFF4 RGN=001 SUBSYSTEM=IM3A     IM4A
DFS0781I ABEND 3303 IN DFSECP20+S203+SP53+810+06/17/02PQ56609 A IM4A
DFS554A DEDLOFF4 00001 G        PSBBA   (2)          000,3303
          2002/329 22:06:36 IM4A
DFS552I BATCH   REGION DEDLOFF4 STOPPED ID=00001 TIME=2206  IM4A
IEF450I DEDLOFF4 G STEP01 - ABEND=S000 U3303 REASON=00000000 489
        TIME=22.06.37
.....
DXR121I IROA003 END-OF-TASK CLEANUP SUCCESSFUL - HI-CSA   2710K  -
HI-ACCT-CSA    132K
IEF450I IM3AIRLM IM3AIRLM - ABEND=S222 U0000 REASON=00000000 459
        TIME=22.06.37
$HASP395 IM3AIRLM ENDED
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=03F4.
.....
```

There are a few important points to mention from the previous message displays:

- Via message DFS2011I IRLM FAILURE - IMS QUIESCING, IM3A drops out of data sharing because it has lost its IRLM. The lock manager status exit is given control and

detects that the IRLM is terminating. This causes the discontinuance of the use of all databases.

IMS pseudo abends all currently scheduled transactions with abend U3303. Full-function and Fast Path databases participating in data sharing (share level 1, 2, or 3) are closed and their authorization is discontinued. All further transaction scheduling, except Fast Path, is prohibited.

- Fast Path AREAs are closed, deallocated, disconnected from the SVSO structure and stopped.
- Connector cleaning occurs from the IRLM lock structure to the IRLM with ID=003 (IM3AIRLM).
- The IRLM image with ID=003 has been disconnected from the data sharing group.
- Local and global lock structure cleanup from the IRLM instances with ID=002 and ID=004 completes.
- DEDLOFF4 that was waiting on the database record lock associated with a GHU call for a CUSTOMER segment within the CUSTDB has to be terminated with an ABU3303, because we don't know when that retained lock will be freed by /ERE or FDBR execution.

Example 8-15 presents the output of command:

```
D XCF,STRUCTURE,STRNAME=LT01
```

This shows that IM3AIRLM is in a FAILED-PERSISTENT state.

Example 8-15 Display of IRLM structure

```
D XCF,STR,STRNM=IM0A_IRLM
IXC360I  22.17.10  DISPLAY XCF 515
STRNAME: IM0A_IRLM
 STATUS: ALLOCATED
.....
MAX CONNECTIONS: 7
 # CONNECTIONS  : 3
CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
---------------- -- -------- -------- -------- ---- ----------------
IM0AIRLM$IR0A002 01 00010025 SC47     IM2AIRLM 00A0 ACTIVE
IM0AIRLM$IR0A003 03 0003000D SC54     IM3AIRLM 03F4 FAILED-PERSISTENT
IM0AIRLM$IR0A004 04 0004000A SC67     IM4AIRLM 00BA ACTIVE
```

A display of another IRLM (IM2AIRLM) shows IM3AIRLM not active in the OS/390 SC54 image. Also in this response to the status of IM2AIRLM, it shows IM3A in SFAIL status.

Example 8-16 Information obtained from a status display of an active IRLM

```
RO *ALL,F IM2AIRLM,STATUS,ALLD
SC54     RESPONSES -----------------------
  IEE341I IM2AIRLM          NOT ACTIVE
.....
DXR102I IR0A002 STATUS
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS    RET_LKS   IRLMID  IRLM_NAME   IRLM_LEVL
        IM2A     UP              0    002      IR0A          009
        IM3A     SFAIL           1    003      IR0A          009
        IM4A     UP              0    004      IR0A          009
    DXR102I End of display
```

In Example 8-17, IRLM has been restarted and completes initialization but does not connect until the modify IMS reconnect message is issued.

Example 8-17 Start of IM3AIRLM

```
S IM3AIRLM
 $HASP100 IM3AIRLM ON STCINRDR
 IEF695I START IM3AIRLM WITH JOBNAME IM3AIRLM IS ASSIGNED TO USER STC
    , GROUP SYS1
 $HASP373 IM3AIRLM STARTED
DXR117I IROA003 INITIALIZATION COMPLETE
DXR172I IROA003 IMOAIRLMIROA003 ARM READY COMPLETED. 572
.....
```

As shown in Example 8-18, the IRLM joins the data sharing group and connects to the lock structure.

Example 8-18 Reconnecting to the IRLM from IM3A

```
F IM3ACTL,RECONNECT
DXR141E IRLM THE LOCK TABLE CF05 WAS ALLOCATED IN A VOLATILE
    FACILITY
    DXR132I IRLM SUCCESSFULLY JOINED THE DATA SHARING GROUP. GLOBAL
            INITIALIZATION IS COMPLETE
    DFS626I - IRLM RECONNECT COMMAND SUCCESSFUL. IM3A
```

8.13.4 Example of an operator initiated rebuild

The operator can initiate a rebuild of any structure using the SETXCF START command. In this example we are running with a VSAM XI structure on CF CF05. We do an operator initiated rebuild and move the VSAM XI structure to CF CF06 and data sharing then resumes. The rebuild process can be performed for both planned and unplanned outages. Figure 8-7 on page 198 presents the status of our test configuration before the rebuild process of the VSAM structure.

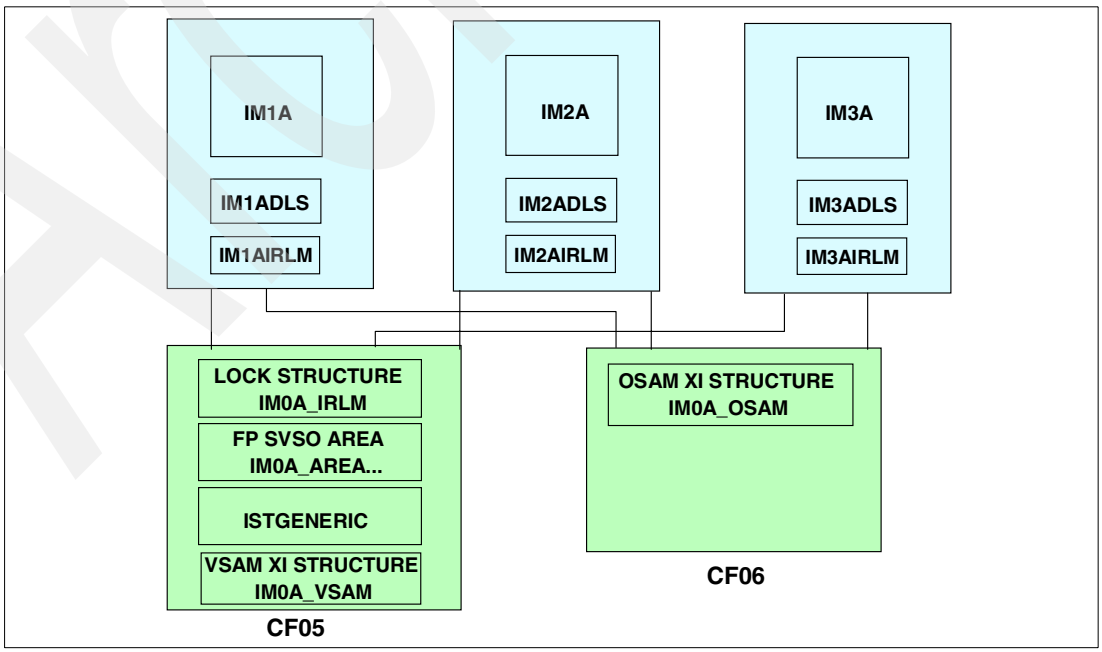

Figure 8-7 Configuration in place before operator rebuild of VSAM structure

As shown in Figure 8-19, the following command is issued to initiate the rebuild:

```
SETXCF START,REBUILD,STRNAME=IMOA_VSAM,LOCATION=OTHER
```

Example 8-19 Message output of the rebuild command process

```
-SETXCF START,REBUILD,STRNAME=IMOA_VSAM,LOCATION=OTHER
 IXC521I REBUILD FOR STRUCTURE IMOA_VSAM
 HAS BEEN STARTED
 IXC367I THE SETXCF START REBUILD REQUEST FOR STRUCTURE
 IMOA_VSAM WAS ACCEPTED.
 IXC526I STRUCTURE IMOA_VSAM IS REBUILDING FROM
 COUPLING FACILITY CF05 TO COUPLING FACILITY CF06.
 REBUILD START REASON: OPERATOR INITIATED
 INFO108: 00000050 00000000.
 IXC521I REBUILD FOR STRUCTURE IMOA_VSAM
 HAS BEEN COMPLETED
```

After the rebuild is complete, the VSAM XI structure has been moved to CF CF06, as shown in Example 8-20.

Example 8-20 Display of the status of the VSAM structure after the rebuild

```
-D XCF,STR,STRNM=IMOA_VSAM
  IXC360I  22.07.24  DISPLAY XCF 001
  STRNAME: IMOA_VSAM
   STATUS: ALLOCATED
.....
  CFNAME          :CF06
.....
MAX CONNECTIONS: 32
   # CONNECTIONS  : 3

  CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
  ---------------- -- -------- -------- -------- ---- ----------------
  IXCL00850001     01 00010074 SC53     IM1ADLS  00C1 ACTIVE
  IXCL00850002     02 00020041 SC47     IM2ADLS  03F9 ACTIVE
  IXCL00850003     03 0003001A SC54     IM3ADLS  03F9 ACTIVE
```

8.14 Problem determination

This section presents tips to help you identify the causes of problems you may encounter.

8.14.1 Generalized problem analysis process

When presented with a failure of part of the Parallel Sysplex, the following steps may help you to determine where and what the problem is:

- Display the status of the CF and the CF links using the command D CF.
- Use the command D XCF,CF,CFNAME=ALL to make sure all of the systems are connected and what structures are allocated on what CFs.
- Display the structure allocations using the command D XCF,STRUCTURE.
- Display the status of a structure using the command D XCF,STRUCTURE,STRNAME=.
- Check to see if the following message is issued:

  ```
  DFS3384I DATA SHARING STOPPED
  ```

- If an IMS is connected to the structure, the following message is issued:

  ```
  DFS3382I CF INITIALIZATION COMPLETE
  ```

- Check to see if the following IRLM(s) connect message appears:

  ```
  DXR141E IRLM THE LOCK TABLE LTO1 WAS ALLOCATED IN A VOLATILE FACILITY
  DXR132I IRLM SUCCESSFULLY JOINED THE DATA SHARING GROUP. GLOBAL
  INITIALIZATION IS COMPLETE
  ```

- Check for retained locks using the command:

  ```
  F irlmproc,STATUS,ALLD
  ```

8.14.2 Advanced analysis

If a more detailed review of the problem is required:

- Make sure that the IMS systems are running with DLI and lock tracing on, and being written to the log or external trace table for every data sharing instance in the IMSplex.
  - To activate the trace, use the following IMS commands in each IMS in the data sharing group:

    ```
    /TRACE SET ON TABLE DL/I OPTION LOG VOLUME HIGH
    /TRACE SET ON TABLE LOCK OPTION LOG VOLUME HIGH
    ```

  - Alternatively, have the following cards in your DFSVSMxx PROCLIB member:

    ```
    OPTIONS,DLOG=ON,SCHD=ON,LATC=ON,STRG=ON,DUMP=YES
    OPTIONS,DISP=HIGH,LOCK=OUTHIGH,DL/I=OUTHIGH
    ```

- Activate the IRLM trace.
- Get a copy of the system log for each OS/390 image.
- Take a console dump of the IMS, IRLM and the DLI/SAS regions associated with the data sharing group.
- Print the IMS x'67FA' log records using exit DFSERA60 in DFSERA10 for each IMS subsystem in the data sharing group.

9

Operational considerations in shared queues environment

In this chapter we review the messages and commands you need when you are operating in a shared queues environment.

9.1 CQS structure management

Common Queue Server (CQS) supports several existing and new z/OS and CQS functions for structures, to help you monitor, tune, and recover your queue structures and resource structures. These functions include:

- CF duplexing
- System-managed rebuild
- CQS structure copy
- Structure alter
- Automatic structure alter
- Structure full threshold monitoring function
- CQS structure full handling

IMS supports a new structure repopulation function for resource structures only. Structure recovery is supported only for the queue structures.

CQS is enhanced in IMS Version 8 to support a new CF list structure called a resource structure, in support of Resource Manager. This broadens CQS's scope beyond only managing queue structures. CQS is therefore poorly named, since resource structures are different from queue structures. CQS is also enhanced to register to SCI, if a CSL is defined, so that the QUERY IMSPLEX command can show CQS as part of the IMSplex.

The new functions have these requirements:

- OS/390 V2 R10 required for IMS Version 8
- z/OS 1.2 required to enable system-managed CF duplexing
- CF level 9 required for IMS Version 8
- CF level 10 required to enable system-managed CF duplexing

9.1.1 Enabling duplexing and system-managed rebuild

These steps are necessary for enabling the structure duplexing:

- Define CFRM couple data set to support duplexing:
  - ITEM NAME(SMDUPLEX) NUMBER(1)
- Perform non-disruptive CFRM couple data set (CDS) migration.
- Define CFRM policy with DUPLEX(ALLOWED) or (ENABLED).
- Activate CFRM policy.
- Starting duplexing (switch to duplex mode) by using the following operator command for CFRM policy defined with DUPLEX(ALLOWED):

  ```
  SETXCF START,REBUILD,DUPLEX,STRNAME=strname
  ```

- Allocate the structure by starting CQS for CFRM policy defined with DUPLEX(ENABLED).

If you need to stop duplexing, you have to select if you want to keep the old or new structure. To stop duplexing (switch to simplex mode) and switch to the old structure can be done by the following command:

```
SETXCF STOP,REBUILD,DUPLEX,STRNAME=strname,KEEP=OLD
```

To stop duplexing and switch to new structure is done by the following command:

```
SETXCF STOP,REBUILD,DUPLEX,STRNAME=strname,KEEP=NEW
```

Planned reconfiguration may be needed while duplexing. Activate new CFRM policy with different structure attributes. Copy structure to another Coupling Facility to take a Coupling Facility offline for maintenance. Structure rebuild is not permitted for duplexed structure.

Planned reconfiguration steps:

- Stop duplexing with operator command:

  ```
  SETXCF STOP,REBUILD,DUPLEX,STRNAME=strname
  ```

- Reconfigure and start duplexing again with operator command:

  ```
  SETXCF START,REBUILD,DUPLEX,STRNAME=strname
  ```

9.2 IMS and XCF commands for shared queues

In this paragraph we look at both IMS commands and XCF commands for IMS shared queues. IMS commands have some new parameters on existing commands as well as changed impact of command. There are also some commands that are specifically only for shared queues, like /CQCHKPT, /CQQUERY and /CQSET.

OS/390 XCF commands are commands to display or change the status of structures or connections, for example D XCF or SETXCF.

9.2.1 Displaying the global and local queue counts

In the shared queues environment, you should be aware of what commands to use to display the global and local queue counts. Global queue count can be displayed using the /DIS QCNT command. It is asking for a count of the messages on the shared queue for the queue type specified and that have been there longer than the age specified by MSGAGE days.

This command reads every message on the queue type specified, so it should be used with caution, especially if there are a lot of messages on the queue. The following command can be used in place of the /DIS USER DEADQ command to find dead letter queues:

```
/DIS QCNT xxxxx MSGAGE n
```

This command displays, by queue name, number of messages on requested queue type which are older than n (number of) days. In this command, xxxxxx = queue type, which can be TRAN, LTERM, BALG, APPC, OTMA, REMOTE, MSGAGE n = number of days, 0 days will display all messages on queue. The /DIS QCNT LTERM MSGAGE 5 displays what you see in Figure 9-1.

As you see in Figure 9-1, a large amount of aged messages are on the LTERM Queue for more than 5 days.

```
  File  Display  View  Options  Help
PLEX1                    IMS Single Point of Control
Command ===> _
                                   Plex . PLEX1  Route . _________  Wait . _____
Log for . . : DIS QCNT LTERM MSGAGE 5                               More:   +>

IMSplex . . . . . :
Routing . . . . . :
Start time. . . . : 2003.119 16:41:08.03
Stop time . . . . : 2003.119 16:41:10.73
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IM1A

MbrName  Messages
-------- -----------------------
IM1A         QUEUENAME          QCNT-TOTAL     QCNT-AGED    TSTMP-OLD    TSTMP-
IM1A         WH000101                   37            37    03091/155055 03091/
IM1A         WH000207                   44            44    03091/155048 03091/
IM1A         WH000503                   62            62    03091/155120 03091/
IM1A         WH000609                   58            58    03091/155108 03091/
IM1A         WH000905                   57            57    03091/155058 03091/
IM1A         WH001203                   57            57    03091/155058 03091/
IM1A         WH001309                    3             3    03091/154939 03091/
IM1A         WH001901                   30            30    03091/154924 03091/
IM1A         WH002009                   55            55    03091/155127 03091/
IM1A         WH002010                    1             1    03092/141122 03092/
IM1A         WH000205                   39            39    03091/155030 03091/
IM1A         WH000501                   59            59    03091/155104 03091/
IM1A         WH000607                   54            54    03091/155110 03091/
```

Figure 9-1 /DIS QCNT LTERM MSGAGE 5 command from SPOC

Doing a /DIS LTERM WH000101 shows only a queue count (QCNT) of zero on IM4A, where the LTERM is in use. As you see in Figure 9-2, this command is against the local queue only.

```
  File  Display  View  Options  Help
PLEX1                    IMS Single Point of Control
Command ===> _
                                   Plex . PLEX1  Route . _________  Wait . _____
Log for . . : DIS LTERM WH000101

IMSplex . . . . . :
Routing . . . . . :
Start time. . . . : 2003.119 17:35:54.06
Stop time . . . . : 2003.119 17:35:54.06
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IM1A

MbrName  Messages
-------- -----------------------
IM1A         LTERM     ENQCT   DEQCT    QCT
IM1A         WH000101 IS INVALID
IM1A         *2003119/173554*
IM4A         LTERM     ENQCT   DEQCT    QCT
IM4A         WH000101      0       0      0
IM4A         *2003119/173554*
IM3A         LTERM     ENQCT   DEQCT    QCT
IM3A         WH000101 IS INVALID
IM3A         *2003119/173554*
```

Figure 9-2 /DISPLAY LTERM WH000101 command from SPOC

A more effective way to display the global queue counts is to issue /DIS LTERM command with QCNT parameter. Example 9-3 shows the /DIS LTERM WH000101 QCNT displaying the global queue counts for LTERM WH000101. The SPOC returns the same reply from all of the IMSs participating the IMSplex.

```
  File  Display  View  Options  Help
PLEX1                    IMS Single Point of Control
Command ===> _
                                   Plex . PLEX1  Route . _________  Wait . _____
Log for . . : DIS LTERM WH000101 QCNT

IMSplex . . . . . :
Routing . . . . . :
Start time. . . . : 2003.119 17:38:50.10
Stop time . . . . : 2003.119 17:38:50.11
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IM1A

MbrName  Messages
-------- -----------------------
IM1A         LTERM         GBLQCT
IM1A         WH000101          21
IM1A         *2003119/173850*
IM4A         LTERM         GBLQCT
IM4A         WH000101          21
IM4A         *2003119/173850*
IM3A         LTERM         GBLQCT
IM3A         WH000101          21
IM3A         *2003119/173850*
```

Figure 9-3 /DIS LTERM WH000101 QCNT command from SPOC

The QCNT parameter can also be used with other resources like LINE, LINK, LUNAME, MSNAME, NODE, TRAN and USER to display the global queue counts. The /DIS TRAN ALL QCNT would show also the affinities if some exist. This is shown in the Figure 9-4.

```
  File  Display  View  Options  Help
PLEX1                    IMS Single Point of Control
Command ===> _
                                   Plex . PLEX1  Route . _________  Wait . _____
Log for . . : DIS TRAN ALL QCNT

IMSplex . . . . . :
Routing . . . . . :
Start time. . . . : 2003.119 17:51:29.82
Stop time . . . . : 2003.119 17:51:29.83
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IM1A

MbrName  Messages
-------- -----------------------
IM1A         TRAN          GBLQCT  AFFINITY
IM1A         TPCCN            404
IM1A         *2003119/175129*
IM4A         TRAN          GBLQCT  AFFINITY
IM4A         TPCCN            404
IM4A         *2003119/175129*
IM3A         TRAN          GBLQCT  AFFINITY
IM3A         TPCCN            404
IM3A         *2003119/175129*
```

Figure 9-4 /DIS TRAN ALL QCNT command from SPOC

9.2.2 Dequeueing transactions from the shared queues

The /DEQ command can now be used with the TRAN keyword to dequeue transactions from the shared queue. This keyword is valid only in a shared queues environment. The format of the command is:

```
/DEQ TRAN xxxxxxxx PURGE | PURGE1 | FIRST
```

Note that the Message Control Error Exit (DFSCMUX0) is not invoked for /DEQ TRAN. With this command you can dequeue transactions from shared queues as follows:

- With keyword PURGE, all messages on the named queue
- With keyword PURGE1, the first message on the named queue

The /DEQUEUE SUSPEND command, when entered from any IMS in the shared queues group will move all transactions on the suspend queue back to the transaction ready queue, regardless of which IMS put them on the suspend queue.

> **Note:** APPC, OTMA, and SERIAL transactions are an exception to this. Those transactions put on suspend queue by other IMSs are not moved back to the ready queue; this must be done by the other IMS.

However, it will reset USTOPPED status only on the IMS on which the command was executed. To reset USTOPPED status on all IMSs, the command should be entered on all IMSs, or the /START command may be used to reset USTOPPED status for individual transaction codes. With the SPOC, the command is executed in all IMSs belonging to the IMSplex by default unless you route the command to the specific IMS. Another option would be to use the OS/390 ROUTE *ALL command to route the command to all IMSs.

9.2.3 Displaying the status of CQS connection

The /DISPLAY CQS command can be used to display information about the Common Queue Server to which IMS is (or was) connected. It returns the following information: the jobname, CQS Version, and the status (connected or disconnected).

Figure 9-5 shows an example of the command entered from the SPOC and routed to IM4A system in our IMSplex.

```
File  Display  View  Options  Help
PLEX1                    IMS Single Point of Control
Command ===>
                                   Plex . PLEX1  Route . IM4A     Wait . 
Log for . . : DISPLAY CQS

IMSplex . . . . . :
Routing . . . . . : IM4A
Start time. . . . : 2003.119 18:22:53.61
Stop time . . . . : 2003.119 18:22:53.61
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IM4A

MbrName  Messages
-------- -----------------------
IM4A         JOBNAME   VERS#  STATUS
IM4A         IM4ACQS     1.3  CONNECTED
IM4A         *2003119/182253*
```

Figure 9-5 /DIS CQS

The /DIS STRUCTURE ALL | xxxxxxxx command displays the status of the named (MSGQ or EMHQ) structures. Example 9-1 shows the command entered from an MCS-E console and the response for this. The possible values for STATUS are:

- AVAILABLE/UNAVAILABLE
- CONNECTED/DISCONNECTED
- IN-OVERFLOW
- SHUTDOWN-STRCHKPT
- STRCHKPT-INPROG
- REBLD-INPROG

Example 9-1 /DIS STRUCTURE ALL

```
IM1ADIS STRUCTURE ALL

DFS4444I DISPLAY FROM ID=IM1A
     STRUCTURE NAME    TYPE  STATUS
     IMOA_MSGP         MSGQ  CONNECTED, AVAILABLE, IN-OVERFLOW
     IMOA_EMHP         EMHQ  CONNECTED, AVAILABLE
     *2003121/180525*
```

In our example, we see that our full function message queue (MSGQ) is in overflow status and the Fast Path message queue (EMHQ) is not in overflow status. The /DIS OVERFLOWQ STRUCTURE ALL | xxxxxx command can be used to identify the queue names of those resources in overflow status. Example 9-2 shows the command and the response. In our case, we have some LTERMs that have their messages moved to the overflow structure.

Example 9-2 **DIS OVERFLOWQ STRUCTURE ALL**

```
IM1ADIS OVERFLOWQ STRUCTURE ALL

DFS4444I DISPLAY FROM ID=IM1A
     STRUC-RSCTYPE    OFLSTRUC-RSCNAME LUNAME-TMEMBER
     IMOA_MSGP        IMOA_MSGO
      LTERM            WH001907
      LTERM            WH001705
      LTERM            WH001909
      LTERM            WH001101
      LTERM            WH001105
      LTERM            WH001805
      LTERM            WH001109
      LTERM            WH001807
      LTERM            WH001605
      LTERM            WH001809
      LTERM            WH001607
      LTERM            WH001407
      LTERM            WH001409
      LTERM            WH001905
     IMOA_EMHP        IS NOT IN OVERFLOW MODE
     *2003121/175526*
```

9.2.4 /CQCHKPT, /CQQUERY and /CQSET

The following commands are related to the usage of CQS and they are valid only in a shared queues environment:

- /CQCHKPT

  /CQCHKPT initiates a CQS checkpoint for a specific coupling facility list structure or all the coupling facility list structures to which the IMS subsystem is connected.

- /CQQUERY

  The /CQQUERY command displays information regarding a specific coupling facility list structure or all the coupling facility list structures holding IMS messages.

- /CQSET

  Use a /CQSET SHUTDOWN SHAREDQ ON|OFF command to tell CQS whether to take a structure checkpoint during normal shutdown.

As an example, let's look at the /CQQUERY STATISTICS STRUCTURE structurename|ALL command. It can be entered from IMS to display the current definition and usage of list entries and data elements in the structure(s).

This command is shown in Figure 9-6, and indicates that of the 5000 list entries allocated, 4000 are currently in use. And of the 15,000 data elements allocated, only 5000 are currently in use. If the OVFLWMAX is 70%, then we will trigger overflow processing when 10,500 data elements are in use. But note that we have only 1000 list entries left, and at the current ration of 4 list entries used for every 5 data elements, we may run out of list entries before we reach the data element threshold. If this happens, the primary structure will become full without ever having the chance to go into overflow processing.

- /CQQUERY STATISTICS STRUCTURE structurename|ALL
  - Obtains statistical information regarding current utilization of shared queue structures

STRUCTURE NAME	LEALLOC	LEINUSE	ELMALLOC	ELMINUSE	LE/EL
MSGQPRIMARY	5000	4000	15000	5000	1/3

LEALLOC	Number of list entries allocated. Maximum number of list entries possible.
LEINUSE	List entries in use (currently on queue).
ELMALLOC	Number of 512-byte list elements allocated.
ELMINUSE	Number of list elements in use. Drives overflow processing.
LE/EL	Ratio of list entries to list elements.

  - If ELMINUSE reaches 10500 overflow processing will be invoked
    - Assuming default overflow threshold of 70%
  - **If LEINUSE reaches 5000 structure will be FULL**
    - **Overflow processing does not detect shortage of List Entries**

Figure 9-6 /CQQUERY

When the structure was sized, an estimate was made as to the average size of a message (for example, 1500 bytes). This was used as the basis for determining the ratio of list entries to 512 data elements (for example, 1:3). If that message size estimate was too big (for example, messages are not as big as we estimated) then this condition can arise. In this example, the estimate produced a LE/EL ratio of 1:3. In reality, messages were not that big.

9.2.5 Other consideration with IMS commands in the sysplex environment

There are some other considerations with IMS commands that you should be aware of when operating in the Parallel Sysplex environment. For example, the /ASSIGN and /MSASSIGN commands should be entered on all IMSs in the IMSplex. For example, don't assign an LTERM to a different NODE on IM1A than on IM2A.

- /ASSIGN and /MSASSIGN

  Must be entered at every IMS sharing queues to keep control blocks synchronized.

- /CHE FREEZE

  CQS remains active after IMS is shut down. Allows CQS to continue taking system checkpoints. A shutdown checkpoint is not allowed in a shared queues environment if the CQS is not available.

  NOCQSSHUT can be used with the /CHE DUMPQ, /CHE FREEZE, or /CHE PURGE commands, to not shut down the CQS address space when the IMS control region terminates. The CQS address space remains active and connected to the message queue structures. NOCQSSHUT is only applicable when IMS is running in a shared-queues environment. The default is to shut down the CQS address space when the IMS control region terminates. However, if there are other CQS clients active (OM and RM in IMS Version 8, other IMS with IMS Version 7 or 8), the CQS is not shut down when IMS control region terminates.

- /CHE DUMPQ|PURGE

  In a shared-queues environment, the DUMPQ and PURGE keywords cause IMS to shut down (as if you entered a /CHECKPOINT FREEZE command), but the message queues are not dumped or purged, because the local IMS subsystem has no local queues. To dump the shared message queues when CQS terminates, use the /CQSET command before issuing the IMS shutdown checkpoint command.

- /CHE SNAPQ

  Has no significance in shared queues environment — ignored. Use the /CQCHKPT command to initiate a CQS structure checkpoint.

- /DISPLAY MODIFY TRS

  Shows SUM of local and global queue counts.

- /DISPLAY SHUTDOWN STATUS

  Shows CQS jobname to start if shutdown hung because CQS is down. IMS will not shut down if CQS is not available.

- /NRE and /ERE

  BUILDQ, FORMAT SM, LM and QC are ignored in the shared queues environment.

- /TRACE SET ON|OFF TABLE QMGR|SQTT

  There are two new traces with shared queues: the queue manager trace and the shared queues trace. Use the /DIS TRACE TABLE QMGR|SQTT command to display the status of the trace.

- /STOP TRAN

  In a shared queues environment, if you issue a /STOP TRANSACTION command for a transaction that is not defined on that IMS subsystem, IMS creates an SMB if the Output Creation user exit routine indicates the destination is a valid transaction. The SMB is marked as “dynamic”. A dynamic SMB created by a /STOP TRANSACTION command can only be used to queue messages for the transaction and place the messages on the shared queues. The transaction cannot be scheduled or assigned. IMS does process checkpoints for the transaction, but does not save them across an IMS restart if they do not have a valid status.

To stop processing of a transaction on a specific IMS without stopping that IMS from queuing transactions on the shared queue, you can use the following commands:

/PSTOP TRAN	Stops scheduling but not queuing

/LOCK TRAN	Stops scheduling but not queuing
/ASSIGN TRAN	Use to assign the transaction to a class that is not active

The point to make here is that /STOP TRAN will stop queuing of that transaction on the system where the command was entered, but not on other systems. The other commands shown allow the transaction to be queued but not scheduled. IMS will deregister interest in the transaction when any of these commands are entered.

9.3 Structure operations

In this section we describe the operations and situations you may experience with shared queue structures.

9.3.1 Structure overflow processing

When IMS goes to the overflow processing, you can expect to see the following kind of messages, indicating that the structure size is being altered:

```
CQS0260I CQS CQS2CQS STARTED OVERFLOW THRESHOLD PROCESSING FOR STRUCTURE IMOA_EMHP
CQS0200I STRUCTURE IMOA_EMHP QUIESCED FOR OVERFLOW THRESHOLD PHASE
CQS0265I STRUCTURE ALTER REQUEST STARTED FOR STRUCTURE IMOA_EMHP
CQS0266I STRUCTURE ALTER REQUEST COMPLETED FOR STRUCTURE IMOA_EMHP
CQS0264I CQS CQS2CQS TERMINATED OVERFLOW THRESHOLD PROCESSING, ALTER SUCCESSFUL FOR
STRUCTURE IMOA_EMHP
CQS0201I STRUCTURE IMOA_EMHP RESUMED AFTER OVERFLOW THRESHOLD PHASE 1
```

Eventually the structure will reach the OVFLWMAX and already be at maximum size, CQS allocates the overflow structure. Watch for a somewhat different set of messages:

```
CQS0260I START OVERFLOW PROCESSING
CQS0200I QUIESCE FOR PHASE 1 (All queues)
CQS0008W CF VOLATILE
IXL014 CONNECT
IXL015 ALLOCATION
CQS0261I PHASE 1 COMPLETE
CQS0201I RESUME
CQS0200I QUIESCE FOR PHASE 2 (Overflow queues)
CQS0201I RESUME
CQS0262I PHASE 2 COMPLETE
CQS0220I QUIESCE FOR STRUCTURE CHECKPOINT
CQS0201I RESUME
CQS0221I STRUCTURE CHECKPOINT COMPLETE
```

Note: There is an automatic structure checkpoint after overflow processing completes.

Once the queue length goes below the overflow threshold, the queue is returned to normal processing. Removal of transactions from the overflow queue is driven by a 15 minute timer. Each 15 minutes, CQS scans the overflows queues and if they are empty at the time of the scan, that queue is moved back to the primary structure. So, even though the queues are empty, when you look at what is in overflow, those empty queues might still be there.

9.3.2 How to handle the structure full condition

Despite all efforts, it is still possible that one or both structures could become full. This may be because either we ran out of list entries or data elements. Eventually the overflow queues will be drained and arriving input messages are rejected until there is space for them in the

structure. When this happens, PUTs to the structure are rejected, but READs, MOVEs, and most important DELETEs are allowed to continue. This should (hopefully) relieve the full condition.

Since DELETEs continue, full is usually both a temporary and sporadic condition. IMS does not stop trying to put messages on the queue, and every time a data element is deleted, another PUT may be allowed.

> **Note:** When structures are full, IMS continues to attempt to process messages. Without shared queues, IMS abends with U0758 when the message queue is full.

The only time an application program should get a QF status code is when it is using multiple queue buffers. For example, when an application program tries to insert a message segment which requires the third buffer, IMS will try to put the second buffer on the staging queue. If this PUT is rejected, the application gets the QF.

If the application is successful at ISRT time, IMS doesn't try again until commit, at which time it is too late to reject the ISRT. In this case, IMS saves the message in its QPOOL and tries again at each system checkpoint.

Input messages are put on the queue when they arrive. If the PUT is rejected, a message will be sent to the terminal, and the system console will get a *CQS0205E* message. This CQS message will only be sent once as long as the structure stays in overflow mode. If the structure gets out of overflow mode, and this situation occurs again, then the message will be sent again.

When the structure is full, input messages are rejected and non-EMH terminal gets *DFS070* message. EMH terminal gets *DFS2194* message, if MSC is involved the related MSC link is stopped, *DFS2140 RC=1945* issued by partner MSC. These messages are shown here:

```
DFS070 UNABLE TO ROUTE MESSAGE
DFS2194 SHARED EMHQ NOT AVAILABLE
DFS2140 DESTINATION MSCLNK1 STOPPED, REASON CODE 1945
CQS0205E STRUCTURE IMOA_MSGP IS FULL
```

9.3.3 If the structure is too small to create EMCs

Another condition related to lack of storage is a shortage of space to create EMCs when a client registers interest in a queue name.

When the structure was allocated, 20% of the size of the structure was allocated for EMCs. Every time a client registers interest, a 64-byte EMC is created in this space. If there is no space left to create another EMC, then that registration is rejected and the queue name is not registered for that client. This means that the client never knows when there is work on the queue, and will never issue a READ to process any messages on that queue.

If this happens, the following CQS message will be sent:

```
CQS0360W QUEUE REGISTRATION FAILED FOR STRUCTURE IMOA_MSGP:
CHANGE CFRM POLICY AND REBUILD.
```

This is an important message, and the structure should be resized and rebuilt as soon as possible.

If the user is running with CF Level 4 or higher, then additional space for EMCs will be dynamically allocated from the space allocated for data elements not currently in use. Too much of this could, of course, lead to a shortage of data elements.

Remember that EMCs are only 64 bytes, so it takes a lot of them to run out of storage. But consider this. Two IMSs with a total of 10,000 resources to register will require 20,000 EMCs. Three IMSs with 10,000 each will require 30,000. Even if the number of messages stays the same across the Parallel Sysplex. In other words, just adding IMSs to the Parallel Sysplex will increase the need for EMC storage.

You can use the following commands to display status of the structures and the individual queue names:

- /DIS STRUCTURE ALL

  Indicates whether primary structure is in overflow mode.
- /DIS OVERFLOWQ STRUCTURE ALL

  Identifies queue names in overflow.
- /DIS LTERM|TRAN|USER xxxxxxxx QCNT (or /DIS QCNT LTERM xxxxxxxx MSGAGE 0)

  Indicates how many messages are on queue xxxxxxxx.
- /DEQ TRANSACTION|LTERM|MSNAME xxxxxxxx PURGE1|FIRST

  Purges 1st message for QNAME xxxxxxxx.
- /DEQ TRANSACTION|LTERM|MSNAME xxxxxxxx PURGE

  Purges all messages for QNAME xxxxxxxx.

Note: Note that the TRANSACTION keyword has been added to the /DEQ command to allow the user to dequeue transactions from the shared queue structures.

If these commands indicate a problem, then one alternative is to use the /DEQUEUE command to delete messages for queue names which have long queues and for which the user knows it is alright to delete them. The other alternative is to make the structure bigger. This can be done dynamically, but of course while it is being rebuilt, all activity to the structure is quiesced. The needed steps for increasing the structure size are:

- Change the CFRM policy to specify a larger SIZE.
- Activate the new changed policy with the following XCF command:

  ```
  SETXCF START,POLICY,TYPE=CFRM,POLNAME=new_policy_name
  ```
- Structure change will be pending until structure is rebuilt. So enter the following command to rebuild structure:

  ```
  SETXCF START,REBUILD,STRNAME=IMOA_MSGP
  ```
- Structure will be rebuilt with new size specification. Keep in mind that the structure is quiesced during the rebuild.

You should consider the Queue Space Notification Exit (DFSQSPC0). In a shared queues environment, the exit does not know how much space is being used or how much remains in the structure, but it does know whether the structure is in overflow mode, and whether the destination of the ISRT is in overflow status. This information could be used by the exit to reject the ISRT, causing an A7 status code to be returned to the application program.

9.3.4 Structure integrity

Because the processing of messages is often done by multiple IMSs in the Parallel Sysplex, the integrity of these messages is the responsibility of CQS — the focal point for all messages across the sysplex.

To accomplish this, CQS performs structure and system functions that have the same intent as equivalent IMS functions. CQS logs all message and structure activity into and out of the shared queues. Since there are multiple CQSs, and just one structure (pair), all CQSs use the System Logger to log this activity to a log stream defined in the LOGR policy. This log stream is a merged log of all CQS log records (unlike IMS logs in a data sharing environment which must be merged by change accumulation prior to using them for recovery).

CQS takes system checkpoints, with the log records also going to the System Logger. The log token of the beginning of the checkpoint log records is kept in a checkpoint data set for each CQS. Another copy is kept on the CONTROLQ in the structure. Think of the checkpoint records like the IMS x'40xx' system checkpoint records, and the checkpoint data set like the IMS restart data set (RDS).

CQS also takes structure checkpoints. These are logical copies of the contents of the structure at the time the structure checkpoint is initiated. We call them logical copies because only the data content is copied, not the entire structure. Unused list entries, for example, are not copied. This is unlike IMS database image copy where a physical copy is taken, including empty blocks.

With the above data, any CQS can recover a structure if it is lost due to structure damage or CF failure. A few messages are written to show the checkpoint status.

CQS system checkpoint is like IMS's system checkpoint. It logs the current status of the CQS system, and in particular, the status of work in progress. This information is used when CQS is restarted. APAR PQ21499 added message CQS0035E to indicate that the CQS system checkpoint failed.

When System Checkpoint completes, message CQS0030I is issued. It names the structure for which the system checkpoint was taken and displays the log token used for restart:

```
CQS0030I SYSTEM CHECKPOINT COMPLETE...., LOGTOKEN.....
CQS0035E SYSTEM CHECKPOINT FAILED...
```

CQS system checkpoints are initiated automatically based on a parameter in each CQS's local parameter list. It is equivalent to the IMS CPLOG parameter and says that after so many log records are written to the log stream by this CQS, then take a system checkpoint. The objective is to limit the amount of log data that must be read when CQS has to be restarted after a failure. It's difficult to say just how large or small this number should be, since the total amount of log data on the log stream depends on what other CQSs are doing.

CQS also initiates the system checkpoint in the following situations:

- At the end of a successful structure checkpoint
- At end of CQS restart
- At end of client resync
- At CQS shutdown

The user may request that a checkpoint be taken by entering the following command from the IMS system that is connected to that CQS:

```
/CQCHKPT SYSTEM STRUCTURE ALL
```

Above command would take two checkpoints, one for MSGQ and the other one for EMHQ. Note that this is NOT a structure checkpoint.

Since CQS restarts like IMS, from the last system checkpoint, its restart time is affected by the amount of log data between that checkpoint record and the end of the log stream. A CQS which has failed, and which is not restarted for a long time, may find that its checkpoint records are a long way back in the log stream.

When a structure checkpoint is taken, some log records can be deleted from the log stream since they are no longer needed for structure recovery. When a structure recovery is taken, then all active CQSs are asked to take a system checkpoint, just in case one hasn't been taken in a while. Of course, if a CQS is not active, then it can't take a system checkpoint, and it is possible that by the time that CQS is restarted, its checkpoint log records will have been deleted. In this case, a CQS cold start is required. IMS has most of the information needed to resync the queues (for instance, to resolve work in progress) but CQS may lose some status information. Checkpoint frequency affects CQS restart time. Remember, all CQSs contribute to the volume of log data CQS is restarted from its last checkpoint record may be way back in the log stream.

9.3.5 Structure checkpoint

Structure checkpoints are image copies of the content of a structure pair (primary + overflow). They are required for structure recovery along with the logs, and also trigger the deletion of log data from the log stream (to keep the log stream from filling up).

Any CQS may perform the structure checkpoint (the checkpoint process is serialized by the CONTROLQ). The process is to copy the data into an OS/390 data space and then copy it to one of two SRDSs which are used in a flip-flop fashion (for example, CQS writes over the oldest SRDS). Messages on both primary and overflow structure are copied to one SRDS.

While the data is being copied to the data space, all structure activity is quiesced. Recoverable data objects are copied to a data space. When copy completes, activity resumes. Copy to SRDS is done from the data space. Only non-empty list entries are copied, because only recoverable data objects are recovered. Fast Path input messages are put on the EMHQ marked non-recoverable, so they are not copied to the SRDS. The same would be true for full function messages defined as non-recoverable (usually not very many).

When structure checkpoint completes, then each active CQS takes a system checkpoint and log records older than the oldest SRDS are purged from the log stream.

Structure checkpoint can be initiated either automatically or by command. Since the impact of a structure checkpoint on the user is potentially significant (structure is quiesced), most structure checkpoints should be initiated by command.

One technique is a combination of issuing a command to cause an automatic structure checkpoint when CQS shuts down. Be careful of this, since there may still be other CQSs active, and when this one shuts down, the queues will be quiesced. We would expect that most users would schedule structure checkpoints during periods of low activity.

Although possible, it is not likely that the System Logger log stream would become full. This can be very large and should happen only in very rare circumstances. But, if it does happen, a structure checkpoint will be taken to delete some of the log records.

To reduce the size of the log stream, two successive structure checkpoints would delete nearly all of it - taken perhaps just before shutting everything down, or during a period of very low activity. Structure checkpoint is initiated by CQS in the following situations:

- System Logger log becomes full, or OS/390 warns of a pending full condition. Pending full means that last spill data set is being used and only the structure remains with any space.
- At completion of successful overflow processing. Messages are moved to overflow structure.
- After successful structure recovery.
- By IMS command /CQCHKPT SHAREDQ STRUCTURE ALL | strname.

- At CQS shutdown if /CQSET SHUTDOWN ON SHAREDQ entered earlier.

When a structure checkpoint is initiated, a series of CQS messages is written to the system console. You can expect to see messages similar to these:

```
CQS0220I CQSA STARTED STRUCTURE CHECKPOINT FOR STRUCTURE MSGQSTR
CQS0200I STRUCTURE MSGQSTR QUIESCED FOR STRUCTURE CHECKPOINT
CQS0201I STRUCTURE MSGQSTR RESUMED AFTER STRUCTURE CHECKPOINT
CQS0221I CQSA COMPLETED STRUCTURE CHECKPOINT FOR STRUCTURE MSGQSTR
CQS0030I SYSTEM CHECKPOINT COMPLETE, STRUCTURE MSGQSTR
LOGTOKEN 00000000002C619E
```

9.3.6 Structure rebuild

A structure rebuild comes in two types:

- Copy

  This is when the structure is fine and available but you just want to copy it — perhaps to move it, or to change some of its properties like the SIZE or REBUILDPERCENT. No SRDS or logs are required.

- Recover

  This is when the structure is damaged or the CF has failed. This requires the SRDS and the logs.

Within the CFRM policy is a parameter called the REBUILDPERCENT. When the number of OS/390 systems losing connectivity to the structure reaches this percentage, then a copy is automatically invoked. XCF notifies each CQS in the CQS shared queues group that a rebuild is required and one of them will do it (using the CONTROLQ to make sure only one actually does it).

Structure copy

When the copy function is performed, all messages are copied, even non-recoverable ones (for example, Fast Path input messages). For instance, if the REBUILDPERCENT=20, the loss of a single connection to the structure would prompt a copy to another CF in the preference list (PREFLIST) defined in the CFRM policy. Note that this automatic rebuild requires a system failure management policy with the specification CONNFAIL(YES).

The likelihood of the occurrence of structure copy can be reduced by having multiple links to the CFs. The structure copy copies the structure to the same or another CF. At least one CQS must have access to the structure. Structure copy can also be invoked manually by using an XCF command. When the command is entered, all CQSs are notified by XES and one of them will perform the copy.

The user may use this command to implement changes in the structure (for example, SIZE, INITSIZE, REBUILDPERCENT or PREFLIST). When the policy is changed, the changes are pending until a rebuild is done. It may also be used to move the structure from one CF to another. When used to move it, LOCATION should be specified. LOCATION=OTHER means to move it to some CF in the preflist other than the one it is currently on, and LOCATION=NORMAL means to rebuild it on its preferred CF defined in the PREFLIST. Here is an example of the command:

```
SETXCF START,REBUILD,STRNM=IMOA_MSGP,LOCATION=OTHER
```

The following messages will be written to the SYSLOG during structure copy:

```
IXC521I REBUILD FOR STRUCTURE MSGQSTR HAS BEEN STARTED
IXC367I THE SETXCF START REBUILD REQUEST FOR STRUCTURE MSGQSTR WAS ACCEPTED
```

```
CQS0200I STRUCTURE MSGQSTR QUIESCED FOR STRUCTURE REBUILD
IXL014I IXLCONN REQUEST FOR STRUCTURE MSGQSTR WAS SUCCESSFUL. JOBNAME: CQSA   ASID: 021B
CONNECTOR NAME: xxxxxx
CQS0240I CQS CQSA START STRUCTURE COPY FOR STRUCTURE MSGQSTR CFNAME: CF02
CQS0241I CQS CQSA COMPLETED STRUCTURE COPY FOR STRUCTURE MSGQSTR
CQS0201I STRUCTURE MSGQSTR RESUMED AFTER STRUCTURE REBUILD
```

Note: These messages come from both XES and CQS. CQS is notified by XES that a REBUILD has been requested. The actual sequence of the CQS and IXC messages may vary somewhat, and some of the CQS messages will be duplicated if multiple CQSs are active at the time of the command.

9.3.7 Structure recovery

Structure recovery is required if no CQSs can access the structure due to a Coupling Facility failure or structure failure (damage). Structures are recovered from the most recent structure checkpoint data set (SRDS1 or SRDS2) with merged CQS log stream. If the most recent SRDS is bad, then the other older one will be used.

CQS detects the problem when it tries to connect, and will automatically invoke recovery. To do this, it acquires a lock on the CONTROLQ, creates a CQSRECOVER entry, and releases the lock. Other CQSs finding the CQSRECOVER entry will know that the creator of the entry is the MASTER and will merely monitor the progress.

9.3.8 Structure delete

For some reason, you may want to delete the structure. This can be, for example, to clean up the message queues. With traditional IMS message queues, when IMS is cold started, its message queues are reinitialized to empty. This doesn't happen with shared queues, an IMS cold start does not delete the queues. If you want to delete the messages, you can use the /DEQUEUE commands as explained earlier, or you can delete the structure. If you choose to delete the structure, you must also delete and reallocate the both SRDSs. If both SRDSs are not deleted, then CQS in its wisdom will recover the structure during the restart. Deleting a structure and the SRDSs, is functional equivalent to IMS cold start (COLDCOMM).

The structure can be deleted by using this command:

```
SETXCF FORCE,STRUCTURE,STRNM=IMSP_MSGP
```

If the structure has been deleted and there has not been two structure checkpoints, then CQS will recover the structure from only the logs. CQS knows this is alright only if the first log record is still in the log stream. This may occur in a test environment where the user is testing and trying to delete the structure to start over again. If this is the case, the user should take two structure checkpoints immediately to cause the first log record to be deleted.

To avoid the need for structure delete, also called structure coldstart, you could use the IMS Queue Control Facility product (product number 5697-E99) that can help to handle situations where messages are put on the cold queue.

Note: Every in-flight UOW messages is internally put on the lock queue. Now if IMS is shut down or abended, those messages are moved to the cold queue. An /NRE or /ERE will recover these messages, but if IMS is cold started, those messages are kept on the cold queue forever. The only way to get rid of them is either to delete the structure or use the QCF or equivalent product.

Recovering messages using QCF

Under normal circumstances IMS and CQS have all the knowledge about the messages if there is any need to recover a message. Unfortunately there are situations which need your manual intervention to recover messages. One situation might be a IMS cold start with messages on the lock queue.

In the shared queues environment, a cold start does not erase messages from the shared queues. Also, it is not possible in this environment for a single IMS to keep track of the status of a message inserted to the shared queues, because another IMS can remove and process that message.

For these reasons, IMS does not log messages to the IMS log during checkpoint processing. DUMPQ and SNAPQ checkpoints are not done, and QCF product cannot perform RECOVERAB or RECOVERDM processing. However, QCF can recover messages on the cold queue, that is, messages that were in process when IMS abended and were then cold started. This is done using the RECOVER function of QCF. Refer to *IMS Queue Control Facility for z/OS User's Guide,* SC26-9685, for more details.

9.3.9 Structure alter

Structure ALTER can be used to change the current size of a structure up to the maximum size. This does not require a rebuild — additional space in the CF is just allocated to the structure. This is done, either automatically by CQS when the overflow threshold is reached, or can be done manually by entering an XCF command.

Only the current size can be changed with this process, and then only if the structure is not already at its maximum SIZE defined in the CFRM policy. If manually invoked by operator command, this can be done for both primary and overflow structure, using the command similar to the following:

```
SETXCF START,ALTER,STRNM=IMOA_MSGP,SIZE=80000
```

9.3.10 Shared queues with XRF

When using XRF with shared queues, it is still necessary to define the local message queue data sets for both the active and the alternate IMS JCL as follows:

```
//QBLKSL DD...
//SHMSGL DD...
//LGMSGL DD...
```

Only the alternate uses them, but the active will not come up XRF enabled if these data sets are not defined. Note that these are not the regular message queue data sets. Note the 'L' at the end of the DDNAME. The local message queues are used for messages entered at the alternate from MTO or system console:

At initialization, the XRF alternate will start a CQS address space, but does not register or connect with it. IMS (XRF) determines whether CQS is active by registering to it. If registration fails, then CQS must be started. If it is successful, then CQS is already active, and XRF will have already registered. XRF does not deregister in this case.

During tracking, only the active system is connected to the shared queue structures.

Messages entered at the alternate are kept on its local message queue data sets.

At takeover, the alternate (new active) will register and connect to CQS, resynchronize, and merge its local messages with the global queues.

9.3.11 RSR

RSR is supported in a shared queues environment with restrictions. Active IMS should be a part of the same global service group. The tracking subsystem does not run with shared queues enabled. So there is no real time tracking of messages. Here are the restrictions:

- Shared queues cannot be recovered by new active.
- /ERE BUILDQ will not work.
- Queues are recoverable only from the SRDS and MVS log stream.

There are no changes to support for database recovery. It is possible, for a planned takeover, to recover the shared queues. This can be done by taking a structure checkpoint when the last CQS shuts down, then sending that SRDS to the remote site. When CQS is started at the remote site, it will discover an empty structure and recover it using the SRDS. The sequence of this activity would be:

- /CQSET SHUTDOWN SHAREDQ
  - Forces structure checkpoint when CQS shuts down normally
- /RTAKEOVER at active site
  - IMS and CQS shut down normally
  - CQS creates structure checkpoint on SRDS
- Send SRDS to remote site
- Send CQS log stream to remote site
- /NRE CHECKPOINT 0 at remote site
  - CQS will recover structures from SRDS

Figure 9-7 shows how the planned takeover might be made to recover the shared queues.

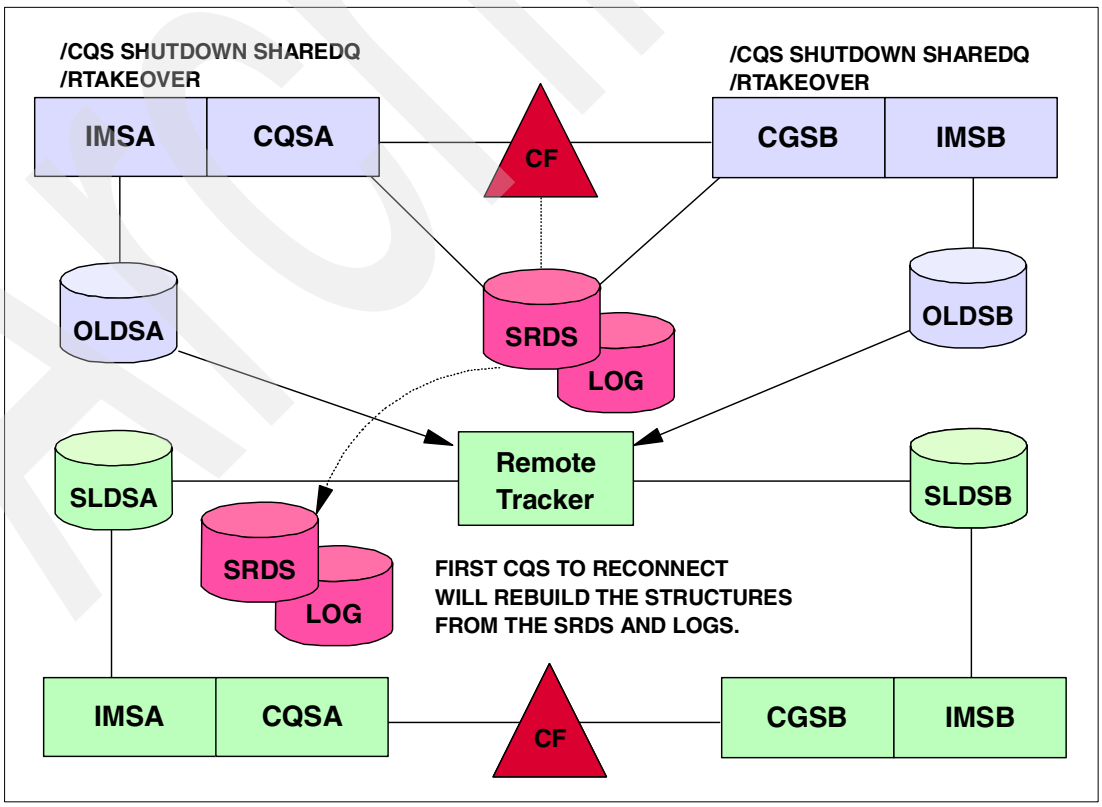

Figure 9-7 Remote Side Recovery (RSR)

9.3.12 How to take a dump of the structure and related address spaces

There are a few ways to get a structure dump either by dump command or via setting a slip as shown in Example 9-3.

Example 9-3 SLIP

```
SLIP SET,IF,N=(IEAVEDS0,00,FF),
     A=SVCD,ENABLE,ID=SCQS,
     STRLIST=(STRNM=IMOA_MSGP,
     (LNUM=ALL,ADJ=CAP,EDATA=UNSER)),END
```

The other way is to create three SYS1.PARMLIB members called IEADMCIA, IEADMCIB, IEADMCIC. Example 9-4 shows the contents of IEADMCIA.

Example 9-4 The contents of IEADMCIA

```
JOBNAME=(j1,j2,j3,j4,j5),SDATA=(CSA,PSA,RGN,SQA,SUM,TRT,GRSQ),
REMOTE=(SYSLIST=(*('j1','j2','j3','j4','j5'),SDATA)),END
```

In the IEADMCIA example, the variables j1-j5 are as follows:

- j1 = IMS control region jobname
- j2 = IMS DLI region jobname
- j3 = DBRC region jobname
- j4 = IRLM region jobname
- j5 = IMS CQS region jobname

Example 9-5 shows the contents of IEADMCIB.

Example 9-5 The contents of IEADMCIB

```
JOBNAME=(j6,j7,j8,j9,j10),SDATA=(CSA,PSA,RGN,SQA,SUM,TRT,XESDATA),
REMOTE=(SYSLIST=(*('j6','j7','j8','j9','j10'),SDATA)),END
```

In the IEADMCIB example, the variables j6-j7 are as follows:

- j6 = APPC/MVS region name
- j7 = APPC scheduler address space name
- j8 = VTAM address space name
- j9 = Other CQS client region jobname
- j10 = other CQS region jobname
- j8 = VTAM address space name

Example 9-6 shows the contents of IEADMCIC.

Example 9-6 The contents of IEADMCIC

```
JOBNAME=(IXGLOGR),DSPNAME=('IXGLOGR'.SYSLOGRO),
SDATA=(COUPLE,ALLNUC,LPA,PSA,RGN,SQA,TRT,CSA,GRSQ,XESDATA)
STRLIST=(STRNAME=ims_msgp,LOCKE,(LISTNUM=ALL,ADJ=CAPTURE,EDATA=UNSER),
STRNAME=ims_msgo,LOCKE,(LISTNUM=ALL,ADJ=CAPTURE,EDATA=UNSER),
STRNAME=ims_logp,LOCKE,ACC=NOLIM,(LISTNUM=ALL,
EDATA=UNSER,ADJ=CAPTURE)),END
```

In the IEADMCIC example, the variables are as follows:

- ims_msgp is the main structure name
- ims_msgo is the overflow structure name

- ims_logp is the associated logger structure

To request a dump from the IEADMCIA, IEADDMCIB and IEADMCIC parmlib members, enter the following MVS command:

```
DUMP TITLE=(DUMP OF IMSplex and Partners),PARMLIB=(IA,IB,IC)
```

Three dump data sets will be created on the OS/390 LPAR from which the command is entered. Two dump data sets will be created on each image in the sysplex matching the remote specifications for the jobnames.

10

Common Service Layer (CSL) operations

This chapter describes the operations management tasks of the Common Service Layer (CSL):

- Starting IMSplex address spaces
- Shutting down IMSplex address spaces
- BPE commands
- Enhanced commands (IMSplex commands)
- IMS classic commands

10.1 CSL operations

There are two issues when discussing operations in an IMSplex CSL environment:

- Startup, execution, and shutdown of the CSL execution environment
- Submission of IMS commands through the OM interface and through the classic IMS interface (MTO, WTOR, IMS terminal, E-MCS console)

The CSL environment consists of a multitude of address spaces and Coupling Facility structures. How these components are started and stopped is the subject of the next few topics.

10.1.1 Starting IMSplex address spaces

First you need to activate the CFRM policy. While this may sound obvious, updates to the sysplex CFRM policy do not take effect until that policy is (re)activated. This is important not only when adding new structures, such as the resource structure, to the policy, but also when making changes to a structure, such as enabling duplexing or autoalter. A policy is activated by this command:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=CFRMPOL1
```

Unlike CQS, the CSL address spaces are not started automatically by IMS. The address spaces must be started either manually or by automation. CSL address spaces can be either jobs or started tasks. All of the CSL address spaces can register with Automatic Restart Manager (ARM), which will restart the address spaces if they terminate abnormally.

You must have at least an SCI running on each z/OS on which there is an IMS component executing. Since SCI is required by all IMS components, it is recommended that you start SCI first. Then start OM and RM, and finally the IMS control regions.

10.1.2 CSL start sequence

The sequence in which components in a CSL are started is significant. Some of them are required to be started in the proper sequence, for others it is just recommended they be started in the proper sequence to avoid warning messages, because some members register with other (target) members and abends may occur if target is not started:

1. Start SCI first
   - All IMSplex members register with SCI
2. Start CQS everywhere there is an RM and structure
   - RM registers with CQS
   - CQS is also needed if shared queues is used (at least one per LPAR, with IMS Version 6, at least one per IMS with shared queues)
3. Start (at least one) OM
   - RM and IMS register with OM
   - Command processing clients register with OM
4. Start (at least one) RM
   - Start only one RM if no resource structure
   - IMS registers with RM
5. Start IMS
   - Nobody registers with IMS

6. Start SPOC
   – Anytime after OM started

10.1.3 IMSplex address spaces relationship

Table 10-1 shows the sequence in which components in a CSL should be started and their relationship.

Table 10-1 IMSplex address spaces relationship

Name	Started by	Needs	Needed by	Reg. with
SCI	User		All IMSplex components. If not active, may cause that component to abend.	
CQS	IMS if serving as shared queues manager, user if used by RM	SCI up on the CQS's z/OS image. If not, a warning message is issued	► RM ► IMS using shared queues	SCI
OM	User	SCI up on the OM's z/OS image. If not, warning message before abend.	RM	SCI
RM	User	► SCI active on the RM's z/OS image. If not, warning message before abend. ► At least one OM active in the IMSplex. If not, issues error message. ► CQS active if resource structure is used. If not, retry register several times before abend.	IMS	CQS, OM
IMS	User	All IMSplex components. If not, issue a WTOR and then wait.		SCI, CQS, OM, RM
DBRC	IMS	SCI if ARLN is enabled		SCI if ARLN is enabled

10.2 Start the CSL address spaces

The CSL address spaces (SCI,CQS, OM and RM) should be started before the IMS address spaces are started, although if any of them are not there when IMS tries to connect, IMS will give a warning message with an option to cancel or retry.

10.2.1 Structured Call Interface

The first address space to be started should always be the Structured Call Interface (SCI) address space. It can be started using the standard z/OS start command (*START procedure_name*) as in this example:

```
S IM1ASC
```

After the SCI address space has been initialized successfully, SCI issues this message:

```
CSL0020I SCI READY IM1ASC
```

You can specify the following parameters to override the start procedure definitions:

- BPECFG

  Name of BPE configuration member in the PROCLIB

- SCIINIT

  Suffix for your CSLSIxxx member in the PROCLIB

- PARM1

  Other override parameters:

ARMRST	Indicates if ARM should be used
SCINAME	Name of the SCI being started

Here is an example of a start command with parameter overrides:

```
S IM1ASC,PARM1 ='ARMRST=Y,SCINAME=IM1ASC'
```

10.2.2 Common Queue Server

After the SCI address space is up, start CQS next. OM does not need CQS but both IMS and RM will register with CQS. CQS may be started by IMS if also serving as shared queues manager, but it must be started by user if used only for RM. It can be started using the standard z/OS start command as in this example:

```
S IM1ACQS
```

After the CQS address space has been started, it issues messages for connecting the CF structures, taking the checkpoints and eventually this message:

```
CQS0020I CQS READY CQ1ACQS
```

Note that the CQS ready message identifies the component name (CQ1A) and the component type (CQS). This is also true for all other CSL address spaces. Example 10-1 shows an example of messages issued when CQS is started and it connects to the shared queues structures and to the resource structure.

Example 10-1 Joblog for a CQS start

```
IEF695I START IM1ACQS  WITH JOBNAME IM1ACQS  IS ASSIGNED TO USER STC     , GROUP SYS1
$HASP373 IM1ACQS  STARTED
IEF403I IM1ACQS - STARTED - TIME=16.18.40 - ASID=0067 - SC42
CQS0008W STRUCTURE IMOA_MSGP        IS VOLATILE; CONSIDER STRUCTURE CHECKPOINT CQ1ACQS
IXL014I IXLCONN REQUEST FOR STRUCTURE IMOA_MSGP  122
WAS SUCCESSFUL.  JOBNAME: IM1ACQS ASID: 0067
CONNECTOR NAME: CQSCQ1ACQS CFNAME: CF05
CQS0008W STRUCTURE IMOA_EMHP        IS VOLATILE; CONSIDER STRUCTURE CHECKPOINT CQ1ACQS
IXL014I IXLCONN REQUEST FOR STRUCTURE IMOA_EMHP  124
WAS SUCCESSFUL.  JOBNAME: IM1ACQS ASID: 0067
CONNECTOR NAME: CQSCQ1ACQS CFNAME: CF05
CQS0008W STRUCTURE IMOA_RSRC        IS VOLATILE; CQ1ACQS
IXL014I IXLCONN REQUEST FOR STRUCTURE IMOA_RSRC  126
WAS SUCCESSFUL.  JOBNAME: IM1ACQS ASID: 0067
CONNECTOR NAME: CQSCQ1ACQS CFNAME: CF05
CQS0008W STRUCTURE IMOA_MSGO        IS VOLATILE; CONSIDER STRUCTURE CHECKPOINT CQ1ACQS
IXL014I IXLCONN REQUEST FOR STRUCTURE IMOA_MSGO  128
WAS SUCCESSFUL.  JOBNAME: IM1ACQS ASID: 0067
CONNECTOR NAME: CQSCQ1ACQS CFNAME: CF05
IXL014I IXLCONN REQUEST FOR STRUCTURE IMOA_EMHO  131
CQS0008W STRUCTURE IMOA_EMHO        IS VOLATILE; CONSIDER STRUCTURE CHECKPOINT CQ1ACQS
```

```
WAS SUCCESSFUL.  JOBNAME: IM1ACQS ASID: 0067
CONNECTOR NAME: CQSCQ1ACQS CFNAME: CF06
IXL015I STRUCTURE ALLOCATION INFORMATION FOR  132
STRUCTURE IMOA_EMH0, CONNECTOR NAME CQSCQ1ACQS
 CFNAME     ALLOCATION STATUS/FAILURE REASON
 --------   --------------------------------
 CF06       STRUCTURE ALLOCATED
 CF05       PREFERRED CF ALREADY SELECTED
CQS0030I SYSTEM CHECKPOINT COMPLETE, STRUCTURE IMOA_MSGP       , LOGTOKEN 000000001262122F CQ1ACQS
CQS0030I SYSTEM CHECKPOINT COMPLETE, STRUCTURE IMOA_EMHP       , LOGTOKEN 0000000003062ADD CQ1ACQS
CQS0020I CQS READY CQ1ACQS
CQS0030I SYSTEM CHECKPOINT COMPLETE, STRUCTURE IMOA_EMHP       , LOGTOKEN 0000000003065369 CQ1ACQS
CQS0030I SYSTEM CHECKPOINT COMPLETE, STRUCTURE IMOA_MSGP       , LOGTOKEN 0000000012623ABB CQ1ACQS
```

CQS will register with SCI. If SCI is not available when CQS tries to register, a warning message is issued but CQS continues initialization. CQS does not require SCI to be available to complete initialization, and will not abend as RM and OM do if SCI is not available.

```
CQS0001E CQS INITIALIZATION ERROR IN ... CSLSCREG ...
```

You can specify the following parameters to override the start procedure definitions:

- BPECFG

  Name of BPE configuration member in the PROCLIB

- CQSINIT

  Suffix for your CQSIPxxx member in the PROCLIB

- PARM1

  Other override parameters:

SSN	CQS subsystem name
STRDEFG	Suffix of CQSSGxxx member
STRDEFL	Suffix of CQSSLxxx member

10.2.3 Operations Manager

Operations Manager only needs SCI to complete initialization. IMS and RM both register with OM. It can be started using the standard z/OS start command as in this example:

```
S IM1AOM
```

After the OM address space has been initialized successfully, OM issues this message:

```
CSL0020I OM READY IM1AOM
```

If SCI is not available when OM is initialized, a warning message will be issued.

```
CSL0003A OM IS WAITING FOR SCI OM1OM
```

OM retries the registration every six seconds and, if SCI is still not available after ten attempts, OM will abend with a U0010.

```
CSL0002E IMSPLEX INITIALIZATION ERROR IN ...
```

You can specify the following parameters to override the start procedure definitions:

- BPECFG

  Name of BPE configuration member in the PROCLIB

- OMINIT

  Suffix for your CSLOIxxx member in the PROCLIB

- PARM1

  Other override parameters:

ARMRST	Indicates if ARM should be used
CMDLANG	Language for command description text
CMDSEC	Command security method
OMNAME	Name of the OM being started

10.2.4 Resource Manager

OM should be brought up before RM, so that RM can register commands with OM. If OM is not up yet, RM issues an error message and proceeds with initialization anyway. RM will register with OMs later as they come up.

Resource Manager registers with SCI, OM, and CQS. IMS registers with RM. CQS must be brought up before RM, for RM initialization to succeed. If the resource structure is used and CQS is not active, RM will try registering to CQS a few times before giving up and abending. Once SCI, OM, and CQS are up, you can bring RM up. It can be started using the standard z/OS start command as in this example:

```
S IM1ARM
```

After the RM address space has been initialized successfully, RM issues this message:

```
CSL0020I RM READY IM1ARM
```

If SCI is not available when RM tries to register, a warning message will be issued. RM will retry six times, 10 seconds apart. If SCI still is not available, RM will abend with a U0010.

```
CSL0003A RM IS WAITING FOR SCI RM1RM
```

If at least one OM is not started before RM, RM takes the following action:

- Issues a warning message and continues initialization
- Registers with any OM to start later

You can specify the following parameters to override the start procedure definitions:

- BPECFG

  Name of BPE configuration member in the PROCLIB

- RMINIT

  Suffix for your CSLRIxxx member in the PROCLIB

- PARM1

  Other override parameters:

ARMRST	Indicates if ARM should be used
RMNAME	Name of the RM being started

10.2.5 Start the IMS control region

When all of the CSL address spaces have been started (SCI, OM, and RM), and CQS is up, IMS can be started. For block level data sharing, you also need an IRLM address space in the same LPAR that IMS can connect to.

IMS will register with SCI, OM, RM, and CQS (for shared queues). A DBRC address space may also register with SCI if automatic RECON loss notification is being enabled. If any one of these is not available, IMS will issue a warning message and continue initialization.

```
DFS3306A CTL REGION IS WAITING FOR SCI|OM|RM
```

If any of these address spaces is not available when IMS completes initialization, IMS will issue a WTOR message indicating which one is not available and then wait.

```
DFS3309A CONTROL REGION WAITING FOR csltype REPLY RETRY OR CANCEL
```

If the reply is CANCEL, IMS abends with a U3309 RC=12.

IMS warm start and emergency start

During an IMS warm start and emergency restart, RM will check the RM structure for all resources with SRM=LOCAL and end-user status. If the resource is no longer owned by that IMS, any local status will be deleted.

IMS cold start

An IMS cold start is required for any of the following reasons:

- First time using CSL (CSLG=xxx in DFSPBxx)
- Restarting after missing one or more global online changes
- When changing the global online change option (OLC=GLOBAL to LOCAL or vice versa)
- When changing the use of the resource structure (using it to not using it, or vice versa)

10.3 Shutting down IMSplex address spaces

CSL address spaces do not shutdown automatically when the related IMS components terminate. There are two ways that CSL address spaces can be stopped:

1. Via the z/OS STOP command (P), by entering:
   - P jobname
2. Via a modify command to the SCI address space, using the SHUTDOWN command. The SHUTDOWN command can shutdown either the CSL address spaces on the local z/OS, or all CSL address spaces in the IMSplex. To shutdown the local CSL members (the CSL address spaces on the same LPAR), use this command:

   ```
   F scijobname,SHUTDOWN CSLLCL
   ```

   To shutdown all CSL members in the IMSplex, use this command:

   ```
   F scijobname,SHUTDOWN CSLPLEX
   ```

> **Note:** The *F SCI* command does not shut down any of the CQS address spaces. CQS can be stopped only after all local RMs are stopped. Use the z/OS stop command.

When shutting down an IMS which is part of an IMSplex, the user should always shut down IMS first, then the CSL address spaces, and CQS. IMS has some cleanup work to do at shutdown which it cannot do if CSL is not active.

After IMS is down, the other address spaces can be stopped. They can be stopped individually or as a group. When shutting them down individually, the recommended sequence is RM, OM, CQS, and SCI, as in the following example:

```
P IM1ARM
P IM1AOM
P IM1ACQS
P IM1ASC
```

After the CSL address spaces have been stopped, the following type of CSL0021E message is issued for each of them (this message is for RM):

```
CSL0021E RM SHUTDOWN COMPLETE IM1ARM
```

10.3.1 IMS shutdown

IMS can be shutdown normally with or without the LEAVEPLEX keyword.

Normal shutdown without LEAVEPLEX keyword

If IMS is shut down normally using the /CHE FREEZE command:

- IMS1 updates RM structure according to SRM and terminal/user status.
- For resources with no end-user or command significant status.
  - Delete node, (static node) user, LTERM, and userid entries.
  - VTAM Generic Resources affinity deleted by IMS or VTAM.
- For resources with command status and no end-user status.
  - Keep entries but clear ownership (except userid always deleted).
  - VTAM Generic Resources affinity deleted by IMS or VTAM.
- For resources with end-user status and not SRM=NONE.
  - Keep entries (except userid always deleted).
  - If SRM=GLOBAL, delete local status, clear ownership.
  - If SRM=LOCAL, keep local status (log records), do not clear ownership.

Shutdown with LEAVEPLEX keyword

If IMS is shut down using the command /CHE FREEZE LEAVEPLEX:

- All resources owned by this IMS are cleared (no ownership). If no significant status exists, the resource entry is deleted. Cleanup RM structure:
  - For all SRMs if no significant status, delete entries. If significant status, keep entries but clear ownership.
  - If SRM=LOCAL, end-user status is lost since IMS1 not coming back.
  - If IMS1 is warm started and rejoins IMSplex, status is deleted since IMS1 is no longer owner of resource.
- IMSs local member is removed from the resource structure (DFSSTMLimsid).
- IMSs entry in the //OLCSTAT online change status data set is removed.
- LEAVEPLEX also implies LEAVEGR, so all VTAM generic resource affinities will be deleted.

You should use the LEAVEPLEX keyword only if your intention is not to restart this IMS.

10.3.2 SCI shutdown while other CSL members are active

If SCI is terminated (normally or abnormally) while other members of the IMSplex are active on the same z/OS system, the other members will be notified that SCI has terminated via the notify exit. They will not be able to communicate with other members of the IMSplex through SCI until SCI is restarted, because the SCI address space must be active on the local z/OS system to communicate within the IMSplex.

Once SCI is restarted, SCI will re-establish any connection or registration that was active when SCI terminated. The SCI token for each connection will still be valid and can be used

when SCI has completed its restart. Each member will be notified when SCI is ready via the notify exit.

10.4 BPE commands

BPE provides a set of commands that you can issue to any IMS component that is running in a BPE environment (CQS, RM, OM, SCI).

10.4.1 BPE command syntax and invocation

BPE supports two command formats:

- Verb only format

  The verb only format consists of a verb, followed by zero or more keyword-value pairs, with the values enclosed in parentheses.

- Verb-resource type format

  The verb-resource type format consists of a verb, a resource type, and zero or more keyword value pairs.

You can only invoke BPE commands through the z/OS MODIFY command:

```
F jobname,command
```

In the MODIFY command, the jobname is the jobname of the address space to which the command is directed, and command is the command being issued.

10.4.2 BPE command examples

Other than starting and stopping, we show here only some examples of commands that affect the operation of the CSL address spaces. See *IMS Version 8: Base Primitive Environment Guide and Reference,* SC27-1290, for details of BPE commands.

DISPLAY TRACETABLE

Display trace level, as seen in Example 10-2, shows a command and the response for all BPE trace tables used for SCI.

Example 10-2 Display trace level

```
F IM1ASC,DIS TRTAB NAME(*)

BPE0030I TABLE  OWNER  LEVEL  #PAGES
BPE0000I AWE    BPE    HIGH        6
BPE0000I CBS    BPE    MEDIUM      6
BPE0000I CMD    BPE    LOW         2
BPE0000I CSL    SCI    HIGH        8
BPE0000I DISP   BPE    HIGH       12
BPE0000I ERPL   SCI    LOW         8
BPE0000I ERR    BPE    HIGH        2
BPE0000I ERR    SCI    HIGH        4
BPE0000I HASH   BPE    MEDIUM      8
BPE0000I INTF   SCI    LOW         8
BPE0000I INTP   SCI    LOW        16
BPE0000I LATC   BPE    LOW         8
BPE0000I MISC   BPE    LOW         1
BPE0000I PLEX   SCI    LOW         8
BPE0000I SCI    SCI    HIGH        8
```

```
BPE0000I SSRV   BPE    LOW         4
BPE0000I STG    BPE    LOW         8
BPE0000I USRX   BPE    LOW         4
BPE0032I DIS TRTAB COMMAND COMPLETED
```

UPDATE TRACETABLE

The UPDATE TRACETABLE command can be used to change the trace level. Example 10-3 shows the command and the response for CQS.

Example 10-3 Update tracetable

```
F IM1ACQS,UPDATE TRACETABLE NAME(DISP),LEVEL(HIGH)

BPE0032I UPDATE TRACETABLE COMMAND COMPLETED CQ1ACQS
```

DISPLAY USEREXIT

DISPLAY USEREXIT command displays user exit information. Example 10-4 shows the command and the response for CQS.

Example 10-4 Display userexit

```
F IM1ACQS,DIS USRX NAME(*)OWNER(CQS) SHOW(CALLS,ETIME,ABLIM)

BPE0030I EXITTYPE MODULE        ABLIM      CALLS      ETIME CQ1ACQS
BPE0032I DIS USRX COMMAND COMPLETED CQ1ACQS
```

REFRESH USEREXIT

The REFRESH USEREXIT command causes the reprocessing of user exit PROCLIB member. Example 10-5 shows the command and the response for RM.

Example 10-5 Refresh userexit

```
F IM1ARM,REF USRX NAME(*)

BPE0032I REF USRX COMMAND COMPLETED IM1ARM
```

DISPLAY VERSION

DISPLAY VERSION displays BPE and component versions. Example 10-6 shows the command and the response for CQS. The output of this command is a single line message containing the IMS component and BPE internal version number.

Example 10-6 Display version

```
F IM1ACQS,DISPLAY VERSION

BPE0000I CQS VERSION = 1.3.0  BPE VERSION = 1.4.0 CQ1ACQS
```

10.5 IMS commands

IMS commands may be entered to IMS either through traditional channels, such as the MTO, the WTOR reply, automated operator programs using the CMD or ICMD call, an E-MCS console, and now in IMS Version 8 through OM, for example, from a SPOC application.

There are several new commands available when running in an CSL environment. We refer to them by *IMSplex commands*. IMSplex commands can only be entered through the OM interface. We refer to traditional IMS commands by *classic commands*. These are those IMS commands starting with the slash (/). Most of these commands can be entered through the OM interface and through traditional means.

Note: New IMS commands were introduced with IMS Version 8, and they were originally referred to by IMSplex commands. However, the terminology is now changing from *IMSplex commands* to *enhanced commands*. This is because you don't have to have an IMSplex to make use of OM, SPOC and these commands. In this book we still are mainly using *IMSplex commands* when referring to these enhanced commands.

This chapter refers to all commands entered through the OM interface as having been entered from the SPOC. Those entered through the traditional interface will be referred to as having been entered from the MTO.

10.6 IMSplex commands

Some IMSplex command characteristics are:

- Can only be entered through OM interface, cannot be entered directly to IMS.
- Command Response returned to SPOC/AOP in XML format and must be translated by SPOC/AOP to display format.
- Filters and wildcards supported for resource name selection:
  - The percent sign (%) wildcard is used for single character substitution.
  - The asterisk (*) wildcard is used for multi character substitution.
- OM API commands are not passed to the AOI exits, DFSAOE00 or DFSAOUE0.

For a complete description of the format and use of these commands, refer to *IMS Version 8: Command Reference,* SC27-1291.

10.6.1 IMSplex commands description

Five new commands have been added to the IMSplex environment. These commands are:

- INITIATE

  Initiate an IMSplex global process. In IMS Version 8, the only global process is global online change.

  INIT OLC starts a coordinated online change, for example:

  ```
  INIT OLC PHASE(PREPARE) TYPE(ALL)
  ```

- TERMINATE

  Terminate process. This command is used to terminate a global process when something goes wrong. For global online change, this would be the equivalent of the /MODIFY ABORT command.

  TERM OLC stops a coordinated online change that is in progress.

- DELETE

  Delete resource. This command is used to delete a resource. In IMS Version 8, the only resource that can be deleted is the dynamic LE runtime parameters added earlier with the UPD LE command.

DEL LE deletes runtime LE options, for example:

```
DEL LE TRAN(TTT) LTERM(LLL) USERID(UUU) PGM(PPP)
```

- UPDATE

  Update resource. This command is used to update a resource. In IMS Version 8, two resources may be updated: the dynamic LE runtime parameters, and some transaction attributes. A new X'22' log record is written for IMSplex UPDATE TRAN command.

  UPD LE updates runtime LE options.

  UPD TRAN updates selected TRAN attributes. For example:

```
UPD TRAN(TTT) LTERM(LLL) USERID(UUU) PGM(PPP) SET(LERUNOPTS(XXXXXX))
UPD TRAN NAME(TTT) STOP(Q,SCHD) SET(CLASS(4))
```

- QUERY

  Query resource. The Query command has several parameters. All of the QRY commands include a SHOW parameter to specify what fields you want returned, or you can say SHOW(ALL).

  QRY IMSPLEX returns the names of the members in the IMSplex. IMSPLEX displays information about the IMSplex itself. It includes for each member of the IMSplex: its OS/390 name, jobname, status, type (for instance, IMS, DBRC, OM, RM, AOP) and subtype (for instance, DBDC, DCCTL, DBCTL, FDBR).

  QRY LE returns runtime LE options. LE displays the dynamic LE runtime parameters set by an earlier UPD LE command, including any filters and what the LERUNOPTS are.

  QRY MEMBER returns status and attributes of the IMS members in the IMSplex. MEMBER displays information about a specific member by member type. For IMS Version 8, TYPE(IMS) is the only type supported. It shows the status some of the attributes of each member of that type, such as the global online change status for each IMS, or whether this IMS is using shared queues.

  QRY OLC returns global online change library and resource information. OLC displays information from the new OLCSTAT data set. This information includes the OLCSTAT data set name, the suffixes for the active online change libraries, the online change modid, the type of the last online change, and all IMS members who are currently in the global online change group and current with the active libraries.

  QRY TRAN returns TRAN info similar to /DIS TRAN. TRAN displays transaction attributes. The information displayed is equivalent to the /DIS TRAN command output, except that it can show the output from multiple IMSs. It also shows the global queue count if those IMSs are in a shared queues group.

  QRY STRUCTURE returns structure information of the RM resource structure, including the number of entries and elements allocated and in use, and the entry-to-element ratio.

All of these commands have global scope. That is, they apply to all IMSs in the IMSplex. For example, if the UPDATE TRAN command is entered to change the MSGCLASS of a transaction, it will be changed on all the IMSs in the IMSplex.

10.6.2 QUERY IMSPLEX command example

Figure 10-1 shows a QUERY IMSPLEX SHOW(ALL) command entered through TSO SPOC and results formatted by TSO SPOC.

```
  File  Display  View  Options  Help
 PLEX1                    IMS Single Point of Control
 Command ===> _____________________________________________________________
 ____________________________________ Plex . PLEX1  Route . __________  Wait . ______
 Response for: QRY IMSPLEX SHOW(ALL)
IMSplex  MbrName    CC Member   JobName  Type  Subtype  Version  OSName   Status
CSLPLEX1 IM1AOM      0 IM3ARM   IM3ARM   RM    MULTRM   1.1.0    SC69     READY,
CSLPLEX1 IM1AOM      0 IM1ADBRC IM1ADBRC DBRC           8.1.0    SC42     READY,
CSLPLEX1 IM1AOM      0 CQ1ACQS  IM1ACQS  CQS            1.3.0    SC42     ACTIVE
CSLPLEX1 IM1AOM      0 IM3ASC   IM3ASC   SCI            1.1.0    SC69     READY,
CSLPLEX1 IM1AOM      0 JOUKO5   JOUKO5   AOP            1.1.0    SC42     ACTIVE
CSLPLEX1 IM1AOM      0 IM3AOM   IM3AOM   OM             1.1.0    SC69     READY,
CSLPLEX1 IM1AOM      0 IM3A     IM3ACTL  IMS   DBDC     8.1.0    SC69     READY,
CSLPLEX1 IM1AOM      0 IM1AOM   IM1AOM   OM             1.1.0    SC42     READY,
CSLPLEX1 IM1AOM      0 IM1A     IM1ACTL  IMS   DBDC     8.1.0    SC42     READY,
CSLPLEX1 IM1AOM      0 IM3ADBRC IM3ADBRC DBRC           8.1.0    SC69     READY,
CSLPLEX1 IM1AOM      0 IM1ASC   IM1ASC   SCI            1.1.0    SC42     READY,
CSLPLEX1 IM1AOM      0 CQ3ACQS  IM3ACQS  CQS            1.3.0    SC69     ACTIVE
CSLPLEX1 IM1AOM      0 IM1ARM   IM1ARM   RM    MULTRM   1.1.0    SC42     READY,
CSLPLEX1 IM1AOM      0 JOUKO4   JOUKO4   AOP            1.1.0    SC42     ACTIVE
CSLPLEX1 IM1AOM      0 CQ4ACQS  IM4ACQS  CQS            1.3.0    SC67     ABENDE
```

Figure 10-1 QUERY IMSPLEX command

10.6.3 QUERY STRUCTURE command example

Figure 10-2 shows a QUERY STRUCTURE command entered through TSO SPOC and results formatted by TSO SPOC.

```
  File  Display  View  Options  Help
 PLEX1                    IMS Single Point of Control
 Command ===> _____________________________________________________________
 ____________________________________ Plex . PLEX1  Route . __________  Wait . ______
 Response for: QRY STRUCTURE NAME(*) SHOW(ALL)
StructureName    MbrName    CC Type     LeAlloc LeInuse ElmAlloc ElmInuse LE/EL>
IM0A_RSRC        IM4ARM      0 RSRC       3651      63     3650       23 0001/0
IM0A_RSRC        IM3ARM      0 RSRC       3651      63     3650       23 0001/0
IM0A_RSRC        IM1ARM      0 RSRC       3651      63     3650       23 0001/0
```

Figure 10-2 QUERY STRUCTURE command

This command is currently supported only for the resource structure. For shared queues structures, use the classic command:

```
/CQQ STATISTICS STRUCTURE ALL
```

10.6.4 QUERY TRAN command example

Figure 10-3 shows a QUERY TRAN command entered through TSO SPOC and results formatted by TSO SPOC.

The query command can display any or all transaction attributes, including QCNT. When QCNT is requested in a shared queues environment, only the master queries CQS for the counts and displays them. Other attributes are displayed by all IMSs receiving the command.

```
  File  Display  View  Options  Help
 PLEX1                    IMS Single Point of Control
 Command ===> _____________________________________________________________
 ____________________________________ Plex . PLEX1  Route . __________  Wait . ______
 Response for: QUERY TRAN NAME(TPCCO) SHOW(ALL)                         More:    >
Trancode MbrName    CC PSBname      QCnt LCls    LQCnt  LLCT LPLCT LCPRI LNPRI
TPCCO    IM4A        0               339
TPCCO    IM1A        0 PSBOS                3        0 65535    10     1     1
TPCCO    IM3A        0 PSBOS                3        0 65535    10     1     1
TPCCO    IM4A        0 PSBOS                3        0 65535    10     1     1
```

Figure 10-3 QUERY TRAN command

10.6.5 UPDATE TRAN

Figure 10-4 shows a UPDATE TRAN command entered through TSO SPOC and results formatted by TSO SPOC.

Transactions have no global status. An UPD TRAN command must be routed to all IMSs where the status update is required.

```
  File  Display  View  Options  Help
 PLEX1                    IMS Single Point of Control
 Command ===> _____________________________________________________________
 ____________________________________ Plex . PLEX1  Route . __________  Wait . ______
 Response for: UPDATE TRAN NAME(TPCCN) START(Q,SCHD,SUSPEND)
Trancode MbrName    CC
TPCCN    IM1A        0
TPCCN    IM3A        0
TPCCN    IM4A        0
```

Figure 10-4 UPDATE TRAN command

10.7 IMS classic commands

As with the IMSplex command, refer to *IMS Version 8: Command Reference,* SC27-1291, for a complete description of the use of these commands in an IMSplex.

Figure 10-5 shows a /DIS STATUS TRAN command.

```
  File  Display  View  Options  Help
 PLEX1                    IMS Single Point of Control
 Command ===> _____________________________________________________________
 ____________________________________ Plex . PLEX1  Route . __________  Wait . ______
 Log for . . : DIS STATUS TRAN

 IMSplex . . . . . :
 Routing . . . . . :
 Start time. . . . : 2003.125 18:49:08.97
 Stop time . . . . : 2003.125 18:49:08.98
 Return code . . . : 00000000
 Reason code . . . : 00000000
 Command master. . : IM4A

 MbrName  Messages
 -------- -----------------------
 IM4A          **TRAN****PSBNAME
 IM4A          TPCCO     PSBOS     USTOPPED
 IM4A          TPCCP     FPSBPA    USTOPPED
 IM4A          *2003125/184908*
 IM3A          **TRAN****PSBNAME
 IM3A          STATUS UNRESTRICTED
 IM3A          *2003125/184908*
 IM1A          **TRAN****PSBNAME
 IM1A          TPCCO     PSBOS     USTOPPED
 IM1A          *2003125/184908*
```

Figure 10-5 /DIS STATUS TRAN command

10.7.1 Command entry

Most IMS classic commands can be entered through the SPOC or the MTO. However, not all classic commands can be entered through the SPOC. Note also that OM API commands are not passed to the AOI exits, DFSAOE00 or DFSAOUE0 and that DL/I CMD or ICMD calls do not support OM API.

In the IMSplex environment, classic IMS commands can be entered locally from IMS terminal, system console, E-MCS console and AO program/exit. That IMS is considered the command

master, and the command is executed locally only. The command may update resource entry if SRM=GLOBAL.

Most, but not all, classic IMS commands can be entered also through OM. Then they may be routed to all active IMSs, or to specific IMSs. One IMS system is considered the command master, commands are executed in one or more IMSs and the command responses may come from multiple IMSs. The responses are concatenated by OM into one buffer exactly as received, and that buffer is passed on to AOP (for example, SPOC). With some responses, like DFS554 message, there is still need existing AOI programs to start resources.

For example, these commands cannot be entered through OM:

- /EXC
- /FORMAT
- /HOLD
- /SIGN
- /SET
- /(UN)LOCK LTERM|NODE|PTERM
- /HOLD

Generally speaking these are the commands that affect the terminal from which they are entered. Since there is no IMS terminal associated with the SPOC, these commands would not make sense.

10.7.2 Command scope

When an IMS is running as part of an IMSplex with sysplex terminal management enabled, the effect of an IMS command can have both local and global impact. For example, a /STOP NODE command will change the command significant status of a NODE and cause a resource entry for the NODE to be created or updated with the STOPPED flag on.

Other commands only have local impact. For example, a /STOP TRAN command would stop a transaction only on the IMS where the command was executed.

Generally speaking, commands which affect terminal resources whose status is maintained on a resource structure have global scope. These are commands which change the command or end-user significant status of a NODE, USER, or LTERM. All others have local scope.

If RM structure does not exist:

- Commands are processed as prior versions of IMS.
- All commands have local significance only.

10.7.3 IMS asynchronous command response

When command response includes a synchronous DFS058I message followed by one or more asynchronous messages (for example, DFS0488I), only the messages indicating the command completed are returned, and the synchronous DFS058I message is not returned through OM API.

10.7.4 Command changes in IMS Version 8

Some IMS classic commands have changed in IMS Version 8. The following commands are no longer supported:

- /TEST LINE | NODE | USER

The statuses set by the following commands are no longer recoverable across session or IMS restart:

- /LOCK LTERM | NODE | PTERM
- /UNLOCK LTERM | NODE | PTERM
- /PSTOP LTERM
- /PURGE LTERM
- /RESET
- /SET LTERM | TRAN
- /TEST (without MFS keyword)

Some commands depend on ownership, for example, the commands that change terminal or user status and the commands that display status.

Some commands function differently; they may be effective locally, globally, or both, they may reset non-recoverable status and set status in RM, or they may display global, local status, or new status (for example, recovery).

Note: All status is recoverable for class 1 terminals across an XRF takeover. New active assumes status of failing IMS based on log records.

10.7.5 Command master

When an IMS classic command is entered through the SPOC to one or more IMSs, OM will select one of those IMSs as the command master. If it is submitted to only one IMS, that IMS is the master. If the command is submitted through the MTO, that IMS is the master.

Some classic commands can only be executed by the command master. Others can only be executed by the resource owner. Still others will be executed by every IMS which receives the command. There are a few basic rules for which IMS will execute the command.

Rules for the commands which update significant status

- If the resource is owned:

  Only that IMS will process the command. Other IMSs, including the master, will reject the command. For example, if NODEA is owned by IMS1, a /STOP NODE NODEA will only be executed by IMS1. It will stop the NODE locally and update the status globally to STOPPED. The NODE would then not be allowed to log on to any IMS in the IMSplex.

- If the resource is not owned:

  Only the master will process the command. Others will reject it. For example, if a NODE does not exist in RM, then it is not owned. A /STOP NODE command will be executed by the master which will create a NODE entry and set the STOPPED flag.

Exceptions to these rules

- ISC parallel session terminals
  - Since parallel sessions can exist on multiple IMSs, each IMS will modify status for parallel sessions (users) active on that IMS.
  - The master system will also modify status relating to all parallel sessions (for example, /TRACE NODE without USER keyword).
  - /TRACE command is processed locally on each system, so that trace status will exist on each system.
- The /ASSIGN LTERM ... TO NODE ... command for static terminals is processed on each system, to ensure consistency

Rules for the commands which display status

- If the resource is owned:

  Only the owning IMS will display the global status. Other IMSs will display any local status. For example, if NODEB is owned by IMS2, then IMS2 would display its global and local status.

- If the resource is not owned:

  The master will display global status and local status. Others will display local status only. In the example above where a NODE has been stopped globally, but is not owned, if a /DIS NODE NODEA command is sent to all IMSs, only the master would display the STOPPED status (this is a global status maintained only in the resource structure). All IMSs, including the master, would reply with their local status, which for a NODE is usually just IDLE.

Exception to these rules

- /DISPLAY NODE for ISC parallel session terminals
  - Since parallel sessions can exist on multiple IMSs, each IMS will display status for parallel sessions active on that IMS.
  - The master system will display local status first (parallel sessions active on that IMS), followed by unowned parallel sessions (known only in RM) and status relating to all parallel sessions (for example, TRACE status in NODE entry).

10.7.6 Commands considerations in a STM environment

Here we describe some of the unique considerations when entering classic IMS commands in a STM environment. However, this document does not describe all of the differences that command processing has when running IMS in a CSL environment with STM enabled. It would be well worth the effort to read the *IMS Version 8: Command Reference,* SC27-1291, very carefully, and then to test all of the commands under a variety of conditions.

Display SRM and RCVY

The /DISPLAY command has been enhanced to display the owner, and the SRM and RCVY values of a resource.

```
/DIS NODE NODE1 RECOVERY

NODE-USR  OWNER     SRM       CONV   STSN   FP
NODE1     IMS1      GLOBAL     Y      Y      Y
```

Assigning LTERMs

There are some rules when assigning LTERMs:

- /ASSIGN LTERM is permitted between BTAM and VTAM, but BTAM status is not maintained.
- /ASSIGN command is not allowed if source and destination are owned by two different IMS systems.
- /ASSIGN command without the SAVE keyword is not allowed if destination does not exist in the RM.
- LTERMs assigned from VTAM to BTAM are deleted from RM.
- LTERMs assigned with SAVE from BTAM to VTAM are added to RM.

Commands not affected by STM

Not all commands are affected by sysplex terminal management. In general, these are the commands which do not reference sysplex terminals. For example, the following commands:

```
/DIS TRAN xxx QCNT
/CQQ STATISTICS STRUCTURE(ALL)
```

These commands do not follow the rules of command master. Each IMS which receives this command would retrieve and display global queue counts or statistics. These commands should only be routed to a single IMS to excessive structure access.

10.8 CSL operations summary

The world of CSL can have a significantly different and often confusing effect on operations. The processing of classic commands depends on:

- Source of the command (classic or OM API)
- Whether RM is active with a resource structure
- Whether the command affects significant status
- Whether the command parameters include ALL or a generic parameter
- Whether the resource exists on this IMS
- Whether the resource is owned by this IMS
- Whether the resource exists in RM
- Whether the resource is owned by another IMS
- Whether the command displays or updates terminal status
- Whether the resource is managed by STM

11

Sysplex terminal management (STM) operations

This chapter describes the operations management tasks of the sysplex terminal management (STM):

- STM operations
- IMS failures in a STM environment
- Managing the resource structure
- SCI, RM, CQS, or structure failures

11.1 STM operations

Sysplex terminal management operations include tasks related to IMS, SCI, OM, RM, and CQS. SCI should be started (on every OS/390 image) first, since every other address space will register with SCI to join the IMSplex.

If SCI is not started when IMS starts, it will put out a WTOR warning message and wait for a reply. If SCI is not available to the other address spaces, then will retry six times and then abend.

When the first IMS joins the IMSplex, IMS will register with RM and RM will register with CQS. CQS will connect to the resource structure, causing it to be created. At this time the IMSplex global entries will be created, as well as the local entry for this first IMS. As each additional component joins the IMSplex, their local entries will be created.

11.1.1 Log on from a static NODE

When a user logs on using a generic resource name, VTAM will check its VTAM Generic Resources affinity table and, if there is no affinity, route the logon request to one of the active IMSs in the generic resources group. The IMS that receives the logon request will check the resource structure for a NODE entry and, if the NODE is not already there, or is there but not owned, it will accept the logon. If the NODE entry exists and is owned by another IMS, the logon will be rejected. The logon will also be rejected if any LTERM assigned to that NODE is already owned. This is true even if only one of several LTERMs is owned by another IMS.

If this is a new logon, IMS will create the appropriate NODE, static NODE user, and LTERM entries. At this time the values for SRM and RCVYxxxx are set according to the system default or the Logon Exit, and if this is a STSN device, STSN significant status will be set in the static NODE user entry. If signon information is provided with the logon request, and if single signon is being enforced, then a userid entry is also created. If signon is done later using the /SIGN ON command, the entry is created (or not created) at that time.

VTAM sets the VTAM Generic Resources affinity to this IMS and IMS tells VTAM that is to manage the VTAM Generic Resources affinity according to the SRM.

11.1.2 Logon from an ETO NODE

Like with static NODEs, IMS will first make sure the NODE entry does not already exist, or if it does, that it is not owned. If it does not exist, it will accept the logon and create a NODE entry with itself as the owner. VTAM Generic Resources affinity management will be set according to the SRM and device type. Little else is done until the user signs on.

11.1.3 Signon from an ETO NODE

Most STM activity for ETO sessions takes place at signon time. At this time, the user and LTERM entries are created to match the user control block structure in IMS. Nearly all recoverable information about the NODE and the user are kept in this user entry. SRM and RCVYxxxx values are set, and if this is a STSN device, STSN status is set.

