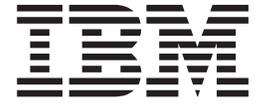


OS/390



MVS Setting Up a Sysplex

OS/390



MVS Setting Up a Sysplex

Note

Before using this information and the product it supports, be sure to read the general information under "Appendix G. Notices" on page 353.

Tenth Edition, September 2000

This is a major revision of GC28-1779-08.

This edition applies to Version 2 Release 10 of OS/390 (5647-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)

Internet e-mail: mhvrdfs@us.ibm.com

World Wide Web: <http://www.ibm.com/s390/os390/webqs.html>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

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

© Copyright International Business Machines Corporation 1988, 2000. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
Tables	xi
About This Book	xiii
Who Should Use This Book	xiii
How to Use This Book	xiii
Where to Find More Information	xiv
Notes on Terminology	xv
Summary of Changes	xvii
Chapter 1. Introduction	1
Characteristics of a Sysplex	1
Recommendations for Redundancy	3
Coupling Facility and the Sysplex	3
Web-Based Assistants for Setting Up a Sysplex	4
Defining Sysplex Parameters and Policies	4
SYS1.PARMLIB Parameters and Members	4
Policies	5
Using Signaling for Sysplex Communications	6
Tuning the Signaling Services and Coupling Facility	7
Installing Systems in a Sysplex	7
Using the Year 2000 Sysplex Test Datesource LPAR Enhancement	8
Sharing a Sysplex Timer Among LPARs	8
Chapter 2. Planning Parmlib Members for a Sysplex	9
Planning the IEASYSxx Member in SYS1.PARMLIB	9
Considerations for the SYSNAME System Parameter	9
Planning the PLEXCFG System Parameter	9
Planning the GRS System Parameter	12
Planning the GRSCNF System Parameter	14
Planning the GRSRNL System Parameter	15
Planning the CLOCK System Parameter	15
Planning the COUPLE System Parameter	19
Planning the COUPLExx Member in SYS1.PARMLIB	19
The COUPLE Statement	20
CLASSDEF, PATHIN, PATHOUT, and LOCALMSG Statements	21
The DATA Statement	21
Modifying COUPLExx Values	21
Planning the CONSOLxx Member in SYS1.PARMLIB	21
Naming Your Consoles	22
Understanding Message Routing	22
Message Recovery in a Sysplex	22
Related SYS1.PARMLB Members to Consider	22
Planning the CLOCKxx Member in SYS1.PARMLIB	23
Planning the XCFPOLxx Member in SYS1.PARMLIB	23
Planning the IEASYMxx Member in SYS1.PARMLIB	23
Specifying the SPINRCVY Statement in the EXSPAT Parmlib Member	24
Chapter 3. Planning the Couple Data Sets	25
Considerations for All Couple Data Sets	25
Considerations for a Sysplex Couple Data Set	30

Formatting a Sysplex Couple Data Set	30
Defining a Sysplex Couple Data Set to MVS	32
Connectivity Requirements for a Sysplex Couple Data Set	32
Considerations for Policy Couple Data Sets	34
Formatting Couple Data Sets.	34
Defining Policy Couple Data Sets to MVS	34
Defining and Activating Policies	35
Updating Policies in a Sysplex	35
Connectivity Requirements for Policy Couple Data Sets	36
Chapter 4. Managing Coupling Facility Resources	37
Planning for a Coupling Facility	37
Defining a Coupling Facility	38
Managing a Coupling Facility.	39
Planning a Coupling Facility Policy	41
Identifying the Coupling Facility	41
Identifying the Coupling Facility Structures	42
Starting a CFRM Policy.	54
Updating a CFRM Policy	54
Summarizing CFRM Policy Setup Requirements	55
The Role of CFRM in a Sysplex	56
Cleaning Up Status Data	56
Performing System Level Initialization	57
Reconciling CFRM Policy Data	57
Transitioning to a New CFRM Policy	58
Resource Allocation in the Coupling Facility	59
Coupling Facility Storage Considerations	59
Coupling Facility Resource Allocation “Rules”.	59
Understanding Coupling Facility Use of Processor Resources.	61
Operating in a Coupling Facility Environment	62
Removing a Coupling Facility from a Sysplex.	63
Transferring a Coupling Facility to Another Sysplex	63
Using the POPULATECF Function to Rebuild Coupling Facility Structures	63
Coupling Facility Structures for IBM Products.	66
Chapter 5. Planning Signaling Services in a Sysplex	69
Recommended Steps for Setting Up Signaling	69
Overview of Available Signaling Options	69
Advantages of Implementing Signaling through List Structures	71
Setting Up Outbound and Inbound Signaling Paths	72
Implementing Signaling through CTC Connections	72
Implementing Signaling Paths through Coupling Facility List Structures	72
Handling Signaling Path Failures	75
Planning for Optimum Signaling Performance	76
XCF Groups	77
Planning the Transport Classes	77
Planning the Message Buffer Space	80
Adding and Deleting Signaling Paths	83
Adding a Signaling Path	84
Deleting a Signaling Path	84
Sample Signaling Scenarios	85
Example 1 — Using COUPLExx Defaults	86
Example 2 — Creating a Transport Class for Large Messages	88
Example 3 — Handling a Range of Message Traffic Considerations	90
Example 4 — Creating a Transport Class for a Group	98
Example 5 — Defining Signaling Paths through the Coupling Facility	100