11.1.4 Commands that change significant status

When one of these commands is entered, then the NODE or (static NODE) user resource entry is updated accordingly. When LTERM status is changed, it is kept in the corresponding LTERM segment in the (Static NODE) User entry. When command significant status is reset

(turned off) then the resource entry is updated to reflect this. For example, a user may be put in MFSTEST mode by the /TEST MFS USER command. The /END command takes this user out of MFSTEST mode. The user entry would be updated accordingly and, if this is the last significant status, the resource entries would be deleted.

Sometimes a command may be entered for a resource which does not exist on the resource structure. When it sets significant status, for example, a /STOP NODE command to a NODE which does not exist, the resource will be created and the significant status set (for static NODEs, the static NODE user and LTERM entries are also created). This then becomes a global status and will prevent that NODE from logging on anywhere in the IMSplex. When a command resets (ends) significant status, if that resource has no other significant status and is not currently active, it will be deleted.

11.1.5 Work which changes end-user significant status

There are several differences between the processing of command and end-user significant status:

- End-user significant status cannot be set when no resource entry exists. It is always the result of end-user activity.
- The installation has the choice about where (and if) the end-user significant status is kept. and xxx for this discussion. When end-user significant status is changed, the appropriate (Static NODE) User entry is updated accordingly.
- End-user significant status, if it exists, causes at least two updates to the resource structure with every transaction.
  - For conversations, the conversational input message in-progress flag is set when the transaction is entered and reset when the response is delivered.
  - For Fast Path, the Fast Path input in-progress flag is set when the Fast Path (EMH) transaction is entered and reset when the response is delivered.
  - For STSN, the sequence numbers are updated for each input and output message.

11.1.6 Commands which change end-user status

Although no command can create end-user status, some commands can delete or change it. For example, if an ETO user is in a conversation but is currently signed off, the user and LTERM structure still exist in the resource structure. If SRM=GLOBAL, the /EXIT CONV USER command will delete the conversational data from the user entry and, if there is no other significant status, the user and LTERM entries will be deleted. Note that when SRM=GLOBAL and the resource is not active, this command can be entered from any IMS. If SRM=LOCAL, then the /EXIT command must be entered from the IMS which owns the resource. But the same rule would apply, when all significant status is gone, the entries are deleted.

11.1.7 Session termination with significant status (not IMS failure)

When a session terminates due to session failure, or because the user logs off (not because of IMS failure), IMS examines that session's resource entries to see if any significant status exists. If either command or end-user significant status exists, the resource entries will not be deleted. If end-user significant status exists, then clearing of ownership depends on the SRM.

If SRM=GLOBAL (or NONE), then ownership will be cleared since everything needed to recover the status is in the structure. In addition, IMS will move all locked messages for that LTERM. This is done to allow the user to log on to another IMS and access those messages.

If SRM=LOCAL, then ownership is not cleared since this is the only IMS that knows how to restore the end-user status. Locked messages remain locked.

11.1.8 Logon from NODE which already exists in resource structure

When a user logs on using a generic resource name, if a VTAM Generic Resources affinity exists, VTAM will route that logon request to the IMS to which it had an affinity. Usually the VTAM Generic Resources affinity will only continue to exist after session termination if SRM=LOCAL and it had significant end-user status when the session terminated, or IMS failed and the clean-up IMS didn't know whether or not it had significant status.

This is intended to route the user's next logon request to the IMS where its significant status is known. However, that user may elect to log on directly to an active IMS, bypassing VTAM Generic Resources. When this happens, the new IMS detects the ownership (RM affinity) in the resource entry and rejects the logon. That user then has to make a choice between these options:

- Wait for the failed IMS to restart.

  When the failed IMS is restarted and the user logs on, its significant status is restored and work continues.

- Log on with *user data* telling the logon exit to steal the NODE if it is owned by an inactive IMS.

  The ownership of that resource is changed and the user loses any end-user significant status that might have existed on the failed IMS. When the failed IMS is restarted, it will discover that it no longer owns the resource and will delete any local status.

  When a user logs on and IMS finds a resource entry without an owner (must have had significant status and not SRM=LOCAL), then the logon request is accepted, the ownership is set to that IMS, and the significant status is recovered. Life is good!

11.2 IMS failure

When an IMS system fails, it has no opportunity to clean up its entries on the resource structure. However, since all the IMSs in the IMSplex are members of the same IMSplex, SCI will inform other surviving IMSs that one has failed. One of those IMSs will get ownership of the failing IMS's DFSSTMLimsid entry and request, from the Resource Manager, a list of all resources owned by the failing IMS. It will then proceed to clean up the resource structure by the following manner:

- If a resource has no significant status and SRM=GLOBAL or NONE, then the resource entry is deleted.
- If a resource has command significant status, or end-user significant status with SRM=GLOBAL, then the resource entry is not deleted but ownership is released.
- If a resource has SRM=LOCAL, the surviving IMS does not know whether or not that user had significant end-user status being maintained by its local (failed) IMS, so it does not delete the resource and it does not clear ownership. That user must wait for the failed IMS to restart before logging back on. Note that in this case, the VTAM Generic Resources affinity would have been IMS-managed and therefore not deleted.

The surviving IMS also does some clean-up work on the shared queues structure. For all resources which were not SRM=LOCAL, IMS requests CQS to move all output messages which were locked (on the LOCKQ) by the failing IMS back to the LTERM Ready Queue (LRQ). These messages would include:

- Output messages which were in the process of being delivered by the failed IMS but which had not completed (Q1 and Q4 messages)
- Active SPA+MSG for the last conversational output message for users who were between iterations of a conversation (Q5 messages).
- SPA+MSG for every held conversation (Q5 messages)

These messages may join output messages which were on the LRQ at the time of the failure:

- Responses to transactions which had not yet been retrieved by the failing IMS (Q1 messages)
- Unsolicited messages not in response to an input transaction (Q4 messages)

Note that input messages (transactions) are not unlocked. This is because only the failed IMS knows the status of that message. It may have been in-flight, or it had reached sync point, but IMS failed before it was able to delete it from the lock queue. The status of that message will not be known until the failed IMS is emergency restarted.

At the end of this process, all output messages locked by the failed IMS have been returned to the LTERM ready queue, including between iteration conversations and held conversations. Output messages on the LTERM ready queue with destinations previously active on the failed IMS are still there. Input messages from the failed IMS which had not yet been scheduled are still on the transaction ready queue and can be scheduled anywhere. Responses to these messages will be put on the LTERM ready queue. This concludes the work of the clean-up IMS.

Sometimes an IMS may fail and there is no surviving IMS to clean up. In this case, of course, no clean up work is done until the next IMS restart, either the failed IMS or another. When restart completes, that IMS will query RM for any IMSs which need to be cleaned up and then performs the clean up.

11.3 IMS emergency restart

While IMS was executing (before the failure) it knew and logged the status of each of its logged on and signed on users. At restart, it recovers this information from log records.

- For those which had SRM=GLOBAL or NONE, it will delete any status, because it was either not recoverable or it was (and perhaps still is) on the structure.
- But if SRM=LOCAL, then it must decide what to do. It knows that there should be resource entries on the structure since the clean-up IMS would not have deleted them if SRM=LOCAL. But it doesn't know if that user was able to successfully log on to another IMS and continue processing. So, it checks the resource structure for any entries and, if they still exist, who the owner is.

  If that IMS is still the owner, then the user must not have logged on to another IMS and the local status is still valid. It is not deleted.

  If the resource is gone from the structure, or if it is there but has a different owner, or no owner, its local status is no longer valid and will be deleted.

In all cases, the restarting IMS does not change the resource entries.

11.4 Recovering significant status

Recovery of command significant status is relatively easy. Simply set the appropriate status in the local resource control blocks. Recovering end-user significant status globally is somewhat more complex.

11.5 Recovering conversations

When a conversational transaction is entered, that user acquires end-user significant status. Where that status is kept depends on the SRM value:

- SRM=GLOBAL

  The status is kept in the resource structure.
- SRM=LOCAL

  The status is kept only in local control blocks.
- If SRM=NONE

  The status is kept only in the local control blocks and is deleted when the session terminates.

In all cases, the value of SRM is kept in the resource structure entry.

The key to being able to recover a conversation is knowing that a user was in conversational mode and what the status of that conversation was at the time of an IMS failure or session termination.

The primary pieces of information kept about a user's conversational status are the conversation ID and transaction code for the active conversation and all held conversations. Also a flag is kept that indicates whether the user has a conversational transaction in progress (as opposed to being between iterations). This flag is important for knowing whether and how the conversation can be recovered on another IMS. We will call this the *conversational input in-progress* (CONV-IP) flag. It is set when a conversational transaction is queued, and is reset when the response is delivered.

11.5.1 Example of conversation recovery

The following example illustrates the events that might occur in establishing and recovering a conversation. Assume that there are two IMSs: IMS1 and IMS2. IMS0 is the GRSNAME for both IMSs. TRANC is defined as a conversational transaction to both IMSs. Both IMSs have static definitions for NODEA and LTERMA.

Normal operation

Here is the normal operation in establishing and recovering a conversation.

- Users logs on and signs on successfully to IMS1 (logon IMS0).
  - NODE, static node USER, and LTERM entries created; owner=IMS1
  - SRM=GLOBAL; RCVYCONV=YES
  - VTAM Generic Resources affinity set to IMS1; VTAM-managed affinity
- User enters conversational transaction (TRANC).
- IMS1 creates SPA and puts SPA+MSG on transaction ready queue (TRQ).
- IMS1 updates static node USER entry.
  - Conversation ID

  - Transaction code
  - CONV-IP ON
- TRANC is scheduled and processed by IMS2 (or any IMS in SQGROUP).
  - Response (SPA+MSG) is put on LTERM ready queue (LRQ) and IMS1 is notified.
- IMS1 retrieves response from LRQ and sends to LTERMA (NODEA).
  - SPA+MSG moved to lock queue.
  - CONV-IP flag turned OFF when NODEA acknowledges receipt.
    - Static node USER still has significant status.
    - SPA+MSG still on LOCKQ.
- User enters another transaction and process is repeated.
  - Static node USER entry gets updated for each input and output (CONV-IP flag).

IMS failure with terminal or user in conversational mode

When an IMS fails, the conversation could be in one of several states. Some of these represent conversational input in progress, and the other is between iterations. Which of these three states the user is in can be determined from the (static node) USER entry in the resource structure and the messages on the shared queues structure.

- User is between iterations of conversation.
  - Last SPA+MSG are on lock queue.
  - CONV-IP flag is OFF.
- Transaction was processing in IMS when IMS failed.
  - Input SPA+MSG are on lock queue.
  - No output messages for this LTERM on lock queue.
  - CONV-IP flag is ON.
- IMS was in the process of delivering the response.
  - Transaction has committed.
  - Input SPA+MSG has been deleted.
  - Output SPA+MSG is on lock queue.
  - CONV-IP flag is ON.
- Last input is still on transaction ready queue, last input is currently processing on IMS2, or response to last input is available on LTERM ready queue; in each case, response to last input will eventually get to LTERM ready queue.
  - Response SPA+MSG are on LTERM ready queue.
  - CONV-IP flag is ON.

Continuing the example - between iterations of conversations

Assume that, in our example, the user is between iterations of a conversation. That is, the response to the last input message has been sent and the CONV-IP flag is OFF. The SPA+MSG are on the lock queue. This is probably the most likely status of the conversation.

When IMS1 fails:

- VTAM deletes VTAM Generic Resources affinity.
- IMS2 is informed by SCI that IMS1 has failed.
- IMS2 queries RM2 for all resources owned by IMS1 and cleans up.
  - Finds NODEA/static node USERA/LTERMA entries with *active* conversational status and SRM=GLOBAL (Conversational status in structure).

- IMS2 cleans up for IMS1.
  - Resources not deleted.
  - Resource ownership cleared.
- IMS2 queries CQS for all output messages on lock queue owned by IMS1.
  - Finds SPA+MSG for last output on lock queue.
  - Moves SPA+MSG from lock queue back to LTERM ready queue for LTERMA.
- IMS2 is done for now.

When user logs and signs back on (using IMS0), VTAM routes logon request to IMS2:

- IMS2 checks resource structure and finds entries for NODEA, static node USERA, and LTERMA.
  - Conversation active; no ownership.
  - CONV-IP flag is OFF.
- Logon accepted; user not in response mode.
  - Response mode not recoverable for full function transactions.
  - IMS2 finds SPA+MSG on LTERM ready queue; moves it back to lock queue (locked by IMS2); also saves locally on Q5.
- User must retrieve last output message to refresh screen and continue conversation:
  - /HOLD
    - Conversation held; static node USERA entry updated.
    - DFS999I HELD CONVERSATION ID IS 0001 message is issued.
  - /RELEASE CONVERSATION 0001
    - Conversation released; static node USERA updated.
    - Last SPA+MSG retrieved from lock queue and sent to NODEA.
  - Screen refreshed; conversation continues.

Continuing the example - conversational input in progress

The CONV-IP flag may be on for several reasons as documented above. When the user logs on to IMS2, IMS2 queries RM for the status or those resources. If it accepts the logon (resources not owned), it will register interest in the logical terminal LTERMA.

- Resource entries show.
  - CONV-IP flag is ON.
  - Conversation active; no ownership.
  - Logon accepted.
- Register interest in LTERMA.
  - IMS2 informed if SPA+MSG on LTERM ready queue.
- If SPA+MSG on LRQ.
  - Must be response to last input.
  - Leave on LTERM ready queue; also keep locally on Q1.
  - Wait for user to request message (for example, PA1).
- If no SPA+MSG on LTERM ready queue.
  - Transaction has not executed (still on transaction ready queue - probably not the case). Or,
  - Transaction was in-progress in IMS1 (most likely case).
  - User must wait for response.

- If no response soon, user has choice.
  - Keep trying until response arrives (may have to wait for IMS1 to restart).
  - /EXIT conversation and do other work. Even though conversation is exited, it will still be scheduled and processed. When the output is queued and retrieved by an interested IMS, the conversational abnormal termination exit (DFSCONE0) will be driven to handle it.

11.6 Recovering Fast Path

When SRM=GLOBAL and RCVYFP=YES, Fast Path response mode can be recovered on any IMS in the IMSplex. This means that, if a user is in Fast Path response mode when IMS fails, that user can log on to any IMS in the IMsplex. When logon and signon is complete, the user session will be put into response mode. When the output message becomes available, it will be delivered to the user.

A detailed example of how this is done is not shown here, but it is similar to recovering a conversation. There is a Fast Path input message in progress (FP-IP) flag that performs the same function that the CONV-IP flag performed.

There is one significant difference, however. When SRM=GLOBAL but the transaction is processed locally without going through the EMHQ (sysplex processing code = local-only or local-first), then its SRM is *temporarily changed* to LOCAL. When this occurs, if IMS fails, ownership of the resource will not be released and the user must wait for the failed IMS to restart.

Note also that, when RCVYFP=NO, all Fast Path output messages are deleted if IMS or a session fails. They are not recoverable and will be deleted when discovered on the shared queue.

11.7 Recovering STSN sequence numbers

Recovery of STSN sequence numbers is much simpler than recovery of conversations or Fast Path. The input and output STSN sequence numbers are maintained in the (static node) USER entry and, when logging on to another IMS, can be used to resynchronize the message traffic with the STSN device. The intent of STSN is to resolve the *in-doubt* message.

For example, if IMS sent a message to a terminal but did not get an acknowledgment, IMS does not know whether the message was received or not. By informing the STSN device of the sequence number of the last output message, the device can ask IMS to resend it, or it can tell IMS to dequeue it. A similar function applies to an input message when the STSN device does not know if it was received by IMS.

By supporting STSN recovery, the user can log on to another IMS. Using the sequence numbers in the resource entry, that IMS can resynchronize its input and output messages with the STSN device.

11.8 Summary of STM in action

The above descriptions and examples, although quite detailed, are meant to give the reader some idea of how sysplex terminal management works, and the impact that various actions and failures can have. It is not complete.

When migrating to a CSL environment where STM is enabled, it is recommended that the installation develop extensive tests to identify all possible scenarios, and adjust end-user procedures accordingly.

11.9 Managing the resource structure

The resource structure and CQS support all the new structure management enhancements. Briefly, they include:

- Alter and autoalter support
- Structure full threshold monitoring
- System managed structure rebuild
- System managed duplexing

11.10 Structure failure

Structure failure can be the result of a complete Coupling Facility failure, or a problem with an individual structure itself. Loss of connectivity to a structure is not considered a structure failure, even when all connectors lose connectivity.

When a resource structure fails, it *must be repopulated*. Because structure checkpoints are not taken (no equivalent to the shared queues structure recovery data set), and changes to the structure are not logged (no equivalent to the shared queues log stream), a failed resource structure *cannot be recovered*.

11.10.1 Structure repopulation

Structure repopulation is the responsibility of CQS, RM, and IMS. When any structure fails, it is the connector that first discovers it.

When CQS discovers that a resource structure has failed, it begins the repopulation process by reallocating the structure and then creating any CQS global and local entries. Currently, there are no CQS entries. It then notifies all of the RMs (using SCI services) to begin their own repopulation. At this time, a message is written indicating that structure repopulation has been requested:

```
CQS210I STRUCTURE strname REPOPULATION REQUESTED
```

Each Resource Manager will repopulate the structure with its own information. RM entries include a RM Global entry (CSLRGBL), which contains the IMSplex name; local RM entries (CSLRLrmid), which contain local RM information, such as the RM version number; and a resource type table entry (CSLRRTYP), which identifies the supported resource types and names types. This is done by each RM, but only the first will add the global entries. As each RM completes, it will issue the message:

```
CSL2020I STRUCTURE strname REPOPULATION SUCCEEDED
```

Note that this does not mean that structure repopulation has completed. It only means that RM has completed it piece of it. RM then directs each of its registered IMSs to repopulate the structure with whatever local information they have.

Each *active IMS* will then add its own entries. This is not a robust process. There are several cases where resource entries will not be repopulated. If it is important to the user that the resource structure be immune from structure failure, then the structure should be duplexed. Causes for resource entries not being repopulated include:

- One or more IMSs are not active.

  If an IMS is not active, then it cannot repopulate. IMS will not attempt to repopulate if it is started later.

- A resource became inactive with command significant status or end-user significant status with SRM=GLOBAL.

  In this case, when the resource became inactive, the local IMS deleted all of its knowledge about the resource. It was available only in the resource entry.

- Global online change is never repopulated.

  If a structure fails during a global process, that process is not repopulated to the structure. In the case of global online change, each IMS must terminate the online change process and then reinitiate it, either at the PREPARE phase or COMMIT phase, depending on whether or not all IMSs had completed commit phase 2.

After the repopulation is completed, IMS issues the following message:

```
DFS4450 RESOURCE STRUCTURE REPOPULATION COMPLETE
```

As noted above, not all resource information is repopulated. If a user with SRM=GLOBAL were in a conversation, and the session were terminated while still in that conversation, then that user's status would not be repopulated and the conversation would, in effect, not exist. When the SPA+MSG is found on the LTERM ready queue, it will be passed to DFSCONE0 for processing.

11.11 Loss of connectivity to a structure

Loss of connectivity is not the same as structure failure, even though all CQSs may have lost connectivity. Loss of connectivity by all CQSs is most likely to happen when there is only one CQS active and its link fails. If this does happen, repopulation is not invoked. The only recourse is to try to fix the problem — perhaps using system managed rebuild to move the structure to a Coupling Facility where the CQS(s) do have connectivity would work — or fix the CF link.

11.12 SCI, RM, CQS, or structure failures

SCI, RM, CQS, or a structure failure will have the following impact until they are restarted or repopulated:

- Any attempt to update the structure will fail.
- Logons and signons will be rejected.
- Commands which affect global status will be rejected.
- Normal activities (for example, conversations, STSN, Fast Path) will continue with the status maintained only locally. When the failed components are again available, the status will be updated in the resource entry (if necessary).
- Logoffs and signoffs will wait until all components are available.

Note: Automatic Restart Manager (ARM) is supported for these address spaces and is recommended to minimize the duration of any outage.

12

Operations for other features utilizing Common Service Layer

This chapter describes the operations management tasks of the other features made possible by the Common Service Layer (CSL):

- Global online change
- Single point of control (SPOC)
- User written interface to Operation Manager (OM)
- Automatic RECON loss notification (ARLN)
- Language Environment (LE) commands

12.1 Global online change

Global online change is also sometimes called coordinated online change, or even coordinated global online change. We call it global online change.

12.1.1 Preparation for global online change

The global online change begins with the usual preparation of libraries. Libraries to be changed must be copied from the staging libraries to the inactive libraries by the online change copy utility. The utility will query OLCSTAT to determine what the inactive libraries are. If different IMSs are using different libraries, then this must be done for each library.

Figure 12-1 shows what happens before global online change begins.

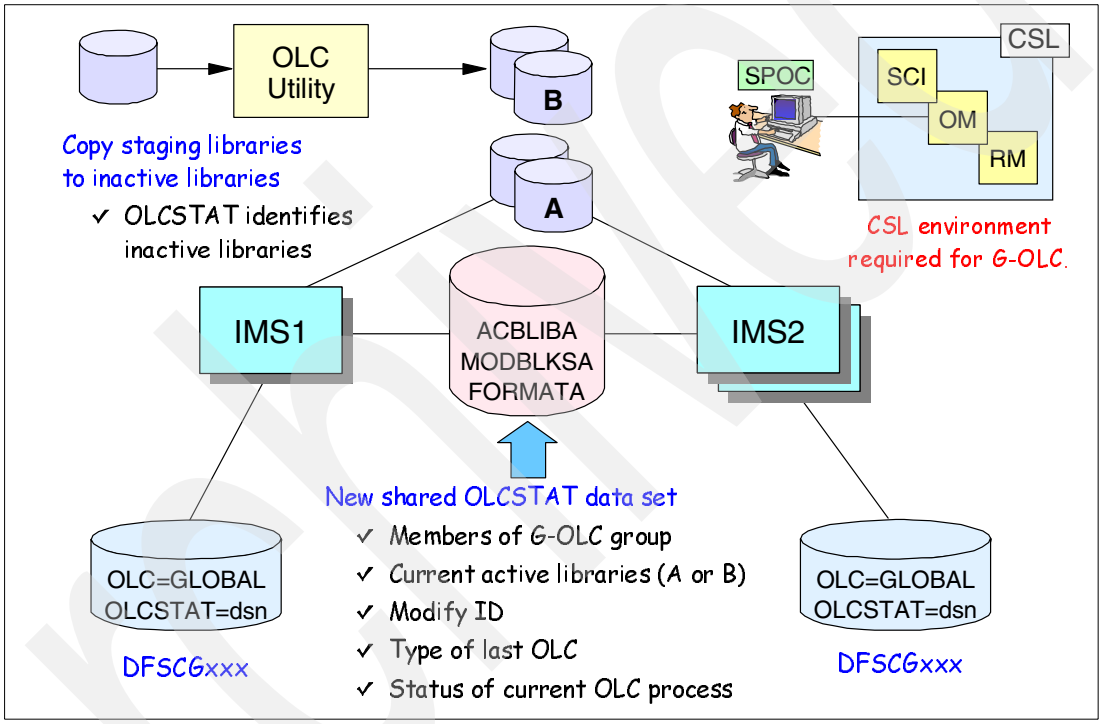

Figure 12-1 Before global online change begins

12.1.2 Executing global online change

- When the copy is done, an operator initiates the global online change process from a SPOC. The command is:

  `INIT OLC PHASE(PREPARE) TYPE(ALL|ACBLIB|MODBLKS|FMTLIB))`

- During the *prepare* phase, all IMSs stop queuing new work and attempt to drain the queues for those resources being changed. When all IMSs complete the prepare phase, the operator enters a command to commit:

  `INIT OLC PHASE(COMMIT)`

- Commit is executed in three phases. Each IMS must complete each phase before any IMS goes on to the next phase. The phases are:
  - Stop scheduling
  - Switch libraries and resume processing
  - Clean up

12.2 Global online change commands

Global online change commands use the OM API. They may be entered using the SPOC. They cannot be entered through the system console, the IMS master terminal, an end user terminal, or OTMA.

The new commands for global online change are given here:

- INITIATE OLC and TERMINATE OLC commands are defined with ROUTE=ANY, even if user specifies a route list.
  - OM picks one IMS to be master of the command.
  - You can select an IMS to be command master on the TSO SPOC by selecting the IMS to route to.
- INITIATE OLC PHASE(PREPARE) command prepares all IMSs for online change.
- INITIATE OLC PHASE(COMMIT) command commits online change on all IMSs.
- TERMINATE OLC command aborts online change on all IMSs.
- QUERY MEMBER TYPE(IMS) command displays online change status for all IMSs participating in online change.
- QUERY OLC LIBRARY(OLCSTAT) command displays online change status dataset contents.
- /DISPLAY MODIFY command displays online change libraries and work in progress.
- /CHE FREEZE LEAVEPLEX removes IMS from OLCSTAT data set.

Table 12-1 shows the new global online change commands, the local online change command equivalents (if any), and the function of the global online change command.

Table 12-1 Online change command table

Local online change command	Global online change command	Function
/MODIFY PREPARE	INITIATE OLC PHASE (PREPARE)	Initiate prepare phase of online change
/MODIFY COMMIT	INITIATE OLC PHASE (COMMIT)	Initiate commit phase of online change
/MODIFY ABORT	TERMINATE OLC	Abort online change
	QUERY OLC LIBRARY (OLCSTAT)	Display OLCSTAT data set contents
/DISPLAY MODIFY	/DISPLAY MODIFY	Displays online change libraries and work in progress
QUERY MEMBER TYPE(IMS)	QUERY MEMBER TYPE(IMS)	Display online change status of IMSs

12.3 PREPARE command

Prepare command coordinates the online change prepare phase across all IMSs in IMSplex listed in OLCSTAT data set. The prepare command is:

```
INIT OLC PHASE(PREPARE) TYPE(xxx) OPTION(yyy)
```

The parameters in this command can be described as follows:

- TYPE

  The TYPE parameter has four valid values. The ACBLIB, FMTLIB, MODBLKS, and ALL values have the same meanings that they have on the /MODIFY PREPARE command for local online change. ACBLIB, FMTLIB, and MODBLKS may be specified together. For example, one may specify TYPE(ACBLIB,MODBLKS).

- OPTION

  The BLDL, PSWD, TERMINAL, and TRANCMDS values have the same meanings that they have on the /MODIFY PREPARE command for local online change. BLDL is used with TYPE(ACBLIB) or TYPE(ALL). PSWD, TERMINAL, and TRANCMDS are used with TYPE(MODBLKS) or TYPE(ALL).

  BLDL sends message for each PSB or DMB not found in ACBLIB.

  PSWD adds SMU password definitions.

  TERMINAL adds SMU terminal security definitions.

  TRANCMDS adds SMU transaction command security definitions.

  FRCNRML is used to invoke online change even though one or more IMS systems which are recorded on the OLCSTAT data set are not active. In this situation, the command is rejected if FRCNRML or FRCABND is not specified.

  FRCABND is used to invoke online change even though one or more IMS systems which are recorded on the OLCSTAT data set have abended and not been restarted. In this situation, the command is rejected if FRCABND is not specified.

The Prepare command is rejected if any IMS is down, unless FRCABND and/or FRCNRML is specified. IMS that is down during online change may have to cold start.

12.3.1 PREPARE command examples

In Example 12-1, the command would start a change for ACBLIB. The BLDL option has the same meaning as it has with the /MODIFY PREPARE command. If any PSB or DMB is not found in ACBLIB, a message would be issued listing the missing member

Example 12-1 Change ACBLIB

```
INIT OLC PHASE(PREPARE) TYPE(ACBLIB) OPTION(BLDL)
```

In Example 12-2, the command would start a change for all of the libraries. It would invoke prepare processing even if an IMS system listed in the OLCSTAT data set is not active.

Example 12-2 Change all libraries

```
INIT OLC PHASE(PREPARE) TYPE(ALL) OPTION(FRCNRML)
```

12.4 COMMIT command

Commit command coordinates online change commit phases 1, 2, and 3 across all IMSs in IMSplex listed in OLCSTAT data set. In each phase, all IMS systems must complete the phase before the next phase is invoked in any IMS system.

The commit command is:

```
INITIATE OLC PHASE(COMMIT)
```

There are no other parameters. This command is rejected if prepare has not completed successfully. If processing fails in an IMS system for a phase, the processing stops for that IMS. Other IMSs may complete the phase. If so, their processing is not backed out.

Online change must be terminated if commit fails for a certain class of errors before the OLCSTAT data set is updated:

- Errors include RM failure, CQS failure, structure failure.
- Subsequent INITIATE OLC PHASE(COMMIT) commands will fail.
- Online change must be terminated with TERMINATE OLC.

12.5 Global online change processing

Global online change processes in a number of phases to coordinate the changes across all of the IMSs in the IMSplex. The phases are: prepare, commit phase 1, commit phase 2, and commit phase 3, or the clean up phase.

12.5.1 Prepare phase

The prepare phase stops (QSTOPs) all incoming messages and transactions for changed resources. This includes changed transactions, changed PSBs, and transactions whose PSBs refer to changed databases. Prepare does not affect other processing. Input messages already on the queue may be processed. All unaffected transactions may be processed. Any PSB may be scheduled. This includes BMPs, IMS transactions, and PSB schedules by CICS and ODBA. These actions occur with both local and global online change.

No scheduling occurs during commit phase 1 or 2.

Figure 12-2 shows the prepare process, and the interaction between the CSL components, and the IMSs.

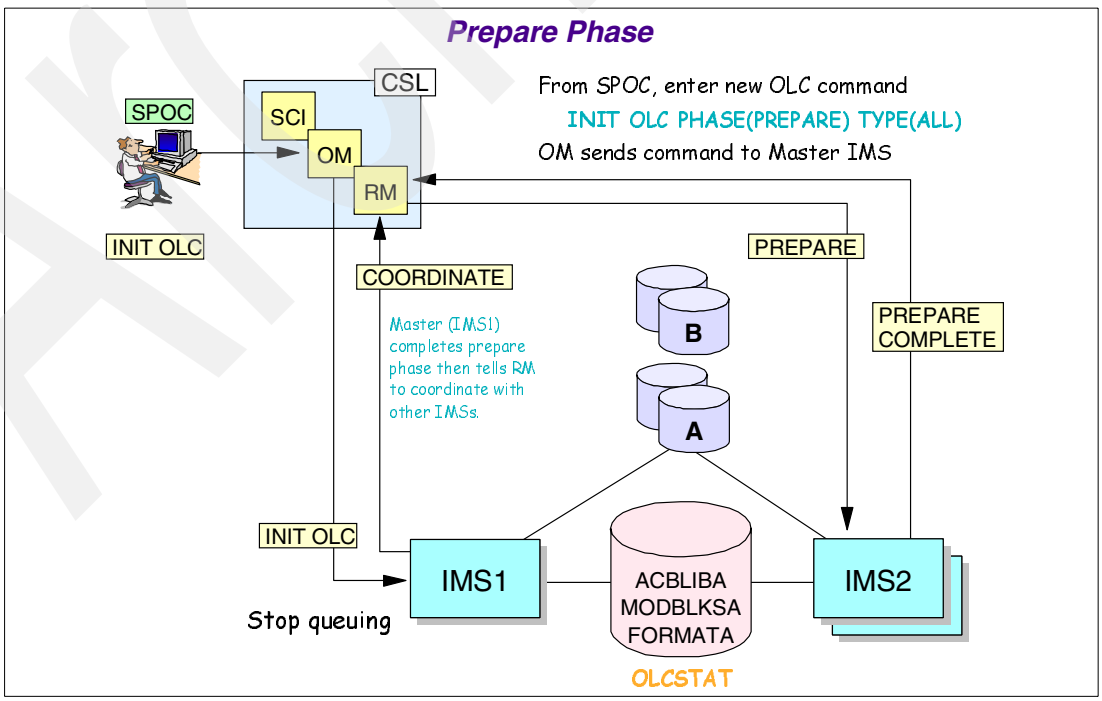

Figure 12-2 Prepare phase

When a prepare command is entered in OM, it is sent to one of the IMSs in the IMSplex. This IMS is the master IMS for prepare processing. It does its prepare processing first. If it succeeds, the master IMS tells RM to coordinate the prepare processing across the other IMS systems. RM invokes prepare processing in these IMSs.

If the prepare processing fails in the master IMS, online change is aborted. The problem may be resolved and the prepare command issued again.

If the prepare processing succeeds in the master IMS, the other IMSs are told to invoke prepare processing. When RM gets a response from all IMSs, it notifies the Prepare Master who then notifies the operator that prepare is complete. See Figure 12-3.

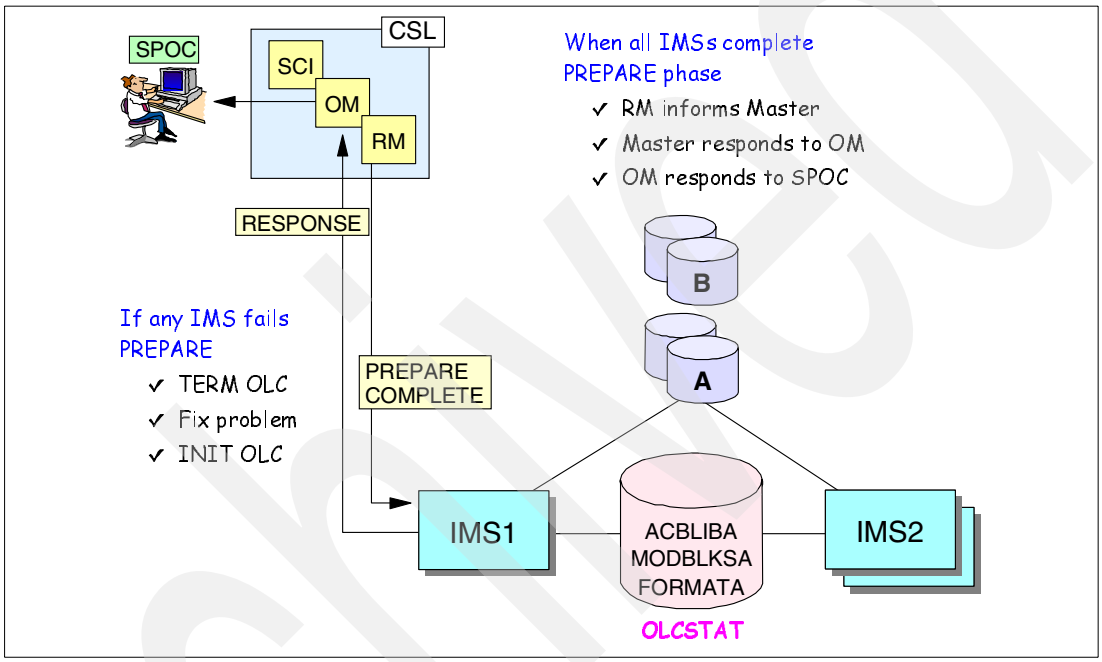

Figure 12-3 Prepare phase completed

The message to the operator indicates the OLC PREPARE completed successfully, or it provides a return code and reason code for why it failed.

If prepare processing fails in one of them, other IMSs are not affected. Their prepare processing is not backed out. In this situation users must cause the back out to occur. This is done by issuing the **`TERMINATE OLC`** command. This will back out prepare processing in all IMSs. Then the prepare may be attempted again. Of course, one should resolve the problem which caused the previous failure.

These are just some of the reasons a prepare command could fail. They are the same problems that could occur with local online change:

- The library could be enqueued, because the online change copy utility is still executing.
- An open error could occur, because the data set name specified for the library is wrong.
- A read error could occur, because of a DASD failure.
- A member of a library could be invalid, because it was generated by ACBGEN from another release of IMS.

12.5.2 Commit phase 1

Once the operator receives an indication that all IMSs have successfully completed the PREPARE phase, the operator must enter the command to commit the change.

```
INIT OLC PHASE(COMMIT)
```

Again, OM selects one IMS to be the master and sends the command. The master tells RM to coordinate Commit Phase 1 and performs phase 1 processing itself. These are done in parallel (the master doesn't complete this phase first before telling RM to coordinate as it does with the prepare phase).

Each IMS performs Commit Phase 1 processing (stops scheduling) and informs RM of success or failure. Commit Phase 1 could fail because, after prepare completed successfully, something was scheduled (for example, a BMP, a CICS transaction, an ODBA connection, ...) which uses resources to be changed.

Commit Phase 1 locks the OLCSTAT data set by updating it to indicate the online change is in progress. New IMS systems with OLC=GLOBAL cannot be initialized while the data set is locked. Phase 1 also checks to ensure that no resources to be changed are in use. When all IMS systems have successfully completed Commit Phase 1, Commit Phase 2 begins.

When a commit command is entered in OM, it is sent to one of the IMSs in the IMSplex. This IMS is the master IMS for commit processing. It is not necessarily the same IMS that was the master for prepare processing. The master does its processing for each commit phase locally. Before it can move to the next commit phase, it invokes the phase in the other IMSs by using RM.

Figure 12-4 shows the processing that occurs when Commit Phase 1 begins.

.

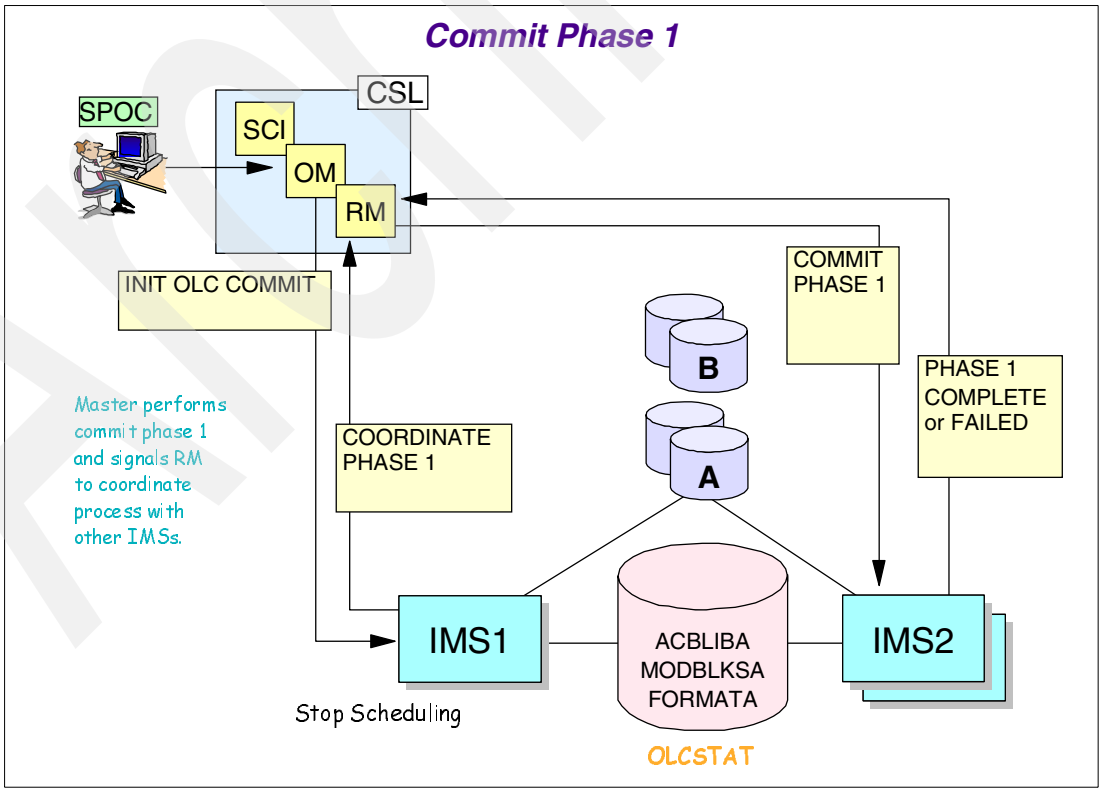

Figure 12-4 Commit phase 1

Assuming all IMSs signal RM that they have completed Commit Phase 1 successfully, the master updates the OLCSTAT data set. Once the master updates the OLCSTAT data set, after which it is no longer possible to terminate online change, it is committed.

- If any IMS fails after OLCSTAT is updated, the problem must be corrected and INIT OLC PHASE(COMMIT) re-entered.
- If any IMS fails, Commit Phase 1 may be terminated, or commit may be attempted again.

Figure 12-5 show the completion of Commit Phase 1.

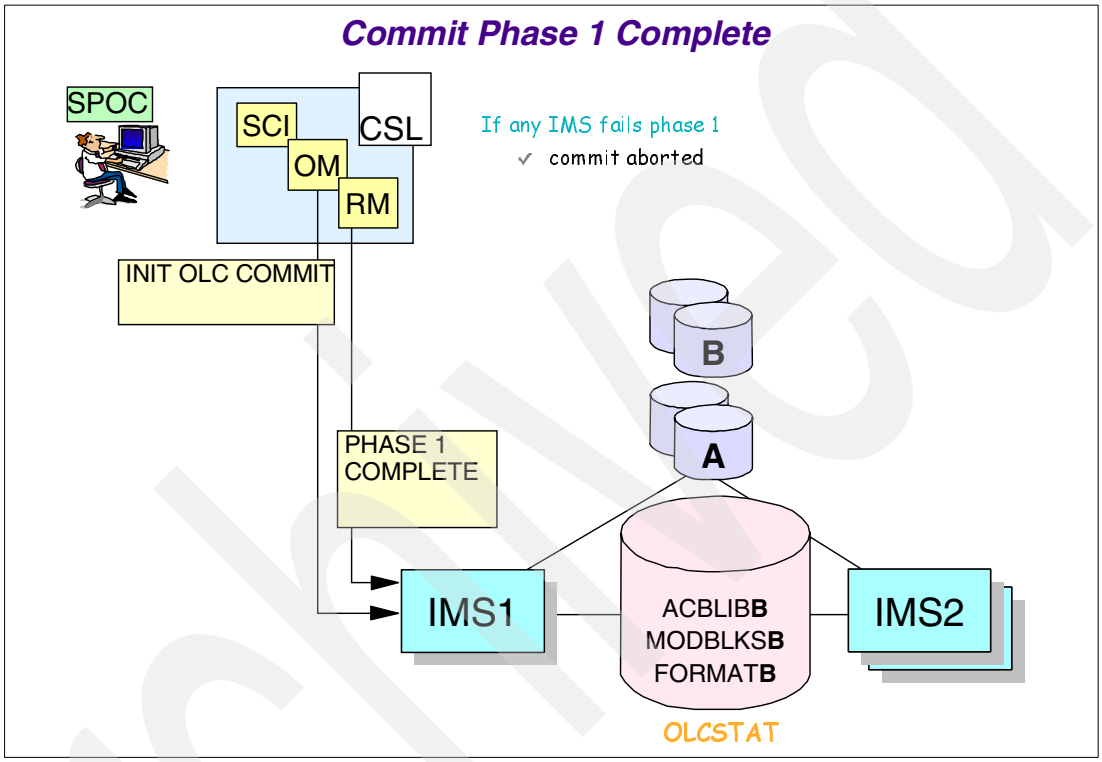

Figure 12-5 Commit phase 1 completed

12.5.3 Commit phase 2

Once the OLCSTAT data set updates are complete, the master signals RM to coordinate Commit Phase 2 with the other IMSs. During this phase, the IMSs switch libraries and resume processing. When all systems have made the change, Commit Phase 2 ends.

Coordination of Commit Phase 2 begins as shown in Figure 12-6.

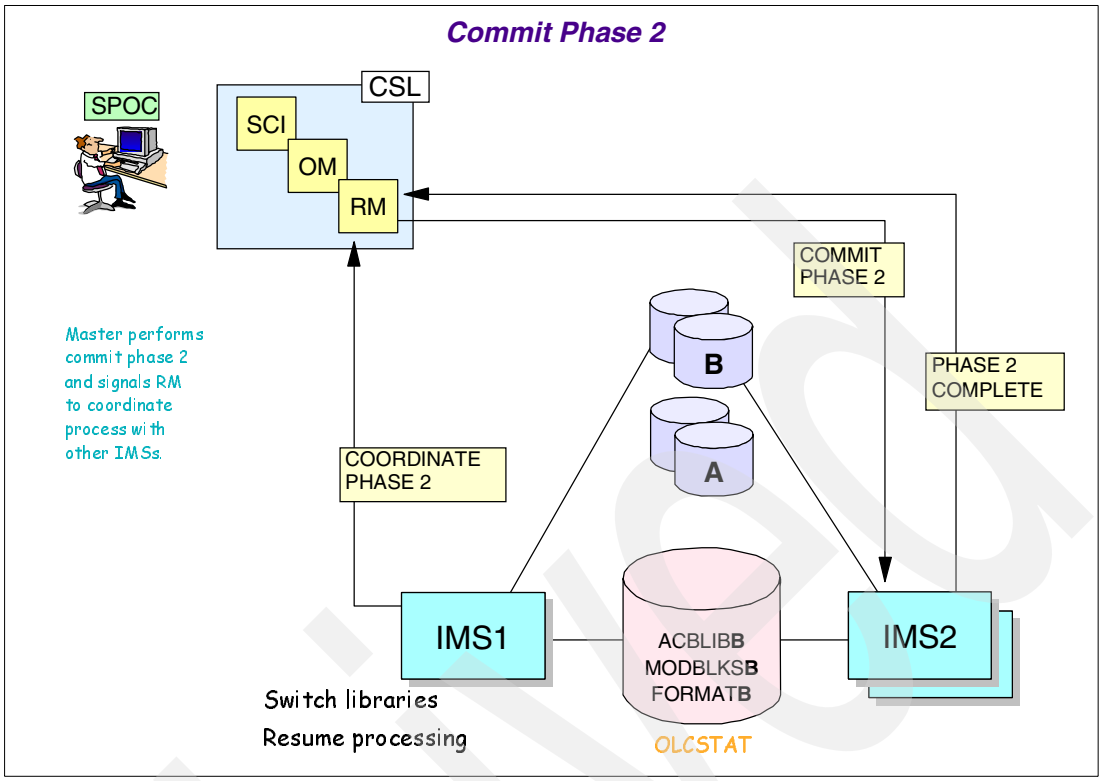

Figure 12-6 Commit phase 2

After completion of Commit Phase 2, the libraries have been switched, and the completion is communicated as shown in Figure 12-7.

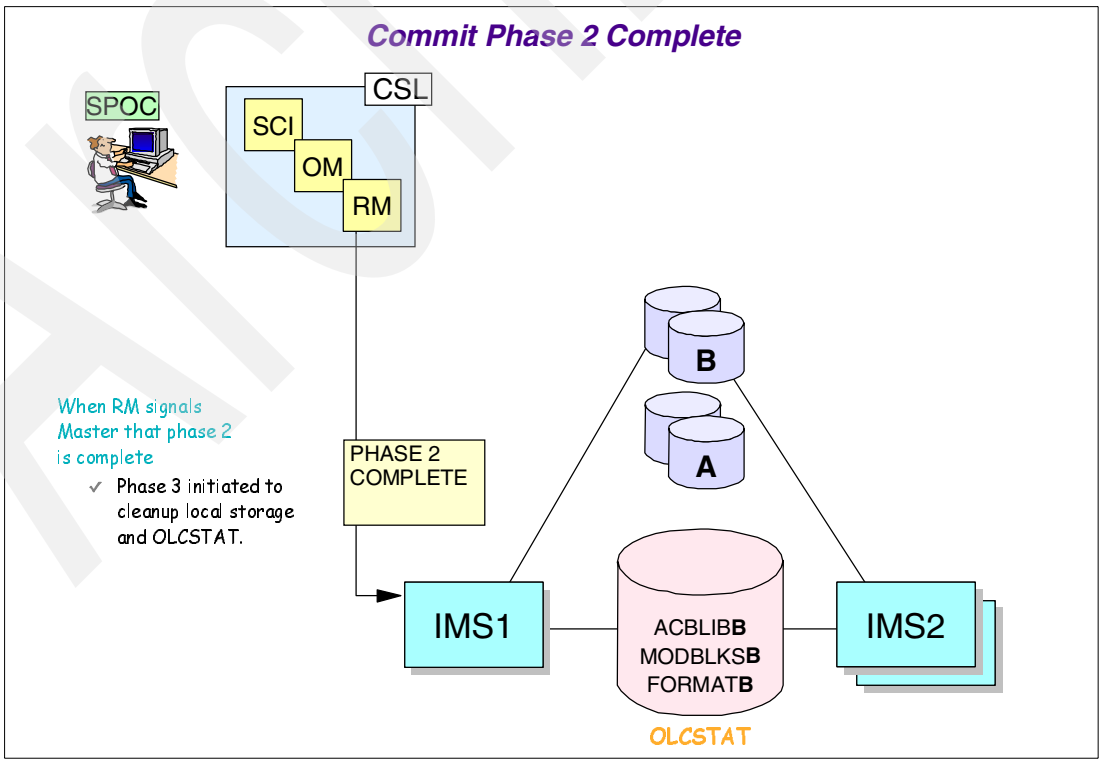

Figure 12-7 Commit phase 2 completed

If a Commit Phase 2 fails in any IMS, it does not cause the processing in any other IMS to be backed out. A return code and reason code explain why and the problem must be corrected and the commit command reentered. Phase 3 cannot begin until all IMSs have completed Phase 2.

Commit processing stops you must take one of two actions.

1. You can issue the **TERMINATE OLC** command, as long as the OLCSTAT has not yet been updated, committing the online change. This aborts all online change processing.
2. You could reissue the **INIT OLC PHASE(COMMIT)** command. This reattempts the commit processing in the phase where it failed.

These are some of the reasons that commit processing could fail. They are problems that could also occur with local online change commit processing:

- It may not be possible to change a PSB, because it is currently in use. Prepare processing would have stopped queuing IMS transactions using the PSB, but the PSB might have been scheduled for another reason.
  - These include the starting of a BMP, the processing of an input transaction which was queued before the prepare, the continued processing of a wait-for-input transaction which was scheduled before the prepare, and the scheduling of the PSB by a CICS transaction, or the scheduling of a PSB via ODBA.
- It may not be possible to change a database, because it is currently in use by a transaction or BMP.

12.5.4 Commit phase 3

In Commit Phase 3 the OLCSTAT data set is unlocked. This allows new IMS systems with OLC=GLOBAL to be initialized.

If the commit failed in Phase 1, reissuing the command will cause Phase 1 to be retried in those IMSs where it failed. If it succeeds, commit processing continues to Phase 2.

If the commit failed in Phase 2, reissuing the command retries Phase 2 in those IMSs where it failed. If it succeeds, commit processing continues to Phase 3.

12.5.5 Reissuing the Commit command

It may be desirable to reissue a Commit command. This is done when a previous commit attempt has failed.

- If the commit failed in Phase 1, reissuing the command will cause Phase 1 to be retried in those IMSs where it failed. If it succeeds, commit processing continues to Phase 2.
- If the commit failed in Phase 2, reissuing the command retries Phase 2 in those IMSs where it failed. If it succeeds, commit processing continues to Phase 3.

12.5.6 COMMIT completion codes

Table 12-2 lists the possible completion codes that may be returned for an IMS participating in the online change commit. These completion codes are defined in the DFSCMDRR macro.

Table 12-2 Commit completion codes

Completion Code (CC)	Meaning
0	Commit phase succeeded for this IMS

Completion Code (CC)	Meaning
1	Command not applicable to this IMS
2	Online change phase not attempted because of error
3	IMS already in correct online change state
4	Online change commit phase incomplete on this IMS
50	CQS is unavailable
51	No resource structure
52	Resource structure full
53	No RM address space
54	No SCI address space
60	IMODULE GETMAIN storage error
61	BCB storage error
62	HIOP storage error
63	WKAP storage error
64	GETSTOR storage error
70	Module load error
71	Module locate error
72	Randomizer load error
80	Data set error
90	Internal error
91	Timeout error
94	RM request error
95	SCI request error
98	CQS request error
B0	Resource definition error
B1	Resource state error (work in progress)
B2	IMS state error (for example, abend or chkpt in progress)

The resource state error (B1) is equivalent to the local online change commit failure due to “work in progress”.

Global online change provides additional information, in that it identifies the resource type, the resource name, and the work in progress that caused the Commit Phase 1 to fail.

Work in progress codes

Table 12-3 shows the message text you may receive for the work in progress. Equivalent information is also returned on /DISPLAY MODIFY.

Table 12-3 Work in progress codes

Resource Type	Error Text	Meaning
DATABASE or DMB		A /DBDUMP command is active
DATABASE or DMB	DBR ACTIVE	A /DBRECOVERY command is active
DATABASE or DMB	IN USE	A database is in use
DATABASE or DMB	DEP REGS ACTIVE	Dependent regions using the FP DEDB are active
DATABASE or DMB	AREA OPEN	An area associated with a FP DEDB is open
DATABASE or DMB	AREA STARTED	An area associated with the FP DEDB is started
DMB	STO ADS OR DBR	/STO ADS or /DBR cmd in progress for FP database
PROGRAM or PSB	SCHEDULED	Program is scheduled
PSB	CONVERSATION	Program is in conversation
RTCODE	ACTIVE	The routing code is active
TRAN	CONVERSATION	A tran, program, or db is in conversation
TRAN	DYNAMIC OR CPIC	Tran already exists and is not CPIC or dynamic
TRAN	INTERNAL ERROR	Resource not found or already exists
TRAN	IN USE	Transaction is in use
TRAN	QUEUEING	Messages are queued to the tran or program
TRAN	SCHEDULED	Transaction is scheduled
TRAN	SUSPENDED	Transaction is on the suspend queue

12.6 TERMINATE OLC command usage

As mentioned before, you may issue the **`TERMINATE OLC`** command after a previous prepare or commit attempt has failed.

12.6.1 Failure in PREPARE or in COMMIT phase 1

If the failure occurred in prepare processing or in Commit Phase 1, the **`TERMINATE OLC`** backs out all online change processing. The IMSs continue processing with the libraries they were using before the online change attempt.

12.6.2 Failure in COMMIT phase 2

If the failure occurred in Commit Phase 2, the **`TERMINATE OLC`** command is rejected. In this case, some of the IMSs have completed commit processing and are using the new libraries. The problem preventing commit success in some of the IMSs should be resolved and the **`INIT OLC PHASE(COMMIT)`** command should be reissued.

12.7 Online change command error handling

Table 12-4 shows a list of the states your IMSplex may be in after a global online change command results in error, and your choices of action.

For example, if an INITIATE OLC PHASE(PREPARE) command fails and QUERY MEMBER TYPE(IMS) shows that all of the IMSs are in an online change prepare state, the prepare is successful, even though the command result was an error. You may proceed with commit or abort the online change if you do not want to continue.

Table 12-4 Command error handling

IMSplex state after online change command error	Action
Some IMSs in a prepare complete state	TERMINATE OLC Correct problem before retry
All IMSs in a prepare complete state	INIT OLC PHASE(COMMIT) or TERMINATE OLC
Mix of IMSs in prepare complete & commit phase 1 complete state	Correct problem & try commit again or TERMINATE OLC
All IMSs in commit phase 1 complete state before OLCSTAT data set updated	Correct problem & try commit again or TERMINATE OLC
All IMSs in commit phase 1 complete state after OLCSTAT data set updated	Correct problem and try commit again. Online change is committed and cannot be aborted.
Mix of IMSs in commit phase 1 & commit phase 2 complete	Correct problem and try commit again.
All IMSs in commit phase 2 complete state	Correct problem and try commit again.
Mix of IMSs in commit phase 2 & not in online change state	Correct problem and try commit again.

12.8 Status display commands

If any process seems to be taking a long time, or to be sure that the prepare phase has completed, IMSplex *QUERY* commands and IMS */DISPLAY* commands can be used to display the status of the online change processing.

- QUERY MEMBER TYPE(IMS)

  This command shows the online change status of the IMSs participating in global online change. It displays the current online change phase and *in progress*, *completed*, or *failed*.

- QUERY OLC LIBRARY(OLCSTAT)

  This command shows the contents of the OLCSTAT data set.

- /DISPLAY MODIFY

  This command shows the same information that it shows for local online change. This is the information about online change processing in the IMS systems where the command is processed. In particular, it shows what resources are to be changed and what processing, if any, is preventing online change from continuing.

12.9 QUERY MEMBER TYPE(IMS) command

If an INITIATE OLC or TERMINATE OLC command fails, you should issue the QUERY MEMBER TYPE(IMS) command to display the online change state of all the IMSs in the IMSplex. This can help you decide how to proceed. You can correct the problem, then either

try the INIT OLC command again, or abort the online change with the TERMINATE OLC command.

Here is the QUERY MEMBER command syntax:

TYPE(IMS) Displays information about the active IMSs in the IMSplex.

SHOW(options**)** Allows user to display only those output fields that are of interest:

ALL

ATTRIB displays attributes, which are static definitions.

STATUS returns status that can change dynamically.

TYPE returns the member type IMS.

12.9.1 QUERY MEMBER TYPE(IMS) status command

Table 12-5 shows the list of possible status you may receive on the QUERY MEMBER TYPE(IMS) command. Most of the status are related to online change. The local online change statuses are also applicable to an IMSplex enabled with local online change.

A scope of local means the status applies to this IMS only. A scope of global means that the status applies to all of the IMSs globally. For example, a local status of OLCPREPC means an IMS has completed the online change prepare phase locally. A global status of OLCPREPC means that all of the IMSs have successfully completed the prepare phase of a global online change.

Table 12-5 QUERY MEMBER TYPE(IMS) status

Status	Meaning	Scope
LEOPT	Language Environment options	LCL
OLCABRTC	OLC abort completed	LCL
OLCABRTI	OLC abort in progress	LCL
OLCCMT1C	OLC commit phase 1 completed	LCL, GBL
OLCCMT1I	OLC commit phase 1 in progress	LCL, GBL
OLCCMT2C	OLC commit phase 2 completed	LCL, GBL
OLCCMT2F	OLC commit phase 2 failed	LCL
OLCCMT2I	OLC commit phase 2 in progress	LCL, GBL
OLCMSTR	OLC command master	GBL
OLCPREPC	OLC prepare phase completed	LCL, GBL
OLCPREPF	OLC prepare phase failed	LCL
OLCPREPI	OLC prepare phase in progress	LCL, GBL
OLCTERMC	OLC terminate completed	GBL
OLCTERMI	OLC terminate in progress	GBL
XRFALT	XRF alternate system	LCL

12.9.2 QUERY MEMBER TYPE(IMS) attributes

Table 12-6 shows the list of possible attributes you may receive as output from the QUERY MEMBER TYPE(IMS) command. Attributes are static definitions associated with the IMS, for example global online change enabled, an RSR tracker IMS, or an IMS enabled with shared queues.

Table 12-6 QUERY IMS attributes

Attribute	Meaning	Scope
GBLOLC	Global online change enabled	LCL
RSRTRK	RSR tracker system	LCL
SHAREDQ	Shared queues enabled	LCL

12.9.3 QUERY MEMBER TYPE(IMS) example

Example 12-3 shows the response to a **QUERY MEMBER TYPE(IMS)** command.

Example 12-3 QUERY MEMBER response

```
MbrName    CC TYPE  STATUS           LclAttr LclStat
IM1A        0 IMS   OLCPREPC,OLCMSTR
IM1A        0 IMS                    GBLOLC  OLCCMT1C
IM3A        0 IMS                    GBLOLC  OLCCMT1C
IM4A        0 IMS                    GBLOLC  OLCPREPC
```

- The first response line with IM1A shows that **INITIATE OLC PHASE(PREPARE)** completed successfully for the IMSplex. The global status is OLCPREPC. IM1A was the master of the prepare.
- The second response line with IM1A shows that IM1A is enabled for global online change and it has completed Commit Phase 1.
- The response line with IM3A shows that it also is enabled for global online change and has completed Commit Phase 1.
- The response line with IM4A shows that it is enabled for global online change and it has completed prepare. This implies that it has failed Commit Phase 1.

The status is reported for both global and local online change. If an IMS were in the process of local online change, its status would be reported.

12.10 QUERY OLC command

Displays global online change status (OLCSTAT) data set contents. Only the command master accesses OLCSTAT data set and returns information.

SHOW(ACTVLIB) Displays the suffixed online change library names that are currently active. This includes ACBLIBA or ACBLIBB, FMTLIBA or FMTLIBB, MODBLKSA or MODBLKSB.

SHOW(DSN) Shows the OLCSTAT data set name.

SHOW(LASTOLC) Shows the last online change that was successfully performed (ACBLIB,FMTLIB,MODBLKS).

SHOW(MBRLIST) Shows a list of IMSs that are current with the online change libraries and will be permitted to warm start. These IMSs either participated in the last online change or cold started since the last online change.

SHOW(MODID) Shows the modify ID, which is the number of global online changes, if initialized to 0 by the global online change utility.

12.10.1 QUERY OLC LIBRARY example

Example 12-4 shows the response to a **QUERY OLC LIBRARY(OLCSTAT) SHOW(ALL)** command. In the response:

A or B Current DDNAME suffix for library

LastOLC Last type of online change: FMTLIB, ACBLIB, and/or MODBLKS

MbrList IMSs using current libraries; they are allowed to warm start

Example 12-4 QUERY OLC response

```
MbrName   CC Library   ACBLIB  FMTLIB  MODBLKS  Modid
IM1A       0 OLCSTAT        A       B        A     02



DSName                        LastOLC      MbrList
IMSPSA.IMS0.OLCSTAT           FMTLIB       IM1A,IM3A,IM4A
```

The current online change active libraries are ACBLIBA, FMTLIBB, MODBLKSA, and MATRIXA. Value 02 in for the Modid field indicates that one online change has been performed. The OLCSTAT data set name IMSPSA.IMS0.OLCSTAT is listed. The last online change was only for FMTLIB. The member list of IMSs that are current with the online change libraries includes IM1A, IM3A, and IM4A.

12.11 /DISPLAY MODIFY command

If a problem occurs on one IMS, the classic /DISPLAY MODIFY command can be used on that IMS to see what the problem might be. It displays work in progress for resources to be changed or deleted by online change.

/DISPLAY MODIFY is the same command and same response format that is used for local online change in IMS Version 8 and previous releases.

12.11.1 /DISPLAY MODIFY example

Example 12-5 show the response from a **/DISPLAY MODIFY** command.

Example 12-5 /Display modify response

```
LIBRARY  IMSACBA  (A)  IMSPSA.IMOA.ACBLIBA
LIBRARY  FORMATA  (A)  IMSPSA.IMOA.MFS.FORMATA
LIBRARY  MODBLKSA (A)  IMSPSA.IMOA.MODBLKSA
LIBRARY  MATRIXA  (A)  IMSPSA.IMOA.MATRIXA
LIBRARY  IMSACBB  (I)  IMSPSA.IMOA.ACBLIBB
LIBRARY  FORMATB  (I)  IMSPSA.IMOA.MFS.FORMATB
LIBRARY  MODBLKSB (I)  IMSPSA.IMOA.MODBLKSB
LIBRARY  MATRIXB  (I)  IMSPSA.IMOA.MATRIXB
DATABASE RXLDB101      PSB SCHEDULED
```

```
TRAN     SNWFT116      QUEUING     6
DISPLAY MODIFY COMPLETE *02182/110535*   IM1A
```

This example shows the current active libraries, marked with "(A)", and the current inactive libraries, marked with "(I)".

Database RXLDB101 is being changed. A PSB using this database is currently scheduled.

Transaction SNWFT116 is being changed. There are currently six instances of this transaction on the queue.

12.12 /CHE FREEZE LEAVEPLEX command

In an IMSplex, this keyword is specified if the IMS that is being shut down is not going to rejoin the IMSplex. Specify the LEAVEPLEX keyword when you do not intend to bring the IMS back up in the IMSplex.

- Type of IMS shutdown where you don't intend to bring IMS back up.
- Removes knowledge of IMS from IMSplex.
- IMS deletes its name from OLCSTAT data set.
  - Next global online change doesn't need to specify FRCNRML, to force the online change in spite of this IMS is being down.

12.13 Add and delete IMS from OLCSTAT data set

When an IMS system defined with global online change is first started, it is added to the list of IMS systems in the OLCSTAT data set. That means that a cold start is done after newly defining OLC=GLOBAL in the systems DFSCGxxx member. A system may also be added to the data set by using the ADD function of the global online change utility (DFSUOLC0), however, this is typically not required. The ADD function would typically be used only if the data set were being rebuilt.

12.13.1 IMS removed from OLCSTAT

An IMS system is removed from the OLCSTAT data set list by either of two actions:

- First, the system may be terminated with a **/CHE FREEZE LEAVEPLEX** command. This means the system is leaving the IMSplex. If you have a system which you do not plan to restart, you should terminate it this way.
- Second, the IMS system may be removed from OLCSTAT by using the DEL function of the global online change utility. You would typically use this if you terminated a system and later decided that you would not restart it.

12.13.2 Deleting IMS subsystem from OLCSTAT

Example 12-6 shows a sample job stream to delete IMS subsystems from the OLCSTAT data set.

Example 12-6 Deleting IMS subsystems from OLCSTAT

```
//JOUKO3X  JOB (999,POK),'OLC',NOTIFY=&SYSUID,
//    CLASS=A,MSGCLASS=T,
//    MSGLEVEL=(1,1)
```

```
//*
//STEP1 EXEC DFSUOLC FUNC=DEL
//SYSIN DD *
IM1A
IM2A
/*
```

12.14 Inactive subsystems

Online change cannot be done with an inactive IMS in the IMSplex, unless the **INIT OLC PHASE(PREPARE)** command includes either OPTION(FRCNRML) or OPTION(FRCABND).

FRCNRML allows online change to be done even though there are normally terminated IMS systems within the IMSplex. FRCABND and FRCNRML specified together allow online change to be done even though there are normally terminated or abended IMS systems within the IMSplex.

IMS restricts the type of restart that may be done by a system when it was not active during an online change. This is done to prevent a restart from processing records created with a different set of libraries than are used by the restart.

12.14.1 Inactive IMS restart options

Table 12-7 shows the types of restarts that are permitted when an IMS was not active during the online change. If multiple online changes occur while and IMS is inactive, the inactive IMS must be cold started (**/NRE CHECKPOINT 0**), regardless of the type of online changes made.

Table 12-7 Inactive IMS restart options

Last online changed type	Restart command permitted
ALL	/NRE CHECKPOINT 0
MODBLKS	/NRE CHECKPOINT 0
ACBLIB	/ERE COLDBASE /NRE CHECKPOINT 0
FORMAT	/NRE /ERE /ERE COLDCOMM /ERE COLDBASE /NRE CHECKPOINT 0

IMS restart is not sensitive to changes made in the MFS FORMAT library. So, an online change only for FORMAT does not restrict the restart. All other online changes require a cold start of the affected part of IMS. A change to ACBLIB requires a cold start of the database manager part of IMS. A change to MODBLKS requires a cold start of all of IMS.

No matter what type of online changes were done, if there have been two or more online changes while an IMS system is not active, it can only be cold started.

12.15 Single point of control (SPOC)

IMS provides the ability to manage a group of IMSs (an IMSplex) from a single point of control (SPOC). That point of control is the OM API. You can write your own application using the OM API, use the IBM-provided TSO SPOC application, or write a REXX SPOC application that uses the REXX SPOC API.

12.15.1 OM automated operator program clients

OM provides an application programming interface (API) for application programs that automate operator actions. These programs are called automated operator programs (AOP). An AOP issues commands that are embedded in an OM API request to an OM. The responses to those commands are returned to the AOP embedded in XML tags.

If you want to use OM to manage commands and command responses in an IMSplex for your own product or service, you can use an AOP client, such as:

- The IMS-supplied AOP client, TSO single point of control (SPOC), which runs on the host. With the TSO SPOC, an automated operator can issue commands to the IMSplex and receive responses to those commands interactively.
- An AOP client that runs on a workstation (called a workstation SPOC).
- The IMS Control Center, an IMS system management application that performs SPOC functions.
- A command processing client, such as IMS.

12.16 ISPF options

The TSO SPOC uses an ISPF panel interface and communicates with a single OM address space.

12.16.1 ISPF application features

As an ISPF application, it also provides other typical ISPF application features:

Print	The command responses and log information can be put to the ISPF list file
Save	Information can be saved to a user specified data set
Find	Command responses can be searched for text strings
Help	Help dialogs are available for application use information
Scrolling	Where applicable, data can be scrolled left and right, or up and down
Split screen	When more than one TSO SPOC is started for by a user, the last one to end displays an exit confirmation panel. As long as you are still using SPOC, older command responses are still viewable from the command status panel. Command responses are discarded when you respond “erase” or when you logoff.

12.16.2 Other ISPF capabilities

Various ISPF capabilities are available from TSO SPOC:

- Other retrieve commands such as RETP and RETF will work.

- Use KEYLIST or KEYS command to customize the function keys.
- Use the PFSHOW TAILOR command to control function key display styles.
- SPLIT NEW and SWAP NEXT can be used to start multiple versions of SPOC.

When more than one TSO SPOC is started for by a user, the last one to end displays an exit confirmation panel. As long as you are still using SPOC, older command responses are still viewable from the command status panel. Command responses are discarded when you respond “erase” or when you logoff.

12.16.3 ISPF /IMS command collisions

ISPF allows commands to be typed in any command line. Some commands are for both IMS and ISPF:

- For example, END, EXIT. With parameters, these are considered to be IMS commands.

ISPF supports various commands that can be typed into any ISPF command line. The command table for SPOC tells ISPF to ignore some of the commands (they are passed on to SCI / OM / IMS).

If you have your own commands that conflict with ISPF commands, you will need to have the installer modify the SPOC command table CSLUCMDS (do not update SMP/E managed data sets):

- ISPF option 3.9 is the command utility table editor. For SPOC the application ID is `CSLU`.

12.17 Working with the TSO SPOC application

With the necessary preferences set you can begin working with the TSO SPOC application. Select **Display —>1. Cmd entry & response** from the action bar to access the command entry and response panel.

The basic operation is to enter a command in the command line and for the command response to be provided in the data area below the command line. The command line is cleared in preparation for the next command; the command issued is shown just above the command response. There are three short fields: plex, route, and wait. These are temporary overrides of the fields in the preferences panel. These values are discarded after you exit SPOC.

12.17.1 IMSplex commands

Figure 12-8 shows the results of a QRY IMSPLEX SHOW(ALL) command which is one of the new format IMSplex commands.

```
  File  Display  View  Options  Help
 ------------------------------------------------------------------------
 PLEX1                    IMS Single Point of Control
 Command ===> ____________________________________________________________
 ------------------------------------------------------------------------
  ------------------------------------ Plex . _____  Route . IM4A_____  Wait . _____
 Response for: QRY IMSPLEX SHOW(ALL)
IMSplex  MbrName    CC Member   JobName  Type  Subtype  Version  OSName   Status
CSLPLEX1 IM1AOM      0 IM1AOM   IM1AOM   OM             1.1.0    SC53     ACTIVE
CSLPLEX1 IM1AOM      0 JOUKO3   JOUKO3   AOP            1.1.0    SC53     ACTIVE
CSLPLEX1 IM1AOM      0 IM4AOM   IM4AOM   OM             1.1.0    SC67     ACTIVE
CSLPLEX1 IM1AOM      0 IM1ARM   IM1ARM   RM    MULTRM   1.1.0    SC53     ACTIVE
CSLPLEX1 IM1AOM      0 IM4ARM   IM4ARM   RM    MULTRM   1.1.0    SC67     ACTIVE
CSLPLEX1 IM1AOM      0 IM3ARM   IM3ARM   RM    MULTRM   1.1.0    SC54     ACTIVE
CSLPLEX1 IM1AOM      0 CQ4ACQS  IM4ACQS  CQS            1.3.0    SC67     ACTIVE
CSLPLEX1 IM1AOM      0 IM3AOM   IM3AOM   OM             1.1.0    SC54     ACTIVE
CSLPLEX1 IM1AOM      0 IM4ASC   IM4ASC   SCI            1.1.0    SC67     ACTIVE
CSLPLEX1 IM1AOM      0 CQ3ACQS  IM3ACQS  CQS            1.3.0    SC54     ACTIVE
CSLPLEX1 IM1AOM      0 IM3ASC   IM3ASC   SCI            1.1.0    SC54     ACTIVE
CSLPLEX1 IM1AOM      0 IM4A     IM4ACTL  IMS   DBDC     8.1.0    SC67     ACTIVE
CSLPLEX1 IM1AOM      0 CQ1ACQS  IM1ACQS  CQS            1.3.0    SC53     ACTIVE
CSLPLEX1 IM1AOM      0 IM1A     IM1ACTL  IMS   DBDC     8.1.0    SC53     ACTIVE
CSLPLEX1 IM1AOM      0 SCI1SC   IM1ASC   SCI            1.1.0    SC53     ACTIVE
CSLPLEX1 IM1AOM      0 IM3A     IM3ACTL  IMS   DBDC     8.1.0    SC54     ABENDE

                    At least one request was successful.
```

Figure 12-8 SPOC IMSplex command results display

IMSplex commands may have log information too if there were some messages to display. For example, one system may say `no resources found` while other systems provide valid resource information. The *no resources* indication would appear in the log. To review the log information select **Display—>2. Cmd entry & log** from the action bar. The command and log panel is displayed as shown in Figure 12-9.

```
  File  Display  View  Options  Help
 ------------------------------------------------------------------------
 PLEX1                    IMS Single Point of Control
 Command ===> ____________________________________________________________
 ------------------------------------------------------------------------
  ------------------------------------ Plex . _____  Route . _________  Wait . _____
 Log for . . : QRY TRAN SHOW(ALL)

 IMSplex . . . . . : PLEX1
 Routing . . . . . : IM1A,IM4A
 Start time. . . . : 2002.154 19:01:38.21
 Stop time . . . . : 2002.154 19:01:38.28
 Return code . . . : 00000000
 Reason code . . . : 00000000
 Command master. . : IM1A
```

Figure 12-9 SPOC command and log panel

12.17.2 IMS classic commands

Figure 12-10 shows the results of an IMS DISPLAY QUEUE TRAN command which is one of the classic IMS commands. The command can be retrieved to the command prompt by

placing the cursor on the command display in the response area and pressing enter. It may then be modified, or resubmitted.

As you can see in the example, for IMS classic commands, you don't need to enter the command recognition character '/' (slash). If you desire, you can still enter the slash, as IMS Operations Manager can handle both formats of classic commands (it removes the slash before actual processing).

Note: The TSO SPOC uses the OM API. The OM API does not support all of the variations in syntax that were acceptable before. IMS has to register with OM and indicate which commands it can process. Although most IMS classic commands are registered and supported through the TSO SPOC, only a few of the variations in syntax are registered by IMS. Refer to *IMS Version 8: Command Reference,* SC27-1291, for the list of the command verbs and primary keywords that can be issued through the OM API. The TSO SPOC uses the OM API. IMSs register the commands they can process with OM.

```
  File  Display  View  Options  Help
 ____________________________________________________________________________
PLEX1                      IMS Single Point of Control
Command ===> ______________________________________________________________
 ____________________________________________________________________________
 ____________________________________ Plex . _____  Route . __________  Wait . _____
Log for . . : DIS Q TRAN

IMSplex . . . . . : PLEX1
Routing . . . . . : IM1A,IM4A
Start time. . . . : 2002.154 14:22:51.08
Stop time . . . . : 2002.154 14:22:51.09
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IM1A

MbrName  Messages
-------- -----------------------
IM1A         CLS     PTY     MSG CT TRAN     PSBNAME
IM1A         *NO QUEUES*
IM1A         *2002154/142251*
IM4A         CLS     PTY     MSG CT TRAN     PSBNAME
IM4A         *NO QUEUES*
IM4A         *2002154/142251*
```

Figure 12-10 SPOC classic command results display

For classic IMS commands, the response is in a sequential format. At top are some execution statistics and log information. Below are the messages produced by the IMS command. The text is in the same format as that of prior releases. The text is prefixed by the member name, and information from each member is grouped together. Each message line is a single XML tagged value.

12.18 Command entry panel

The basic operation is to enter a command in the command line and for the command response to be provided in the data area below the command line.

The command line is cleared in preparation for the next command; the command issued is shown just above the command response. Figure 12-11 shows a command entry panel.

There are three short fields: plex, route, and wait. These are temporary overrides of the fields in the preferences panel. These values are discarded after you exit SPOC.

```
 File  Display  View  Options  Help
-------------------------------------------------------------------
PLX0              IMS Single Point of Control
Command ===>                                   command line
---------------------- Plex . _____ Route . IMS13____ Wait . ____
Response for:

                                               command response



F13=Help F15=Exit F16=Showlog F18=Expand  F21=Retrieve  F24=Cancel
```

Figure 12-11 Command entry panel

IMSplex command and response pages with a lot of data will contain a scrolling indicator: "More: <-+> ".

< More data to the left

- More data below

+ More data above

> More data to the right

Some IMSplex command responses will have key columns that do not scroll horizontally, so the user can have a point of reference. The columns are defined as keys by the XML.

12.18.1 Command entry using shortcuts

Figure 12-12 shows an example of a command entered using a shortcut:

QRY TRAN This was defined as a shortcut to be **QRY TRAN NAME(A*) SHOW(ALL)**

And this is the command that would be sent to OM and IMS.

```
 File  Display  View  Options  Help
-----------------------------------------------------------------
PLX0                IMS Single Point of Control
Command ===> QRY TRAN
----------------------- Plex . _____ Route . IMS13____ Wait . ____
Response for:
```

QRY TRAN command without parameters uses defaults from SHORTCUTS.

```
F13=Help F15=Exit F16=Showlog F18=Expand  F21=Retrieve  F24=Cancel
```

Figure 12-12 Command entered using a shortcut

When a short cut is used with additional parameters, the PREFERENCEs determine whether these parameters are to be added to, or replace, the short-cut parameters.

Since in the earlier example, it was opted to override defaults when explicit parameters are entered, the command shown in Figure 12-13 would be the one entered to OM and IMS.

```
 File  Display  View  Options  Help
-----------------------------------------------------------------
PLX0                IMS Single Point of Control
Command ===> QRY TRAN NAME(B*) SHOW(QCNT)
----------------------- Plex . _____ Route . IMS13____ Wait . ____
Response for:
```

QRY TRAN command with parameters may override or be merged with parameters defined in SHORTCUT.

Depends on PREFERENCE.

```
F13=Help F15=Exit F16=Showlog F18=Expand  F21=Retrieve  F24=Cancel
```

Figure 12-13 Command entered using a shortcut with additional parameters

12.19 Command response

The different commands responses are:

- IMSplex command response
- IMS command response

12.19.1 IMSplex command response

IMSplex command response is in a tabular format. The data is displayed in columns. Character fields are typically left aligned. Numeric fields are typically right aligned.

The list is sorted according to key fields and priority set in the XML created by the command processor. The user can sort the list by positioning the cursor on a column heading and pressing Enter. Another option is to use the action bar to sort: use View -> Sort and select the column heading.

To retrieve a command use the ISPF retrieve key or position the cursor on the command text below the command line and press Enter.

The panel includes fields for the name of the IMSplex, the specific members within the IMSplex, and the wait time for the command execution. These are temporary overrides of defaults set in the 'preferences' panel. These values are discarded after you exit SPOC.

Figure 12-14 is an example of a IMSplex command response which does not fit a single screen. The + and > signs indicate more data below (+) and to the right (>).

The command entry line has been cleared, but the command text is shown in the response section.

```
  File  Display  View  Options  Help
-----------------------------------------------------------------
                   IMS Single Point of Control
Command ===> ____________________________________________________
----------------------- Plex . _____ Route . IMS13____ Wait . ____
Response for: QRY TRAN NAME(A*) SHOW(ALL)                 More:   +>
Trancode MbrName    CC PSBname      QCnt LCls    LQCnt  LLCT LPLCT
ADDINV   IMS1        0                 0
ADDINV   IMS1        0 DFSSAM04            4        0     2 65535
ADDPART  IMS3        0                 0
ADDPART  IMS3        0 DFSSAM04            4        0     2 65535
AOBMP    IMS1        0                 0
AOBMP    IMS1        0 TS2IAOB0           23        0 65535 65535
etc.
```

Display formatted by SPOC from XML response.

```
F13=Help F15=Exit F16=Showlog F18=Expand  F21=Retrieve  F24=Cancel
```

Figure 12-14 IMSplex command response

12.19.2 IMS command response

The command response for IMS commands is in a sequential format. See Figure 12-15.

- At top are some execution statistics and information.
- Below are the messages produced by the IMS command. The text is in the same format as that of prior releases. The text is prefixed by the member name. Information from each member is grouped together.

```
 File  Display  View  Options  Help
-----------------------------------------------------------------
                    IMS Single Point of Control
Command ===> ____________________________________________________
----------------------- Plex . _____ Route . IMS13____ Wait . ____
Log for . . .: /DIS STATUS DATABASE
IMSplex . . . . . : PLX0
Routing . . . . . : IMS13
Start time. . . . : 2001.199 16:43:53.31
Stop time . . . . : 2001.199 16:43:54.47
Return code . . . : 00000000

Reason code . . . : 00000000
Command master. . : IMS4

MbrName      Messages
IMS1         **DATABASE**
IMS1         STATUS UNRESTRICTED
IMS3         **DATABASE**
IMS3         BANKATMS  NOTINIT, NOTOPEN, STOPPED

F13=Help F15=Exit F16=Showlist F18=Expand  F21=Retrieve F24=Cancel
```

Command message log can be shown at top or bottom of display.

Display formatted by SPOC as received from IMS. Each line is one XML tag.

Figure 12-15 IMS command response

IMSplex commands may have log information too if there were some messages to display. For example, one system may say `no resources found` while other systems provide valid resource information. The *no resources* indication would appear in the log.

12.20 Action bar options

You can display different command oriented panels using the action bar. See Figure 12-16. Position the cursor on the “Display” word in the action bar and press Enter:

```
  File  Display  View  Options  Help
----- +----------------------------+ -------------------------
PLX0  |    1. Cmd entry & response | oint of Control
Comma |    2. Cmd entry & log      |
      |    3. Command status       |
      |    4. Command shortcuts    | x .        Route .
Respo |    5. Expand command...    |
      +----------------------------+
```

Cmd Entry and Log	shows error or non-response messages
Command Status	shows recent commands and responses
Expand Command	is used for long commands
FILE	includes SAVE AS and PRINT
VIEW	includes FIND and SORT
OPTIONS	includes setting PREFERENCES and GROUPS

Figure 12-16 Action bar options

Choose an option by typing the number or by positioning the cursor and pressing Enter. For example, “Cmd status” pulls up recent commands and their responses.

12.20.1 Hierarchy and functions of the SPOC application menu

Table 12-8 shows the hierarchy of the TSO SPOC application menu options and their functions.

Table 12-8 SPOC application menu structure

Menu item	Sub-menu item	Function
File	1. Save As	Saves the current (displayed) command response to an ISPF data set. When you save a command response, you are prompted to input a data set name. The command entered, IMSplex name, and a timestamp are included in the heading of the saved file.
	2. Print	Sends only the command response information that you have chosen to view on the screen to the ISPF list data set. When the data has been successfully sent, a confirmation message appears at the bottom of the screen. After the information has been sent to the ISPF list data set, you can manipulate the list data set file from within ISPF.
	3. Print All	Sends both log information and IMSplex command response to the ISPF list data set. When the data has been successfully sent, a confirmation message appears at the bottom of the screen. After the information has been sent to the ISPF list data set, you can manipulate the list data set file from within ISPF.
Display	1. Cmd entry & response	Changes the IMS SPOC display so that IMSplex command input area is at the top of the screen, and the command response information is at the bottom of the screen. To toggle the display so that the command message log is shown at the bottom of the screen, choose Showlog (F4).

Menu item	Sub-menu item	Function
	2. Cmd entry & log	Changes the IMS SPOC display so that IMSplex command input area is at the top of the screen, and the command message log is shown at the bottom of the screen. To toggle the display so that the command response information is shown at the bottom of the screen, choose Showlist (F4).
	3. Command status	Enables you to re-issue or delete any previously entered command from any IMS SPOC session. For commands issued during this IMS SPOC session, you can also re-display command responses. All previously entered commands are stored in your personal ISPF ISPTABL file. When you select this command, a table listing all of your previously entered commands displays. The table shows the command issued and the command status.
	4. Command shortcuts	Enables you to maintain a table of frequently used commands to save time and keystrokes.
	5. Expand command	Use this command when you need to issue an IMSplex command that is longer than 145 character command line. Using Expand command, you can issue IMSplex commands up to 1024 characters long.
View	1. Find	Enables you to search the current (displayed) command response for a specific character or series of characters. All fields that are in the command response are searched, not just the fields displayed on the screen.
	2. Sort	Enables you to sort the current (displayed) command response by a specific column, in ascending or descending order.
Options	1. Preferences	Use this command to set default IMSplex routing and IMS SPOC operating preferences.
	2. Set IMS groups	Enables you to add, delete, and set default names of IMSplex group members.
Help	1. Help for Help	Explains the general function of the help dialogs
	2. Extended help	Explains how to use the SPOC application to issued commands
	3. Keys help	Explains current function key processes
	4. Help Index	Provides an index of help topics
	5. Tutorial	Provides access to an online guide to help you use the SPOC application.
	6. About	Displays release and copyright information

Tip: Take the time to review the function key settings on each of the screens. The TSO SPOC is a CUA® compliant application, but some of the function key assignments may not be what you expect. The application uses a number of ISPF dialog keylists to provide context specific function key processing.

12.20.2 Expanded command

Use the expand command to issue commands longer than 145 characters. Long commands are edited using the ISPF editor. You can issue commands as long as 1024 bytes. The command in the editor can span multiple lines. Standard TSO continuation rules apply.

To access the expand command select **Display—>5. Expand command** from the action bar.

The user uses the normal editor commands. The copy, create, and replace commands are used to get the command from a data set and to save the command to a data set. Figure 12-17 shows a /DIS NODE command example in a few lines.

```
PLEX1      IMS Single Point of Control                    Columns 00001 00072
Command ===> _____________________________________________ Scroll ===> PAGE

 ________________________________ Plex . _____  Route . _________  Wait . _____
****** **************************** Top of Data ******************************
000001 DIS NODE  -
000002 WZ099525  -
000003 WZ1Q0005
****** **************************** Bottom of Data ***************************
```

Figure 12-17 Expanded command input

Comments can be included by using REXX comment delimiters (/* this is a comment. */). Comments are ignored when the command is sent to the IMSplex member.

ISPF editor conventions apply:

- F3 ends the editor with the information preserved and executes the command.
- If the user types `cancel`, the command is not executed.

Figure 12-18 shows the answer to the earlier command.

```
 ____________________________________________________________________________
PLEX1                    IMS Single Point of Control
Command ===> ______________________________________________________________
 ____________________________________________________________________________
 ________________________________ Plex . _____  Route . _________  Wait . _____
Log for . . : DIS NODE WZ099525 WZ1Q0005                           More:    >

IMSplex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2002.319 17:16:50.25
Stop time . . . . : 2002.319 17:16:50.26
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : IM3A

MbrName  Messages
-------- ----------------------
IM3A         NODE-USR TYPE   CID       RECD ENQCT DEQCT   QCT  SENT
IM3A         WZ099525 3277   00000000     0     0     0     0     0 IDLE STATIC
IM3A         WZ1Q0005 3277   00000000     0     0     0     0     0 IDLE STATIC
IM3A         *2002319/171650*
IM4A         NODE-USR TYPE   CID       RECD ENQCT DEQCT   QCT  SENT
IM4A         WZ099525 3277   00000000     0    14     0    14     0 IDLE STATIC
IM4A         WZ1Q0005 3277   00000000     0     0     0     0     0 IDLE STATIC
IM4A         *2002319/171650*
IM1A         NODE-USR TYPE   CID       RECD ENQCT DEQCT   QCT  SENT
IM1A         WZ099525 3277   00000000     0    18     0    18     0 IDLE STATIC
IM1A         WZ1Q0005 3277   00000000     0     0     0     0     0 IDLE STATIC
IM1A         *2002319/171650*
```

Figure 12-18 Expanded command answer

12.20.3 Defining groups

The TSO SPOC application provides an IMS group function. To access the group definition panel select **Options—>2. Set IMS groups** from the action bar. Use the group definition dialog to create user defined groupings of command processors. When a command is routed to this group, only the command processors listed will execute the command. Figure 12-19 shows the definition of several groups.

Use the blank line to create a new group. Use the “s” to select a default group, or “d” to delete a group definition.

If you delete a group name that is in the default routing list, you will be prompted to confirm the delete.

```
  Help
PLEX1              Single Point of Control Group Definitions    Row 1 to 4 of 4
COMMAND ===>

Enter a group name and member names to add a new group. Enter 's' to select a
default, or 'd' to delete a group.

Default routing . . . : IM1A,IM4A


Act   Group      IMSplex members

___   GROUP13    IM1A,IM3A

___   GROUP14    IM1A,IM4A

___   GROUP24    IM2A,IM4A

******************************** Bottom of data ********************************
```

Figure 12-19 SPOC group definition panel

12.20.4 Defining command shortcuts

The TSO SPOC application allows you to create and manage a table of selectable commands that consist of predefined parameters for your convenience.

If you issue one of the commands in the table, the additional parameters are appended to the entered command. This occurs before default routing is added.

To access and manage shortcuts, select **Display—>4. Command shortcuts** from the action bar. Figure 12-20 shows the command shortcuts screen and several defined shortcuts.

To use the command shortcuts, you must first indicate that shortcuts should be used on the preference panel as shown in Figure 12-20.

```
 File  Display  View  Options  Help
PLEX1                        SPOC Command Shortcuts            Row 1 to 6 of 6
Command ===> _____________________________________________________________________
 ____________________________________ Plex . _____  Route . IM4A     Wait . _____
Use the blank line to add commands or 'd' to delete a command.  The additional
parameters will be appended to the command when it is issued. Prefix the
command with an ampersand ('&') to replace the entire command.

Act   Cmd & resource     Additional Parameters
___   _________________  ______________________________________________________
___   &DIS A            DIS ACT REGION
___   &QP               QRY IMSPLEX SHOW(STATUS)
___   &QPA              QRY IMSPLEX SHOW(ALL)
___   /DIS ACT          REGION
___   QRY USER          SHOW(ALL)
******************************** Bottom of data ********************************
```

Figure 12-20 SPOC command shortcut definition panel

In addition to IMSplex commands and the supported classic commands, you can create a shortcut by using the ampersand (&) as the first character. In this case, the command parameters are not appended to the shortcut, but they replace the shortcut entirely.

A blank line is provided for you to define new commands. As you add new commands, they are translated to upper case. Adjacent blanks are eliminated so that only one blank is included. The command shortcuts table has the following field columns with these possibilities:

- An *action field* is provided to allow you to manage the command parameter list by entering these actions:

  / Retrieve the command (slash). The command appears in command area of the panel and is ready for execution.

  'd' Delete the command shortcut from the list.

  'i' Issue the command.

- The *Cmd & resource field* is considered the shortcut and is used as the key field. The table is sorted by this field. When you issue a command, it is compared against the shortcuts in this table, before submitting it to IMS. If there is a match, the additional parameters are appended to your command, or command replacement occurs if the shortcut begins with an ampersand (&).
- The *additional parameters field* allows you to set a default for parameters to be added to the entered command, or the command that is to replace the shortcut for shortcuts beginning with an ampersand (&).

12.20.5 Saving and printing

The TSO SPOC application includes options to print and save the currently displayed information. To print the information to your ISPF list data set, select **File—> 1. Print** or **File—> 1. Print all** from the action bar. If you select print, TSO SPOC sends only the command response information that you have chosen to view on the screen to the ISPF list

data set. When the data has been successfully sent, a confirmation message appears at the bottom of the screen. If you select print all, both log information and IMSplex command response is sent to the ISPF list data set.

You can also save the information to a data set. Select **File—> 1. Save as** from the action bar, and you will be presented with the save response and log panel as shown in Figure 12-21.

```
  File  Display  View  Options  Help
 PLEX1                              Save As
 Command ===> ____________________________________________________________

 Type the name of the dataset that is to receive the command response, then
 press Enter.

 Information to save . . 3_  1.  IMSplex information
                             2.  Log information
                             3.  All
 Dataset name  . . SAVED.SPOC1____________________________________________
 Disposition . . . 1_  1.  Old
                       2.  Mod

 SPINV    IM4A        0 DFSSAM03            1        0     2 65535     7     7
PART      IM4A        0                 0
PART      IM4A        0 DFSSAM02            1        0     2 65535     7     7
TPCCB0    IM4A        0                 2
TPCCB0    IM4A        0 PSBBA               1        0 65535 65535     0     0
TPCCB1    IM4A        0                 9
TPCCB1    IM4A        0 PSBBA               1        0 65535 65535     0     0
TPCCB2    IM4A        0                 3
TPCCB2    IM4A        0 PSBBA               1        0 65535 65535     0     0
```

Figure 12-21 SPOC save response and log panel

This panel gives you the option to save the currently displayed information. You may save command response information from IMSplex commands, or log information which will typically have command response from classic IMS commands, or both.

12.20.6 Sorting and searching results

The TSO SPOC application provides a basic text search function that applies to both classic and IMSplex command responses. A sort function is available for IMSplex responses.

Figure 12-22 shows the command entry and response for a QRY TRAN SHOW(ALL) command.

```
  File  Display  View  Options  Help
 ____________________________________________________________________________
 PLEX1                    IMS Single Point of Control
 Command ===> ______________________________________________________________
 ____________________________________________________________________________
  ___________________________________ Plex . _____  Route . __________  Wait . _____
 Response for: QRY TRAN SHOW(ALL)                                      More:   +>
Trancode MbrName    CC PSBname      QCnt LCls    LQCnt  LLCT LPLCT LCPRI LNPRI
ADDINV   IM1A        0                 0
ADDINV   IM1A        0 DFSSAM04            1        0     2 65535     7     7
ADDINV   IM4A        0 DFSSAM04            1        0     2 65535     7     7
ADDPART  IM1A        0                 0
ADDPART  IM1A        0 DFSSAM04            1        0     2 65535     7     7
ADDPART  IM4A        0 DFSSAM04            1        0     2 65535     7     7
CLOSE    IM1A        0                 0
CLOSE    IM1A        0 DFSSAM05            1        0     2 65535     7     7
CLOSE    IM4A        0 DFSSAM05            1        0     2 65535     7     7
DISBURSE IM1A        0                 0
DISBURSE IM1A        0 DFSSAM06            1        0     2 65535     7     7
DISBURSE IM4A        0 DFSSAM06            1        0     2 65535     7     7
DLETINV  IM1A        0                 0
DLETINV  IM1A        0 DFSSAM04            1        0     2 65535     7     7
DLETINV  IM4A        0 DFSSAM04            1        0     2 65535     7     7
DLETPART IM1A        0                 0
DLETPART IM1A        0 DFSSAM04            1        0     2 65535     7     7
DLETPART IM4A        0 DFSSAM04            1        0     2 65535     7     7
DSPALLI  IM1A        0                 0
DSPALLI  IM1A        0 DFSSAM07            1        0     2 65535     7     7
DSPALLI  IM4A        0 DFSSAM07            1        0     2 65535     7     7
DSPINV   IM1A        0                 0
DSPINV   IM1A        0 DFSSAM03            1        0     2 65535     7     7
DSPINV   IM4A        0 DFSSAM03            1        0     2 65535     7     7
```

Figure 12-22 SPOC results from a QRY TRAN command

To search for a particular text string, from the action bar select **View—>1. Find** to invoke the search command responses dialog shown in Figure 12-23. Enter the search string and press Enter.

```
  File  Display  View  Options  Help
 _ ┌──────────────────────────────────────────────────────────┐ ──────────────────
 P │ PLEX1        Search the Command Response                 │
 C │                                                          │
   │ Type text to find in the command response, then press    │
   │ Enter.                                                   │        Wait .
 R │                                                          │        More:   +>
Tr │                                                          │ PLCT LCPRI LNPRI
AD │ Find . . . PSBSL                                         │
AD │                                                          │ 5535     7     7
AD │                                                          │ 5535     7     7
AD │                                                          │
AD │                                                          │ 5535     7     7
AD │                                                          │ 5535     7     7
CL │                                                          │
CL │                                                          │ 5535     7     7
CL │                                                          │ 5535     7     7
DI │                                                          │
DI └──────────────────────────────────────────────────────────┘ 5535     7     7
DISBURSE IM4A        0 DFSSAM06            1        0     2 65535     7     7
DLETINV  IM1A        0                 0
DLETINV  IM1A        0 DFSSAM04            1        0     2 65535     7     7
DLETINV  IM4A        0 DFSSAM04            1        0     2 65535     7     7
DLETPART IM1A        0                 0
DLETPART IM1A        0 DFSSAM04            1        0     2 65535     7     7
DLETPART IM4A        0 DFSSAM04            1        0     2 65535     7     7
DSPALLI  IM1A        0                 0
DSPALLI  IM1A        0 DFSSAM07            1        0     2 65535     7     7
DSPALLI  IM4A        0 DFSSAM07            1        0     2 65535     7     7
DSPINV   IM1A        0                 0
DSPINV   IM1A        0 DFSSAM03            1        0     2 65535     7     7
DSPINV   IM4A        0 DFSSAM03            1        0     2 65535     7     7
```

Figure 12-23 SPOC find text dialog box

The command response display will be scrolled to the location of the search string as shown in Figure 12-24.

```
  File  Display  View  Options  Help
 ──────────────────────────────────────────────────────────────────────────────
 PLEX1                    IMS Single Point of Control                Text found
 Command ===> ____________________________________________________________________
──────────────────────────────────────────────────────────────────────────────
  ──────────────────────────────── Plex . _____  Route . _________  Wait . _____
 Response for: QRY TRAN SHOW(ALL)                                   More:  - >
Trancode MbrName    CC PSBname      QCnt LCls    LQCnt  LLCT LPLCT LCPRI LNPRI
TPCCS    IM1A        0 PSBSL               5        0 65535    10     1     1
TPCCS    IM4A        0 PSBSL               5        0 65535    10     1     1
```

Figure 12-24 results of a find command

To sort on a particular IMSplex response field, select **View—>2. Sort** from the action bar to present the sort column name selection dialog as shown in Figure 12-25. Select the column using either a '/', or the character 'A' for ascending order, or 'D' for descending order and press Enter.

```
  File  Display  View  Options  Help
PLEX1
Command ===>
 ________________
 Response for: QRY
Trancode MbrName
ADDINV   IM1A
ADDINV   IM1A
ADDINV   IM4A
ADDPART  IM1A
ADDPART  IM1A
ADDPART  IM4A
CLOSE    IM1A
CLOSE    IM1A
CLOSE    IM4A
DISBURSE IM1A
DISBURSE IM1A
DISBURSE IM4A
DLETINV  IM1A
DLETINV  IM1A
DLETINV  IM4A
DLETPART IM1A
DLETPART IM1A
DLETPART IM4A
DSPALLI  IM1A        0                0
DSPALLI  IM1A        0 DFSSAM07            1        0     2 65535     7     7
DSPALLI  IM4A        0 DFSSAM07            1        0     2 65535     7     7
DSPINV   IM1A        0                0
DSPINV   IM1A        0 DFSSAM03            1        0     2 65535     7     7
DSPINV   IM4A        0 DFSSAM03            1        0     2 65535     7     7
```

```
PLEX1                Column names     Row 1 to 14 of 18
Command ===> ________________________________________

Select a column to sort.  Use A or d to choose
ascending or descending.

/     Column name
___   Trancode
/__   MbrName
___   CC
___   PSBname
___   QCnt
___   LCls
___   LQCnt
___   LLCT
___   LPLCT
___   LCPRI
___   LNPRI
___   LLPRI
___   LSegSz
___   LSegNo
```

Figure 12-25 SPOC sort selection panel

As shown in Figure 12-26, the command response area will be sorted by the selected column. In our example, the command response output has been sorted by MbrName.

```
  File  Display  View  Options  Help
 ------------------------------------------------------------------------
 PLEX1                    IMS Single Point of Control                  Sorted
 Command ===> _____________________________________________________________
 ------------------------------------------------------------------------
  ------------------------------------ Plex . _____  Route . __________  Wait . _____
 Response for: QRY TRAN SHOW(ALL)                                        More:   +>
Trancode MbrName    CC PSBname      QCnt LCls    LQCnt  LLCT LPLCT LCPRI LNPRI
ADDINV   IM1A        0                 0
ADDINV   IM1A        0 DFSSAM04            1        0     2 65535     7     7
ADDPART  IM1A        0                 0
ADDPART  IM1A        0 DFSSAM04            1        0     2 65535     7     7
CLOSE    IM1A        0                 0
CLOSE    IM1A        0 DFSSAM05            1        0     2 65535     7     7
DISBURSE IM1A        0                 0
DISBURSE IM1A        0 DFSSAM06            1        0     2 65535     7     7
DLETINV  IM1A        0                 0
DLETINV  IM1A        0 DFSSAM04            1        0     2 65535     7     7
DLETPART IM1A        0                 0
DLETPART IM1A        0 DFSSAM04            1        0     2 65535     7     7
DSPALLI  IM1A        0                 0
DSPALLI  IM1A        0 DFSSAM07            1        0     2 65535     7     7
DSPINV   IM1A        0                 0
DSPINV   IM1A        0 DFSSAM03            1        0     2 65535     7     7
PART     IM1A        0                 0
PART     IM1A        0 DFSSAM02            1        0     2 65535     7     7
TPCCB0   IM1A        0                 2
TPCCB0   IM1A        0 PSBBA               1        0 65535 65535     0     0
TPCCB1   IM1A        0                 9
TPCCB1   IM1A        0 PSBBA               1        0 65535 65535     0     0
TPCCB2   IM1A        0                 3
TPCCB2   IM1A        0 PSBBA               1        0 65535 65535     0     0
```

Figure 12-26 Results from the sort request

12.20.7 Command status

The TSO SPOC application maintains a table of the previously entered commands. To access the command and command status panel select **Display—>3. Command** status from the action bar. Figure 12-27 shows the command and command status panel.

```
_ File  Display  View  Options  Help
 ____________________________________________________________________________
 PLEX1                    IMS Single Point of Control        Row 1 to 23 of 76
 Command ===> ______________________________________________________________
 ____________________________________________________________________________
  ________________________________ Plex . _____  Route . _________  Wait . _____
 Enter '/' to view command response, 'i' to reissue a command, or 'd' to delete
 a command.
 Act   Status      Command
 ___   Complete    QUERY TRAN NAME(*) SHOW(ALL)
 ___   Complete    RML DBRC='DB DBD(WAREDB) DBDS'.
 ___   Complete    DIS AREA ALL
 ___   Complete    DIS PSB FPSBPA
 ___   Complete    QUERY IMSPLEX SHOW(ALL)
 ___   Complete    /DEQ LTERM WH000509 PURGE
 ___   Complete    /STO USER WH000509
 ___   Complete    /STO LTERM WH000509
 ___   Complete    DIS TRAN ALL QCNT
 ___   Complete    DIS LTERM ALL QCNT
 ___   Complete    QRY IMSPLEX SHOW(ALL)
 ___   Complete    DIS ACT
 ___   Complete    CQQ STATISTICS STRUCTURE ALL
 ___               CQQ STRUCTURE STATISTICS
 ___               DIS ACT
 ___               DIS LTERM ALL QCNT
```

Figure 12-27 SPOC command and command status panel

The commands are displayed with the most recent commands at the top of the list. A status column indicates if a command has completed, is still executing, or was issued sometime in the past. The table includes an input field to allow you to resubmit a command, delete the command completely or display the command response.