	Example 6 — Signaling through a Coupling Facility and CTC Connections	102
	WSC Flash — Parallel Sysplex Performance: XCF Performance	
	Considerations	106
	Tuning XCF	106
	Transport Classes	106
	Message Buffers	106
	Signaling Paths	110
	Case Study	114
	Chapter 6. Tuning a Sysplex	117
	RMF Reports and the Sysplex	117
	XCF Activity Report	117
	Coupling Facility Activity Report	118
	Tuning the Signaling Service	118
	Tuning the Message Buffer Size (Class Length)	118
	Tuning the Maximum Message Buffer Space	121
	Tuning the Signaling Paths	124
	Tuning the Transport Classes	127
	Capacity Planning	128
	RMF XCF Activity Report — Sample	129
	XCF Activity Report — Usage by System	130
	XCF Activity Report — Usage by Member	130
	XCF Activity Report — Path Statistics	131
	Tuning OS/390 UNIX Performance in a Sysplex	131
	Tuning Coupling Facilities	131
	Service Time	132
	CHNGD Requests	132
	Potential Sources of Delay	134
	Chapter 7. Planning Sysplex Availability and Recovery	139
	Controlling Availability and Recovery through COUPLExx	139
	Planning the Failure Detection Interval and Operator Notification Interval	140
	Planning the Cleanup Interval	141
	Controlling System Availability and Recovery through the SFM Policy	141
	Overview and Requirements of SFM Policy	141
	Planning for a Status Update Missing Condition	142
	Handling Signaling Connectivity Failures	144
	Handling Coupling Facility Connectivity Failures	146
	Planning PR/SM Reconfigurations	146
	Setting Up an SFM Policy	147
	Controlling Availability and Recovery through XCFPOLxx (PR/SM Only)	148
	Implementing XCFPOLxx	148
	Examples	148
	Example 1: Failure Management in a PR/SM Environment — One	
	Processor	149
	Example 2: Failure Management in a PR/SM Environment — Two	
	Processors and a Coupling Facility	150
	Example 3: Failure Management in a PR/SM Environment — One	
	Processor	151
	Example 4: Failure Management with SFM Policy and a Coupling Facility	152
	Controlling Job Availability and Recovery through Automatic Restart	
	Management	154
	Requirements for Participating in Automatic Restart Management	154
	Using Automatic Restart Management	155
	Handling Concurrent System and Couple Data Set Failures	157

Chapter 8. Adding MVS Systems to a Sysplex	159
XCF-Local Mode Configuration	159
Single-System Sysplex Configuration	160
Multisystem Sysplex Configuration without an Existing Global Resource Serialization Complex	161
Multisystem Sysplex Configuration with an Existing Global Resource Serialization Complex	163
The First System IPL	164
The Second System IPL	165
The Third System IPL	166
The Fourth System IPL	166
Multisystem Sysplex Configuration on One Processor under VM	168
Multisystem Sysplex Configuration on PR/SM	168
Multisystem Sysplex Configuration with a Coupling Facility	169
Multisystem Sysplex Configuration of Greater than Eight Systems	170
Removing a System from the Sysplex	171
Multisystem Application Considerations	172
Rejoining a Sysplex.	172
Chapter 9. Planning for System Logger Applications	173
What is System Logger?	173
Coupling Facility Log Stream	174
DASD-Only Log Stream	175
The System Logger Configuration	176
The System Logger Component	177
The LOGR Couple Data Set (LOGR Policy)	178
Log Data on The Coupling Facility	179
Log Data On DASD Log Data Sets	179
Duplexing Log Data.	179
Offloading Log Data to DASD	180
Finding Information for System Logger Applications	184
Finding Information for CICS Log Manager	185
Finding Information for OPERLOG Log Stream	186
Finding Information for Logrec Log Stream	186
Finding Information for APPC/MVS	187
Finding Information for IMS Common Queue Server Log Manager	188
Preparing to Use System Logger Applications	188
Understand the Requirements for System Logger.	190
Plan the System Logger Configuration	191
Determine the Size of Each Coupling Facility Structure.	197
Develop a Naming Convention for System Logger Resources	203
Plan DASD Space for System Logger	206
Managing Log Data: How Much? For How Long?.	213
Define Authorization to System Logger Resources	215
Format the LOGR Couple Data Set and Make it Available to the Sysplex	216
Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy	217
Deleting Log Streams From the LOGR Policy	224
Define the Coupling Facility Structures Attributes in the CFRM Policy Couple Data Set	225
Activate the LOGR Subsystem	226
Deleting Log Data and Log Data Sets	227
Deleting Log Data	227
Deleting Log Data Sets	228
When Is My Log Data or Log Data Set Physically Deleted?	228
Finding and Deleting Orphaned Log Data Sets.	228