There are a few line commands supported:

'**/**'	If the command was issued in this SPOC session, you can display the command response using the '/' line command.
'**d**'	You can delete a command from the list using 'd'.
'**I**'	You can reissue a command.
'**E**'	You can edit a command using the command editor.

12.21 Leaving the SPOC application

When you end your session with the TSO SPOC application, you are presented with the following options:

1. Do not exit
2. Exit and keep the command responses
3. Exit and erase the command responses

Option 1 returns you to the previous SPOC panel, and is provided in case you pressed the exit or cancel key accidentally and do not want to leave the application.

Selecting option 2 exits the SPOC application, but the responses of any commands entered in SPOC will still be available if you start SPOC again later in this TSO session. The command responses are discarded if you log off of TSO.

Option 3 discards the command responses immediately.

12.21.1 Automatic RECON loss notification (ARLN)

When an I/O error on a RECON or a CHANGE.RECON REPLACE command is issued, DBRC begins a reconfiguration process. This is true for all releases of IMS. The loss of a RECON causes the remaining good RECON to be copied to the spare. This makes the spare a member of the good RECON pair. It also leaves the RECONs without a spare.

IMS V7 added the DSP0388I message. It is sent by the system which begins the RECON reconfiguration process. The message identifies the subsystems using the RECONs. It is intended to be used by operators or automation processes. The message is of the form:

```
DSP0388I nnnn SSYS RECORD(S) IN THE RECON AT RECONFIGURATION
DSP0388I SSID=ssidname FOUND
```

The installation needs to create a new spare so that it may handle any failure of a member of the new pair. The bad RECON data set name is used for the new spare. The bad RECON must be deallocated from all subsystems which were using it before it may be deleted and redefined. In previous releases each subsystem deallocates the bad RECON the next time it attempts to access the RECONs. At that time, it will discover the change in the RECONs and reconfigure. This may take a long time for batch jobs or utilities which use DBRC.

12.22 Examples

Here are some examples referred to by explanations and messages mentioned above.

12.22.1 SCI registration

Example 12-7 SCI registration

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E

17.08.02 JOB10316 ---- THURSDAY,  13 JUN 2002 ----
17.08.02 JOB10316  IRR010I  USERID JOUKO1   IS ASSIGNED TO THIS JOB.
17.08.03 JOB10316  ICH70001I JOUKO1   LAST ACCESS AT 17:06:13 ON THURSDAY, JUNE
17.08.03 JOB10316  $HASP373 LISTRCN  STARTED - INIT A    - CLASS A - SYS SC53
17.08.03 JOB10316  IEF403I LISTRCN - STARTED - TIME=17.08.03 - ASID=03F3 - SC53
17.08.03 JOB10316  +DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1
17.08.04 JOB10316  -                                        --TIMINGS (MINS.)--
17.08.04 JOB10316  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
17.08.04 JOB10316  -LISTRCN           D           00    232    .00    .00     .0
17.08.04 JOB10316  IEF404I LISTRCN - ENDED - TIME=17.08.04 - ASID=03F3 - SC53
17.08.04 JOB10316  -LISTRCN  ENDED.  NAME-LISTINGS             TOTAL CPU TIME=
17.08.04 JOB10316  $HASP395 LISTRCN  ENDED
------ JES2 JOB STATISTICS ------
...
        2 //D        EXEC PGM=DSPURX00,PARM=('IMSPLEX=PLEX1')
        3 //STEPLIB  DD DISP=SHR,DSN=IMSPSA.IMS0.SDFSRESL
        4 //         DD DISP=SHR,DSN=IMSPSA.IMOA.MDALIB
...
DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1
         IMS VERSION 8 RELEASE 1  DATA BASE RECOVERY CONTROL          PAGE 0002
  LIST.RECON STATUS
2002.164 17:08:03.2 -04:00          LISTING OF RECON                  PAGE 0003
-------------------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS V8R1
  DMB#=13                            INIT TOKEN=02015F0058438F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
```

```
  TAPE UNIT=3480      DASD UNIT=3390      TRACEOFF   SSID=IM1A
  LIST DLOG=YES                CA/IC/LOG DATA SETS CATALOGED=YES
  MINIMUM VERSION = 6.1
  LOG RETENTION PERIOD=00.001 00:00:00.0
  COMMAND AUTH=NONE  HLQ=**NULL**
  SIZALERT DSNUM=15      VOLNUM=16     PERCENT= 95
  LOGALERT DSNUM=3       VOLNUM=16
  TIME STAMP INFORMATION:
    TIMEZIN = %SYS             -LABEL- -OFFSET-
                                UTC     +00:00
OUTPUT FORMAT:  DEFAULT = LOCORG LABEL  PUNC YYYY
                  CURRENT = LOCORG LABEL  PUNC YYYY
  IMSPLEX = PLEX1

 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          IMSPSA.IMOA.RECON1
  RECON2        COPY2          IMSPSA.IMOA.RECON2
  RECON3        SPARE          IMSPSA.IMOA.RECON3
```

If you use the provided sample SCI registration user exit, you get the following SCI registration message DSP1123I with the additional info `USING EXIT`:

```
19.00.40 JOB10437  +DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1 USING EXIT
```

You get this message even though you have stated the correct IMSPLEX parameter on the exec statement. It indicates that the final decision is made by the exit.

12.22.2 RECON access denied

If there is already an IMSplex name in the RECON ("PLEX1"), and you are trying to access this RECON without any execution parameter or an SCI user exit, the access is denied and return code 12 is issued as shown in Example 12-8.

Example 12-8 RECON access denied

```
...
19.16.52 JOB09571  IEF403I LISTRCN - STARTED - TIME=19.16.52 - ASID=0035 - SC54
19.16.53 JOB09571  +DSP1136A  RECON ACCESS DENIED, IMSPLEX NAME       NOT VALID
19.16.53 JOB09571  +DSP1136A  RECON ACCESS DENIED, IMSPLEX NAME       NOT VALID
19.16.53 JOB09571  -                                      --TIMINGS (MINS.)--
19.16.53 JOB09571  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
19.16.53 JOB09571  -LISTRCN           D           12    170    .00    .00     .0
19.16.53 JOB09571  IEF404I LISTRCN - ENDED - TIME=19.16.53 - ASID=0035 - SC54
...
1 //LISTRCN JOB (999,POK),'LISTINGS',NOTIFY=&SYSUID,
          //    CLASS=A,MSGCLASS=T,TIME=1439,
          //    REGION=0M,MSGLEVEL=(1,1)
          /*JOBPARM SYSAFF=SC54
          //*JOBPARM SYSAFF=SC47
          //*******************************************************************
          //* SC53 IM1SC , SC54 IM3SC , SC67 IM4SC
          //* IMS0 RESLIB WITH DSPSCIX0 MODIFIED WITH UMSCIX0
          //* IMOA RESLIB WITHOUT
          //*******************************************************************
          IEFC653I SUBSTITUTION JCL - (999,POK),'LISTINGS',NOTIFY=JOUK01,CLASS=A
          MSGLEVEL=(1,1)
        2 //D        EXEC PGM=DSPURX00                    <====  no parm string !
...
```

12.22.3 SCI registration failed

If the IMSplex name stated on execution parameter is wrong (no SCI address space found responsible for an IMSplex called `PLEX4` on this LPAR), the `SCI registration failed` message is issued as shown in Example 12-9.

Example 12-9 SCI registration failed

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E

20.03.21 JOB10471 ---- THURSDAY,  13 JUN 2002 ----
20.03.21 JOB10471  IRR010I  USERID JOUKO1   IS ASSIGNED TO THIS JOB.
20.03.22 JOB10471  $HASP373 LISTRCN  STARTED - INIT A    - CLASS A - SYS SC53
20.03.22 JOB10471  IEF403I LISTRCN - STARTED - TIME=20.03.22 - ASID=03F3 - SC53
20.03.22 JOB10471  +DSP1135A  SCI REGISTRATION FAILED, IMSPLEX NAME=PLEX4,
         RC=01000010, RSN=00004000
20.03.22 JOB10471  -                                        --TIMINGS (MINS.)--
20.03.22 JOB10471  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
20.03.22 JOB10471  -LISTRCN           D           12    129    .00    .00     .0
20.03.22 JOB10471  IEF404I LISTRCN - ENDED - TIME=20.03.22 - ASID=03F3 - SC53
20.03.22 JOB10471  -LISTRCN  ENDED.  NAME-LISTINGS            TOTAL CPU TIME=
20.03.22 JOB10471  $HASP395 LISTRCN  ENDED
...
          //    CLASS=A,MSGCLASS=T,TIME=1439,
          //    REGION=0M,MSGLEVEL=(1,1)
          /*JOBPARM SYSAFF=SC53
          //*******************************************************************
          //* LPAR & SCI asid STARTED :
          //* SC53    IM1ASC  PLEX1 (CSLSIxxx mbr with stmt IMSPLEX(NAME=PLEX1)
          //*******************************************************************
          IEFC653I SUBSTITUTION JCL - (999,POK),'LISTINGS',NOTIFY=JOUKO1,CLASS=A
          MSGLEVEL=(1,1)
        2 //D        EXEC PGM=DSPURX00,PARM=('IMSPLEX=PLEX4')
        3 //STEPLIB  DD DISP=SHR,DSN=IMSPSA.IMS0.SDFSRESL
```

12.22.4 SCI registration and RECON denied

If there are two SCI address spaces running on the same LPAR (another SCI participating in another IMSplex, `PLEXC` for instance), and you specify the wrong IMSplex name (PLEXC) not matching the really intended RECON (IMSPLEX value is `PLEX1`), you get the registration message (issued by the other but unwanted SCI), however, the `RECON ACCESS DENIED` message also appears because of the mismatch between IMSPLEX name `PLEX1` already inserted into your RECON and the wrong name passed by your execution parameter.

Example 12-10 SCI registration even so RECON access denied

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E

20.10.37 JOB10479 ---- THURSDAY,  13 JUN 2002 ----
20.10.37 JOB10479  IRR010I  USERID JOUKO1   IS ASSIGNED TO THIS JOB.
20.10.38 JOB10479  ICH70001I JOUKO1   LAST ACCESS AT 20:10:15 ON THURSDAY, JUNE
20.10.38 JOB10479  $HASP373 LISTRCN  STARTED - INIT A    - CLASS A - SYS SC53
20.10.38 JOB10479  IEF403I LISTRCN - STARTED - TIME=20.10.38 - ASID=03F3 - SC53
20.10.38 JOB10479  +DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEXC USING EXIT
20.10.38 JOB10479  +DSP1136A  RECON ACCESS DENIED, IMSPLEX NAME PLEXC NOT VALID
20.10.38 JOB10479  +DSP1136A  RECON ACCESS DENIED, IMSPLEX NAME PLEXC NOT VALID
20.10.39 JOB10479  -                                        --TIMINGS (MINS.)--
20.10.39 JOB10479  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
20.10.39 JOB10479  -LISTRCN           D           12    163    .00    .00     .0
```

```
20.10.39 JOB10479  IEF404I LISTRCN - ENDED - TIME=20.10.39 - ASID=03F3 - SC53
20.10.39 JOB10479  -LISTRCN  ENDED.  NAME-LISTINGS            TOTAL CPU TIME=
20.10.39 JOB10479  $HASP395 LISTRCN  ENDED
...
        2 //D        EXEC PGM=DSPURX00,PARM=('IMSPLEX=PLEXC')
...
```

Again, the DSP1123I message issues with `... USING EXIT` if the provided IBM user exit is used.

12.22.5 CHANGE RECON REPLACE

In the following example you can see the DSP0388I message listing the subsystems up and running identified by the active subsystem records flagged in RECON. We have used the **`CHANGE.RECON REPLACE(RECON1)`** control statement running the DBRC batch utility to force the notification message. This statement caused the copy of RECON2 into the SPARE and the RECON1 to be discarded.

Example 12-11 RECON change forced by REPLACE(RECON1)

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E

15.27.34 JOB10976 ---- FRIDAY,    14 JUN 2002 ----
15.27.34 JOB10976  IRR010I  USERID JOUK01   IS ASSIGNED TO THIS JOB.
15.27.35 JOB10976  ICH70001I JOUK01   LAST ACCESS AT 14:50:40 ON FRIDAY, JUNE 14
15.27.35 JOB10976  $HASP373 CHNGRCN  STARTED - INIT A    - CLASS A - SYS SC53
15.27.35 JOB10976  IEF403I CHNGRCN - STARTED - TIME=15.27.35 - ASID=03F3 - SC53
15.27.35 JOB10976  +DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1 USING EXIT
15.27.36 JOB10976  +DSP0380I  RECON2   COPY TO RECON3   STARTED
15.27.36 JOB10976  +DSP0388I  SSID=IM1A     FOUND
15.27.36 JOB10976  +DSP0388I  SSID=IM3A     FOUND
15.27.36 JOB10976  +DSP0388I  0002 SSYS RECORD(S) IN THE RECON AT RECONFIGURATION
15.27.36 JOB10976  +DSP0381I  COPY COMPLETE, RC = 000
15.27.36 JOB10976  -                                      --TIMINGS (MINS.)--
15.27.36 JOB10976  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
15.27.36 JOB10976  -CHNGRCN          D           00    241    .00    .00     .0
15.27.36 JOB10976  IEF404I CHNGRCN - ENDED - TIME=15.27.36 - ASID=03F3 - SC53
15.27.36 JOB10976  -CHNGRCN  ENDED.  NAME-LISTINGS            TOTAL CPU TIME=
15.27.36 JOB10976  $HASP395 CHNGRCN  ENDED
------ JES2 JOB STATISTICS ------
  14 JUN 2002 JOB EXECUTION DATE
           21 CARDS READ
          104 SYSOUT PRINT RECORDS
            0 SYSOUT PUNCH RECORDS
            6 SYSOUT SPOOL KBYTES
         0.02 MINUTES EXECUTION TIME
        1 //CHNGRCN  JOB (999,POK),'LISTINGS',NOTIFY=&SYSUID,
          //    CLASS=A,MSGCLASS=T,TIME=1439,
          //    REGION=0M,MSGLEVEL=(1,1)
          /*JOBPARM SYSAFF=SC53
DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1 USING EXIT
         IMS VERSION 8 RELEASE 1  DATA BASE RECOVERY CONTROL          PAGE 0002
  CHANGE.RECON REPLACE(RECON1)
DSP0380I  RECON2   COPY TO RECON3   STARTED
DSP0388I  SSID=IM1A     FOUND
DSP0388I  SSID=IM3A     FOUND
DSP0388I  0002 SSYS RECORD(S) IN THE RECON AT RECONFIGURATION
DSP0381I  COPY COMPLETE, RC = 000
DSP0242I  RECON1 DSN=IMSPSA.IMOA.RECON1
```

```
DSP0242I  REPLACED BY
DSP0242I  RECON3 DSN=IMSPSA.IMOA.RECON3
DSP0203I  COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I  COMMAND COMPLETION TIME 2002.165 15:27:36.4 -04:00
         IMS VERSION 8 RELEASE 1  DATA BASE RECOVERY CONTROL          PAGE 0003
DSP0211I  COMMAND PROCESSING COMPLETE
DSP0211I  HIGHEST CONDITION CODE = 00
******************************** BOTTOM OF DATA ********************************
```

12.22.6 RECON loss notification

In the following joblog example you can see the notification message in one of these registered DBRC instances.

Example 12-12 RECON loss notification

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E

12.34.19 STC10813 ---- FRIDAY,    14 JUN 2002 ----
12.34.19 STC10813  IEF695I START IM1ADBRC WITH JOBNAME IM1ADBRC IS ASSIGNED TO USER
12.34.19 STC10813  $HASP373 IM1ADBRC STARTED
12.34.19 STC10813  IEF403I IM1ADBRC - STARTED - TIME=12.34.19 - ASID=00C7 - SC53
12.34.21 STC10813  DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1 USING EXIT
12.34.21 STC10813  DFS3613I - DRC TCB INITIALIZATION COMPLETE  IM1A
15.27.36 STC10813  DSP1141I  RECON LOSS NOTIFICATION RECEIVED
        1 //IM1ADBRC JOB MSGLEVEL=1
        2 //STARTING EXEC IM1ADBRC,IMSID=IM1A
        3 XXIM1ADBRC PROC RGN=6M,DPTY='(14,15)',SOUT=A,
          XX           IMSID=,SYS='IMS0.',SYS2='IMOA.'
          XX*          IMSID=,SYS='IM1A.',SYS2='IMOA.'
        4 XXIEFPROC  EXEC PGM=DFSMVRC0,REGION=&RGN,
          XX           DPRTY=&DPTY,PARM=(DRC,&IMSID)
...
```

12.22.7 RECON IMSplex change denied

If you try to change your RECON IMSplex value while any other DBRC instance which is SCI registered and running within this IMSplex you will get following informational message as shown in Example 12-13.

Example 12-13 DSP1137I message

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E

12.54.41 JOB10833 ---- FRIDAY,    14 JUN 2002 ----
12.54.41 JOB10833  IRR010I  USERID JOUKO1   IS ASSIGNED TO THIS JOB.
12.54.41 JOB10833  ICH70001I JOUKO1   LAST ACCESS AT 12:08:16 ON FRIDAY, JUNE 14
12.54.41 JOB10833  $HASP373 LISTRCN  STARTED - INIT A    - CLASS A - SYS SC53
12.54.41 JOB10833  IEF403I LISTRCN - STARTED - TIME=12.54.41 - ASID=03F3 - SC53
12.54.41 JOB10833  +DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1 USING EXIT
12.54.42 JOB10833  +DSP1137I  IMSPLEX MAY NOT BE CHANGED, DBRC ACTIVE FOR
12.54.42 JOB10833  +DSP1137I  IM1ADBRC
12.54.42 JOB10833  +DSP1137I  IM3ADBRC
12.54.42 JOB10833  -                                        --TIMINGS (MINS.)--
12.54.42 JOB10833  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
12.54.42 JOB10833  -LISTRCN           D           12    211    .00    .00     .0
12.54.42 JOB10833  IEF404I LISTRCN - ENDED - TIME=12.54.42 - ASID=03F3 - SC53
12.54.42 JOB10833  -LISTRCN  ENDED.  NAME-LISTINGS             TOTAL CPU TIME=
...
```

```
DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1 USING EXIT
         IMS VERSION 8 RELEASE 1  DATA BASE RECOVERY CONTROL          PAGE 0002
    CHANGE.RECON NOPLEX
DSP1137I  IMSPLEX MAY NOT BE CHANGED, DBRC ACTIVE FOR
DSP1137I  IM1ADBRC
DSP1137I  IM3ADBRC
DSP0209I  PROCESSING TERMINATED WITH CONDITION CODE = 12
DSP0217I  THE FOLLOWING SYSIN RECORDS HAVE BEEN SKIPPED:
    LIST.RECON STATUS
DSP0218I  END OF SKIPPED SYSIN RECORDS
******************************** BOTTOM OF DATA ********************************
```

Any subsequent command after the **CHANGE.RECON IMSPLEX(...) | NOPLEX** control statement will not executed, indicated by the DSP0217I message. The message issued at the bottom of Example 12-13 for the LIST.RECON command.

12.22.8 SCI user exits

In the following examples we use a different SCI user exit modified with some RECON data set entries, maintained by SMP/E and intended for global use, for example, across your test and development environments (and their RECONs), ignoring any IMSplex value specified in the execution parameter. The following examples are shown only for documentation purposes, your version should be modified to suit your own environment.

In this simple user exit modification there is only one set of RECONs (assigned for PLEX1) intended to register with SCI and use ARLN, whereas the other entries (IMSPSA.IM0B.RECON* , IMS810.RECON*) are intended to bypass any SCI registration.

The first example is a job including sample JCL for compile and link of the sample exit. It is only intended to use as input for an SMP JCLINREPORT job to create the necessary SMP/E target zone elements preparing the target for the apply step (of course, in substitute you can define the MOD and LMOD element explicitly with the known SMP/E statements).

Example 12-14 JCLINREPORT input

```
******************************** Top of Data ***************************
//IMSEXITS JOB (999,POK),
// 'HK',
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
// NOTIFY=&SYSUID,
// REGION=64M
//*
//*  JCLLIB ORDER=(IMS810C.PROCLIB)
/*JOBPARM L=9999,SYSAFF=*
//*********************************************************************
//* IMS EXIT FROM SAMPLE LIB
//* !! THIS IS ONLY THE DECK FOR JCLINREPORT !!
//* C.SYSIN's LLQ DSNAME IS MAPPED TO SRC ENTRY DISTLIB(DDD)
//* L.SYSLMOD's LLQ DSNAME IS MAPPED TO LMOD SYSLMOD(DDD)
//* INCLUDE DDDEF(MODUL) IS CREATING MOD ENTRY
//*********************************************************************
//C        EXEC PGM=ASMA90,PARM='OBJECT,NODECK'
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,DSN=IMS810C.ADFSMAC
//         DD DISP=SHR,DSN=SYS1.MACLIB
//         DD DISP=SHR,DSN=ASM.SASMMAC2
//SYSLIN   DD ...
...
//SYSIN    DD DISP=SHR,DSN=IMS810C.ADFSSMPL(DSPSCIX0)
```

```
//L        EXEC PGM=IEWL,
//         PARM='XREF,LIST,RENT',COND=(0,LT,C)
//SYSPRINT DD SYSOUT=*
//SYSLMOD  DD DISP=SHR,DSN=IMS810C.SDFSRESL
//SYSUT1   DD UNIT=(SYSALLDA,SEP=(SYSLMOD,SYSLIN)),
//            DISP=(,DELETE,DELETE),SPACE=(CYL,(1,1))
//SYSLIN   DD DISP=(OLD,DELETE,DELETE),
//            DSN=*.C.SYSLIN,VOL=REF=*.C.SYSLIN
//         DD *
  INCLUDE  ADFSLOAD(DSPSCIX0)
    ENTRY  DSPSCIX0
     MODE  AMODE(31),RMODE(ANY)
     NAME  DSPSCIX0(R)
/*
```

After creation of the MOD and LMOD elements (and after the RECEIVE process) the APPLY (CHECK at first!) of the following USERMOD shown in Example 12-15 will link the changed (++SRCUPD.ated) SCI user exit into your target SDFSRESL.

Example 12-15 SMP/E USERMOD for an SCI user exit change

```
++ USERMOD(UMSCIX0)  /* DSPSCIX0 USER EXIT      */.                    00010002
++ VER(P115) FMID(HMK8800) .                                           00020002
++ SRCUPD(DSPSCIX0) DISTLIB(ADFSSMPL).                                 00030004
./ CHANGE NAME=DSPSCIX0                                                00040002
*/*01*   CHANGE-ACTIVITY: 05/24/02 table entries for                */ 07551000
*/*                      IMSPSA.IMOA.RECON1,2,3  using PLEX1        */ 07552000
*/*                      IMSPSA.IMOB.RECON1,2,3  no ARLN            */ 07553007
*/*                      IMS810C.RECON1,2,3      no ARLN            */ 07554007
*---  DONT CARE ABOUT EXEC VALUE , ONLY TABLE IN USE                   08849907
*        L     R4,8(,R1)          R4 = A(IMSPLEX VALUE FROM EXEC CARD) 08850000
*-----------------------------------------------------------------------  08900000
* IF IMSPLEX= SPECIFIED ON EXEC CARD, IGNORE IT !                       08950007
*-----------------------------------------------------------------------  09000000
*        LTR   R4,R4              IMSPLEX= ON EXEC STATEMENT?          09050000
*        BZ    NEXECPRM           IF NOT SPECIFIED, BRANCH             09100000
*        MVC   0(PNL,R3),0(R4)    ELSE COPY VALUE TO RETURN AREA       09150000
*        SR    R15,R15            SET RC00                             09200000
*        B     EXIT               AND RETURN TO DBRC                   09250000
         DC    CL(DSNL)'IMSPSA.IMOA.RECON1'         RECON NAME          11811000
         DC    CL(PNL)'PLEX1'         IMSPLEX NAME                      11812000
         DC    XL(RCL)'00000000'      RC00 = USE THE IMSPLEX NAME       11813000
         DC    CL(DSNL)'IMSPSA.IMOA.RECON2'         RECON NAME          11814000
         DC    CL(PNL)'PLEX1'         IMSPLEX NAME                      11815000
         DC    XL(RCL)'00000000'      RC00 = USE THE IMSPLEX NAME       11816000
         DC    CL(DSNL)'IMSPSA.IMOA.RECON3'         RECON NAME          11817000
         DC    CL(PNL)'PLEX1'         IMSPLEX NAME                      11818000
         DC    XL(RCL)'00000000'      RC00 = USE THE IMSPLEX NAME       11819000
         DC    CL(DSNL)'IMSPSA.IMOB.RECON1'         RECON NAME          11820000
         DC    CL(PNL)'*****'         IMSPLEX NAME                      11821000
         DC    XL(RCL)'00000004'      RC04 = NO SCI REGISTRATION        11822000
         DC    CL(DSNL)'IMSPSA.IMOB.RECON2'         RECON NAME          11823000
         DC    CL(PNL)'*****'         IMSPLEX NAME                      11824000
         DC    XL(RCL)'00000004'      RC04 = NO SCI REGISTRATION        11825000
         DC    CL(DSNL)'IMSPSA.IMOB.RECON3'         RECON NAME          11826000
         DC    CL(PNL)'*****'         IMSPLEX NAME                      11827000
         DC    XL(RCL)'00000004'      RC04 = NO SCI REGISTRATION        11828000
         DC    CL(DSNL)'IMS810C.RECON1'         RECON NAME              11828100
         DC    CL(PNL)'*****'         IMSPLEX NAME                      11828200
         DC    XL(RCL)'00000004'      RC04 = NO SCI REGISTRATION        11828300
```

```
         DC    CL(DSNL)'IMS810C.RECON2'             RECON NAME          11828400
         DC    CL(PNL)'*****'         IMSPLEX NAME                      11828500
         DC    XL(RCL)'00000004'      RC04 = NO SCI REGISTRATION        11828600
         DC    CL(DSNL)'IMS810C.RECON3'             RECON NAME          11828700
         DC    CL(PNL)'*****'         IMSPLEX NAME                      11828800
         DC    XL(RCL)'00000004'      RC00 = USE THE IMSPLEX NAME       11828900
./ ENDUP                                                                11829000
```

This exit has been modified to match RECON data set names with IMSplex values regardless of any IMSPLEX execution parameter. If any DBRC instance is running with a customized user exit as shown in previously in Example 12-15, it will get access to the right RECON despite the "wrong" IMSPLEX name being stated in the execution parameter (Example 12-16).

Example 12-16 modified SCI user exit ignoring execution parameter

```
J E S 2  J O B  L O G  --  S Y S T E M  S C 5 3  --  N O D E
21.07.47 JOB10501  ICH70001I JOUK01   LAST ACCESS AT 20:10:38 ON THURSDAY, JUNE
21.07.47 JOB10501  $HASP373 LISTRCN  STARTED - INIT A    - CLASS A - SYS SC53
21.07.47 JOB10501  IEF403I LISTRCN - STARTED - TIME=21.07.47 - ASID=03F3 - SC53
21.07.47 JOB10501  +DSP1123I  DBRC REGISTERED WITH IMSPLEX PLEX1 USING EXIT
21.07.48 JOB10501  -                                      --TIMINGS (MINS.)--
21.07.48 JOB10501  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU    SRB  CLOCK
21.07.48 JOB10501  -LISTRCN          D           00    223    .00    .00     .0
21.07.48 JOB10501  IEF404I LISTRCN - ENDED - TIME=21.07.48 - ASID=03F3 - SC53
21.07.48 JOB10501  -LISTRCN  ENDED.  NAME-LISTINGS            TOTAL CPU TIME=
21.07.48 JOB10501  $HASP395 LISTRCN  ENDED
------ JES2 JOB STATISTICS ------
        1 //LISTRCN  JOB (999,POK),'LISTINGS',NOTIFY=&SYSUID,
          //    CLASS=A,MSGCLASS=T,TIME=1439,
          //    REGION=0M,MSGLEVEL=(1,1)
          /*JOBPARM SYSAFF=SC53
          //*********************************************************************
          //* LPARS & SCI asids STARTED :
          //* SC53    IM1ASC  PLEX1 (CSLSIxxx mbr with stmt IMSPLEX(NAME=PLEX1)
          //* SC54    IM3ASC  PLEX1 (CSLSIxxx mbr with stmt IMSPLEX(NAME=PLEX1)
          //* SC67    IM4ASC  PLEX1 (CSLSIxxx mbr with stmt IMSPLEX(NAME=PLEX1)
          //*
          //* SC53    IMCCSC  PLEXC (CSLSIxxx mbr with stmt IMSPLEX(NAME=PLEXC)
          //*********************************************************************
...
        2 //D        EXEC PGM=DSPURX00,PARM=('IMSPLEX=PLEXC')
        3 //STEPLIB  DD DISP=SHR,DSN=IMSPSA.IMS0.SDFSRESL
        4 //        DD DISP=SHR,DSN=IMSPSA.IMOA.MDALIB
...
IMS VERSION 8 RELEASE 1  DATA BASE RECOVERY CONTROL
   LIST.RECON STATUS
2002.164 21:07:47.4 -04:00          LISTING OF RECON
-----------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS V8R1
  DMB#=13                           INIT TOKEN=02015F0058438F
...
  LOGALERT DSNUM=3       VOLNUM=16
  TIME STAMP INFORMATION:
    TIMEZIN = %SYS             -LABEL- -OFFSET-
                                UTC     +00:00
    OUTPUT FORMAT:  DEFAULT = LOCORG LABEL  PUNC YYYY
                    CURRENT = LOCORG LABEL  PUNC YYYY
  IMSPLEX = PLEX1
```

```
-DDNAME-      -STATUS-       -DATA SET NAME-
 RECON1        COPY1          IMSPSA.IMOA.RECON1
 RECON2        COPY2          IMSPSA.IMOA.RECON2
 RECON3        SPARE          IMSPSA.IMOA.RECON3
```

Since any DBRC instance, including batch and utility jobs running with DBRC=Y, needs SCI up and running for the registration and the use of ARLN, you have to make sure of the following:

1. On every LPAR available to schedule your jobs, there is one SCI active and a member of the same IMSplex.

   Or,

2. You might define SCHEDULE environment(s) in your WLM policy describing SCI started tasks based on unique IMSplex name(s) as elements to control your batch scheduling across your sysplex and to avoid any batch job starting outside of your *IMSplex* LPARs.

12.23 Language Environment (LE) commands

Commands issued through the SPOC allow you to specify LE run time options with “filters”. Filters are one or more of the keywords on the command (that is, TRAN, USERID, PGM, or LTERM). The commands are used to change the LE run time options. Commands are issued from and responses received at the SPOC.

For complete information on the new commands, refer to *IMS Version 8: Command Reference,* SC27-1291.

12.23.1 Enhanced DL/I INQY call

The DL/I INQY call with the new LERUNOPT subfunction is used to retrieve the dynamic LE run time options.

12.23.2 UPDATE LE command

The **UPDATE LE** command is used to change LE run time options for a transaction code, logical terminal (LTERM name), user ID and/or application program. At least one of the keywords and filters (TRAN, LTERM, USERID, or PGM) must be specified on the UPDATE LE command.

The SET keyword is used to:

- Specify the LE run time options that you want to dynamically change. The new or changed run time options are designated by the LERUNOPTS specification(s).
- Specify whether IMS is to enable (YES) or disable (NO) the ability to dynamically override LE run time options. The LEOPT keyword is used for this purpose. The UPD LE SET(LEOPT(YES)) command enables LE dynamic run time options for every IMS in the IMSplex. The command *cannot* be directed to a specific IMS in the IMSplex.

When dynamic run time option support is disabled (LEOPT(NO)), UPDATE LE command issued to change run time options result in run time option updates in an area of storage used internally by IMS. However, the updates are not used until dynamic option support is enabled SET(LEOPT(YES)). The UPDATE command is entered from the SPOC. The command is routed to all IMS systems in the IMSplex. The dynamic run time option overrides supplied by the UPDATE LE command are only allowed when dynamic LE overrides have been enabled.

The syntax for the UPDATE LE command is as follows:

```
UPDATE LE TRAN(trancode) LTERM(ltermname) USERID(userid) PGM(pgmname)
SET(LERUNOPTS(xxxxxxxx))
```

Example 12-17 shows two sample UPDATE LE commands. Note that generic parameters on the filters are not usable with the UPDATE LE command.

Example 12-17 Update LE command

```
UPD LE TRAN(PART) LTERM(TERM1) USERID(USER1) SET(LERUNOPTS(xxxxxxxx))
UPD LE SET(LEOPT(YES))
```

Note: The LE override function only takes place at program scheduling time, so filtering on LTERM or USERID is only effective for the first message processed following program scheduling.

12.23.3 DELETE LE command

The **DELETE LE** command is used to delete LE run time options for a transaction code, logical terminal (LTERM name), user ID and/or application program. At least one of the keywords and filters (TRAN, LTERM, USERID, or PGM) must be specified on the DELETE LE command.

The DELETE command is entered from the SPOC. The command is routed to all IMS systems in the IMSplex. The run time option overrides supplied by the DELETE LE command are only allowed when dynamic LE run time option overrides have been enabled.

The syntax for the DELETE LE command is as follows:

```
DELETE LE TRAN(trancode) LTERM(ltermname) USERID(userid) PGM(pgmname)
```

Example 12-18 shows a sample DELETE LE command. Note that you can use generic parameters on the filters with this command.

Example 12-18 Delete LE command

```
DEL LE TRAN(PART*) LTERM(TERM1)
```

12.23.4 QUERY LE command

The QUERY LE command is used to query LE run time parameters in effect for a transaction code, logical terminal (LTERM name), user ID and/or application program.

The SHOW parameter specification displays output fields selected in the filter. At least one SHOW (ALL | TRAN | LTERM | USERID | PGM | LERUNOPTS) field is required.

The syntax for the QUERY LE command is as follows:

```
QUERY LE TRAN(trancode) LTERM(ltermname) USERID(userid) PGM(pgmname) SHOW(ALL |
TRAN  | LTERM |  USERID |  PGM |  LERUNOPTS)
```

Example 12-19 shows a sample QUERY LE command. Note that you can use generic parameters on the filters with this command.

Example 12-19 Query LE command

```
QRY LE LTERM(TERM2) USERID(USER2) SHOW(ALL)
```

12.23.5 DFSBXITA

CEEBXITA is an existing LE exit and continues to function as it has in previous releases. When CEEBXITA is linked with the Language Environment initialization/termination library routines during installation, it functions as an installation-wide user exit. When CEEBXITA is linked in your load module, it functions as an application-specific user exit. The application-specific exit is used only when you run that application. The installation-wide assembler user exit is not executed.

What is new in IMS Version 8 is an IMS supplied version of CEEBXITA. The exit is delivered as DFSBXITA but must be reassembled and linked as a part of CEEBXITA, and, in simple terms, builds a table to be used with the default options table to contain the overrides. This exit will be invoked when a program is scheduled into a dependent region. Once program scheduling completes, the options remain in effect for all subsequent transactions processed during that PSB schedule — regardless of which userid or LTERM the subsequent transactions came from.

Code in the exit checks to see if the environment is IMS or not. If it is IMS then an INQY call is made to retrieve the LE run time overrides; otherwise for non-IMS environments, the code is bypassed.

In Java regions, the enclave initialization takes place before the application is known or given control. Therefore, if DFSBXITA is linked with the Java application, it will not be invoked. If DFSBXITA is linked with the LE libraries, it will be invoked, but the ECP (pointer to parameters that made the last call to DL/I) will not be allocated at this point of a Java region initialization. The Java region CREATE THREAD will make the DLI INQY LERUNOPT call. If parameters are returned as a result of the LERUNOPT call, the C SETENV call will be issued to pass the overrides parameters to the application.

DFSBXITA can be modified by user if needed but the exit must be linked as CEEBXITA. If you have an existing CEEBXITA, you will need to incorporate the logic from DFSBXITA into your existing exit to provide dynamic LE options support.

- Incorporate DFSBXITA source into to CEEBXITA.
- Assemble and link-edit CEEBXITA.
- If exit invoked from IMS environment:
  - Execute DFSBXITA logic prior to returning control to LE.
- If exit invoked from non-IMS environment:
  - Decide where to branch to in CEEBXITA for non-IMS related processing.

12.23.6 DL/I INQY LERUNOPT call

The DL/I INQY LERUNOPT call is issued by DFSBXITA to retrieve the LE run time option overrides for an application. If an override parameter string is found, the address of the run time overrides is returned by the DL/I INQY LERUNOPT call to the exit routine. The output from the DL/I INQY LERUNOPT call contains AIBRETRN and AIBREASN codes. The AIBRSA2 field contains either an address of the LE run time parameter string or zero. AIBRSA2 contains the address of the LE run time options parameter string when:

- Overrides are enabled and the override string was successfully found.

AIBRSA2 contains zero for each of the following conditions:

- Overrides are disabled for the IMS system.
- Overrides are allowed for the system, but the override table has not yet been initialized.
- Overrides are allowed for the system, but there is no applicable override string for the caller.

Run time options are defined to IMS using the UPDATE LE command. In the following command, the SET(LERUNOPTS(xxxxxxxx)) specification is used to define the run time overrides:

```
UPD LE TRAN(tttt) LTERM(llll) USERID(uuuu) PGM(pppp) SET(LERUNOPTS(xxxxxxxx)).
```

MPP and JMP region overrides are based on the combination of transaction name, LTERM name, user ID, or program name. IFP, BMP and JBP region overrides are based on the program name. Message driven BMP regions may also have overrides specified by transaction name.

If more than one override is appropriate for a transaction, the override value is chosen based on which table entry matched the most filters. Therefore, if only a TRAN is specified on one entry (and it matches), and on another entry the LTERM and PGM are specified and match, the options in the second entry will be returned because it specifies more filters. In the case of a tie, the first entry to appear in the table is used. No filter type takes precedence over another. Therefore if two entries match, one specifying just a TRAN and the other specifying just a PGM, the entry that gets used is the one that appears first in the table.

12.23.7 Run time options override illustration

Figure 12-28 follows the logic (at a high level) when the TERMTHDACT option is set for TRANA. IMS must be enabled for this capability by having LEOPT=Y or setting it on with the UPD LE command.

When the command gets to IMS, a *Runtime Option Overrides Table* is updated with the new/changed options. In this case, TRANA is updated for TERMTHDACT. When TRANA is scheduled into any dependent region, DFSBXITA will be driven and the LE option set for that transaction.

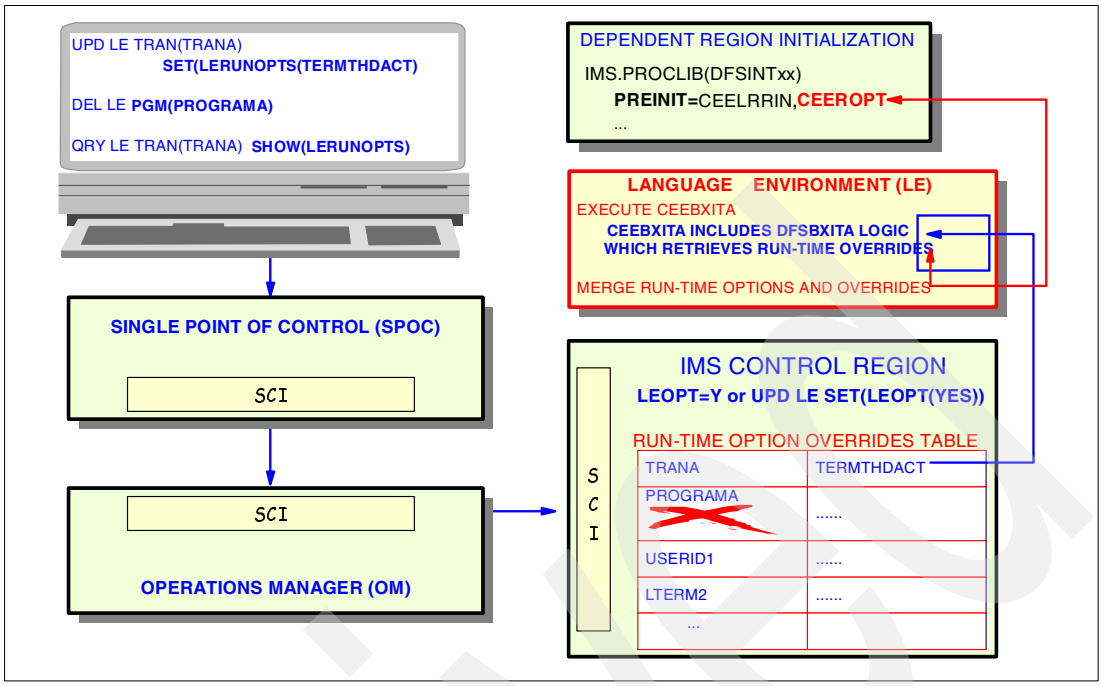

Figure 12-28 Run time options override illustration

13

Operating VTAM Generic Resources

When you start using a new feature, such as VTAM Generic Resources for IMS, you will need to determine if there are any changes to existing operating procedures, or if there are any new operating procedures that must be put in place. VTAM Generic Resources should be transparent to the user, but there are some new procedures necessary to operate a VTAM Generic Resources environment in IMS, such as these topics of our discussion:

- Activating VTAM Generic Resources for IMS
- Starting IMS
- Session establishment with VTAM Generic Resources
- Stopping VTAM Generic Resources
- Stopping IMS
- Removing affinities in VTAM Generic Resources

13.1 Activation of VTAM Generic Resources for IMS

You can activate VTAM Generic Resources in two ways:

1. Use the GRSNAME parameter in the DFSPBxxx member, and recycle IMS.
2. Issue the /START VGR GRSNAME command.

It is currently not possible to alter the GRSNAME while IMS is running. It would need a recycle of IMS. If the /START VGR command was entered and the GRSNAME was keyed in incorrectly, you would have to recycle IMS to remove the assignment to that generic resources group to that IMS.

A generic resource name is dynamically allocated when activated, and does not need to be pre-defined to VTAM. Once the GRSNAME is activated, VTAM dynamically builds an entry in the Coupling Facility. As soon as terminals begin connecting to IMS using the generic resource name, then affinities are created in the VTAM Generic Resources table for that terminal to the IMS system that it is connecting to.

13.1.1 IMS startup

When IMS is cold started (or COLDCOMM), then **all** session affinities to the IMS are deleted at startup time.

After IMS has been restarted, a /START DC command is issued. This command opens the VTAM ACB with the APPLID for that IMS subsystem. If GRSNAME= has been specified on the IMS execution parameters, then IMS joins the GRG specified. If this is the first IMS to be started from the group, then VTAM creates the GRG and adds this IMS(APPLID) to it.

As more IMS subsystems are started, VTAM adds them to the member list in the ISTGENERIC structure. The member list holds the names of all IMS subsystems in this GRG. Against each member is held the number of sessions currently active. Within the ISTGENERIC structure is also a list of affinities: terminals that are attached or must reattach to a particular IMS subsystem. Although VTAM maintains the affinity table, it does not own it. Each IMS subsystem owns and controls its affinities.

13.1.2 Session establishment

When a user selects to log on to IMS using a generic resource name, VTAM first checks to see if an affinity with a particular IMS subsystem is outstanding for this LU.

If an affinity exists, VTAM establishes a session between the LU and the IMS subsystem named in the affinity table. If no affinity exists, VTAM connects the user to any IMS subsystem in the GRG, based on VTAM resource resolution methods.

The order followed to assign a session to an APPLID is as follows:

1. Existing session affinity
2. VTAM Generic Resource Resolution Exit
3. MVS Workload Manager (WLM)

VTAM checks session counts for all members in the GRG to decide which IMS to establish a session with.

An end user may want to establish a session with a particular IMS in a Parallel Sysplex. In this case, VTAM connects the user to the specified IMS subsystem and performs no session-resolution processing.

If the MVS Work Load Manager (WLM) is invoked by VTAM to assist in session resolution, then WLM takes priority over VTAM session resolution. The MVS WLM must be in goal mode for it to be used.

If an affinity is outstanding, VTAM attempts to reconnect the session, even if the IMS subsystem is unavailable. Session setup fails if the IMS subsystem is unavailable, otherwise a normal session is re-established. The user has now established a session to an IMS subsystem from within the GRG and an affinity has been created between the LU and the IMS subsystem to which it is attached.

13.1.3 Stopping VTAM Generic Resources

An IMS system can be removed from a generic resources group by issuing the /STOP VGR command in IMS. This command prevents any new logons to that IMS if those terminals logging on use the GRS name, and have no affinity to that IMS.

If a static terminal was not connected to that IMS at the time the /STOP VGR command was issued, and the terminal has an affinity to that IMS, then the next logon to the GRS name connects the terminal to that IMS.

This command only prevents a terminal logging onto that IMS system using the generic resource name if no affinity to that IMS exists for that terminal. The /STOP VGR command stops future connections to the IMS of any terminals logging onto the GRS name, but does not stop terminals from logging onto that IMS using the APPLID, nor any terminal that has an affinity to that IMS. The /STOP VGR command does not remove any terminal affinities to that IMS.

A start command, /START VGR, would allow the IMS to rejoin the generic resources group. The GRSNAME parameter does not need to be respecified on the start command that follows the stop command.

13.1.4 Backing out VTAM Generic Resources during IMS shutdown

For a permanent de-activation of VTAM Generic Resources, the GRSNAME, GRAFFIN and GRESTAE parameters must be taken out, and any changes to the logoff/signoff exits backed out. This should be followed by cycling IMS. IMS should be taken down with the /CHE... LEAVEGR command to ensure a removal of all affinities to that IMS.

13.1.5 Removing affinities in VTAM Generic Resources environments

An affinity is created whenever an LU connects to an IMS subsystem. These affinities are normally deleted at session termination. Session termination can consist of a user logoff, a VTAM-driven LOSTERM exit, IMS normal shutdown, or IMS failure. Affinities can remain after session termination if the terminal has a significant status. A significant status indicates that an affinity should remain after session termination to maintain a consistency between a terminal and that IMS. These are each considered a significant status:

- Terminal in response mode
- Terminal in conversation mode
- Exclusive (/EXCLUSIVE)
- Test (/TEST)
- Preset Destination (/SET xxxxxxxx)
- MFS Test (/TEST MFS)
- 3600 Financial/SLUP(LU0) devices
- MVS failure

An affinity remains in the case of an MVS failure because the IMS extended subtask abend exit (ESTAE) is not driven. This is distinct from an IMS failure in which affinities are deleted by the abend routines.

If necessary, there are ways that affinity deletion can be forced. Two IMS exits have been enhanced to allow deletion of affinities, the signoff exit (DFSSGFX0) and the logoff exit (DFSLGFX0).

Affinity deletion can also be accomplished when shutting down IMS. The LEAVEGR option on the /CHECKPOINT command, signifies that this IMS will leave the GRG and all affinities are to be cleared. If the DC portion of IMS is cold started, then existing affinities are deleted.

Existing affinities may cause problems in the establishment of later sessions. If there is an existing affinity for a terminal and a user then attempts to connect to a specific APPLID, a session is created with that APPLID. Affinities are not globally known; they are held by each individual IMS. When the user later re-establishes a session with the original IMS, significant status is also re-established.

13.2 Command processing considerations

In an IMSplex environment, commands issued to the IMS Version 8 Operations Manager (OM), can behave differently than commands issued to a single IMS system.

13.2.1 Command enhancements for VTAM Generic Resources with IMS

With the introduction of VTAM Generic Resources in IMS, changes have been made to assist in the operation of IMS. There are new parameters to existing commands, and the output of some commands has changed to include some information related to VTAM Generic Resources. There are also commands in VTAM that might facilitate operating in a VTAM Generic Resources environment.

IMS commands

The commands that have specific information for a VTAM Generic Resources environment are discussed briefly below. Refer to the *IMS Version 8: Command Reference,* SC27-1291, for more information.

/DIS AFFIN NODE nodename

The /DIS AFFIN command retrieves information from the ISTGENERIC structure on the Coupling Facility. The terminal does not need to be attached to that IMS system that the command is issued on, for the information to be shown in the display. There are no wildcard characters in the display, or generic commands, nor is there an IMS command to display all the affinities.

This command makes it easier to operate in a VTAM Generic Resources environment. If a user has a problem with an IMS session, the /DIS AFFIN command can be issued on any IMS system in the same generic resources group to find out which system the user is connected to. The operator can then know where to start looking at any of the users' problems.

If the data in the APPLID field is N/A, then there is no information in the ISTGENERIC structure in the Coupling Facility regarding that node.

/DIS ACTIVE

The GRSNAME, if one is specified or active, is included in the /DIS ACTIVE display. This command can be used to see what the GRSNAME is for a specific IMS as well as to show

whether the generic resource name is active or stopped for the IMS on which the command was entered.

/DIS APPC

This output has been changed to include a field to describe the generic resource group to which the APPC LU is connected to. There will only be a value in the generic resource group field, if VTAM Generic Resources for APPC has been activated. VTAM Generic Resources is defined for the specific APPC LUs in APPC, and is not in any way connected to IMS.

/CHECKPOINT... LEAVEGR

The LEAVEGR keyword can only be used with a shutdown checkpoint. If this keyword is used on the shutdown command, then all the VTAM affinities to that IMS system are deleted from the VTAM affinity table. It is important to remember that the nodes or users might still have a significant status in IMS, even though the affinities have been deleted from the VTAM affinity table. It is recommended that a LEAVEGR should only be issued if the IMS DC system will be restarted with a cold start.

/START VGRS GRSNAME grsname

This command can be used to activate VTAM Generic Resources if GRSNAME= is not specified in the DFSPBxxx member. This command can also be used to reintroduce the IMS to the VTAM generic resource group after it was removed using the /STOP VGRS command. This command is not executed if there are existing VTAM sessions with that IMS.

You must first issue a /CLSDST NODE command from a non-VTAM terminal (such as, the system console) for any active VTAM terminals, including the MTO, before the command takes effect. If there are any VTAM terminals connected to IMS, the following error message is issued:

```
DFS3691W 17:12:11 GENERIC RESOURCES START/STOP FAILED, SETLOGON RC=14, FDB2=86
```

You must first issue a /CLSDST NODE command from a non-VTAM terminal (such as, the system console) for any active VTAM terminals, including the MTO, before the command takes effect.

/STOP VGRS

The /STOP VGRS command can be used to remove an IMS from a generic resources group. This does not prevent a terminal from logging onto the IMS using the APPLID name. A terminal that has an affinity for the IMS is able to logon to the IMS using the GRS name.

VTAM commands

You can use the following VTAM DISPLAY NET command to display all members of a generic resource group, if you give the generic resource name as the ID parameter:

```
/D NET,ID=vgrname
```

If you specify the IMS APPLID as the ID parameter, then you can identify the possible generic resource name from the reply.

Displaying affinities in VTAM

/D NET,GRAFFIN,GRNAME=xxx,,,,

This command is only available from OS/390 Version 2.7 and higher. Information on the command can be found in the *OS/390 V2R7.0 eNetwork CS SNA Operation* manual. Make sure that this command is used efficiently, otherwise there could be VTAM slowdowns as a result of the amount of information requested.

13.3 IMS commands for terminating affinities that persist

When you terminate a session by logging off, IMS terminates the affinity between your node and the IMS generic resource member. However, in some situations, affinities persist. You can use several methods to terminate these affinities. To display the list of IMSs and their affinities with specific nodes, you can use this command:

```
/DISPLAY AFFINITY
```

A terminal cannot participate in session balancing if it has a persisting affinity with a particular IMS from a previous session. When VTAM manages the VTAM Generic Resources affinity, affinity is automatically reset at session termination. When IMS manages the VTAM Generic Resources affinity, an affinity persists after a logoff or IMS failure in each of the following situations:

- The terminal is static with LOCAL status recovery mode, and has end-user significant status (Conversation, STSN, or Fast Path).
- The terminal is ETO SLUP or 3600 Finance terminal with LOCAL status recovery mode and RCVYSTSN=YES.
- The terminal is an ISC parallel-session terminal with LOCAL status recovery mode that has not quiesced.
- The terminal is an ISC parallel-session terminal with LOCAL status recovery mode that has not quiesced.
- The terminal is an ISC parallel-session terminal, and one or more of the parallel sessions has not quiesced.

You can terminate an affinity using any of the following methods:

- Enter the /CHECKPOINT command with the LEAVEGR keyword.

  ```
  /CHE FREEZE LEAVGER
  ```

- Use the logoff or signoff exit routine to reset terminal status.
- Cold start the IMS TM subsystem, and specify the GRSNAME when restarting IMS.

13.3.1 Dropping an IMS system from a generic resource group

You can use the /STOP VGRS command to drop an IMS out of a generic resource group. Similarly, you can use the /START VGRS command to bring that IMS back into the generic resource group.

13.3.2 Resetting terminal status

You can reset terminal status (and therefore terminate affinities) using one or more of the following:

- An authorized command
- The logoff exit routine (DFSLGFX0)
- The signoff exit routine (DFSSGFX0)
- The RCVYFP, RCVYSTSN, and RCVYCONV parameters
- The NONE status recovery mode

The logoff exit routine (DFSLGFX0) resets a terminal's status during a logoff. Similarly, you can use the signoff exit routine (DFSSGFX0) to reset a user's status during a signoff. Resetting terminal or user status enables IMS to terminate an affinity with the terminal, enabling the terminal to log on again and participate in session balancing.

13.3.3 Logging on after IMS failure with affinity

Suppose you are operating in a shared-queues environment, and your terminal is logged onto an IMS in a generic resource group. If that IMS fails, the affinity between that IMS and your terminal might persist. Consequently, you can have output messages waiting for you on that terminal. To obtain your messages, you can log onto another IMS within the generic resource group, provided that all of the following conditions exist:

- You log on using the APPLID name of another IMS within the generic resource group.
- Your output messages are not locked on the failed IMS, and those messages are not responses to either an IMS conversation or a Fast Path response-mode transaction.
- If you have a resource structure and Resource Manager, the status recovery mode is not LOCAL and there is not end-user significant status on the failed IMS (no RM affinity).

13.3.4 Controlling generic resource member selection

You can control VTAM's selection of a generic resource member for a terminal when the terminal logs on. To control the selection process, use either the VTAM generic resource resolution exit routine (ISTEXCGR) or the OS/390 Workload Manager. Choosing between these facilities depends on the needs of your application program and on your installation requirements. The criteria that VTAM uses to select a generic resource member are:

1. Existing affinity
2. VTAM generic resource resolution exit routine (ISTEXCGR)
3. OS/390 Workload Manager
4. Current session counts

13.3.5 Bypassing affinity management during the IMS ESTAE process

If you use IMS to manage generic resource affinities, you can control whether or not IMS uses or bypasses the VTAM generic resource logic in the IMS ESTAE exit. To use the IMS ESTAE process, indicate GRESTAE=Y in the DFSDCxxx PROCLIB member. GRESTAE=Y indicates that IMS should follow the existing ESTAE logic to delete affinity for nodes where no status remains before closing the ACF/VTAM ACB. GRESTAE=N indicates that IMS should close the ACF/VTAM ACB immediately to expedite IMS termination, and leave affinity for all nodes set. IMS VTAM generic resource logic in the IMS ESTAE exit attempts to delete generic resource affinity if no terminal status remains. This action requires a serial and synchronous VTAM CLSDST loop for all active terminals in IMS. Under normal conditions, the amount of time that this action requires is not significant. However, each CLSDST issued by the IMS ESTAE can incur the maximum defined VTAM I/O timeout time, leading to a significant delay in the final termination and subsequent restart of IMS.

Table 7 summarizes how generic resource affinities are managed based on whether affinities are VTAM or IMS managed and the GRESTAE options indicated in the DFSDCxxx PROCLIB member.

Table 13-1 GRAFFIN and GRESTAE options

Affinity manager (GRAFFIN=)	GRESTAE option	Generic resource affinities management
IMS	Y	IMS manages generic resource affinities, including during ESTAE processing. IMS resets affinity during the ESTAE process that occurs during IMS failure.

Affinity manager (GRAFFIN=)	GRESTAE option	Generic resource affinities management
IMS	N	IMS manages generic resource affinities, except during ESTAE processing. IMS does not reset affinity during the ESTAE process that occurs during IMS failure.
VTAM	Y or N	VTAM manages non-ISC generic resource affinities. IMS does not reset ISC affinity because (at the very least) the failing (parallel) session will not be properly quiesced.

14

IMS connectivity recovery restart

The question of how network recovery, as it applies to applications running many different environments, and connected to the OS/390 host via many different methods, is a complex one.

We discuss connectivity recovery for these topics:

- Recovery with VTAM Generic Resources
- Recovery when ISC connections are utilized
- Recovery in the Virtual IP addressing environment
- Sysplex Distributor backup and recovery
- IMS Connect load balancing and failover

14.1 Recovery with VTAM Generic Resources

Use of generic resources increases the need to manage the association of LUs and VTAM APPLIDs. An LU may now have affinity to a specific resource in the generic group which must be managed to ensure that no important information is lost. Depending on how the VTAM Generic Resources is defined, either IMS or VTAM may manage this affinity.

Affinities that should persist through a VTAM failure include LU 6.1 sessions, LU 6.2 sessions with SYNCPT synchronization support, and multi-node persistent sessions.

14.2 Affinity scenarios

The affinity scenarios should answer most of the “what happens if...?” questions about affinities. The scenarios discussed are:

- Local VTAM failure
- Logoff and signoff
- OS/390 or CEC failure
- IMS failure
- IMS cold-type start

14.2.1 Local VTAM failure

If VTAM is managing affinities (GRAFFIN=VTAM), and one of the VTAMs in a generic resource group abends, then another VTAM cleans up the affinities to the IMS associated with the VTAM failure for each terminal other than ISC links. Terminals that were in session though that VTAM to IMS at time of failure could logon again to the generic resource name and be routed to any available IMS system in the generic resource group.

If IMS is managing affinities (GRAFFIN=IMS), affinities remain until VTAM is restarted and a /START DC performed. Until such time, logons to IMS are rejected regardless of significant status in IMS. Once the /START DC is performed, IMS is able to accept logons, and all terminals having an affinity to that IMS are able to connect to IMS.

If VTAM is managing sessions, and one of the VTAMs in a generic resource group abends, then another VTAM cleans up the affinities to the IMS associated with the VTAM failure for each terminal other than ISC links. Terminals that were in session though that VTAM to IMS at time of failure could logon again to the generic resource name and be routed to any available IMS system in the generic resource group.

Table 14-1 shows the results of session termination that was caused by a VTAM failure.

Table 14-1 Table of affinity status after a VTAM failure

Types of Terminals	GRAFFIN=IMS	GRAFFIN=IMS	GRAFFIN=VTAM
	GRESTAE=Y	GRESTAE=N	
ISC Links / LU61	Affinity NOT deleted	Affinity NOT deleted	Affinity NOT deleted
LU0 Financial Terminals	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted
Static Terminals (not LU0 or Financial)	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted

Types of Terminals	GRAFFIN=IMS	GRAFFIN=IMS	GRAFFIN=VTAM
	GRESTAE=Y	GRESTAE=N	
ETO Terminals (not LU0 or Financial)	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted

14.2.2 Logoff and signoff

Terminals that logoff or signoff have their affinities deleted, with the exception of LU0 terminals and FINANCIAL terminals. These terminals do not have their affinity reset if GRAFFIN=IMS. The reason for this is so that the STSNs are not synchronized between each IMS system, and if the terminal attempted to logon to another IMS system, the STSN numbers would not correlate, and resynchronization using STSN would not be successful.

The logoff and signoff exits can be used to alter the affinities of terminals. These exit changes are not required if VTAM is chosen to manage affinities.

Table 8 shows a summary of affinity resolution at logoff/signoff time.

Table 14-2 Table of affinity status after session termination

Types of Terminals	GRAFFIN=IMS	GRAFFIN=IMS	GRAFFIN=VTAM
	GRESTAE=Y	GRESTAE=N	
ISC Links / LU61	Affinity NOT deleted	Affinity NOT deleted	Affinity NOT deleted
LU0 Financial Terminals	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted
Static Terminals (not LU0 or Financial)	Affinity deleted if no significant status	Affinity deleted if no significant status	Affinity deleted
ETO Terminals (not LU0 or Financial)	Affinity deleted	Affinity deleted	Affinity deleted

14.2.3 OS/390 or CEC failure

All the IMSs on different OS/390 images, and all other VTAMs, are still active in the sysplex. Any terminals with an affinity to the IMS on the failed system are not able to logon or connect to any IMS system using the generic resource name until the IMS system is available. Terminals will be able to logon to another IMS if they use the IMS APPLID, and not the generic resource name.

In the case of an OS/390 failure, the IMS ESTAE is not driven, and can be seen to have a similar result as if IMS is managing affinities, and the ESTAE exit is not used to resolve affinities during abend time.

If VTAM is managing affinities, then the cleanup is similar to that of a VTAM failure. If there is another VTAM, then the affinities are removed by another VTAM, but the ISC affinities will still remain. Once IMS and VTAM are restarted, IMS is available for session establishment, but until that time, all terminals other than the ISC terminal can logon to any other IMS in the same generic resource group. Table 14-3 summarizes activity when the IMS ESTAE cannot be invoked as is the case of a OS/390 or CEC failure.

Table 14-3 Table of affinity status after any failure when IMS cannot run ESTAE

Types of Terminals	GRAFFIN=IMS	GRAFFIN=IMS	GRAFFIN=VTAM
	GRESTAE=Y	GRESTAE=N	
ISC Links / LU61	Affinity NOT deleted	Affinity NOT deleted	Affinity NOT deleted
LU0 Financial Terminals	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted
Static Terminals (not LU0 or Financial)	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted
ETO Terminals (not LU0 or Financial)	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted

14.2.4 IMS failure

If IMS abends, then affinity processing depends on the VTAM Generic Resources related parameters in IMS. If GRAFFIN=IMS, then the affinity processing is determined at ESTAE processing time. If GRESTAE=Y, then all affinities are deleted for terminals without a significant status or a persistent session. Affinities will remain for terminals with a significant status, all LU0 and FINANCIAL terminals and ISC sessions. ETO terminals still have the significant status in the terminal control block until /ERE time, and therefore their affinities are not resolved until session establishment to the failed IMS system.

Table 14-4 shows the results of session termination that was caused by a failure in IMS when the IMS ESTAE processing was driven.

Table 14-4 Table of affinities status after IMS failure when ESTAE was driven

Types of Terminals	GRAFFIN=IMS	GRAFFIN=IMS	GRAFFIN=VTAM
	GRESTAE=Y	GRESTAE=N	
ISC Links / LU61	Affinity NOT deleted	Affinity NOT deleted	Affinity NOT deleted
LU0 Financial Terminals	Affinity NOT deleted	Affinity NOT deleted	Affinity deleted
Static Terminals (not LU0 or Financial)	Affinity deleted if no significant status	Affinity NOT deleted	Affinity deleted
ETO Terminals (not LU0 or Financial)	Affinity deleted if no significant status	Affinity NOT deleted	Affinity deleted

14.2.5 IMS cold-type start

If IMS is cold-type started, that is either COLD or COLDCOMM, then all affinities are deleted during restart.

It is advisable to shutdown with the LEAVEGR command, to remove affinities at IMS shutdown, if you are planning to start IMS with a cold start. This ensures that all terminals with an affinity to that IMS are able to establish session with any other IMS system in the same generic resource group while IMS is unavailable.

14.2.6 VTAM Persistent Sessions

VTAM persistent session support was introduced in VTAM V3R4. This capability allowed sessions to survive application failures and then be restored as soon as the application was

restarted. The value of this was that it eliminated the overhead of session termination and re-establishment. The support, however, as initially implemented in VTAM V3R4, was limited to application failures/restarts on the same CEC (single node).

Persistent session support was further enhanced in VTAM V4R4 to provide nondestructive session recovery from any failure including network, application, VTAM, MVS or CEC in a Parallel Sysplex environment. This enhancement allowed the failed application (IMS) to be restarted on a different VTAM system in the sysplex (multi-node support). As a result of this added capability, the terminology associated with persistent sessions was expanded to include: Single node persistent sessions (SNPS), and Multi-node persistent sessions (MNPS).

Although RNR support in IMS requires VTAM persistent session capability, the installation can select the use of either single or multi-node persistent sessions.

Note that RNR does not apply to APPC. The support for APPC persistent sessions is provided by APPC/MVS.

Rapid Network Reconnect (RNR) support in IMS V7 implements VTAM persistent session (single node and multi-node) support in IMS. This function provides the ability for sessions to be maintained across a failure (for example, IMS) and automatically reconnected when the failed component has been restarted.

The DFSDCxxx IMS option RNR=NRNR | ARNR implements VTAM persistent session support where NRNR indicates no reconnect and ARNR specifies automatic session reconnect. There are two types of VTAM persistent sessions:

- VTAM Single Node Persistent Session (SNPS)
  - Reconnect must be on the same CEC as the original IMS
  - Supports only application (IMS) failure and reconnect
- VTAM Multiple Node Persistent Session (MNPS)
  - Reconnect may be on other CEC in the sysplex
  - Support failures and reconnects, including IMS, OS/390, VTAM, CEC failures

Persistent sessions support applies to all VTAM terminals except MSC links. Persistent session support for APPC is provided by APPC/MVS. Figure 14-1 provides a brief overview of what occurs when SNPS support is invoked.

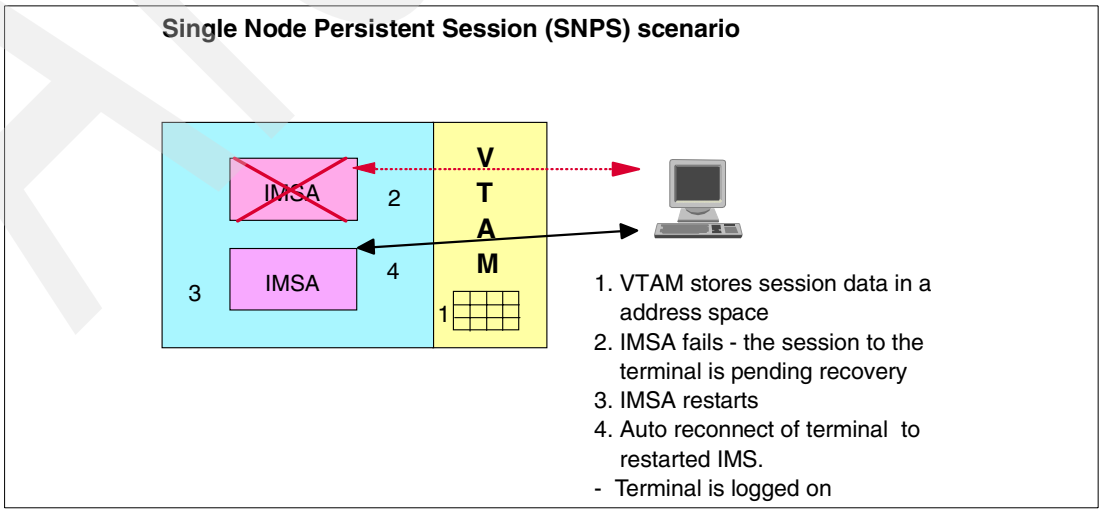

Figure 14-1 Operation in SNPS environments

When a persistence-enabled application program (for example, IMS with RNR=ARNR specified) fails, VTAM closes the ACB on IMS's behalf but retains the LU-LU sessions, saves the allocated resources and control blocks in a data space, and shields the network from knowledge of the application program failure.

VTAM stores the incoming data so that the network views the session as active but not currently responding. When the failed IMS restarts, VTAM reconnects the sessions. If the PSTIMER option is not specified, the session can remain pending recovery for an indefinite period. In this case, a VTAM `VARY NET INACT` command can be issued.

Figure 14-2 presents MultiNode Persistent Session operation. In an MNPS environment, session data is stored by VTAM in the MNPS structure of the Coupling Facility.

In this example CECA fails. When VTAM fails with CECA, the VTAM running in CECB in the same sysplex detects the error and IMSA is marked pending recovery. VTAM operating in CECB, connected to the same MNPS structure in the CF, starts a timer based on the DFSDCxxx member PSTIMER value. If the PSTIMER expires before IMSA recovery is successful, VTAM will perform cleanup of the failed IMS's session info in the CF structure. Otherwise, if IMSA is restarted on CEC2 either by operator intervention or by the Automatic Restart Manager (ARM), then IMSA's sessions are restored using the information from the MNPS structure in the CF.

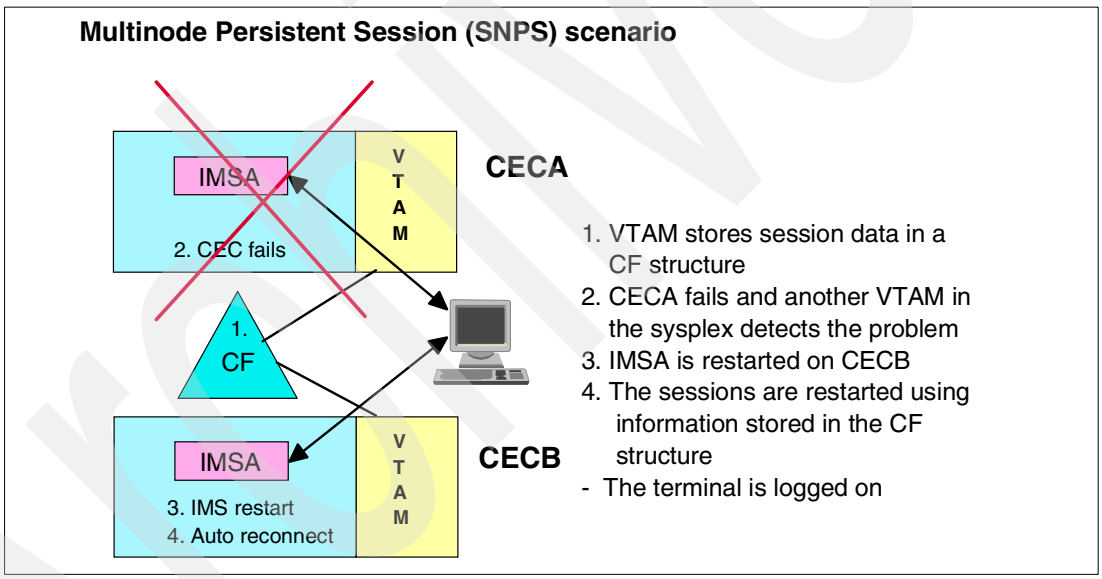

Figure 14-2 MNPS operation

Now let's examine network recovery using VTAM Generic Resources alone and then VTAM Generic Resources and RNR together.

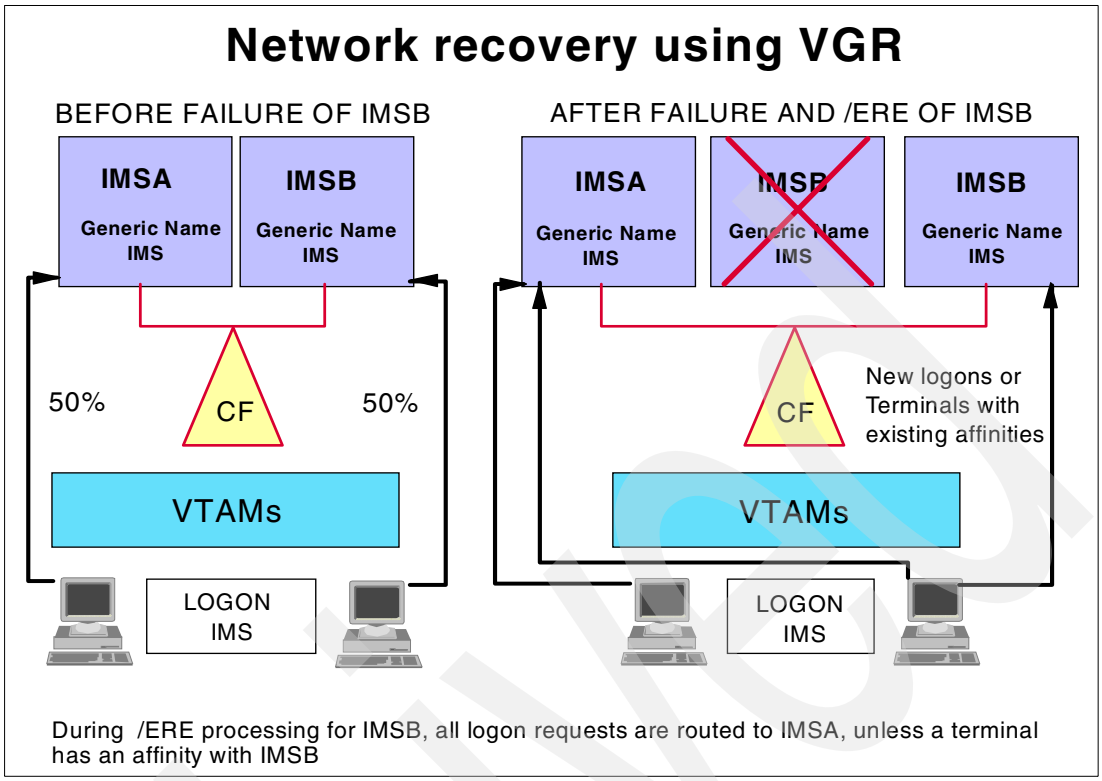

Figure 14-3 Network recovery using VTAM Generic Resources

As presented in Figure 14-3 if IMSB fails, with VTAM Generic Resources active without RNR, a remote user can issue a logon request but appropriate affinity and terminal status management is performed based on the GRAFFIN selection.

VTAM Generic Resources answers the need for load balancing, application availability, capacity growth, and allows end users to immediately reconnect to another IMS after a failure. But RNR is required to ensure persistence is established. Figure 14-4 presents network recovery operation when VTAM Generic Resources is used with RNR.

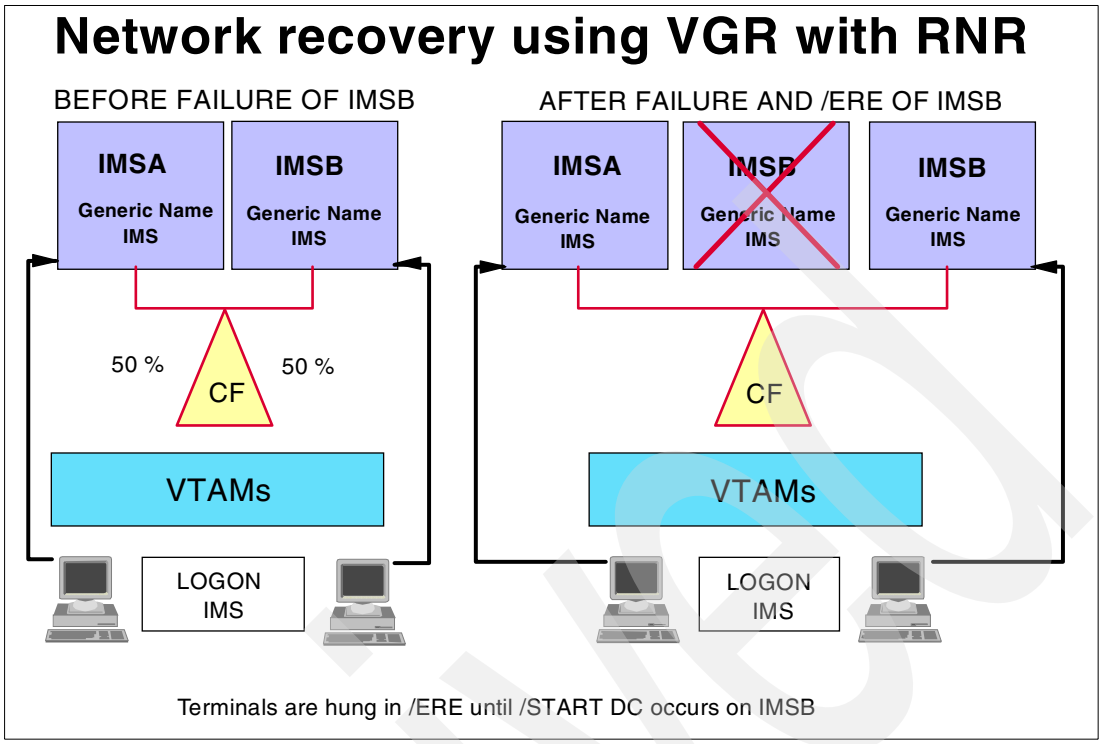

Figure 14-4 IMS terminal recovery using VTAM Generic Resources with RNR

With RNR active and automatic reconnection selected:

- During the /STARTDC after IMSB fails, session reconnect is established.
- Remote users must wait for the reconnection of an established session.
  - VTAM Generic Resources is not invoked to establish a new session with another IMS.

Therefore, if you are running two or more IMS systems in a Parallel Sysplex, with VTAM Generic Resources and RNR enabled, then in an IMS failure scenario, the user session will be suspended until the failed IMS is restarted. Care should be taken to set the correct level of RNR and VTAM Generic Resources settings to deliver the support for your installation.

14.3 ISC recovery

Like SLUTYPEP, the ISC protocol provides for session resynchronization between IMS and its partner subsystem, following session and/or system failures. The VTAM STSN command and protocols are used for this purpose. Unlike SLUTYPEP, ISC session resynchronization is not required every time a session is established between IMS and its ISC partner.

An ISC session can be normally terminated through an IMS command such that session resynchronization is not required when restarted. When sessions are terminated using the /QUIESCE command or when the QUIESCE option is used with the shut down checkpoint command (CHE FREEZE QUIESCE), a session, or sessions are terminated without any sequence numbers in doubt, and resynchronization is not required upon session restart.

Another difference between SLUTYPEP and ISC is that ISC parallel sessions can be cold started without a cold start of IMS. The cold start of an ISC session simply means that sequence numbers required for session restart have been discarded. Keep in mind that the cold start of an ISC session might mean that messages between systems can be duplicated;

no messages will be lost. The messages queued to an ISC session that is cold started are not discarded.

If ISC resynchronization is important, one should ensure that ISC partners always log back on to the same IMS system within the IMSplex following a session or IMS failure whether shared queues are used or not. This can be guaranteed by using specific VTAM APPLIDS in all system definitions between ISC partners or whenever a session request is made in an ETO environment.

14.4 Recovery in the Virtual IP addressing environment

Virtual IP addressing (VIPA) allows for the definition of a virtual IP address that does not correspond to any physical interface but rather is associated with a stack of addresses. There are considerations related to recovery that are unique to different VIPA implementations.

14.4.1 Static VIPA and recovery

Static VIPA was the first VIPA implementation. It eliminates an application's dependence on a particular network interface (IP address) being available. Figure 14-5 shows the static VIPA.

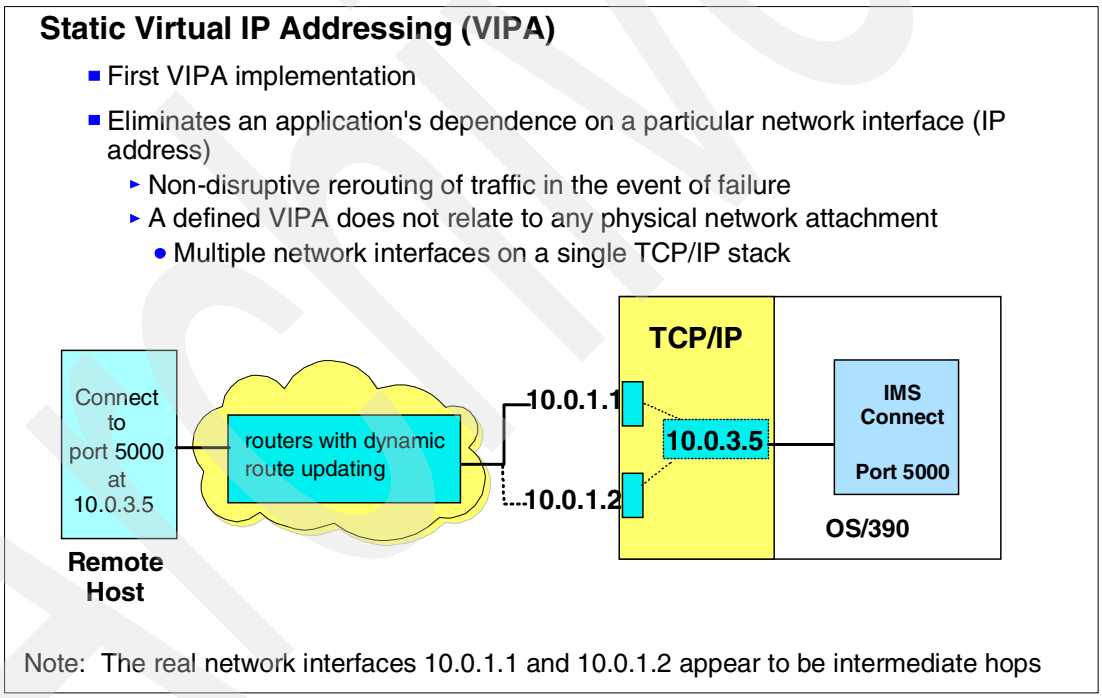

Figure 14-5 Static VIPA

For example, the use of Static VIPA allows a remote host to access IMS Connect at port 5000 and IP address 10.0.3.35 without knowing if either network interface 10.0.1.1 or 10.0.1.2 are available.

Only in the event of a physical interface failure associated with 10.0.1.1 the traffic will be routed to 10.0.1.2. This rerouting will be done non-disruptively without the client application receiving any notification of a connection failure. Static VIPA only addresses non-disruptive fault tolerance in the case of a specific network interface failure on a stack. If the stack itself fails, VIPA fails. To move the VIPA address to a different stack, a manual vary obey command must be issued.

14.4.2 Dynamic VIPA and failover recovery

If a stack or its underlying OS/390 or z/OS fails, all other stacks in the sysplex are informed of the failure. The VIPA address is automatically moved to the backup stack which receives information regarding the connections from the original stack. All new connection requests to the VIPA address are processed by the backup which becomes the new active. Instances of the server applications, for example, IMS Connect systems, listening on the same ports should be automatically started if they are not already active on the backup.

Additionally, the network routers are informed about the change. From a client application perspective, a connection failure is received when the primary stack fails but the client can immediately resubmit a new connection request which will be processed by the backup (new active) stack.

VIPA and network failure recoveries

In TCP/IP networks a primary area of concern is the outage of an IP network interface, more commonly known as an IP address. This is the network access point to the TCP/IP stack. By default, therefore, the failure of an interface prevents access to/from applications using that IP address. When the interface that fails is that of a client TCP/IP stack, the impact is the isolation of the client application from the network. On the other hand, if the interface that fails is that of a server then the impact is greater, because it affects access to all server applications on that stack.

The implementation of VIPA in the mainframe environment has evolved over several releases of OS/390 and can address the failure of a specific interface on a single TCP/IP stack (Static VIPA), the failure of the entire stack and all its interfaces (Dynamic VIPA takeover/takeback) or the failure of the server application on a specific host (application-initiated Dynamic VIPA).

Dynamic VIPA takeover

Beginning with OS/390 V2R8 the VIPA concept was extended to be more dynamic and to support recovery of a failed TCP/IP stack. Dynamic VIPA Takeover was introduced to automate the movement of the VIPA to a surviving backup stack. The use of this capability presumes that server application instances, IMS Connect instances, exist on the backup stack and can serve the clients formerly connected to the failed stack. In the example shown in Figure 14-6, the Dynamic VIPA IP address 10.0.3.5 is defined as having a home stack on TCPIPA and a backup on TCPIPB.

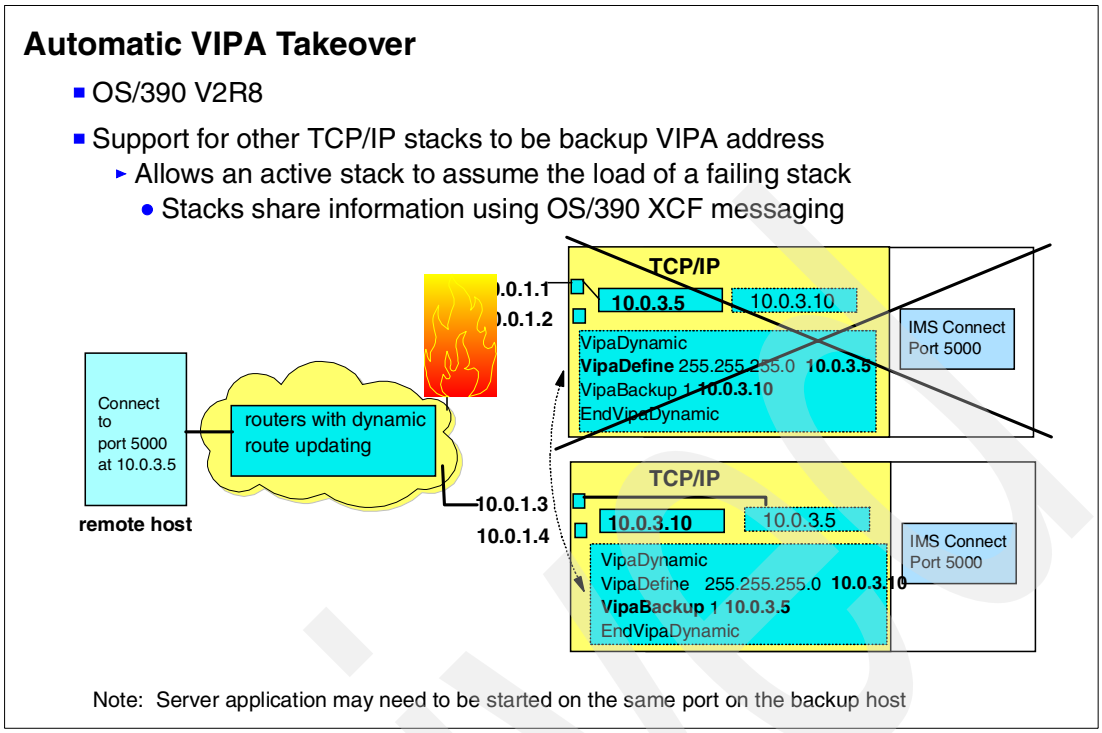

Figure 14-6 Dynamic VIPA failover support

Likewise, TCPIPB is defined as the primary for 10.0.3.10 and TCPIPA as the backup. Both stacks share information regarding the Dynamic VIPAs through the use of XCF messaging services. Each TCP/IP stack, therefore, is aware of all the Dynamic VIPA addresses and the associated primary and backup order.

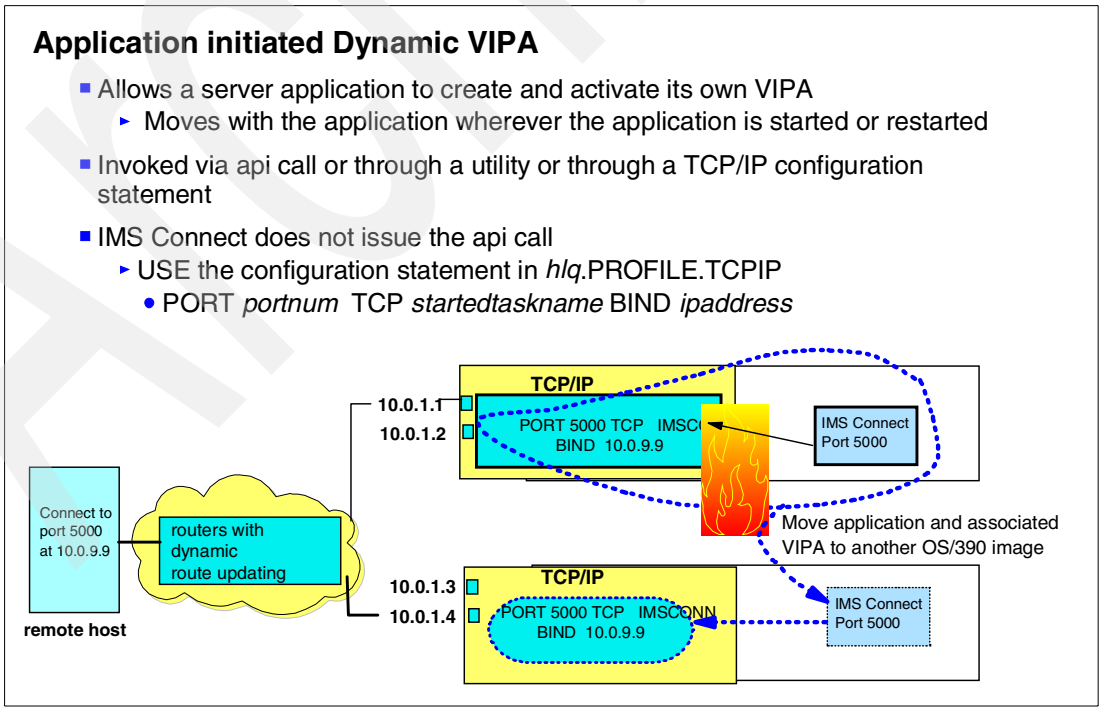

Figure 14-7 Application initiated dynamic VIPA

Dynamic VIPA takeback

The corollary to VIPA Takeover support is VIPA Takeback which is the ability for a reactivated primary or home stack to re-establish ownership of the VIPA address. Depending on the release of OS/390 the behavior of this support may vary.

In OS/390 V2R8, the Dynamic VIPA cannot be moved back to the primary until all active connections to that address on the backup are terminated. This means that the Dynamic VIPA may stay on the backup stack far beyond the reactivation of the primary.

OS/390 V2R10 lifts this restriction. By default, once the primary is reactivated the VIPA is reacquired and new connection requests are immediately routed to it. Connections that are still active on the backup remain there and the backup sends information about these still active connections to the primary. Since subsequent data packets associated with all connections using the VIPA flow to the primary, the primary has the responsibility to detect where the data should be sent. If the data is associated with a connection owned by the primary then it is sent to the appropriate server application on its stack. If, however, the data is associated with an old connection that is still associated with the backup then the primary sends the data to the backup stack.

14.4.3 ARM and application initiated dynamic VIPA support

Figure 14-8 shows an example how automatic restart manager (ARM) and dynamic VIPA could be used for IMS Connect to automatically move IMS Connect to another system.

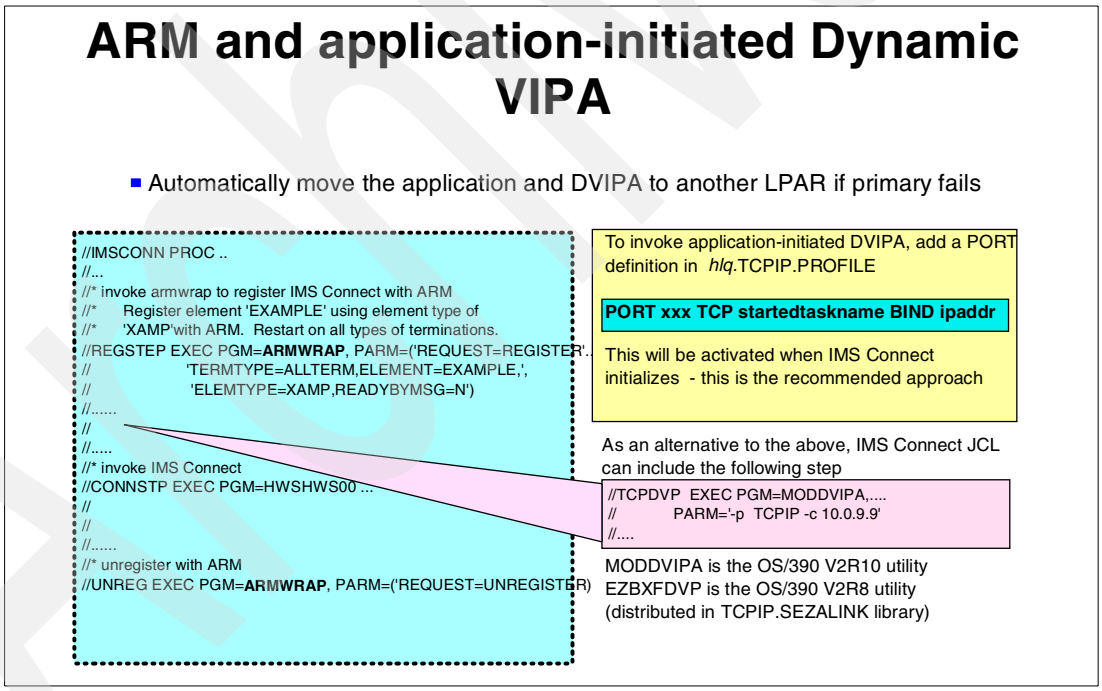

Figure 14-8 ARM and application initiated dynamic VIPA

14.5 Sysplex Distributor

A single IP address can be provided for a cluster of hosts with this sysplex-wide VIPA function available with OS/390 Version 2 Release 10.

To support high availability and to prevent a single point of failure, the Network Dispatcher function also supports a backup capability and redundancy. Two Network Dispatchers can be configured with connectivity to each other, the same clients and cluster of servers. A heartbeat function exists that allows each to detect the failure of the other along with a synchronization mechanism for the applicable databases (connectivity tables, reach ability tables, and so on) along with logic to elect the active Network Dispatcher and a mechanism to perform fast IP takeover to switch from the active to the standby.

To obtain more information on the Network Dispatcher and the WebSphere Edge Server, refer to this Web site:

http://www-4.ibm.com/software/network/dispatcher/

14.5.1 Sysplex Distributor recovery

For example let's assume we have five hosts within a defined sysplex. In the eventuality that host 1 fails, all participating stacks in the sysplex are notified of the failure.

Dynamic VIPA takeover is activated and host 2 defines itself as the new distributing stack. All the participating stacks on host 3 to 5 send their connection information regarding the DVIPA to host 2 to facilitate the rebuilding of the connection table. Existing connections between remote clients and IMS Connect systems on hosts 3 to 5 are not affected. Data flows are simply rerouted through host 2.

Once host 1 is reactivated, Dynamic VIPA takeback begins. This is again non-disruptive to existing connections. Host 2 sends its connection information to host 1 so that it can rebuild its table and resume its activities.

In this environment, the only time that remote clients see a connection failure is when the backend host running IMS Connect fails. This is because that host is the terminating point of the failure. Failures of other hosts or stacks including the distributing stack are transparent. When a connection failure is received, the remote client can immediately resend a new connection request. As long as an IMS Connect is active on a listening port, the request can be satisfied.

14.6 IMS Connect load balancing and failover

After a message destination has been resolved to a particular host system and IMS Connect, the next set of configuration decisions deals with connectivity between IMS Connect and the IMS systems in the sysplex. IMS Connect can be partnered with a single IMS or it can be given access to several as shown previously.

When provided with access to multiple IMS systems, code can be added to the IMS Connect user message exits to perform load balancing and failover. The code to do this would have to be user written, but an interface is provided to enable this capability. IMS Connect provides a *datastore table* that keeps track of the status, active or inactive, of all the IMS systems that are defined in the HWSCFG file. The table is updated as events occur. When a message reaches IMS Connect, an appropriate user message exit is invoked. All user message exits have access to the datastore table and can therefore take action based upon the information.

Refer to the *IMS Connect Guide and Reference*, SC27-0946, for more information on the datastore table and the user message exits.

A

IMS Parallel Sysplex migration task lists

This appendix provides sample task lists for implementing various IMS Parallel Sysplex features:

A.1 Migration task list for implementing data sharing

Table A-1 shows a sample task list for data sharing implementation and gives references to more information.