Upgrading a Log Stream from DASD-Only to Coupling Facility	229
Plan for a DASD-Only Log Stream Upgrade.	229
Match the MAXBUFSIZE Values for a DASD-only Log Stream Upgrade	229
Staging Data Sets and Upgrading a DASD-Only Log Stream	230
System Logger Recovery	230
Operations Log and Logrec Application Recovery.	230
Recovery Performed for DASD-Only Log Streams	231
When an MVS System Fails	231
When a Sysplex Fails	232
When the System Logger Address Space Fails or Hangs	233
When the Coupling Facility Structure Fails	236
When the Coupling Facility Space for a Log Stream Becomes Full	237
When a Staging Data Set Becomes Full	237
When a Log Stream is Damaged.	238
When DASD Log Data Set Space Fills.	238
When Unrecoverable DASD I/O Errors Occur	239
Appendix A. Format Utility for Couple Data Sets	241
Using the Couple Data Set Format Utility	241
Understanding Couple Data Set Coexistence	242
Coding the Couple Data Set Format Utility	242
Sysplex Parameters for Format Utility	246
Automatic Restart Management Parameters for Format Utility	246
CFRM Parameters for Format Utility	246
LOGR Parameters for Format Utility	248
SFM Parameters for Format Utility	248
OS/390 UNIX Parameters for Format Utility	249
WLM Parameters for Format Utility	249
Return Codes	250
Examples of Using the IXCL1DSU Format Utility	251
Formatting an Automatic Restart Management Couple Data Set	252
Formatting a CFRM Couple Data Set	254
Formatting a LOGR Couple Data Set	255
Formatting an SFM Couple Data Set	256
Formatting an OS/390 UNIX System Services Couple Data Set	257
Formatting a WLM Couple Data Set	259
Sample Output	261
Appendix B. Administrative Data Utility	263
Using the Administrative Data Utility.	263
Authorizing Use of the Utility	263
Coding the Administrative Data Utility	264
Automatic Restart Management Parameters for Administrative Data Utility	267
CFRM Parameters for Administrative Data Utility	275
LOGR Parameters for Administrative Data Utility	281
SFM Parameters for Administrative Data Utility.	297
Return Codes	301
Examples of Using the IXCMIAPU Utility	302
Administrative Data Utility for Automatic Restart Manager Policy Data	302
Administrative Data Utility for CFRM Policy Data	305
Administrative Data Utility for the LOGR Policy.	307
Administrative Data Utility for SFM Policy Data.	307
Appendix C. Calculating Sizes for Signaling Services	309
Calculating the Size of a List Structure for Signaling.	309
Additional Considerations	309

Message Information for Some MVS Groups	311
Calculating Message Buffer Space — An Example	312
Appendix D. Sysplex Configurations	315
Under VM - Physical Configuration of a Sysplex With MVS Guests	315
Controlling the Verification of CPU Identification	316
On PR/SM - Physical Configuration of a Sysplex on LPARs	317
Configuration of a Sysplex on LPAR and Non-LPAR Systems	318
Appendix E. Coupling Facility Guidelines.	321
Understanding Some Basic Concepts	321
Reviewing the Use of Structure Data	321
Reviewing the Concept of Persistence	321
Reviewing the Role of CFRM	323
Guidelines for Planned Outages	325
Adding a Coupling Facility to an Active Sysplex (MVS Image Side)	325
Adding a Coupling Facility to an Active Sysplex (Coupling Facility Image Side)	326
Planning for a Coupling Facility Shutdown	326
Removing a Coupling Facility When Multiple Coupling Facilities Exist	327
Removing the Only Coupling Facility in the Sysplex	329
Considerations for Removing an ICMF Partition	330
Replacing or Moving Coupling Facilities in an Active Sysplex	331
Actions to Remove a Structure from a Coupling Facility	331
Restoring the Coupling Facility to the Sysplex	338
Structure Rebuild Recovery	338
Guidelines for Unplanned Outages	340
Guidelines for Recovering from Coupling Facility Failures	341
Handling Failing or Disconnecting Connector States	344
Appendix F. Defining an IEFAUTOS Structure for Automatic Tape Switching	349
Recommendations for Enhancing the Availability of Automatic Tape Switching	349
Determining the Size of an IEFAUTOS Structure	350
Updating SYS1.PARMLIB for Automatic Tape Switching	351
Appendix G. Notices	353
Trademarks.	354
Glossary	357
Sources of Terms and Definitions.	357
Index	361