Table A-1 IMS data sharing migration checklist

Implementation Activities	References
Set the RECON to SHARECTL if operating IMS Version 6. This implements authorization processing in block level data sharing environments. There is no SHARECTL option in IMS Version 7 and 8 since it is mandatory.	1.2, “Ensure that SHARECTL is set in the RECON” on page 4
Register databases at SHARELVL(1) if not performed yet.	1.3, “Register databases at SHARELVL(1)” on page 4
Review and modify the parameters in the IMS procedures if necessary.	1.4, “Parameters to review in IMS procedures” on page 4
Since the IRLM is operational at SCOPE=LOCAL and locking and deadlock detection have changed, measure system performance for database access times and numbers of deadlocks.	1.5, “Comments on data sharing overhead” on page 10
Ensure that all IMS and batch subsystems that will be sharing data point to the same RECONs.	
Set VSAM data sets to Shareoptions(3 3). This is required before SHARELVL(2 or 3) and IRLM SCOPE=GLOBAL can be implemented.	1.6, “Defining VSAM share options” on page 13
Set VSAM data sets to DISP = SHR using DD cards or dynamic allocation. This is required before SHARELVL(2 or 3) and IRLM SCOPE=GLOBAL can be implemented.	1.7, “Set VSAM and OSAM data set dispositions to share” on page 13
Choose appropriate ACCESS = xx parameters in the Database macros. Set ACCESS = UP for databases to be shared.	1.8, “Adjust ACCESS=xx in the DATABASE macro” on page 13
Decide on which databases will not be shared and isolate them from the block level data sharing environment.	1.9, “Excluding a database from data sharing” on page 14
Examine the DBRC skeletal JCL currently in place.	1.10, “Examine DBRC skeletal JCL” on page 14
Examine the expected increases in size and use of the database buffer pools.	1.11, “Buffer pool use in a data sharing environment” on page 14
Calculate the size of the OSAM and VSAM cache and IRLM lock structures.	1.12, “Calculate the size of the OSAM/VSAM XI structures” on page 15
Define the CF structures in the XCF Administrative Data utility. Defining the structures will have no effect before they are specified to IMS or IRLM.	1.14, “Defining the OSAM, VSAM and IRLM CF structures” on page 18

Implementation Activities	References
Update DFSVSMxx with CFNAMES parameters. This causes the OSAM and VSAM structures to be built. Since data sharing is not yet performed yet, no buffer invalidataions will occur; however, the overhead or read-and-register operations will be added to the system.	1.15, “Update DFSVSMxx with CFNAMES parameters” on page 18
Register databases at SHARELVL(2) in phases. This invokes block locks, eliminates Fast Path lock management, and creates possible lock conflicts for key-sequenced data sets. This may be done in phases to limit the effects of the lock changes. With SHARELVL(2), sharing is at the block level but only within the scope of a single IRLM. IRLM uses its local control blocks rather than accessing lock structures in the CF.	1.16, “Registering databases at SHARELVL(2)” on page 19
Customize the IRLM procedure for n-way block level data sharing. This includes setting SCOPE = GLOBAL or NODISCON which causes the lock structure to be built when IRLM is initialized. Since no databases are registered yet at SHARELVL (3), only busy locks, data set reference locks, and command locks are placed in the lock structure. The overhead of this step will depend on the number of updates to KSDSs since this requires busy locks.	1.17, “Update IRLM start procedures for global data sharing” on page 19
Start the CFRM policies which define the OSAM and VSAM cache and IRLM lock structures.	1.18, “Starting the CFRM policy” on page 20
Register databases at SHARELVL(3) in phases. This causes locks to be placed in the lock structure. This may be done in phases to limit the effects of locking changes. Check compatibility of SHARELVL(3) with the following parameters: DFSMDA =, PROCOPT =, ACCESS = , and VSAM share options at (3 3).	1.19, “Registering databases at SHARELVL(3)” on page 21
Create a new IRLM on the target data sharing partner system.	1.20, “Cloning your IMS subsystems” on page 21
Clone a IMS subsystem to become a data sharing partner and cold start both subsystems.	1.21, “Starting the second IMS subsystem” on page 22
Start databases and AREAs with ACCESS UP via /START commands or use of TCO scripts.	1.21, “Starting the second IMS subsystem” on page 22
Test data sharing with sample application runs. This invokes sharing and may result in lock conflicts and buffer invalidations.	1.22, “Testing data sharing with simple applications” on page 23

A.2 Migration task list for implementing shared queues

Table A-2 shows a sample task list for shared queues implementation.

Table A-2 Shared queues implementation task list

Shared queues task list
Setup - Startup Procedures Update IM1A & IM2A started procedures in SYS1.xxx.PROCLIB
Create CQS procedures
BPECFG=BPECONFG Name of BPE configuration member in PROCLIB
Update program properties table
CQSINIT=01Q/02Q Suffix to CQS Initialization member in PROCLIB
Setup - Review parm member relationships

Shared queues task list
Update IMS Runtime Parameters for SQ
Create IMS shared queues PROCLIB member DFSSQ0nQ
Create CQS initialization PROCLIB members CQSIP01Q and CQSIP02Q
Create CQSn local PROCLIB members CQSSL01L and 2L
Create CQS global PROCLIB member CQSIP0nQ
CQS Started Procedures as specified in DFSSQ0nQ
Update the Sysplex Logger Policy CQSSG00G LOGNAME and LOGSTREAM NAME must agree
The CFRM Policy must be updated to define the new structures The names of these have been defined in the global parameters CQSSG00G
Define and allocate CQS checkpoint data sets and structure recovery data sets
Spill data sets
Staging data sets
Update the local members (CQSSL01L and CQSSL02L) regarding the previous step
Define and update global member CQSSGxxx
Structure Security
ARM policy update

A.3 Task list to set up a CSL environment

Table A-3 shows a sample task list to set up a CSL environment and gives references to more information.

Table A-3 CSL task list

CSL TASK LIST	REFERENCES
Basic rules to simplify the process	3.2.4 on page 57
Hardware and software currency	3.2.4 on page 57
Formatting the CFRM couple data set (CDS)	3.3.1 on page 58
Sample resource structure definition	3.3.2 on page 58
BPE task list	**References**
Update the program properties table (PPT)	3.4.2 on page 60
Update the BPE configuration parameter PROCLIB member	3.4.4 on page 61
Update the BPE user exit PROCLIB member	3.4.5 on page 62
SCI task list	**References**
Create SCI startup procedure	3.5.1 on page 64
Specify the execution parameters in the SCI startup procedure	3.5.2 on page 65

CSL TASK LIST	REFERENCES
Create the SCI initialization PROCLIB member CSLSIxxx	3.5.3 on page 65
Define profiles to control SCI registration Authorize connections to IMSplex for all SCI connectors	3.5.4 on page 66
Define SCI user exit routines	3.6 on page 67
OM task list	**References**
Create OM startup procedure	3.7.1 on page 68
Specify the execution parameters in the OM startup procedure	3.7.2 on page 68
Create OM initialization PROCLIB member CLSOIxxx	3.7.3 on page 69
Define OM command security	3.7.4 on page 70
Define OM user exit routines	3.8 on page 72
RM task list	**References**
Create RM startup procedure	3.10.1 on page 74
Specify the execution parameters in the RM startup procedure	3.10.2 on page 75
Create the RM initialization PROCLIB member CSLRIxxx	3.10.3 on page 75
Define RM user exits routines	3.11 on page 76
Authorize RMs to access RM structure	3.10.4 on page 76
IMS task list	**References**
Update the IMS execution parameters PROCLIB member DFSPBxxx	3.13.1 on page 77
Create the IMS CSL related parameters PROCLIB member DFSCGxxx	3.13.2 on page 78
Update the IMS DC execution parameters PROCLIB member DFSDCxxx	3.13.3 on page 79
Update the OPTION statement in PROCLIB member DFSVSMxx	3.13.4 on page 79
Choose IMSplex security issues for an IMS in an IMSplex	3.15 on page 80

A.4 Task list to set up an STM environment

Table A-4 shows a sample task list to set up an STM environment and gives references to more information.

Table A-4 STM task list

STM task list	References
Set the status recovery mode (SRM) in PROCLIB member DFSDCxxx	4.2.2 on page 86
Set the status recoverability (RCVYxxxx) in PROCLIB member DFSDCxxx	4.2.3 on page 86
Override SRMDEF and RCVYxxxx defaults	4.3 on page 87
Use of CFSIZER tool for resource structure definition	4.4.1 on page 89
Methods to calculate the resource number	4.6.1 on page 91
Methods to calculate the data element number	4.7.1 on page 92

STM task list	References
Calculate an initial resource structure	4.8 on page 92
CFRM policy definitions options	4.9.1 on page 93
Adjust the size of the resource structure	4.10 on page 93

A.5 Task list to set up a global online change environment

Table A-5 shows a task list to set up a global online change environment and gives references to more information.

Table A-5 Global online change task list

Global online change task list	References
Preparation for global online change	5.1.4 on page 101
Update the IMS global online change related parameters PROCLIB member DFSCGxxx	5.1.5 on page 101
Definition of OLCSTAT data set	5.3.1 on page 103
Initialize OLCSTAT data set using DFSUOLC procedure	5.4.2 on page 106
Execute the online change copy utility using OLCUTL procedure	5.5 on page 107
Migration to global online change	5.5.2 on page 108
Fallback to local online change	5.5.3 on page 108

A.6 Task list to set up a TSO SPOC environment

Table A-6 shows a task list to set up a SPOC environment and gives references to more information.

Table A-6 SPOC task list

SPOC task list	References
Set up TSO SPOC program	5.7 on page 109
Set up SPOC preferences	5.7.4 on page 111

A.7 Task list to set up a REXX user interface to OM environment

Table A-7 shows a task list to set up a REXX user AOP interface to OM environment and gives references to more information.

Table A-7 REXX SPOC task list

REXX SPOC task list	References
Allocation of REXX SPOC file to the TSO session	5.8.3 on page 114
Set up the REXX host command environment	5.10 on page 117

A.8 Task list to set up a ARLN environment

Table A-8 shows a sample task list to enable an ARLN environment and gives references to more information.

Table A-8 ARLN task list

ARLN task list	References
Choice how to enable ARLN environment	5.11 on page 118
Implement ARLN	5.12.1 on page 120
Specify ARLN in batch and utility regions	5.12.2 on page 120
Migrate and fallback from ARLN	5.13 on page 121
Use the IMS provided default exit	5.14.1 on page 122

A.9 Task list to set up a dynamic LE environment

Table A-9 shows a sample task list to set up an IMS Version 8 dynamic LE run time environment and gives references to more information.

Table A-9 LE dynamic run time task list

LE dynamic run time task list	References
Enable or disable the dynamic run time option in member DFSCGxxx	5.17.1 on page 125
Choose the DFSBXITA IMS exit options	5.19.2 on page 126
Set up LE run time options for IMS environments	5.19.3 on page 127

B

Sample DFSDDLT0 programs used to test data sharing

These sample BMPs use databases and segments that were available to us during the creation of this redbook, however you can replace the PSB= value, the names of the databases, and segments so they retrieve and replace real data using real test structures at your site.

DFSDDLT0 BMPs used to test data sharing

Two jobs, BMP DDLOCK1 and BMP DDLOCK2, are described here with examples, and a locking demonstration is given.

BMP DDLOCK1

The first job, called DDLOCK1, was run on OS/390 image SC53TS, under an IMS Version 8 system IM1A. Database DISTRICT is a VSO DEDB and CUSTDB is HDAM full function database. The sequence of calls created for this DFSDDLT0 execution are:

- GHU to segment DISTRICT in the database DISTDB with Key=002110.
- REPL the data in that particular segment.
- GHU and REPL to change the data again in the segment DISTRICT in database DISTDB with the same Key=002110.
- GHU to segment CUSTOMER in the database CUSTDB with Key=0021100001.
- REPL the data in that particular segment.
- GHU and REPL to change the data again in the segment CUSTOMER in the database CUSTDB with the same Key=0021100001.
- WTOR with the message 'ENTER NN. TO RESUME IM1A'. This ensures that DDLOCK1 waits until the operator takes action.
- After the WTOR is responded to a SYNC call is issued within the DFSDDLT0 control statement stream.

When the WTOR is issued, IRLM will have obtained locks that will be reflected in the lock structure of the Coupling Facility. The JCL and control statements follow in Example B-1.

Example: B-1 DFSDDLT0 BMP job DDLOCK1

```
//DDLOCK1  JOB CLASS=A,MSGCLASS=O,REGION=4096K,TIME=1440
/*JOBPARM SYSAFF=*
//*
//*DDLTO JOB TO VERIFY LOCKING
//*
//BMP    PROC  MBR=TEMPNAME,PSB=,IN=,OUT=,
//             OPT=N,SPIE=0,TEST=0,DIRCA=000,
//             PRLD=,STIMER=,CKPTID=,PARDLI=,
//             CPUTIME=,NBA=,OBA=,IMSID=IM1A,AGN=,
//             SSM=,PREINIT=,RGN=512K,SOUT=*,
//             ALTID=IM2A,APARM=
//*
//G      EXEC  PGM=DFSRRC00,REGION=&RGN,
//             PARM=(BMP,&MBR,&PSB,&IN,&OUT,
//             &OPT&SPIE&TEST&DIRCA,&PRLD,
//             &STIMER,&CKPTID,&PARDLI,&CPUTIME,
//             &NBA,&OBA,&IMSID,&AGN,&SSM,
//             &PREINIT,&ALTID,
//             '&APARM')
//STEPLIB  DD DSN=IMSPSA.IMS0.SDFSRESL,DISP=SHR
//         DD DSN=IMSPSA.IMS0.PGMLIB,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR                  <--- LE/370
//PROCLIB  DD DSN=IMSPSA.IM0A.PROCLIB,DISP=SHR
//SYSUDUMP DD  SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//             SPACE=(125,(2500,100),RLSE,,ROUND)
//             PEND
//*-------------------------------------------------
//STEP01   EXEC BMP,RGN=0M,
```

```
//         MBR=DFSDDLT0,PSB=FPSBPA,NBA=10
//G.PRINTDD  DD SYSOUT=*
//G.SYSIN    DD *
S11 1 1 1 1    DISTDB
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110XXXXXXXXXXXXXXXXXXXXXXXXXX
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110AAAAAAAAAAAAAAAAAAAAAAAAAA
S11 1 1 1 1    CUSTDB
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001XXXXXXXXXXXXXXXXXXXXXXXXX
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001AAAAAAAAAAAAAAAAAAAAAAAAA
WTOR ENTER NN. TO RESUME IM1A
L        SYNC
/*
//
```

BMP DDLOCK2

The second job, called DDLOCK2, would be run on IM2A, an IMS Version 7 system running on an OS/390 image SC47TS. The sequence of calls is:

- GHU to segment DISTRICT in the DISTDB with Key=002110
- REPL the data in that particular segment
- GHU and REPL to segment DISTRICT in the DISTDB with Key=002110 adding different data than the REPL call.
- GHU to segment CUSTOMER in Data Base CUSTDB with Key= 0021100001
- REPL the data in that particular segment.
- GHU and REPL to segment CUSTOMER in the CUST with Key=0021100001 adding different data than the REPL call.
- WTOR with the message 'ENTER NN TO RESUME IM2A'. This will ensure that DDLOCKB waits until the operator takes action. When WTOR is responded to this application ends with a SYNC call. The JCL and control statements for DDLOCKB follow in Example B-2.

Example: B-2 DFSDDLT0 BMP job DDLOCK2

```
//DDLOCK2  JOB CLASS=A,MSGCLASS=O,REGION=4096K,TIME=1440
/*JOBPARM SYSAFF=*
//*
//*DDLTO JOB TO VERIFY LOCKING
//*
//BMP    PROC  MBR=TEMPNAME,PSB=,IN=,OUT=,
//             OPT=N,SPIE=O,TEST=O,DIRCA=000,
//             PRLD=,STIMER=,CKPTID=,PARDLI=,
//             CPUTIME=,NBA=,OBA=,IMSID=IM2A,AGN=,
//             SSM=,PREINIT=,RGN=512K,SOUT=*,
//             ALTID=IM2A,APARM=
//*
//G      EXEC  PGM=DFSRRC00,REGION=&RGN,
//             PARM=(BMP,&MBR,&PSB,&IN,&OUT,
//             &OPT&SPIE&TEST&DIRCA,&PRLD,
```

```
//             &STIMER,&CKPTID,&PARDLI,&CPUTIME,
//             &NBA,&OBA,&IMSID,&AGN,&SSM,
//             &PREINIT,&ALTID,
//             '&APARM')
//STEPLIB  DD DSN=IMSPSA.IMS0.SDFSRESL.V7,DISP=SHR
//         DD DSN=IMSPSA.IMS0.PGMLIB,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR                    <--- LE/370
//PROCLIB  DD DSN=IMSPSA.IM2A.PROCLIB,DISP=SHR
//SYSUDUMP DD  SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//             SPACE=(125,(2500,100),RLSE,,ROUND)
//             PEND
//*-------------------------------------------------------------------*//
//STEP01   EXEC BMP,RGN=0M,
//         MBR=DFSDDLT0,PSB=FPSBPA,NBA=10
//G.PRINTDD  DD SYSOUT=*
//G.SYSIN    DD *
S11 1 1 1 1    DISTDB
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110YYYYYYYYYYYYYYYYYYYYYYYY
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110AAAAAAAAAAAAAAAAAAAAAAAA
S11 1 1 1 1    CUSTDB
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001YYYYYYYYYYYYYYYYYYYYYYYY
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001AAAAAAAAAAAAAAAAAAAAAAAA
WTOR ENTER NN. TO RESUME IM2A
L        SYNC
/*
//
```

If DDLOCK2 is run after DDLOCK1, it will wait for the locks held by DDLOCK1 since DDLOCK2 is accessing the same database record with update intent. Lets run through this exercise to present the changing status of the lock structure elements as updating programs are scheduled within IMS.

Locking demonstration

As commands are issued we will display the output of informational commands:

- Start DDLOCK1 on IMA1 via a /STA REGION DDLOCK1
  - DDLOCK1 acquires locks and issues a WTOR as displayed in Example B-3:

Example: B-3 WTOR from DDLOCK1

```
JOB16693 @2395 ENTER NN. TO RESUME IM1A
```

  - Do not reply to this WTOR at the moment.
- Start DDLOCK2 on IM2A via command /STA REGION DDLOCK2.
  - DDLOCK2 will request the same initial lock that DDLOCK1 requested because the database record is the same in the same database so will have to wait.
  - You should NOT see the WTOR from DDLOCK2 yet.

First issue a F IM1AIRLM,STATUS to see that locks are being held for the IM1A client.

Example: B-4 Display of IM1AIRLM status after DDLOCK1 placed in WTOR wait

```
F IM1AIRLM,STATUS
DXR101I IR0A001 STATUS SCOPE=NoDISC 565
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS       UNITS      HELD    WAITING    RET_LKS
        IM1A     UP               3        13          0          0
    DXR101I End of display
```

Then issue command F IM2AIRLM,STATUS on OS/390 image SC47TS.

Example: B-5 Output of F IM2AIRLM,STATUS command

```
F IM2AIRLM,STATUS
DXR101I IR0A002 STATUS SCOPE=NoDISC 785
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS       UNITS      HELD    WAITING    RET_LKS
        IM2A     UP               3         7          1          0
    DXR101I End of display
```

Notice that the waiting count is 1 for IM2A, since BMP job DDLOCK2 is waiting for DDLOCK1 running on IM1A to complete after the WTOR is responded to.

- Respond to the WTOR from DDLOCK1 and note that it completes and that a new WTOR for DDLOCK2 appears, as shown in Example B-6.

Example: B-6 WTOR outstanding response from DDLOCK2

```
JOB16697 @2402 ENTER NN. TO RESUME IM2A
```

- Don't reply to this message yet. Instead issue another F IM2AIRLM,STATUS.

Example: B-7 Output of F IM2AIRLM,STATUS

```
DXR101I IR0A002 STATUS SCOPE=NoDISC 816
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS       UNITS      HELD    WAITING    RET_LKS
        IM2A     UP               3        14          0          0
    DXR101I End of display
```

- Note that IM2A as an user of IM2AIRLM services, is now not waiting on any locks but does hold 14 as compared to 7 before it was allowed to process up to this point.
- Response to the WTOR from DDLOCK2 and allow it to complete.

Example: B-8 Status of IM2AIRLM after DDLOCK2 completes.

```
DXR101I IR0A002 STATUS SCOPE=NoDISC 888
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS       UNITS      HELD    WAITING    RET_LKS
        IM2A     UP               2         7          0          0
    DXR101I End of display
```

This demonstrates that both of the updating BMPs running on separate IMS images have been able to update common resources with integrity.

DFSDDLT0 BMPs to demonstrate deadlocks

The setup is similar to the two DFSDDLT0 jobs above, but the intent here is to create a deadlock so that we can examine the resulting recovery activity.

Deadlocks with Fast Path databases included in the PSB

BMP DEDLOCK1

DEDLOCK1 would be run on IM1A executing in an initiator in OS/390 system SC53TS. The sequence of calls is:

- GHU to segment DISTRICT in the DISTDB with Key=002110.
- REPL the data in that particular segment.
- A repeat of the GHU and the REPL call (with different data but still running under control of the same application).
- WTOR with the message 'ENTER NN. TO RESUME IM1A'. This ensures that DEDLOCK1 waits until the operator takes action.
- When the WTOR is responded to a GHU to segment CUSTOMER in Data Base CUSTDB with Key= 00211100001 is issued.
- REPL the data in that particular segment.
- A repeat of the GHU and the REPL call (with different data but still running under control of the same application).
- A SYNC call to terminate the BMP.

The JCL and control statements for DEDLOCK1 follow in Example B-9

Example: B-9 JCL and DFSDDLT0 control statements for BMP DEDLOCK1

```
//DEDLOCK1 JOB CLASS=A,MSGCLASS=O,REGION=4096K,TIME=1440
/*JOBPARM SYSAFF=*
//*
//*DDLTO JOB TO VERIFY LOCKING
//*
//BMP    PROC  MBR=TEMPNAME,PSB=,IN=,OUT=,
//             OPT=N,SPIE=0,TEST=0,DIRCA=000,
//             PRLD=,STIMER=,CKPTID=,PARDLI=,
//             CPUTIME=,NBA=,OBA=,IMSID=IM1A,AGN=,
//             SSM=,PREINIT=,RGN=512K,SOUT=*,
//             ALTID=IM1A,APARM=
//*
//G      EXEC  PGM=DFSRRC00,REGION=&RGN,
//             PARM=(BMP,&MBR,&PSB,&IN,&OUT,
//             &OPT&SPIE&TEST&DIRCA,&PRLD,
//             &STIMER,&CKPTID,&PARDLI,&CPUTIME,
//             &NBA,&OBA,&IMSID,&AGN,&SSM,
//             &PREINIT,&ALTID,
//             '&APARM')
//STEPLIB  DD DSN=IMSPSA.IMS0.SDFSRESL,DISP=SHR
//         DD DSN=IMSPSA.IMS0.PGMLIB,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR                  <--- LE/370
//PROCLIB  DD DSN=IMSPSA.IM0A.PROCLIB,DISP=SHR
//SYSUDUMP DD  SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//             SPACE=(125,(2500,100),RLSE,,ROUND)
//             PEND
//STEP01   EXEC BMP,RGN=0M,
```

```
//         MBR=DFSDDLT0,PSB=FPSBPA,NBA=10
//G.PRINTDD  DD SYSOUT=*
//G.SYSIN    DD *
S11 1 1 1 1    DISTDB
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110XXXXXXXXXXXXXXXXXXXXXXXXXX
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110AAAAAAAAAAAAAAAAAAAAAAAAAA
WTOR ENTER NN. TO RESUME IM1A
S11 1 1 1 1    CUSTDB
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001XXXXXXXXXXXXXXXXXXXXXXXXX
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001AAAAAAAAAAAAAAAAAAAAAAAAA
L        SYNC
/*
//
```

BMP DEDLOCK2

DEDLOCK2 would be run on IM2A executing in an initiator in the OS/390 image SC47TS. The sequence of calls is:

- GHU to segment CUSTOMER in Data Base CUSTDB with Key= 0021100001.
- REPL the data in that particular segment.
- A repeat of the GHU and the REPL call (with different data but still running under control of the same application).
- WTOR with the message ‘ENTER NN. TO RESUME IM2A”. This ensures that DEDLOCK2 waits until the operator takes action.
- When the WTOR is responded to, a GHU to segment DISTRICT in the DISTDB with Key=0021101 is used.
- REPL the data in that particular segment.
- A repeat of the GHU and the REPL call (with different data but still running under control of the same application).
- A SYNC call to terminate the BMP.

The JCL and control statements for job DEDLOCK2 follow in Example B-10.

Example: B-10 JCL and DFSDDLT0 control statements for BMP DEDLOCK2

```
//DEDLOCK2  JOB CLASS=A,MSGCLASS=O,REGION=4096K,TIME=1440
/*JOBPARM SYSAFF=*
//*
//*DDLTO JOB TO VERIFY LOCKING
//*
//BMP    PROC  MBR=TEMPNAME,PSB=,IN=,OUT=,
//             OPT=N,SPIE=O,TEST=O,DIRCA=000,
//             PRLD=,STIMER=,CKPTID=,PARDLI=,
//             CPUTIME=,NBA=,OBA=,IMSID=IM2A,AGN=,
//             SSM=,PREINIT=,RGN=512K,SOUT=*,
//             ALTID=IM2A,APARM=
//*
//G      EXEC  PGM=DFSRRC00,REGION=&RGN,
```

```
//              PARM=(BMP,&MBR,&PSB,&IN,&OUT,
//              &OPT&SPIE&TEST&DIRCA,&PRLD,
//              &STIMER,&CKPTID,&PARDLI,&CPUTIME,
//              &NBA,&OBA,&IMSID,&AGN,&SSM,
//              &PREINIT,&ALTID,
//              '&APARM')
//STEPLIB  DD DSN=IMSPSA.IMS0.SDFSRESL.V7,DISP=SHR
//         DD DSN=IMSPSA.IMS0.PGMLIB,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR                    <--- LE/370
//PROCLIB  DD DSN=IMSPSA.IM2A.PROCLIB,DISP=SHR
//SYSUDUMP DD  SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//             SPACE=(125,(2500,100),RLSE,,ROUND)
//             PEND
//*-------------------------------------------------------------------*//
//STEP01   EXEC BMP,RGN=0M,
//         MBR=DFSDDLT0,PSB=FPSBPA,NBA=10
//G.PRINTDD  DD SYSOUT=*
//G.SYSIN    DD *
S11 1 1 1 1    CUSTDB
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001YYYYYYYYYYYYYYYYYYYYYYYY
L        GHU   CUSTOMER(CKEY     =0021100001)
L        REPL
L        DATA  0021100001AAAAAAAAAAAAAAAAAAAAAAAA
WTOR ENTER NN. TO RESUME IM2A
S11 1 1 1 1    DISTDB
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110YYYYYYYYYYYYYYYYYYYYYYYY
L        GHU   DISTRICT(DKEY     =002110)
L        REPL
L        DATA   Ç002110AAAAAAAAAAAAAAAAAAAAAAAA
L        SYNC
/*
//
```

When DEDLOCK1 is executed, it updates a segment in the DISTDB and goes into a wait on a WTOR. When job DEDLOCK2 is executed, it updates a segment in the CUSTDB and also waits on a WTOR.

If either of the jobs is allowed to continue by a reply to its WTOR, it waits on the locks held on behalf of the other application. It does so because the retrieve call is for the segment updated by the other application before the system has had a chance to commit the data. When the second application is allowed to start, a deadlock occurs and IMS must begin deadlock detection and recovery.

- Start DEDLOCK1 via a /STA REGION DEDLOCK1 on SC53TS. Example B-11 presents the WTOR that came from DEDLOCK1.

Example: B-11 WTOR from DEDLOCK1 after updating the DISTDB database

```
JOB16709 @2397 ENTER NN. TO RESUME IM1A
```

- Start DEDLOCK2 via a /STA REGION DEDLOCK2 on SC47TS. Example B-12 presents the WTOR that came from DEDLOCK2.

Example: B-12 WTOR from DEDLOCK2 after updating the CUSTDB database

```
JOB16711 @2403 ENTER NN. TO RESUME IM2A
```

- Reply to the WTOR for DEDLOCK1
- A /DIS A for IM1A indicates that although we have released DEDLOCK1 from its WTOR wait it cannot complete since it is waiting for the release of locks held on behalf of DEDLOCK2. A display of the IM1AIRLM status indicates we are still waiting on locked resources.

Example: B-13 Display of M1AIRLM status output

```
DXR101I IROA001 STATUS SCOPE=NoDISC 878
         SUBSYSTEMS IDENTIFIED
         NAME     STATUS       UNITS      HELD    WAITING    RET_LKS
         IM1A     UP               3         7          1          0
     DXR101I End of display
```

- Reply to the WTOR for DEDLOCK2
- You might expect that one of the BMPs would abend with an user 777 code. This results when applications enter into deadlocks but one of the DBPCBs databases in question in the PSB; DISTRICT, is a DEDB VSO database. This makes the PSB we used mixed mode.

  When a deadlocks between these types of BMPs occur, one BMP will receive an FD status code, have its in-flight work backed out internally to the end of the last checkpoint and if the application has no code to provide alternate processing it will continue on till normal termination. The other BMP will continue to normal termination.

So lets see if we can simulate an abend U0777 with two BMPs running on different IMS images in different OS/390 systems accessing full function databases.

Deadlocks when only full function databases included in PSB

An IMS Version 7 system, IM2A running on OS/390 image SC47 will execute DEDLOFF2 to first perform a GHU call to segment ORDER in the ORDRDB database, wait on a WTOR and then update a CUSTOMER segment in the CUSTDB database.

Example: B-14 DFSDDLT0 running as a BMP to demonstrate deadlocks

```
//DEDLOFF2 JOB CLASS=A,MSGCLASS=O,REGION=4096K,TIME=1440
/*JOBPARM SYSAFF=SC47
//*
//*DDLTO JOB TO VERIFY LOCKING
//*
//BMP    PROC  MBR=TEMPNAME,PSB=,IN=,OUT=,
//             OPT=N,SPIE=0,TEST=0,DIRCA=000,
//             PRLD=,STIMER=,CKPTID=,PARDLI=,
//             CPUTIME=,NBA=,OBA=,IMSID=IM2A,AGN=,
//             SSM=,PREINIT=,RGN=512K,SOUT=*,
//             ALTID=IM2A,APARM=
//*
//G      EXEC  PGM=DFSRRC00,REGION=&RGN,
//             PARM=(BMP,&MBR,&PSB,&IN,&OUT,
//             &OPT&SPIE&TEST&DIRCA,&PRLD,
//             &STIMER,&CKPTID,&PARDLI,&CPUTIME,
//             &NBA,&OBA,&IMSID,&AGN,&SSM,
//             &PREINIT,&ALTID,
//             '&APARM')
```

```
//STEPLIB  DD DSN=IMSPSA.IMS0.SDFSRESL.V7,DISP=SHR
//         DD DSN=IMSPSA.IMS0.PGMLIB,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR                    <--- LE/370
//PROCLIB  DD DSN=IMSPSA.IM2A.PROCLIB,DISP=SHR
//SYSUDUMP DD  SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//             SPACE=(125,(2500,100),RLSE,,ROUND)
//             PEND
//*-------------------------------------------------------------------*//
//STEP01   EXEC BMP,RGN=0M,
//         MBR=DFSDDLT0,PSB=PSBGW
//G.PRINTDD  DD SYSOUT=*
//G.SYSIN    DD *
S11 1 1 1 1    ORDRDB
L        GHU   ORDER
WTOR ENTER NN. TO RESUME IM2A
S11 1 1 1 1    CUSTDB
L        GHU   CUSTOMER(CKEY     =0001010001)
L        REPL
L        DATA  0001010001XXXXXXXXXXXXXXXXXXXXXXXXX
L        SYNC
/*
//
```

An IMS Version 8 system, IM3A running on OS/390 image SC69 will execute DEDLOFF3 to first perform an update a CUSTOMER segment in the CUSTDB database, wait on a WTOR and then perform a GHU call to segment ORDER in the ORDRDB database.

Example: B-15 DFSDDLT0 running as a BMP to demonstrate deadlocks

```
//DEDLOFF3 JOB CLASS=A,MSGCLASS=0,REGION=4096K,TIME=1440
/*JOBPARM SYSAFF=SC69
//*
//*DDLT0 JOB TO VERIFY LOCKING
//*
//BMP    PROC  MBR=TEMPNAME,PSB=,IN=,OUT=,
//             OPT=N,SPIE=0,TEST=0,DIRCA=000,
//             PRLD=,STIMER=,CKPTID=,PARDLI=,
//             CPUTIME=,NBA=,OBA=,IMSID=IM3A,AGN=,
//             SSM=,PREINIT=,RGN=512K,SOUT=*,
//             ALTID=IM3A,APARM=
//*
//G      EXEC  PGM=DFSRRC00,REGION=&RGN,
//             PARM=(BMP,&MBR,&PSB,&IN,&OUT,
//             &OPT&SPIE&TEST&DIRCA,&PRLD,
//             &STIMER,&CKPTID,&PARDLI,&CPUTIME,
//             &NBA,&OBA,&IMSID,&AGN,&SSM,
//             &PREINIT,&ALTID,
//             '&APARM')
//STEPLIB  DD DSN=IMSPSA.IMS0.SDFSRESL,DISP=SHR
//         DD DSN=IMSPSA.IMS0.PGMLIB,DISP=SHR
//         DD DSN=CEE.SCEERUN,DISP=SHR                    <--- LE/370
//PROCLIB  DD DSN=IMSPSA.IM0A.PROCLIB,DISP=SHR
//SYSUDUMP DD  SYSOUT=&SOUT,DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//             SPACE=(125,(2500,100),RLSE,,ROUND)
//             PEND
//*-------------------------------------------------------
//STEP01   EXEC BMP,RGN=0M,
//         MBR=DFSDDLT0,PSB=PSBGW
//*        MBR=DFSDDLT0,PSB=FPSBPA,NBA=10
//G.PRINTDD  DD SYSOUT=*
```

```
//G.SYSIN    DD *
S11 1 1 1 1    CUSTDB
L        GHU   CUSTOMER(CKEY     =0001010001)
L        REPL
L        DATA  0001010001XXXXXXXXXXXXXXXXXXXXXXXXX
WTOR ENTER NN. TO RESUME IM3A
S11 1 1 1 1    ORDRDB
L        GHU   ORDER
L        SYNC
/*
//
```

Each BMP is forced to execute on a specific OS/390 image because of the SYSAFF job statements. Also the PSB ‘PSBGW’ only has sensitivity to the two full function CUSTDB and ORDER databases.

First we submit both BMPs and note that they both wait on their respective WTORs.

Here is a display of the activity for IM2A after the WTOR from DEDLOFF2 is responded to. The reason that DEDLOFF2 is still executing is that it waits for the release of locks held via calls from DEDLOFF3

Example: B-16 Display of IM2A activity after WTOR for DEDLOFF2 replied to

```
?DIS A
DFS4444I DISPLAY FROM ID=IM2A
     REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS          CLASS
           MSGRGN    TP    NONE
         1 DEDLOFF2  BMP   G         PSBGW
           FPRGN     FP    NONE
           DBTRGN    DBT   NONE
           IM2ADBRC  DBRC
           IM2ADLS   DLS
     VTAM ACB CLOSED
     IMSLU=N/A.N/A            APPC STATUS=DISABLED  TIMEOUT=   0
     OTMA GROUP=N/A       STATUS=NOTACTIVE
     APPLID=IMS2A     GRSNAME=IMSOA    STATUS=DISABLED
     LINE ACTIVE-IN -    1 ACTIV-OUT -    0
     NODE ACTIVE-IN -    0 ACTIV-OUT -    0
     *2002330/171050*
```

Finally we respond to the outstanding WTOR from DEDLOFF3, when IM2A and IM3A detect that a deadlock exists. Example B-17 contains the output of the abend U0777 (deadlock detection, backout, and termination) for DEDLOFF3 and normal completion for DEDLOFF2.

Example: B-17 Status of both BMPs after the final WTOR from DEDLOFF3 is responded to

```
DFS554A DEDLOFF3 00001 G        PSBGW   (2)          000,0777
          2002/330 17:56:32 IM3A
DFS552I BATCH   REGION DEDLOFF3 STOPPED ID=00001 TIME=1756  IM3A
IEF450I DEDLOFF3 G STEP01 - ABEND=S000 U0777 REASON=00000000 294
        TIME=17.56.32
.....
DFS552I BATCH   REGION DEDLOFF2 STOPPED ID=00001 TIME=1756  IM2A
```

Since a pseudo abend has been taken, we should be able to obtain the type x’67ff’ records off the OLDS that was active for IM3A when the ABU0777 occurred. Here is the JCL that will pull those records off and format the deadlock report.

Example: B-18 DFSERA10 with exit DFSERA30 for type X'67ff' records

```
//LOGPRINT JOB CLASS=A,MSGCLASS=X,REGION=4096K,
//           NOTIFY=JOUKO4
//PRINT  EXEC PGM=DFSERA10
//STEPLIB   DD  DSN=IMSPSA.IMS0.SDFSRESL,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//TRPUNCH   DD  SYSOUT=*,DCB=BLKSIZE=80
//SYSUT1   DD DSN=IMSPSA.IM3A.OLP04,DISP=SHR,
//         DCB=BUFNO=5
//SYSIN    DD  *
CONTROL  CNTL   H=EOF
OPTION   PRINT  E=DFSERA30,V=67FF,T=X,L=2,O=5
/*
//
```

When the output of the LOGPRINT job is examined, there is a section of the formatted report that contains a summary of the deadlock event.

Example: B-19 Deadlock analysis report from DFSERA10 with exit DFSERA30

```
DEADLOCK ANALYSIS REPORT - LOCK MANAGER IS IRLM
.......................................................................
RESOURCE DMB-NAME LOCK-LEN LOCK-NAME
01 OF 02 CUSTDB      08    00004004800701D7
LOCKING ON HDAM ANCHOR, KEY DISPLAYED IS HDAM KEY REQUESTED
KEY=(0001010001)
       IMS-NAME TRAN/JOB PSB-NAME PCB--DBD PST#  RGN  CALL LOCK  LOCKFUNC STATE
WAITER IM2A     DEDLOFF2 PSBGW    CUSTDB   00001 BMP  GET  GRIDX 30400358  06-P
BLCKER IM3A     DEDLOFF3 PSBGW    -------- 00001 BMP  ---- ----- --------  06-P
.......................................................................
RESOURCE DMB-NAME LOCK-LEN LOCK-NAME     - WAITER FOR THIS RESOURCE IS VICTIM
02 OF 02 ORDRDB      08    00002004800901D7
KEY FOR RESOURCE IS NOT AVAILABLE
       IMS-NAME TRAN/JOB PSB-NAME PCB--DBD PST#  RGN  CALL LOCK  LOCKFUNC STATE
WAITER IM3A     DEDLOFF3 PSBGW    ORDRDB   00001 BMP  GET  GRIDX 30400358  06-P
BLCKER IM2A     DEDLOFF2 PSBGW    -------- 00001 BMP  ---- ----- --------  06-P

DEADLOCK ANALYSIS REPORT - END OF REPORT
```

The report indicates that the waiter for the resource in ORDRDB database; DEDLOFF3 running in IM3A was the victim. The locks in question are the database record locks requested at private update level.

This completes our review of how simple DFSDDLT0 BMPs can be used to test locking and recovery scenarios in a BLDS environment.

C

Sample uservar exit for network balancing

Note: This appendix applies only IMS Transaction Manager.

The following sample exit is provided as an example of a USERVAR exit that could be used to implement workload balancing through the network balancing approach. It is supplied primarily for users who do not have support for VTAM generic resources.

This sample exit has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either expressed or implied. The use of this sample exit depends on the customer's ability to evaluate and integrate it into the customer's operational environment.

The sample will simulate generic logon support for IMS by distributing logon requests by hashing or randomizing the node name requesting logon. Since a repeatable hashing technique is used (the same LU name always hashes to the same value), successive logon requests from the same node will be directed to the same IMS system (APPLID). The exit routine is table driven (the table is assembled with the exit) and can handle translation requests for up to five different USERVARs. The ability to handle multiple USERVARs is critical since there is only one VTAM USERVAR exit within a VTAM system. Logon requests for each USERVAR managed by the exit can be distributed among up to ten different IMS system (APPLIDs).

The limits on the number of USERVARs managed and the number of APPLIDs per USERVAR are easily changed. The sample exit also allows the logon distribution characteristics to be changed through operator command for one of the managed USERVARs.

For information regarding the VTAM USERVAR exit, ISTEXCUV, see the *VTAM Customization* manual for your VTAM release. For information regarding the VTAM definition of USERVARs and a description of the VTAM commands used to specify the use of, activation and replacement of, and how to pass parameter information to a USERVAR exit, see the *VTAM Operation* manual for your VTAM release.