Figures

1. Sysplex with Coupling Facility and Couple Data Sets	3
2. COUPLExx Parmlib Member	20
3. Signaling on a Four-System Sysplex, Implemented through CTC Connections	70
4. Example of SYSA's PATHIN and PATHOUT Specifications in COUPLExx	70
5. Signaling on a Four-System Sysplex, Implemented through a Coupling Facility List Structure	71
6. Example of SYSW's PATHIN and PATHOUT Specifications in COUPLExx	71
7. Syntax for CLASSDEF Statement of COUPLExx Parmlib Member	78
8. Syntax for PATHIN, PATHOUT, LOCALMSG on COUPLExx Parmlib Member	80
9. SYSA with Inbound, Local and Outbound Message Buffers	82
10. Example 1 — Diagram of PLEXOF2.	87
11. Example 1 — SYSA's COUPLExx Parmlib Member	87
12. Example of SYSB's COUPLExx Parmlib Member	88
13. Example 2 — Diagram of PLEXOF2B	89
14. Example 2 — SYSA's COUPLExx Parmlib Member	89
15. Example 3 — Diagram of PLEXOF3.	92
16. Example of SYSA's COUPLExx Parmlib Member	93
17. Example 4 — Diagram of PLEXOF2A	99
18. Example of SYSA's COUPLExx Parmlib Member	99
19. Example 5 — Diagram of PLEXOF2C	101
20. Example 5 — SYSA's COUPLExx Parmlib Member	101
21. Example 6 — Diagram of PLEXOF2D	104
22. Example of SYSA's COUPLExx Parmlib Member	104
23. XCF Activity Report - Usage by System	130
24. XCF Activity Report - Usage by Member.	130
25. XCF Activity Report - Path Statistics	131
26. Example of RMF CF Structure Report	132
27. Example of RMF CF Structure Report	133
28. Example of RMF CF Usage Report	134
29. Example of RMF CF Subchannel Activity Report.	136
30. Example of RMF CF Subchannel Activity Report.	136
31. Three-System Sysplex with Active SFM Policy	144
32. Signaling Connectivity Failure Between SYSA and SYSB	145
33. Three-System Sysplex on Two Processors with Active SFM Policy	147
34. Failure Management in a PR/SM Environment — One Processor	149
35. Failure Management in a PR/SM Environment — Two Processors and a Coupling Facility	150
36. Failure Management in a PR/SM Environment — One Processor	152
37. Failure Management with SFM Policy and a Coupling Facility	153
38. XCF-Local Mode Configuration	160
39. Configuration with a Single-System Sysplex and Three Independent Systems	161
40. Multisystem Sysplex Configuration	163
41. One System in the Multisystem Sysplex	164
42. Two Systems in the Multisystem Sysplex	165
43. Three Systems in the Multisystem Sysplex	166
44. Four Systems in the Multisystem Sysplex	167
45. Multisystem Sysplex Configuration with a Coupling Facility	170
46. Log Stream Data on the Coupling Facility and DASD	174
47. Log Stream Data in Local Storage Buffers and DASD Log Data Sets	175
48. A Complete Coupling Facility Log Stream Configuration	176
49. A DASD-Only Configuration	177
50. Example: Merging Log Data from Systems Across a Sysplex	194
51. Example: Maintaining Multiple Log Streams in the Same Coupling Facility Structure	195
52. Example: Recovery for a Failing System - All Data Recovered	196
53. Example: Recovery for a Failing System - Recovery Delayed for Some Data	197

54. Example: Using Staging Data Sets When the Coupling Facility Shares a CPC with a System	223
55. Example: Using Staging Data Sets When the Coupling Facility is Volatile	224
56. IXCSYSPF: Sample OS/390 JCL for Formatting Sysplex Couple Data Set	251
57. SAMPXCF: Sample SP Version 4 JCL for Formatting Couple Data Sets	252
58. IXCARMF - JCL for Utility to Format ARM Couple Data Set	253
59. IXCCFRMF - JCL for Utility to Format CFRM Couple Data Set	255
60. JCL for Utility to Format LOGR Couple Data Set	256
61. IXCSFMF - JCL for Utility to Format SFM Couple Data Set	257
62. BPXISCDs: Sample OS/390 JCL for Formatting an OS/390 UNIX Couple Data Set	258
63. Sample JCL to Format WLM Couple Data Set	260
64. Sample Output of Couple Data Set Format Utility	261
65. IXCARMP0 - Administrative Policy Data in ARM Couple Data Set (Part 1)	303
66. IXCARMP0 - Administrative Policy Data in ARM Couple Data Set (Part 2)	304
67. IXCARMP0 - Administrative Policy Data in ARM Couple Data Set (Part 3)	305
68. IXCCFRMP - Administrative Policy Data in CFRM Couple Data Set	306
69. Administrative Inventory Data in LOGR Couple Data Set	307
70. IXCSFMP - Administrative Policy Data in SFM Couple Data Set	308
71. Under VM - Physical Configuration of a Sysplex With MVS Guests	316
72. On PR/SM - Physical Configuration of a Sysplex on LPARs	318
73. Configuration of a Sysplex on LPAR and Non-LPAR Systems	319
74. Message Output from a Rebuild Hang	340
75. Variables Required to Determine a Precise Size of IEFAUTOS	351