Example: C-1 Sample USERVAR exit

```
UVAR     TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX00010000
                                TIPLE IMS SYSTEMS'                                  00020000
                      ISTEXCUV CSECT ,                                              00030000
                      ISTEXCUV AMODE 31                                             00040000
                      ISTEXCUV RMODE ANY                                            00050000
                               PRINT NOGEN                                          00060000
                               SPACE 1                                              00070000
                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00080000
                      *                                                             * 00090000
                      *  MODULE NAME:  ISTEXCUV                                     * 00100000
                      *                                                             * 00110000
                      *  ATTRIBUTES:  AMODE 31, RMODE ANY, REENTRANT                * 00120000
                      *                                                             * 00130000
                      *  PURPOSE:  THIS EXIT WILL SIMULATE GENERIC LOGON SUPPORT FOR IMS BY * 00140000
                      *            DISTRIBUTING LOGON REQUESTS BASED ON HASHING THE NODE    * 00150000
                      *            NAME REQUESTING LOGON.  SINCE A REPEATABLE HASHING       * 00160000
                      *            TECHNIQUE IS USED, SUCCESSIVE LOGON REQUESTS REQUESTS    * 00170000
                      *            FROM THE SAME LU WILL BE DIRECTED TO THE SAME IMS SYSTEM * 00180000
                      *            (ASSUMING THAT THIS EXIT HAS NOT BEEN UPDATED WITH NEW   * 00190000
                      *            DISTRIBUTION PARAMETERS).                                * 00200000
                      *                                                                     * 00210000
                      *            THE EXIT IS TABLE DRIVEN AND CAN HANDLE REQUESTS FOR UP  * 00220000
                      *            TO 5 DIFFERENT USERVARS.  FOR EACH USERVAR THAT IS       * 00230000
                      *            CONTROLLED, LOGON REQUESTS CAN BE DISTRIBUTED AMONG UP   * 00240000
                      *            TO 10 DIFFERENT IMS SYSTEMS (APPLIDS).                   * 00250000
                      *                                                                     * 00260000
                      *            THIS SAMPLE EXIT HAS NOT BEEN SUBMITTED TO ANY FORMAL    * 00270000
                      *            IBM TEST AND IS DISTRIBUTED ON AN 'AS IS' BASIS WITHOUT  * 00280000
                      *            ANY WARRANTY EITHER EXPRESSED OR IMPLIED.  THE USE OR    * 00290000
                      *            IMPLEMENTATION OF THIS SAMPLE EXIT IS A CUSTOMER         * 00300000
                      *            RESPONSIBILITY AND DEPENDS ON THE CUSTOMER'S ABILITY TO  * 00310000
                      *            EVALUATE AND INTEGRATE IT INTO THE CUSTOMER'S            * 00320000
                      *            OPERATIONAL ENVIRONMENT.                                 * 00330000
                      *                                                                     * 00340000
                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00350000
                               TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX00360000
                                TIPLE IMS SYSTEMS - INITIALIZATION'                 00370000
                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00380000
                      *                                                             * 00390000
                      *  DEFINE REGISTER EQUATES                                    * 00400000
                      *  SAVE REGISTERS                                             * 00410000
                      *  ESTABLISH EXIT BASE                                        * 00420000
                      *  DETERMINE TYPE OF CALL AND BRANCH TO APPROPRIATE ROUTINE   * 00430000
                      *  ISSUE WTO IF TYPE OF CALL UNKNOWN AND RETURN               * 00440000
                      *                                                             * 00450000
                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00460000
                               SPACE 1                                              00470000
                      R0       EQU   0                                              00480000
                      R1       EQU   1                                              00490000
                      R2       EQU   2                                              00500000
                      R3       EQU   3                                              00510000
                      R4       EQU   4                                              00520000
                      R5       EQU   5                                              00530000
                      R6       EQU   6                                              00540000
                      R7       EQU   7                                              00550000
                      R8       EQU   8                                              00560000
                      R9       EQU   9                                              00570000
                      R10      EQU   10                                             00580000
                      R11      EQU   11                                             00590000
```

```
R12      EQU   12                                                  00600000
R13      EQU   13                                                  00610000
R14      EQU   14                                                  00620000
R15      EQU   15                                                  00630000
         SPACE 1                                                   00640000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00650000
*                                                                  * 00660000
*  THE EYECATCHER IN THE SAVE MACRO 'V001' SHOULD BE CHANGED WHEN THE * 00670000
*  VERSION OF THIS EXIT IS CHANGED.  THIS WILL ALLOW YOU TO        * 00680000
*  CORRELATE A DUMP WITH AN EXIT ASSEMBLY.                         * 00690000
*                                                                  * 00700000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00710000
         SPACE 1                                                   00720000
         SAVE  (14,12),,&SYSDATE..V001                             00730000
         USING ISTEXCUV,R12        DEFINE EXIT BASE                00740000
         LR    R12,R15             LOAD EXIT BASE                  00750000
         CH    R0,HEX04            IS ENTRY FOR EXIT INVOCATION?   00760000
         BE    INVOKE              YES, BRANCH TO ROUTINE          00770000
         CH    R0,HEX10            IS ENTRY FOR EXIT ACTIVATION?   00780000
         BE    ACT                 YES, BRANCH TO ROUTINE          00790000
         CH    R0,HEX18            IS ENTRY FOR EXIT REPLACEMENT?  00800000
         BE    REPL                YES, BRANCH TO ROUTINE          00810000
         CH    R0,HEX20            IS ENTRY FOR EXIT DEACTIVATION? 00820000
         BE    DEACT               YES, BRANCH TO ROUTINE          00830000
         CH    R0,HEX40            IS ENTRY FOR VTAM TERMINATION?  00840000
         BE    INITRET             YES, BRANCH TO ROUTINE          00850000
         SPACE 1                                                   00860000
         WTO   'REGISTER 0 CONTENTS TO USERVAR EXIT UNEXPECTED',  X00870000
               ROUTCDE=(10),       SYSTEM ERROR INFORMATION        X00880000
               DESC=(12),          IMPORTANT INFORMATION MESSAGE   X00890000
               LINKAGE=BRANCH      USE BRANCH ENTRY                 00900000
         SPACE 1                                                   00910000
INITRET  DS    0H                                                  00920000
         RETURN (14,12),,RC=0      RETURN                          00930000
         TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX00940000
               TIPLE IMS SYSTEMS - EXIT INVOCATION'                00950000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00960000
*                                                                  * 00970000
*  IF NON-TRANSLATION REQUEST THEN                                 * 00980000
*    RETURN                                                        * 00990000
*  IF TRANSLATION REQUEST FOR USERVAR NOT MANAGED BY EXIT THEN     * 01000000
*    SET NOT-TRANSLATED                                            * 01010000
*    RETURN                                                        * 01020000
*  HASH THE OLU NAME GIVING RESULT 1 - 100                         * 01030000
*  SEARCH DIST TABLE FOR CUMULATIVE PERCENT GREATER THAN OR EQUAL  * 01040000
*    TO HASH VALUE                                                 * 01050000
*  SET TRANSLATED NAME TO DIST TABLE ENTRY                         * 01060000
*  SET TRANSLATED                                                  * 01070000
*  RETURN                                                          * 01080000
*                                                                  * 01090000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 01100000
         SPACE 1                                                   01110000
INVOKE   DS    0H                 EXIT INVOCATION ROUTINE         01120000
         LR    R2,R1              GET PARM ADDR                   01130000
         USING PARM2,R2           SET PARAMETER BASE              01140000
         L     R3,INVOKEA         GET ADDR OF INVOKE FLAGS        01150000
         CLI   0(R3),X'04'        IS THIS A TRANSLATION REQUEST?  01160000
         BNE   INVOKRET           NO, RETURN                      01170000
         L     R3,USRVPRMS        GET ADDR OF USERVAR PARMS       01180000
         USING USRVPARM,R3        SET USERVAR PARMS BASE          01190000
```

```
         L     R4,USERADD1         GET ADDR OF USER FIELD               01200000
         USING USERFLD,R4          SET BASE FOR USER FIELD              01210000
         L     R4,STORADDR         GET ADDR OF GETMAINED STORAGE        01220000
         DROP  R4                  DROP USER FIELD BASE                 01230000
         LA    R4,WORKLEN(,R4)     GET ADDR OF USERVAR TBL              01240000
         USING USRVDSCT,R4         SET BASE FOR USERVAR ENTRY           01250000
INVOKE1  DS    0H                  START OF SEARCH LOOP                 01260000
         CLC   USRVNAME,GENERIC    USERVAR NAME FOUND IN TABLE?         01270000
         BE    INVOKE2             YES, PROCESS                         01280000
         L     R4,USRVNEXT         GET ADDR OF NEXT USERVAR ENTRY       01290000
         LTR   R4,R4               IS THERE A NEXT USERVAR ENTRY?       01300000
         BNZ   INVOKE1             YES, LOOP                            01310000
         B     INVOKRET            NO, USERVAR NOT FOUND, RETURN        01320000
INVOKE2  DS    0H                  USERVAR FOUND, NOW HASH OLU NAME     01330000
         SPACE 1                                                        01340000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 01350000
*  VERY SIMPLE HASHING TECHNIQUE OF LOGICALLY ADDING THE FIRST 4      * 01360000
*  BYTES OF THE OLU TO THE LAST 4 BYTES OF THE OLU, DIVIDING BY 100   * 01370000
*  AND ADDING 1 TO THE REMAINDER.                                     * 01380000
*                                                                     * 01390000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 01400000
         SPACE 1                                                        01410000
         L     R7,OLUNAME          GET FIRST 4 BYTES OF OLU             01420000
         AL    R7,OLUNAME+4        ADD LAST 4 BYTES OF OLU              01430000
         N     R7,HIOFF            CLEAR HIGH BIT                       01440000
         XR    R6,R6               CLEAR R6                             01450000
         D     R6,=F'100'          DIVIDE BY 100                        01460000
         LA    R6,1(,R6)           ADD 1 TO REMAINDER                   01470000
         SPACE 1                                                        01480000
         L     R5,USRVDIST         GET ADDR OF DISTRIBUTION TABLE       01490000
         USING DISTDSCT,R5         SET DISTRIBUTION TABLE ENTRY BASE    01500000
INVOKE3  DS    0H                  START OF DIST TBL SEARCH LOOP        01510000
         C     R6,DISTHIGH         COMPARE HASH VALUE TO CUMM PERCENT   01520000
         BNH   INVOKE4             BRANCH IF LESS THAN OR EQUAL         01530000
         LA    R5,DISTLEN(,R5)     POINT TO NEXT ENTRY                  01540000
         B     INVOKE3             LOOP                                 01550000
INVOKE4 DS     0H                  CORRECT DISTRIBUTION ENTRY FOUND     01560000
         MVC   TRANVAL,DISTAPPL    MOVE APPLID TO PARM LIST             01570000
         OI    FLAGS,TRANSLAT      SET TRANSLATE FLAG ON                01580000
INVOKRET DS    0H                  RETURN                               01590000
         RETURN (14,12),,RC=0                                           01600000
         DROP  R2                  DROP PARAMETER BASE                  01610000
         DROP  R3                  DROP USERVAR PARMS BASE              01620000
         DROP  R4                  DROP USERVAR TABLE ENTRY BASE        01630000
         DROP  R5                  DROP DISTRIBUTION TABLE ENTRY BASE   01640000
         TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX01650000
               TIPLE IMS SYSTEMS - EXIT ACT/REPL'                       01660000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 01670000
*                                                                     * 01680000
*  OBTAIN STORAGE FOR USERVAR LIST, DISTRIBUTION LIST, AND USER DATA  * 01690000
*    WORKING STORAGE                                                  * 01700000
*  IF STORAGE REQUEST FAILED THEN                                     * 01710000
*    ISSUE ERROR WTO AND RETURN                                       * 01720000
*  BUILD USERVAR AND DISTRIBUTION LISTS IN OBTAINED STORAGE           * 01730000
*  IF USER DATA PROVIDED THEN                                         * 01740000
*    VALIDATE USERVAR NAME AND DISTRIBUTION PARAMETERS                * 01750000
*    IF PARAMETER ERROR THEN                                          * 01760000
*      ISSUE ERROR WTO AND RETURN                                     * 01770000
*    ELSE                                                             * 01780000
*      OVER-RIDE SPECIFIED USERVAR PARAMETER DATA                     * 01790000
```

```
*  RETURN                                                        * 01800000
*                                                                * 01810000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 01820000
         SPACE 1                                                  01830000
ACT      DS    0H                                                 01840000
REPL     DS    0H                                                 01850000
         LR    R7,R1               SAVE PARAMETER ADDR            01860000
         L     R2,WRKSPCLN         GET SIZE OF REQUIRED STORAGE   01870000
         STORAGE OBTAIN,COND=YES,  OBTAIN STORAGE FOR LISTS       X01880000
               LENGTH=(2),ADDR=(3)                                01890000
         LTR   R15,R15             WAS STORAGE REQUEST SUCCESSFUL? 01900000
         BZ    ACT1                YES, BRANCH OVER ERROR MESSAGE 01910000
         SPACE 1                                                  01920000
         WTO   'STORAGE OBTAIN FAILED IN USERVAR EXIT',           X01930000
               ROUTCDE=(10),       SYSTEM ERROR INFORMATION       X01940000
               DESC=(12),          IMPORTANT INFORMATION MESSAGE  X01950000
               LINKAGE=BRANCH      USE BRANCH ENTRY               01960000
         B     ACTRET              RETURN                         01970000
         SPACE 1                                                  01980000
ACT1     DS    0H                  RELOCATE TABLE ENTRIES         01990000
         LA    R4,USRVTBL1         GET ADDR OF 1ST USERVAR ENTRY DEF 02000000
         LA    R5,WORKLEN(,R3)     GET ADDR OF USERVAR TBL        02010000
         USING USRVDSCT,R5         DEFINE USERVAR ENTRY BASE      02020000
ACT2     DS    0H                  START OF RELOCATION LOOP       02030000
         MVC   0(USRVLEN+DISTLEN*DISTMAX,R5),0(R4) MOVE ENTIRE ENTRY 02040000
         LA    R6,USRVLEN(,R5)     GET ADDR OF DISTRIBUTION LIST  02050000
         ST    R6,USRVDIST         STORE IN USERVAR ENTRY         02060000
         L     R6,USRVNEXT         GET ADDR OF NEXT USERVAR ENTRY 02070000
         LTR   R6,R6               IS THERE ANOTHER?              02080000
         BZ    ACT3                NO, GO CHAIN USERVAR ENTRIES   02090000
         LA    R4,USRVLEN+DISTLEN*DISTMAX(,R4) POINT TO NEXT USERVAR 02100000
         LA    R5,USRVLEN+DISTLEN*DISTMAX(,R5) POINT TO NEXT SPACE 02110000
         B     ACT2                GO RELOCATE NEXT ENTRY         02120000
         DROP  R5                  DROP USERVAR ENTRY BASE        02130000
         SPACE 1                                                  02140000
ACT3     DS    0H                                                 02150000
         LA    R5,WORKLEN(,R3)     GET ADDR OF USERVAR TBL        02160000
ACT4     DS    0H                  START OF USERVAR ENTRY CHAINING LOOP 02170000
         USING USRVDSCT,R5         SET USERVAR BASE TO RELOCATED ENTRY 02180000
         L     R6,USRVNEXT         GET ADDR OF NEXT ENTRY         02190000
         LTR   R6,R6               IS THERE A NEXT ENTRY?         02200000
         BZ    ACT5                NO,                            02210000
         LA    R6,USRVLEN+DISTLEN*DISTMAX(,R5) GET ADDR OF NXT REL ENT 02220000
         ST    R6,USRVNEXT         STORE IN RELOCATED ENTRY       02230000
         DROP  R5                  DROP USERVAR BASE              02240000
         LR    R5,R6               GET ADDR OF NEXT ENTRY         02250000
         B     ACT4                PROCESS NEXT ENTRY             02260000
ACT5     DS    0H                  STORE ADDR OF OBT STOR IN USER FIELD 02270000
         USING PARM1,R7            DEF BASE FOR INPUT PARM LIST   02280000
         L     R2,USERADDR         GET ADDR OF USER FIELD         02290000
         L     R8,PARMADDR         GET ADDR OF USER DATA          02300000
         DROP  R7                  DROP INPUT PARM LIST BASE      02310000
         USING USERFLD,R2          DEF BASE FOR USER FIELD        02320000
         ST    R3,STORADDR         SAVE OBTAINED STORAGE ADDR     02330000
         MVC   VERID,VERSION       SAVE EXIT VERSION ID           02340000
         DROP  R2                  DROP INPUT PARM LIST BASE      02350000
         SPACE 1                                                  02360000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 02370000
*                                                                * 02380000
*  THIS SECTION PROCESSES USER DATA PASSED AS PART OF THE ACTIVATE OR * 02390000
```

```
*  REPLACE COMMAND                                                     * 02400000
*                                                                      * 02410000
*  THE USER DATA MUST BE IN THE FORMAT OF                              * 02420000
*                                                                      * 02430000
*    USER-VAR,XXX,XXX,XXX,XXX,XXX,XXX,XXX,XXX,XXX,XXX                  * 02440000
*                                                                      * 02450000
*    WHERE USER-VAR IS THE 8 CHARACTER (PAD WITH BLANKS) USER NAME     * 02460000
*    TO BE OVER-RIDEN                                                  * 02470000
*                                                                      * 02480000
*    XXX IS A 3 DIGIT (MUST BE 3 DIGITS) DISTRIBUTION PERCENTAGE       * 02490000
*    TO BE OVER-RIDDEN                                                 * 02500000
*                                                                      * 02510000
*    AT LEAST ONE OVER-RIDE PERCENTAGE MUST BE SPECIFIED AND UP TO     * 02520000
*    10 CAN BE SPECIFIED.  OVER-RIDE PERCENTAGE DISTRIBUTION ENTIRES   * 02530000
*    APPLY TO THE CORRESPONDING ENTRIES IN THE APPROPRIATE USERVAR     * 02540000
*    DISTRIBUTION TABLE COMPILED IN THIS EXIT ROUTINE.                 * 02550000
*                                                                      * 02560000
*    NEW ENTRIES IN THE DISTRIBUTION TABLE CANNOT BE ADDED VIA USER    * 02570000
*    DATA.  IN ADDITION, THE MERGING OF THE OVER-RIDE AND COMPILED     * 02580000
*    DISTRIBUTION ENTRIES MUST RESULT IN A TABLE WHERE EACH            * 02590000
*    DISTRIBUTION ENTRY IS GREATER THAN OR EQUAL TO THE PRIOR AND AT   * 02600000
*    LEAST ONE DISTRIBUTION PERCENTAGE IS 100.                         * 02610000
*                                                                      * 02620000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 02630000
        SPACE 1                                                         02640000
        LTR   R8,R8             IS ADDR OF USER DATA ZERO?              02650000
        BZ    ACTRET            YES, RETURN                             02660000
        LH    R2,0(,R8)         GET USER DATA LENGTH                    02670000
        CH    R2,=H'0'          IS USER DATA LENGTH ZERO?               02680000
        BE    ACTRET            YES, RETURN                             02690000
        CH    R2,=H'12'         IS USER DATA LENGTH AT LEAST 12?        02700000
        BL    PARMERR           NO, INVALID FORMAT                      02710000
        CH    R2,=H'48'         IS USER DATA LENGTH GREATER THAN 48?    02720000
        BH    PARMERR           YES, INVALID FORMAT                     02730000
        TM    1(R8),X'03'       IS USER DATA LENGTH A MULT OF 4?        02740000
        BNZ   PARMERR           NO, INVALID FORMAT                      02750000
        LA    R4,2(,R8)         SET R4 TO USER DATA                     02760000
        LA    R5,0(R2,R4)       SET R5 TO END OF USER DATA + 1          02770000
*  R3 POINTS TO GETMAINED STORAGE, R4 POINTS TO USER DATA, AND R5       02780000
*  POINTS TO BYTE AFTER USER DATA                                       02790000
        USING WORKSTOR,R3       DEFINE WORKING STORAGE BASE             02800000
        MVI   WSSWITCH,X'00'    CLEAR WORKING STORAGE SWITCH            02810000
        LA    R7,DISTMAX        SET LOOP COUNTER                        02820000
        LA    R6,WSDIST         POINT FOR FIRST WS DIST ENTRY           02830000
ACT6    DS    0H                INIT WS LOOP                            02840000
        MVC   0(L'WSDIST,R6),=F'-1' SET WS DIST ENTRY TO -1             02850000
        LA    R6,L'WSDIST(,R6)  POINT TO NEXT WS DIST ENTRY             02860000
        BCT   R7,ACT6           LOOP DISTMAX TIMES                      02870000
*  SAVE USERVAR NAME FROM USER DATA AND CHECK DIST SYNTAX               02880000
        MVC   WSUSRVAR,0(R4)    MOVE USERVAR NAME TO WS                 02890000
        LA    R6,8(,R4)         GET ADDR OF FIRST OVER-RIDE             02900000
        LA    R7,WSDIST         GET ADDR OF FIRST WS DIST ENTRY         02910000
ACT7    DS    0H                SYNTAX CHECKING LOOP                    02920000
        CLI   0(R6),C','        IS 1ST CHAR A COMMA?                    02930000
        BNE   PARMERR           NO, ERROR                               02940000
        TRT   1(3,R6),TRTABLE   ARE NEXT 3 CHARS NUMERIC?               02950000
        BNZ   PARMERR           NO, ERROR                               02960000
        PACK  WSDOUBLE,1(3,R6)  PACK DIST OVER-RIDE                     02970000
        CVB   R2,WSDOUBLE       CONVERT DIST OVER-RIDE TO BINARY        02980000
        ST    R2,0(R7)          STORE IN WS DIST ENTRY                  02990000
```

```
         LA    R6,4(,R6)           GET ADDR OF NEXT OVER-RIDE          03000000
         LA    R7,L'WSDIST(,R7)    GET ADDR OF NEXT WS DIST ENTRY      03010000
         CR    R6,R5               POINTING BEYOND USER DATA?          03020000
         BL    ACT7                NO, LOOP                            03030000
*  CHECK IF USERVAR SPECIFIED IS CONTROLLED                            03040000
         LA    R11,WORKLEN(,R3)    GET ADDR OF 1ST USERVAR             03050000
         USING USRVDSCT,R11        DEFINE USERVAR ENTRY BASE           03060000
ACT8     DS    0H                  SEARCH USERVAR TABLE LOOP           03070000
         CLC   USRVNAME,WSUSRVAR   SPECIFIED USERVAR NAME FOUND?       03080000
         BE    ACT9                YES, EXIT LOOP                      03090000
         L     R11,USRVNEXT        NO, GET ADDR OF NEXT USERVAR ENTRY  03100000
         LTR   R11,R11             IS IT ZERO?                         03110000
         BNZ   ACT8                NO, LOOP                            03120000
         B     PARMERR             YES, ERROR                          03130000
*  CHECK WS DIST ENTRIES FOR INCREASING VALUES AND VALUES BETWEEN      03140000
*  0 AND 100                                                           03150000
ACT9     DS    0H                                                      03160000
         LA    R6,WSDIST           GET ADDR OF FIRST DIST ENTRY        03170000
         XC    WSOLDHI,WSOLDHI     CLEAR CURRENT HIGH ENTRY HOLDER     03180000
         LA    R7,DISTMAX          SET LOOP COUNTER                    03190000
ACT10    DS    0H                  CHECK DIST VALUE LOOP               03200000
         L     R8,0(,R6)           GET WS DISTRIBUTION ENTRY           03210000
         LTR   R8,R8               IS IT NEGATIVE?                     03220000
         BM    ACT11               YES, GET OUT OF LOOP                03230000
         C     R8,=F'100'          IS IT GREATER THAN 100?             03240000
         BH    PARMERR             YES, ERROR                          03250000
         CH    R8,WSOLDHI          IS IT LESS THAN PRIOR HIGH VALUE?   03260000
         BL    PARMERR             YES, ERROR                          03270000
         STH   R8,WSOLDHI          SAVE CURRENT VALUE AS NEW HIGH      03280000
         LA    R6,L'WSDIST(,R6)    BUMP TO NEXT ENTRY                  03290000
         BCT   R7,ACT10            LOOP DISTMAX TIMES                  03300000
*  CHECK TO SEE IF MERGE OF STATIC DISTRIBUTION TABLE AND OVER-RIDES   03310000
*  RESULTS IN A GOOD DISTRIBUTION TABLE                                03320000
ACT11    DS    0H                                                      03330000
         L     R10,USRVDIST        GET ADDR OF STATIC DIST TABLE       03340000
         USING DISTDSCT,R10        DEFINE BASE FOR STATIC DIST TABLE   03350000
         LA    R6,WSDIST           GET ADDR OF WS DIST TABLE           03360000
         LA    R7,DISTMAX          SET LOOP COUNTER                    03370000
ACT12    DS    0H                  MERGE LOOP                          03380000
         CLC   DISTAPPL,=CL8'NULLAPPL' IS STATIC APPLID NULL?          03390000
         BE    ACT13               YES, SKIP                           03400000
         L     R8,0(,R6)           GET WS DIST VALUE                   03410000
         LTR   R8,R8               IS IT NEGATIVE                      03420000
         BNM   ACT13               NO, SKIP                            03430000
*  SLOT IN STATIC TABLE NOT OVER-RIDDEN, MOVE STATIC ENTRY TO WS       03440000
         MVC   0(L'WSDIST,R6),DISTHIGH  MOVE STATIC TO WS DIST ENTRY   03450000
ACT13    DS    0H                                                      03460000
         CLC   DISTAPPL,=CL8'NULLAPPL' IS STATIC APPLID NULL?          03470000
         BNE   ACT14               NO, SKIP                            03480000
         L     R8,0(,R6)           GET WS DIST VALUE                   03490000
         LTR   R8,R8               IS WS DIST VALUE NEGATIVE?          03500000
         BM    ACT14               YES, SKIP                           03510000
*  NULL ENTRY IN STATIC TABLE OVER-RIDDEN, ERROR                       03520000
         B     PARMERR             ERROR                               03530000
ACT14    DS    0H                                                      03540000
         L     R8,0(R6)            GET WS DIST VALUE                   03550000
         C     R8,=F'100'          IS IT 100?                          03560000
         BNE   ACT15               NO, SKIP                            03570000
         OI    WSSWITCH,WS100FND   YES, SET SWITCH                     03580000
ACT15    DS    0H                                                      03590000
```

```
         CLC   DISTAPPL,=CL8'NULLAPPL' IS STATIC APPLID NULL?          03600000
         BNE   ACT16               NO, SKIP                           03610000
         L     R8,0(,R6)           GET WS DIST VALUE                  03620000
         LTR   R8,R8               IS IT NEGATIVE?                    03630000
         BNM   ACT16               NO, SKIP                           03640000
*  NULL ENTRY IN STATIC TABLE NOT OVER-RIDDEN, SET TO 100 IN WS        03650000
         MVC   0(L'WSDIST,R6),=F'100' SET WS DIST VALUE TO 100        03660000
ACT16    DS    0H                                                     03670000
         LA    R6,L'WSDIST(,R6)    POINT TO NEXT WS DIST ENTRY        03680000
         LA    R10,DISTLEN(,R10)   POINT TO NEXT STATIC DIST ENTRY     03690000
         BCT   R7,ACT12            LOOP DISTMAX TIMES                 03700000
*  IF WS100FND NOT ON, ERROR                                           03710000
         TM    WSSWITCH,WS100FND   IS WS100FND ON?                    03720000
         BZ    PARMERR             NO, ERROR                          03730000
*  WS COPY OF DIST TABLE IS GOOD, MOVE IT TO STATIC COPY IN WS         03740000
         L     R10,USRVDIST        GET ADDR OF STATIC DIST TABLE      03750000
         LA    R6,WSDIST           GET ADDR OF WS DIST TABLE          03760000
         LA    R7,DISTMAX          SET LOOP COUNTER                   03770000
ACT17    DS    0H                  COPY DIST TABLE LOOP               03780000
         MVC   DISTHIGH,0(R6)      MOVE WS DIST ENTRY TO STATIC ENTRY 03790000
         LA    R6,L'WSDIST(,R6)    POINT TO NEXT WS DIST ENTRY        03800000
         LA    R10,DISTLEN(,R10)   POINT TO NEXT STATIC DIST ENTRY     03810000
         BCT   R7,ACT17            LOOP DISTMAX TIMES                 03820000
         DROP  R3                                                     03830000
         DROP  R10                                                    03840000
         DROP  R11                                                    03850000
ACTRET   DS    0H                                                     03860000
         RETURN (14,12),,RC=0      RETURN                             03870000
         SPACE 1                                                      03880000
PARMERR  DS    0H                                                     03890000
         WTO   'ERROR DETECTED IN USER DATA, USER DATA IGNORED',      X03900000
               ROUTCDE=(10),       SYSTEM ERROR INFORMATION           X03910000
               DESC=(12),          IMPORTANT INFORMATION MESSAGE      X03920000
               LINKAGE=BRANCH      USE BRANCH ENTRY                    03930000
         B     ACTRET              RETURN                             03940000
         TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX03950000
               TIPLE IMS SYSTEMS - EXIT DEACTIVATION'                 03960000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 03970000
*                                                                     * 03980000
*  IF STORAGE OBTAINED BY EXIT ACTIVATION OR REPLACE ROUTINES THEN    * 03990000
*    FREE STORAGE OBTAINED                                            * 04000000
*  RETURN                                                             * 04010000
*                                                                     * 04020000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 04030000
         SPACE 1                                                      04040000
DEACT    DS    0H                  EXIT DEACTIVATION ROUTINE          04050000
         USING PARM1,R2            DEFINE BASE FOR DEACT PARM LIST    04060000
         LR    R2,R1               LOAD BASE FOR DEACT PARM LIST      04070000
         L     R2,USERADDR         GET ADDR OF USER FIELD             04080000
         DROP  R2                  DROP BASE FOR DEACT PARM LIST      04090000
         USING USERFLD,R2          DEFINE BASE FOR USER FIELD AREA    04100000
         L     R3,STORADDR         GET ADDR OF OBTAINED STORAGE       04110000
         DROP  R2                  DROP BASE FOR USER FIELD AREA      04120000
         LTR   R3,R3               IS ADDR ZERO (NO STORAGE OBTAINED)? 04130000
         BZ    DEACTRET            YES, RETURN                        04140000
         L     R2,WRKSPCLN         GET SIZE OF ACQUIRED STORAGE       04150000
         STORAGE RELEASE,COND=YES, FREE OBTAINED STORAGE              X04160000
               LENGTH=(2),ADDR=(3)                                    04170000
         LTR   R15,R15             DID STORAGE RELEASE WORK?          04180000
         BZ    DEACTRET            YES, RETURN                        04190000
```

```
         SPACE 1                                                        04200000
         WTO   'STORAGE RELEASE FAILED IN USERVAR EXIT',               X04210000
               ROUTCDE=(10),       SYSTEM ERROR INFORMATION            X04220000
               DESC=(12),          IMPORTANT INFORMATION MESSAGE       X04230000
               LINKAGE=BRANCH      USE BRANCH ENTRY                     04240000
         SPACE 1                                                        04250000
DEACTRET DS    0H                                                      04260000
         RETURN (14,12),,RC=0      RETURN                               04270000
         TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX04280000
               TIPLE IMS SYSTEMS - MACRO DEFINITION'                    04290000
         MACRO                                                          04300000
         VARDEF &NAME,&P1,&P2,&P3,&P4,&P5,&P6,&P7,&P8,&P9,&P10,        X04310000
               &LAST=NO                                                 04320000
         GBLA  &NMACRO                                                  04330000
         LCLA  &LCTR,&DPCT(10),&HIGHVAL,&TCTR                           04340000
         LCLB  &GOODEND                                                 04350000
         LCLC  &DNAM(10),&VARNAM,&TLABEL                                04360000
&NMACRO  SETA  &NMACRO+1           INCREMENT COUNT OF VARDEF MACROS     04370000
         AIF   (&NMACRO LE 5).GOOD1                                     04380000
         MNOTE 16,'MORE THAN 5 VARDEF MACROS SPECIFIED, MACRO IGNORED'  04390000
         MEXIT                                                          04400000
.GOOD1   ANOP                                                           04410000
&VARNAM  SETC  ' '                 INITIALIZE USERVAR NAME              04420000
&LCTR    SETA     1                                                     04430000
.LOOP1   ANOP                                                           04440000
&DNAM(&LCTR) SETC 'NULLAPPL'       INITIALIZE APPLID                    04450000
&DPCT(&LCTR) SETA 100              INITIALIZE DIST PCT TO 100           04460000
&LCTR    SETA  &LCTR+1             INCREMENT LOOP COUNTER               04470000
         AIF   (&LCTR LE 10).LOOP1 LOOP                                 04480000
         AIF   (T'&NAME NE '0').GOOD2                                   04490000
         MNOTE 16,'USERVAR NAME PARAMETER MUST BE SPECIFIED'            04500000
         MEXIT                                                          04510000
.GOOD2   ANOP                                                           04520000
&VARNAM  SETC  '&NAME'             SET USERVAR NAME                     04530000
         AIF   (T'&P1 NE '0').GOOD3                                     04540000
         MNOTE 16,'2ND PARAMETER NOT SPECIFIED'                         04550000
         MEXIT                                                          04560000
.GOOD3   ANOP                                                           04570000
         AIF   (N'&P1 EQ 2).GOOD4                                       04580000
         MNOTE 16,'2ND PARAMETER MUST HAVE TWO VALUES SPECIFIED'        04590000
         MEXIT                                                          04600000
.GOOD4   ANOP                                                           04610000
&DNAM(1) SETC '&P1(1)'             SET APPLID                           04620000
&DPCT(1) SETA &P1(2)               SET PERCENTAGE                       04630000
         AIF   (T'&P2 EQ '0').GENERAT                                   04640000
&DNAM(2) SETC '&P2(1)'             SET APPLID                           04650000
&DPCT(2) SETA &P2(2)               SET PERCENTAGE                       04660000
         AIF   (T'&P3 EQ '0').GENERAT                                   04670000
&DNAM(3) SETC '&P3(1)'             SET APPLID                           04680000
&DPCT(3) SETA &P3(2)               SET PERCENTAGE                       04690000
         AIF   (T'&P4 EQ '0').GENERAT                                   04700000
&DNAM(4) SETC '&P4(1)'             SET APPLID                           04710000
&DPCT(4) SETA &P4(2)               SET PERCENTAGE                       04720000
         AIF   (T'&P5 EQ '0').GENERAT                                   04730000
&DNAM(5) SETC '&P5(1)'             SET APPLID                           04740000
&DPCT(5) SETA &P5(2)               SET PERCENTAGE                       04750000
         AIF   (T'&P6 EQ '0').GENERAT                                   04760000
&DNAM(6) SETC '&P6(1)'             SET APPLID                           04770000
&DPCT(6) SETA &P6(2)               SET PERCENTAGE                       04780000
         AIF   (T'&P7 EQ '0').GENERAT                                   04790000
```

```
&DNAM(7) SETC '&P7(1)'             SET APPLID                          04800000
&DPCT(7) SETA &P7(2)               SET PERCENTAGE                      04810000
         AIF   (T'&P8 EQ 'O').GENERAT                                  04820000
&DNAM(8) SETC '&P8(1)'             SET APPLID                          04830000
&DPCT(8) SETA &P8(2)               SET PERCENTAGE                      04840000
         AIF   (T'&P9 EQ 'O').GENERAT                                  04850000
&DNAM(9) SETC '&P9(1)'             SET APPLID                          04860000
&DPCT(9) SETA &P9(2)               SET PERCENTAGE                      04870000
         AIF   (T'&P10 EQ 'O').GENERAT                                 04880000
&DNAM(10) SETC '&P10(1)'           SET APPLID                          04890000
&DPCT(10) SETA &P10(2)             SET PERCENTAGE                      04900000
.GENERAT ANOP                                                          04910000
&LCTR SETA     1                   INITIALIZE LOOP COUNTER             04920000
&HIGHVAL SETA  0                                                       04930000
.LOOP2   ANOP                                                          04940000
         AIF   (&DPCT(&LCTR) GE &HIGHVAL).GOOD5                        04950000
         MNOTE 16,'DISTRIBUTION PERCENTAGES ARE NOT INCREASING'        04960000
         MEXIT                                                         04970000
.GOOD5   ANOP                                                          04980000
&HIGHVAL SETA  &DPCT(&LCTR)        SET NEW HIGH VALUE                  04990000
         AIF   ((&DPCT(&LCTR) GE 0) AND                               X05000000
               (&DPCT(&LCTR) LE 100)).GOOD6                            05010000
         MNOTE 16,'DISTRIBUTION PERCENTAGES NOT BETWEEN 0 AND 100'     05020000
         MEXIT                                                         05030000
.GOOD6   ANOP                                                          05040000
         AIF   ((&DPCT(&LCTR) NE 100) OR                              X05050000
               ('&DNAM(&LCTR)' EQ 'NULLAPPL')).GOOD7                   05060000
&GOODEND SETB  1                                                       05070000
.GOOD7   ANOP                                                          05080000
&LCTR    SETA  &LCTR+1             INCREMENT LOOP COUNTER              05090000
         AIF   (&LCTR LE 10).LOOP2 LOOP                                05100000
         AIF   (&GOODEND).GOOD8                                        05110000
         MNOTE 16,'100 MUST BE SPECIFIED FOR AT LEAST ONE ENTRY'       05120000
         MEXIT                                                         05130000
.GOOD8   ANOP                                                          05140000
         PUSH  PRINT                                                   05150000
         PRINT GEN                                                     05160000
&TLABEL  SETC  'USRVTBL'.'&NMACRO' BUILD LABEL NAME                    05170000
&TLABEL  DS    0F                                                      05180000
         DC    CL8'&VARNAM'        USERVAR NAME TO BE CONTROLLED       05190000
&TLABEL  SETC  'DISTTBL'.'&NMACRO' BUILD LABEL NAME                    05200000
         DC    A(&TLABEL)          ADDR OF DIST TABLE                  05210000
&TCTR    SETA  &NMACRO+1                                               05220000
&TLABEL  SETC  'USRVTBL'.'&TCTR'   BUILD LABEL NAME                    05230000
         AIF   ('&LAST' EQ 'YES').GOOD9                                05240000
         DC    A(&TLABEL)          ADDR OF NEXT USERVAR ENTRY          05250000
         AGO   .GOODA                                                  05260000
.GOOD9   ANOP                                                          05270000
         DC    A(0)                ADDR OF NEXT USERVAR ENTRY          05280000
.GOODA   ANOP                                                          05290000
&TLABEL  SETC  'DISTTBL'.'&NMACRO' BUILD LABEL NAME                    05300000
&LCTR    SETA  1                                                       05310000
.LOOP3   ANOP                                                          05320000
         AIF   (&LCTR EQ 1).GOODB                                      05330000
&TLABEL  SETC  ' '                                                     05340000
.GOODB   ANOP                                                          05350000
&TLABEL  DC    F'&DPCT(&LCTR)'     CUMMULATIVE DIST PERCENTAGE         05360000
         DC    CL8'&DNAM(&LCTR)'   APPLID                              05370000
&LCTR    SETA  &LCTR+1             INCREMENT LOOP CTR                  05380000
         AIF   (&LCTR LE 10).LOOP3                                     05390000
```

```
         POP   PRINT                                                    05400000
         MEND                                                           05410000
         TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX05420000
               TIPLE IMS SYSTEMS - TABLE DEFINITIONS'                   05430000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 05440000
*                                                                       * 05450000
*  THE VARDEF MACRO IS USED TO SPECIFY THE USERVARS THAT THIS EXIT IS * 05460000
*  TO CONTROL AND HOW THE USERVAR TRANSLATIONS ARE TO BE HANDLED      * 05470000
*  (REAL APPLIDS AND DISTRIBUTION).  UP TO 5 DIFFERENT VARDEF MACROS  * 05480000
*  CAN BE SPECIFIED.  THE VARDEF MACROS SHOULD BE CONTIGUOUS (NOT     * 05490000
*  ABSOLUTELY REQUIRED).  THE VARDEF PARAMETER CONTAINS UP TO 11      * 05500000
*  POSITIONAL PARAMETERS AND ONE KEYWORD PARAMETER (LAST=YES - MUST   * 05510000
*  BE SPECIFIED ON THE LAST VARDEF MACRO).  THE FORMAT OF VARDEF      * 05520000
*  MACRO IS AS FOLLOWS:                                               * 05530000
*                                                                     * 05540000
*      VARDEF NAME,D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,LAST=YES            * 05550000
*                                                                     * 05560000
*  WHERE                                                              * 05570000
*                                                                     * 05580000
*    NAME:  USERVAR NAME TO BE TRANSLATED (REQUIRED)                  * 05590000
*                                                                     * 05600000
*    D1 THROUGH D10:  APPLID AND CUMULATIVE PERCENTAGE, ENCLOSED IN   * 05610000
*                     PARENTHESES.  D1 IS REQUIRED.  D2 THROUGH D10   * 05620000
*                     ARE OPTIONAL.  THE PERCENTAGE CAN RANGE FROM 0  * 05630000
*                     TO 100.  CUMULATIVE PERCENTAGES MUST INCREASE   * 05640000
*                     AND THE LAST PARAMETER SPECIFIED MUST HAVE A    * 05650000
*                     CUMULATIVE PERCENTAGE OF 100.                   * 05660000
*                                                                     * 05670000
*                                                                     * 05680000
*    LAST=YES:  MUST (AND ONLY SHOULD) BE SPECIFIED FOR THE LAST      * 05690000
*               VARDEF MACRO                                          * 05700000
*                                                                     * 05710000
*  EXAMPLES:                                                          * 05720000
*                                                                     * 05730000
*  1.  ASSUME THAT LOGON REQUESTS AGAINST USERVAR IMSPLEX1 ARE TO BE  * 05740000
*      DISTRIBUTED 50 PERCENT TO IMS1 AND 50 PERCENT TO IMS2.  ONLY   * 05750000
*      ONE USERVAR IS TO BE MANAGED BY THIS EXIT.  THE APPROPRIATE    * 05760000
*      VARDEF WOULD BE:                                               * 05770000
*                                                                     * 05780000
*       VARDEF IMSPLEX1,(50,IMS1),(100,IMS2)                          * 05790000
*                                                                     * 05800000
*  2.  ASSUME THAT LOGON REQUESTS AGAINST USERVAR IMSPLEX1 ARE TO BE  * 05810000
*      DISTRIBUTED 40 PERCENT TO IMS1, 30 PERCENT TO IMS2, 20 PERCENT * 05820000
*      TO IMS3, AND 10 PERCENT TO IMS4.  ONLY ONE USERVAR IS TO BE    * 05830000
*      MANAGED BY THIS EXIT.  THE VARDEF MACRO REQUIRED TO IMPLEMENT  * 05840000
*      THIS DISTRIBUTION COULD BE CODED MANY WAYS.  AMONG THE MANY    * 05850000
*      WAYS ARE:                                                      * 05860000
*                                                                     * 05870000
*       VARDEF IMSPLEX1,(40,IMS1),(70,IMS2),(90,IMS3),(100,IMS4)      * 05880000
*       VARDEF IMSPLEX1,(10,IMS4),(30,IMS3),(60,IMS2),(100,IMS1)      * 05890000
*       VARDEF IMSPLEX1,(20,IMS3),(60,IMS1),(70,IMS4),(100,IMS2)      * 05900000
*                                                                     * 05910000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 05920000

         SPACE 1                                                        05930000
         VARDEF IMSPLEX,(IMS1,50),(IMS2,100)                            05940000
         SPACE 1                                                        05950000
         VARDEF IMSPLEX2,(IMS100,40),(IMS200,70),(IMS300,90),          X05960000
               (IMS400,100),LAST=YES                                    05970000
         SPACE 1                                                        05980000
```

```
WRKSPCLN DC    A((USRVMAX*USRVLEN)+(DISTMAX*DISTLEN*USRVMAX)+WORKLEN)   05990000
WORKLEN  EQU   WSLEN+DISTMAX*L'WSDIST                                   06000000
HIOFF    DC    X'7FFFFFFF'                                              06010000
HEX04    DC    H'04'              HEX 04                                06020000
HEX10    DC    H'16'              HEX 10                                06030000
HEX18    DC    H'24'              HEX 18                                06040000
HEX20    DC    H'32'              HEX 20                                06050000
HEX40    DC    H'64'              HEX 40                                06060000
         SPACE 1                                                        06070000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 06080000
*                                                                       * 06090000
*  THE VALUE OF VERSION SHOULD BE CHANGED WHEN THE VERSION OF THIS     * 06100000
*  EXIT IS CHANGED.  IT SHOULD AGREE WITH THE EYECATCHER IN THE        * 06110000
*  SAVE MACRO.  THE VERSION WILL BE DISPLAYED IN THE VTAM DISPLAY      * 06120000
*  EXIT COMMAND.  USING UNIQUE VERSION AND EYECATCHER VALUES WILL      * 06130000
*  AVOID PROBLEM DETERMINATION CONFUSION.                              * 06140000
*                                                                       * 06150000
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 06160000
         SPACE 1                                                        06170000
VERSION  DC    CL4'V001'          VERSION ID                            06180000
         SPACE 1                                                        06190000
TRTABLE  DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'00' - X'0F'      06200000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'10' - X'1F'      06210000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'20' - X'2F'      06220000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'30' - X'3F'      06230000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'40' - X'4F'      06240000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'50' - X'5F'      06250000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'60' - X'6F'      06260000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'70' - X'7F'      06270000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'80' - X'8F'      06280000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'90' - X'9F'      06290000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'A0' - X'AF'      06300000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'B0' - X'BF'      06310000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'C0' - X'CF'      06320000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'D0' - X'DF'      06330000
         DC    X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'  X'E0' - X'EF'      06340000
         DC    X'00000000000000000000FFFFFFFFFFFF'  X'F0' - X'FF'      06350000
         SPACE 1                                                        06360000
         LTORG                                                          06370000
         TITLE 'SAMPLE USERVAR EXIT TO DISTRIBUTE LOGON REQUESTS TO MULX06380000
               TIPLE IMS SYSTEMS - DSECT DEFINITIONS'                   06390000
USRVDSCT DSECT ,                USERVAR NAME ENTRY (5 OCCURRENCES)     06400000
USRVNAME DS    CL8              USERVAR NAME                           06410000
USRVDIST DS    A                ADDR OF DISTRIBUTION TABLE             06420000
USRVNEXT DS    A                ADDR OF NEXT ENTRY                     06430000
USRVLEN  EQU   *-USRVDSCT       LENGTH OF USERVAR NAME ENTRY           06440000
USRVMAX  EQU   5               MAXIMUM NUMBER OF USERVAR ENTRIES   06450000
         SPACE 1                                                        06460000
DISTDSCT DSECT ,                USERVAR DIST TABLE (10 OCCURRENCES)    06470000
DISTHIGH DS    F                HIGH RANGE FOR APPLID                  06480000
DISTAPPL DS    CL8              APPLID TO ASSIGN                       06490000
DISTLEN  EQU   *-DISTDSCT       LENGTH OF DIST TABLE ENTRY             06500000
DISTMAX  EQU   10               MAXIMUM NUMBER OF DIST ENTRIES         06510000
         SPACE 1                                                        06520000
WORKSTOR DSECT ,                WORKING STORAGE (USER DATA PROC)       06530000
WSDOUBLE DS    D                DOUBLE WORD WORK AREA                  06540000
WSSWITCH DS    XL1              SWITCHES                               06550000
WS100FND EQU   X'01'            VALUE OF 100 FOUND                     06560000
         DS    XL1              FILLER                                 06570000
WSOLDHI  DS    H                HALF WORD WORK AREA                    06580000
```

```
WSUSRVAR DS    CL8                 USER DATA USERVAR NAME              06590000
WSLEN    EQU   *-WORKSTOR          LENGTH OF WS BASE SECTION           06600000
WSDIST   DS    OF                  USER DATA DISTRIBUTION ENTRY        06610000
*                                  (DISTMAX ENTRIES)                   06620000
         SPACE 1                                                        06630000
PARM1    DSECT ,                   PARM LIST DEFINITION FOR ACTIVATE,  06640000
*                                  DEACTIVATE, AND REPLACE             06650000
USERADDR DS    A                   ADDR OF USER FIELD                  06660000
         DS    XL8                 RESERVED                            06670000
PARMADDR DS    A                   ADDR OF USER PARMS                  06680000
         SPACE 1                                                        06690000
USERFLD  DSECT ,                   USER FIELD DSECT                    06700000
STORADDR DS    A                   ADDR OF OBTAINED STORAGE            06710000
VERID    DS    CL4                 EXIT VERSION ID                     06720000
         SPACE 1                                                        06730000
PARM2    DSECT ,                   PARM LIST DEFINITION FOR INVOKE     06740000
USERADD1 DS    A                   ADDR OF USER FIELD                  06750000
         DS    XL8                 RESERVED                            06760000
INVOKEA  DS    A                   ADDR OF INVOKE FLAGS                06770000
USRVPRMS DS    A                   ADDR OF USERVAR PARMS               06780000
         SPACE 1                                                        06790000
USRVPARM DSECT ,                   USERVAR PARMS                       06800000
OLUNAME  DS    CL8                 OLU NAME                            06810000
OLUADR   DS    XL4                 OLU SUBAREA ADDR                    06820000
SESSADDR DS    A                   ADDR OF SESSION PARTNERS TABLE      06830000
COSNAME  DS    CL8                 COS NAME FOR SESSION                06840000
GENERIC  DS    CL8                 USERVAR NAME                        06850000
FLAGS    DS    XL1                 FLAGS                               06860000
TRANSLAT EQU   X'20'               USERVAR TRANSLATED                  06870000
VOLATILE EQU   X'18'               VOLATILE USERVAR                    06880000
         DS    XL3                 RESERVED                            06890000
TRANVAL  DS    CL8                 TRANSLATED USERVAR VALUE            06900000
         DS    CL8                 RESERVED                            06910000
         END                                                            06920000
```

Abbreviations and acronyms

ACB	application control block
AGN	application group name
AOI	automated operator interface
APAR	authorized program analysis report
APF	authorized program facility
APPC	advanced program to program communication
ARLN	automatic RECON loss notification
ARM	Automatic Restart Manager
AWE	Asynchronous Work Element
BMP	batch message program
BPE	Base Primitive Environment
BSDS	boot strap data set
CBPDO	custom built product delivery offering
CI	control interval
CICS	Customer Information Control System
CF	Coupling Facility
CQS	Common Queue Server
CSA	common system area
CSI	consolidated software inventory
CSL	Common Service Layer
CST	Consolidated Service Test
CTC	channel-to-channel
DASD	direct access storage device
DB/DC	database/data communications
DB2	DATABASE 2
DBA	database administrator
DBCTL	database control
DBD	database description
DBDS	database data set
DBRC	database recovery control
DEDB	data entry database
DL/I	Data Language/I
DLI/SAS	DL/I separate address space
DLT	database level tracking (RSR)
DRA	database resource adapter
ECSA	extended common system area
EMCS	extended multiple consoles support
EMEA	Europe, Middle East and Africa
ESAF	external subsystem attach facility
ESCON®	Enterprise System Connection
ESO	Extended Service Offering
ETO	Extended Terminal Option
EX	execution (IVP)
FDBR	Fast Database Recovery
FICON™	Fiber Connection
FMID	function modification identifier
FT	file tailoring (IVP)
FTP	File Transfer Protocol
GRG	generic resource group
GSAM	generalized sequential access method
HALDB	High Availability Large Database
HFS	hierarchical file system
HLQ	high-level qualifier
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines Corporation
IFP	IMS Fast Path program
ILS	Isolated Log Sender
IMS	Information Management System
IPCS	Interactive Problem Control System
IPL	initial program load
IRLM	Internal Resource Lock Manager
ISC	intersystem communication
ISD	independent software delivery
ISPF	Interactive Systems Productivity Facility
ITSO	International Technical Support Organization
IVP	installation verification program
J2C	J2EE Connector Architecture
J2EE	Java 2 Platform, Enterprise Edition
JBP	Java batch processing region
JCL	job control language
JDBC	Java database connectivity
JDK	Java Development Kit
JES	job entry subsystem (JES2 or JES3)

JMP	Java message processing region
JVM	Java Virtual Machine
KSDS	key sequenced data set
LE	Language Environment
LMOD	load module
LPA	link pack area
LPAR	logical partition
LTERM	logical terminal
LU	logical unit
LU2	Logical Unit 2
MCS	multiple consoles support
MFS	message format services
MLPA	modifiable link pack area
MNPS	Multi-Node Persistent Sessions
MOD	message output descriptor (MFS)
MOD	module (SMP/E)
MPP	message processing program
MPR	message processing region
MSC	multiple systems coupling
MSDB	Main Storage Database
MTO	master terminal operator
MVS	Multiple Virtual System
N/A	not applicable
ODBA	open database access
OLDS	online log data set
OM	Operations Manager
ORS	Online Recovery Service
OSAM	overflow sequential access method
OTMA	open transaction manager access
OTMA C/I	OTMA callable interface
PCB	program communication block
PDS	partitioned data set
PDSE	partitioned data set extended
PIA	package input adapter
PIC	Product Introduction Centre
PMR	problem management record
PPT	program properties table
PSB	program specification block
PSP	preventive service planning
PST	program specification table
PTF	program temporary fix
QPP	Quality Partnership Program
RACF	Resource Access Control Facility
RDS	restart data set
RIM	related installation material
RLDS	recovery log data set
RLT	recovery level tracking (RSR)
RM	Resource Manager
RMF	Resource Measurement Facility
RNR	Rapid Network Recovery
RRS	Resource Recovery Service
RSR	Remote Site Recovery
RSU	recommended service upgrade
SAF	Security Authorization Facility
SCI	structured call interface
SDM	System Data Mover
SDSF	spool display and search facility
SLDS	system log data set
SMP/E	System Modification Program/Extended
SMQ	shared message queues
SMTO	secondary master terminal operator
SMU	security maintenance utility
SPOC	single point of control
SRDS	structure recovery data set
SSA	sub-system alias
STSN	set and test sequence numbers
SVC	supervisor call
SVL	Silicon Valley Laboratories
TCB	task control block
TCO	time controlled operations
TCP/IP	Transmission Control Protocol/Internet Protocol
TMS	Transport Manager System
TPNS	Teleprocessing Network Simulator
TSO	Time Sharing Option
USS	Unix System Services
USS	unformatted system services (SNA)
VG	variable gathering (IVP)
VOLSER	volume serial (number)
VSAM	virtual storage access method
VSCR	Virtual Storage Constraint Relief
VSO	Virtual Storage Option (DEDB VSO)
VTAM	virtual telecommunication access method
WADS	write ahead data set
WAS	WebSphere Application Server
WWW	World Wide Web

XML	eXtensible Markup Language
XRC	eXtended Remote Copy
XRF	eXtended Recovery Facility
WLM	Workload Manager

Related publications

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

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 362. Note that some of these documents referenced here may be available in softcopy only.

- *IMS Primer,* SG24-5352
- *IMS/ESA V6 Parallel Sysplex Migration Planning Guide for IMS TM and DBCTL,* SG24-5461
- *IMS Version 7 High Availability Large Database Guide,* SG24-5751
- *IMS Version 7 Release Guide,* SG24-5753
- *A DBA’s View of IMS Online Recovery Service,* SG24-6112
- *IMS Version 7 Performance Monitoring and Tuning Update,* SG24-6404
- *IMS e-business Connectors: A Guide to IMS Connectivity,* SG24-6514
- *IMS Version 7 Java Update,* SG24-6536
- *Ensuring IMS Data Integrity Using IMS Tools,* SG24-6533
- *IMS Installation and Maintenance Processes*, SG24-6574
- *IMS Version 8 Implementation Guide - A Technical Introduction of the New Features,* SG24-6594
- *IMS DataPropagator Implementation Guide,* SG24-6838
- *Using IMS Data Management Tools for Fast Path Databases,* SG24-6866
- *IMS in the Parallel Sysplex, Volume I: Reviewing the IMSplex Technology,* SG24-6908
- *IMS in the Parallel Sysplex, Volume II: Planning the IMSplex,* SG24-6928

Other publications

These publications are also relevant as further information sources:

- *IMS Version 8: Base Primitive Environment Guide and Reference,* SC27-1290
- *IMS Version 8: Command Reference,* SC27-1291
- *IMS Version 8: Release Planning Guide,* GC27-1305
- *Parallel Sysplex Overview: Introducing Data Sharing and Parallelism in a Sysplex,* SA22-7661
- *OS/390 V1R3.0 OS/390 System Commands*, GC28-1781
- *DFSMS/MVS V1R5 DFSMSdss Storage Administration Guide*, SC26-4930
- *DFSMS/MVS V1R5 DFSMSdss Storage Administration Reference*, SC26-4929
- *IMS Connect Guide and Reference*, SC27-0946

- *IMS Queue Control Facility for z/OS User's Guide,* SC26-9685
- *MVS/DFP V3 R3 Access Method Services for Integrated Catalog Facility*, SC26-4562
- *IMS Installation Volume 2: System Definition and Tailoring*, GC27-1298
- *z/OS MVS Setting Up a Sysplex,* SA22-7625
- *IMS Version 8: Customization Guide,* SC27-1294
- *IMS Version 8 DBRC Guide and Reference*, SC27-1295

Online resources

These Web sites and URLs are also relevant as further information sources:

- IMS Web site

  http://www.ibm.com/ims
- CF sizer tool

  http://www.ibm.com/servers/eserver/zseries/cfsizer
- IBM publications

  http://ehone.ibm.com/public/applications/publications/cgibin/pbi.cgi
- WebSphere Edge Server

  http://www-4.ibm.com/software/network/dispatcher

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs at this Web site:

ibm.com/redbooks

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

IMS in the Parallel Sysplex, Volume III: IMSplex Implementation and Operations

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

IMS in the Parallel Sysplex

Volume III: IMSplex Implementation and Operations

Implement and operate IMS data sharing

Implement and operate IMS shared message queues

Implement and operate the new IMS CSL features

This IBM Redbook is the third volume of a series of redbooks called *IMS in the Parallel Sysplex*. These redbooks describe how IMS exploits the Parallel Sysplex functions and how to plan for, implement, and operate IMS systems working together in a Parallel Sysplex.

This redbook encompasses two parts. Part 1, *Implementation*, is a description of the tasks an IMS installation must execute to enable each of the functions. Part 2, *Operations*, addresses the operational issues associated with running an IMSplex that include startup and shutdown, steady state operations, online change, and recovery from failure.

Both of the parts are divided into chapters addressing the sysplex services exploited by IMS to support block level data sharing, connecting to the IMSplex, shared queues, and the systems management functions enabled with the Common Service Layer introduced in IMS Version 8 — operations management and resource management. Additional topics are addressed in appendices or incorporated within these major topics.

The other volumes in this series are:

- *IMS in Parallel Sysplex, Volume I: Reviewing the IMSplex Technology,* SG24-6908
- *IMS in the Parallel Sysplex, Volume II: Planning the IMSplex,* SG24-6928

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6929-00 ISBN 0738453455