Tables

1. Structure Size Allocation in OS/390	44
2. Summary of MVS Commands for Managing a Coupling Facility.	65
3. Coupling Facility Structures in OS/390 for IBM Products	66
4. Sizes for Signaling Structures	74
5. MAXMSG Values on COUPLE Statements	81
6. Automatic Restart Management Parameters for IXCMIAPU and IXCARM Parameter Overrides	156
7. Finding CICS Log Manager Information	185
8. Finding OPERLOG Information	186
9. Finding Logrec Log Stream Information	186
10. Finding APPC/MVS Information	187
11. Finding IMS (CQS) Log Manager Information	188
12. Reviewing System Logger Setup Activities	189
13. Determining the Size of IEFAUTOS	350

About This Book

This book contains information you need to set up an MVS/ESA system to run in a sysplex.

The book assumes that you have a good knowledge of MVS and that you are familiar with the enhanced capabilities of data sharing and parallel processing presented in the OS/390 Parallel Sysplex Library.

Who Should Use This Book

System programmers should use this book for information that they need to plan, install, operate, modify, and tune an MVS system in a sysplex.

How to Use This Book

This book is organized as follows:

Chapter 1. Introduction — describes the characteristics of a sysplex, introduces basic sysplex concepts and provides an overview of the topics covered in this book.

Chapter 2. Planning Parmlib Members for a Sysplex — provides an overview of the SYS1.PARMLIB members that you need to set up a sysplex.

Chapter 3. Planning the Couple Data Sets — describes how to plan couple data sets for the sysplex.

Chapter 4. Managing Coupling Facility Resources — provides an overview of the coupling facility and contains information on planning and setting up a coupling facility resource management policy.

Chapter 5. Planning Signaling Services in a Sysplex — describes MVS signaling to communicate in the sysplex.

Chapter 6. Tuning a Sysplex — contains tuning information for a sysplex.

Chapter 7. Planning Sysplex Availability and Recovery — describes availability and recovery in a sysplex.

Chapter 8. Adding MVS Systems to a Sysplex — illustrates various sysplex configurations.

Chapter 9. Planning for System Logger Applications — contains information for planning a system logger application.

Appendix A. Format Utility for Couple Data Sets — contains the information needed to format couple data sets.

Appendix B. Administrative Data Utility — contains the information needed to define policies.

Appendix C. Calculating Sizes for Signaling Services — contains formulas for calculating the size of coupling facility list structures and message buffers for your sysplex.

Appendix D. Sysplex Configurations — contains diagrams of some typical sysplex configurations.

Appendix E. Coupling Facility Guidelines — contains guidance and procedures for changing the configuration of a coupling facility in a sysplex.

Appendix F. Defining an IEFAUTOS Structure for Automatic Tape Switching — contains information for defining a structure for automatic tape switching.

Appendix G. Notices — contains the Notices and Trademarks list.

This book also contains a glossary.

Where to Find More Information

This book references information in other books, using shortened versions of the book titles. For complete titles and order numbers of the books for OS/390 products, see *OS/390 Information Roadmap*.

The following table lists the additional publications referenced in this book.

Short Title Used in This Book	Title	Order Number
<i>CICS Transaction Server for OS/390 Release Guide</i>	<i>CICS Transaction Server for OS/390 Release Guide</i>	GC33-1570
<i>CICS System Definition Guide</i>	<i>CICS System Definition Guide</i>	SC33-1682
<i>DATABASE2 V4 Data Sharing: Planning and Administration</i>	<i>DB2 for MVS/ESA V4 Data Sharing: Planning and Administration</i>	SC26-3269
<i>DFDSS User's Guide</i>	<i>DFDSS: User's Guide</i>	SC26-4388
<i>ES/9000 and ES/3090 PR/SM Planning Guide</i>	<i>ES/9000 and ES/3090 PR/SM Planning Guide</i>	GA22-7123
<i>PR/SM Planning Guide</i>	<i>Processor Resource/Systems Manager Planning Guide (S/390 processors only)</i>	GA22-7236
<i>Introducing Enterprise Systems Connection</i>	<i>Introducing Enterprise Systems Connection</i>	GA23-0383
<i>IOCP User's Guide</i>	<i>IOCP User's Guide and ESCON Channel-to-Channel Reference</i>	GC38-0401
<i>Planning for the 9037 Sysplex Timer</i>	<i>Planning for the 9037 Sysplex Timer</i>	GA23-0365
<i>Storage Management Subsystem Migration Planning Guide</i>	<i>MVS/ESA SML: Storage Management Subsystem Migration Planning Guide</i>	SC26-4659
<i>System/390 MVS Parallel Sysplex Test Report</i>	<i>S/390 MVS Parallel Sysplex Test Report</i>	GC28-1236
<i>OS/390 MVS Parallel Sysplex Configuration Volume 1: Overview</i>	<i>OS/390 MVS Parallel Sysplex Configuration Volume 1: Overview</i>	SG24-2075
<i>OS/390 MVS Parallel Sysplex Configuration Volume 2: Cookbook</i>	<i>OS/390 MVS Parallel Sysplex Configuration Volume 2: Cookbook</i>	SG24-2076
<i>OS/390 MVS Parallel Sysplex Configuration Volume 3: Connectivity</i>	<i>OS/390 MVS Parallel Sysplex Configuration Volume 3: Connectivity</i>	SG24-2077
<i>3088 Product Description</i>	<i>IBM 3088 Multisystem Channel Communication Unit Product Description</i>	GA22-7081

For sysplex-related performance information, you might also be interested in *System/390 MVS Parallel Sysplex Performance*, SG24-4356.

Notes on Terminology

References in this book to “MVS” refer either to the MVS/ESA product or to the MVS element of OS/390. The OS/390 MVS product is based on MVS/ESA SP Version 5.2. When necessary, other version and release levels of the MVS/ESA product are discussed.

Summary of Changes

Summary of Changes for GC28-1779-09 OS/390 Version 2 Release 10

The book contains information previously presented in GC28-1779-08 which supports OS/390 Version 2 Release 9.

New Information

- Support for the automatic altering of a structure is described. Through this function, the system automatically alters a structure when it reaches an installation-defined or defaulted-to percent full threshold is determined by structure full monitoring.
- The Administrative Data Utility provides new support for the definition of CFRM policies.
 - The MINSIZE parameter specifies the smallest size to which a structure can ever be altered, either as a result of automatic alter processing or any alter process.
 - The ALLOWAUTOALT parameter specifies the installation's request to allow system-initiated alter processing for a structure.
 - The FULLTHRESHOLD parameter specifies a percentage value to be used by automatic alter processing and structure full monitoring.
- Additional information about coupling facility configuration and recovery has been added to "Appendix E. Coupling Facility Guidelines" on page 321.
- Portions of a Washington Systems Center Flash, "Parallel Sysplex Performance: XCF Performance Considerations" have been added to "Chapter 5. Planning Signaling Services in a Sysplex" on page 69.
- Support for the following APARs, which have been rolled into Release 10, is included:
 - OW41959, which changes the default value of REBUILDPERCENT from 100 to 1. This allows MVS to initiate a structure rebuild if ANY system loses connectivity to the coupling facility in which the structure resides (if structure rebuild is supported).
 - OW39921, which provides Automatic Restart Management support for OS/390 Component Broker.
 - OW43778, which clarifies the effects on structure sizing when migrating to a coupling facility of CFLEVEL=8 or higher.
 - OW43278, which provides a new keyword in the COUPLExx parmlib member that specifies the name of the CFRM policy to be started at IPL-time if there is no previously-activated CFRM policy.

This book includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of Changes for GC28-1779-08 OS/390 Version 2 Release 9

The book contains information previously presented in GC28-1779-07 which supports OS/390 Version 2 Release 8.

New Information

- The Couple Data Set Format Utility (IXCL1DSU) provides new support for the definition of more than 255 structures in a coupling facility when formatting CFRM couple data sets. See “CFRM Parameters for Format Utility” on page 246. (This support was previously available as OW19974.)
- Support for monitoring structure utilization is provided, by which the system issues a warning message when a structure full condition is imminent. See “Monitoring Structure Utilization” on page 49.
- The Couple Data Set Format Utility (IXCL1DSU) provides new support for the use of a shared Hierarchical File System (HFS) by OS/390 UNIX System Services.
- Information about tuning a sysplex when a shared HFS is implemented is added to “Chapter 6. Tuning a Sysplex” on page 117. See “Tuning OS/390 UNIX Performance in a Sysplex” on page 131.
- The Parallel Sysplex Configuration Assistant and the Coupling Facility Structure Sizer are described. See “Web-Based Assistants for Setting Up a Sysplex” on page 4.
- Support for the following APAR is included:
 - OW33615

Summary of Changes for GC28-1779-07 OS/390 Version 2 Release 8

The book contains information previously presented in GC28-1779-06 which supports OS/390 Version 2 Release 7.

New Information

- The Couple Data Set Format Utility (IXCL1DSU) provides new support for system-managed processes when formatting CFRM couple data sets. See “CFRM Parameters for Format Utility” on page 246.
- Automatic Restart Manager (ARM) provides a default value for restarting TCP/IP. The value is added to the ARM RESTART_ORDER parameter for the Administrative Data Utility, IXCMIAPU. See “Automatic Restart Management Parameters for Administrative Data Utility” on page 267.
- The JES2 checkpoint list structure now supports system-managed rebuild.
- Support for the following APAR is included:
 - OW30926

Summary of Changes for GC28-1779-06 OS/390 Version 2 Release 7

The book contains information previously presented in GC28-1779-05 which supports OS/390 Version 2 Release 6.

New Information

- Support for the following APARs is included:
 - OW32876, which affects systems running as VM guests in a sysplex. See “Rejoining a Sysplex” on page 172.

- OW34184, which provides additional information about performance implications of using certain XCF and XES macros with regard to the CFRM couple data set.

Changed Information

- “Chapter 6. Tuning a Sysplex” on page 117 has been updated with more recent information about tuning coupling facilities.

Summary of Changes for GC28-1779-05 OS/390 Version 2 Release 6

The book contains information previously presented in GC28-1779-04 which supports OS/390 Version 2 Release 5.

New Information

- Support for a coupling facility at CFLEVEL=5 is included.
- Information about the new POPULATECF function to rebuild coupling facility structures is added.
- Support for the following APARs is included:
 - OW29849, which optimizes the control interval size for DASD log data sets.
 - OW30814, which provides the capability for MVS-initiated structure rebuild when loss of connectivity to a structure occurs and there is not active SFM policy for the sysplex.

Changed Information

- CFRM parameters for the Administrative Data Utility are updated to support user-managed structure duplexing.

Summary of Changes for GC28-1779-04 OS/390 Version 2 Release 5

The book contains information previously presented in GC28-1779-03 which supports OS/390 Version 2 Release 4.

New Information

- Support for the following APARs is included:
 - OW21511, which increases the limit for the number of members in an XCF group to 1023. The previous limit was 511.
 - OW24532, which provides support for a coupling facility at CFLEVEL=4.
 - OW27741, which provides a new optional COUPLExx parmlib member keyword, VMCPUIDTOLERATION(NOIYES). The purpose of this keyword is to control the verification of CPU identification information that is done at XCF initialization when SIMETRID is being used.
 - OW26854, which provides a new system parameter, COUPLE=**. Specifying COUPLE=** causes XCF to initialize the system in an XCF-local environment.

Summary of Changes for GC28-1779-03 OS/390 Version 2 Release 4

The book contains information previously presented in GC28-1779-02 which supports OS/390 Version 1 Release 3.

Changed Information

- “Chapter 9. Planning for System Logger Applications” on page 173 contains changes, including:
 - Planning for DASD-only log streams.
 - Finding information for IMS Common Queue Server Log Manager system logger application.
 - Specifying whether a log stream is DASD-only in the LOGR policy.
 - Upgrading a log stream from DASD-only to a coupling facility based one.
 - Recovery for DASD-only log streams.
- “Appendix B. Administrative Data Utility” on page 263 contains changes, including:
 - New DASDONLY parameter to indicate whether a log stream is a DASD-only or coupling facility based one.
 - New use of the MAXBUFSIZE parameter on the DEFINE LOGSTREAM request to specify the largest log block that can be written to a DASD-only log stream.
 - New information on how system logger determines staging data set size if you do not specify a value on the STG_SIZE parameter.
 - You can now specify the STRUCTNAME parameter on the UPDATE LOGSTREAM request to upgrade a log stream from a DASD-only to a coupling facility based one.

Summary of Changes for GC28-1779-02 as Updated June, 1997 online only for SK2T-6700-05

The following changes appear only in the online version of this publication.

New Information

“Finding Information for System Logger Applications” on page 184 includes information about the APPC/MVS log stream.

“Finding Information for System Logger Applications” on page 184 includes pointers to *OS/390 MVS Programming: Resource Recovery* for related information about resource recovery services (RRS) log streams.

Summary of Changes for GC28-1779-02 OS/390 Version 1 Release 3

The book contains information previously presented in GC28-1779-01 which supports OS/390 Version 1 Release 2.

Changed Information

- “Chapter 9. Planning for System Logger Applications” on page 173 contains changes, including:
 - System logger management of the average log block size.

- Automatic allocation of staging data sets for a coupling facility structure failure followed by a rebuild failure.
- Automatic log data archival and deletion. System logger now allows you to set up a retention period and automatic deletion for managing DASD log data set space and archival.
- Removal of the 168 limit on DASD log data sets.
- Associating a log stream with a resource manager in the LOGR policy.
- Deleting log streams from the LOGR policy.
- “Appendix A. Format Utility for Couple Data Sets” on page 241 contains new parameters for system logger.
- “Appendix B. Administrative Data Utility” on page 263 contains new parameters for system logger.

**Summary of Changes
for GC28-1779-01
OS/390 Version 1 Release 2**

This book contains information previously presented in GC28-1779-00 which supports OS/390 Version 1 Release 1.

The following summarizes the changes to that information.

New Information

- New chapter, Chapter 9. Planning for System Logger Applications. Some of this information was previously presented in *OS/390 MVS Programming: Assembler Services Guide*.
- A new appendix, “Appendix E. Coupling Facility Guidelines” on page 321, is added. The information was previously presented in APAR OW18365.

Changed Information

- Default values for Sysplex and CFRM parameters in the couple data set format utility (IXCL1DSU) are increased.

**Summary of Changes
for GC28-1779-00
OS/390 Version 1 Release 1**

This book contains information previously presented in *MVS/ESA Setting Up a Sysplex*, GC28-1449 which supports MVS/ESA System Product Version 5.

Chapter 1. Introduction

A **sysplex** is a collection of MVS/ESA systems that cooperate, using certain hardware and software products, to process workloads. The products that make up a sysplex cooperate to provide higher availability, easier systems management, and improved growth potential over a conventional computer system of comparable processing power.

The cross system coupling facility (XCF) component of MVS/ESA provides simplified multisystem management. XCF services allow authorized programs on one system to communicate with programs on the same system or on other systems. If a system fails, XCF services also provide the capability for batch jobs and started tasks to be restarted on another eligible system in the sysplex.

In a sysplex, central processing complexes (CPCs) are connected through channel-to-channel (CTC) communications or through a *coupling facility*.

A coupling facility enables parallel processing and improved data sharing for authorized programs running in the sysplex. The cross-system extended services (XES) component of MVS enables applications and subsystems to take advantage of the coupling facility.

A shared sysplex couple data set records status information for the sysplex.

A Sysplex Timer is required to synchronize the time-of-day (TOD) clocks for systems in a sysplex that run on different processors.

For an overview of a sysplex that uses a coupling facility and software products for data sharing, see *OS/390 Parallel Sysplex Overview*.

Characteristics of a Sysplex

A sysplex can include the following software and hardware:

- MVS/ESA SP Version 4 or Version 5 or OS/390

These products include the cross-system coupling facility (XCF) component, which enables authorized programs in a sysplex to communicate with programs on the same MVS system or other MVS systems; and the global resource serialization component, which serializes sysplex resources.

MVS/ESA SP Version 5 and OS/390 provide support for the coupling facility, which provides high speed access to shared data across applications and subsystems running on different MVS systems.

- Signaling paths between MVS systems

There must be at least two operational signaling paths (one inbound and one outbound path) between each of the MVS systems in the sysplex.

The signaling paths can be defined through one or more of the following:

- Coupling facility list structure (on OS/390 or Version 5 systems)
- An ESCON channel operating in CTC mode
- The 3088 Multisystem Channel Communication Unit

- Sysplex couple data set

MVS requires a DASD data set (and an alternate data set is recommended for availability) to be shared by all systems in the sysplex. On the sysplex couple data set, MVS stores information related to the sysplex, systems, XCF groups, and their members. An XCF group is the set of related members that a

Introduction

multisystem application defines to XCF. A multisystem application can be an installation-defined program, an MVS component or subsystem, or a program product.

A sysplex couple data set is always required for a multisystem sysplex and for most single-system sysplexes. However, you can define a single system sysplex that does not require a sysplex couple data set.

- Common time reference

When the sysplex consists of multiple MVS systems running on two or more processors, MVS requires that the processors be connected to the same Sysplex Timer. MVS uses the Sysplex Timer to synchronize TOD clocks across systems.

For a multisystem sysplex defined on a single processor (under PR/SM or VM) the SIMETRID parameter in the CLOCKxx parmlib member must specify the simulated Sysplex Timer identifier to synchronize timings for the MVS systems.

TOD Clocks: In a configuration with more than one CP, each CP may have a separate TOD clock (as in the S/390 Parallel Enterprise G4 and later servers) or more than one CP may share the same clock. To assist in the synchronization of the TOD clocks in a multisystem sysplex, a new architected 128-bit extended time-of-day clock is available. The extended clock format was required to provide the improved resolution necessary for the faster processors as they become available. The extended time-of-day architecture ensures that when an application in a multisystem sysplex environment requests a TOD value, XCF will always return a clock value that is unique across the sysplex, regardless of the number of systems in the sysplex.

To manage certain aspects of your sysplex, you can install one or more additional couple data sets, for:

- Coupling facility resource management (CFRM), to define how the system is to manage coupling facility resources.
- Sysplex failure management (SFM), to define how the system is to manage system and signaling connectivity failures and PR/SM reconfiguration actions.
- Workload management (WLM), to define service goals for workloads.
- Automatic restart management (ARM), to define how to process restarts for started tasks and batch jobs that have registered with automatic restart management.
- System logger (LOGR), to define, update, or delete structure or log stream definitions.
- OS/390 UNIX System Services (OMVS), to contain file information when using a shared Hierarchical File System (HFS) in a sysplex (available with OS/390 Release 9).

Figure 1 on page 3 shows a sysplex of two MVS systems, SYSA and SYSB, where both systems are connected to a Sysplex Timer and a coupling facility and can access the sysplex couple data set and the CFRM couple data set. The sysplex might also be configured so the systems can access one or more of the following couple data sets: ARM (automatic restart management), LOGR, SFM, WLM, or OMVS.

(For diagrams of additional configurations, see “Chapter 8. Adding MVS Systems to a Sysplex” on page 159 and “Appendix D. Sysplex Configurations” on page 315.)

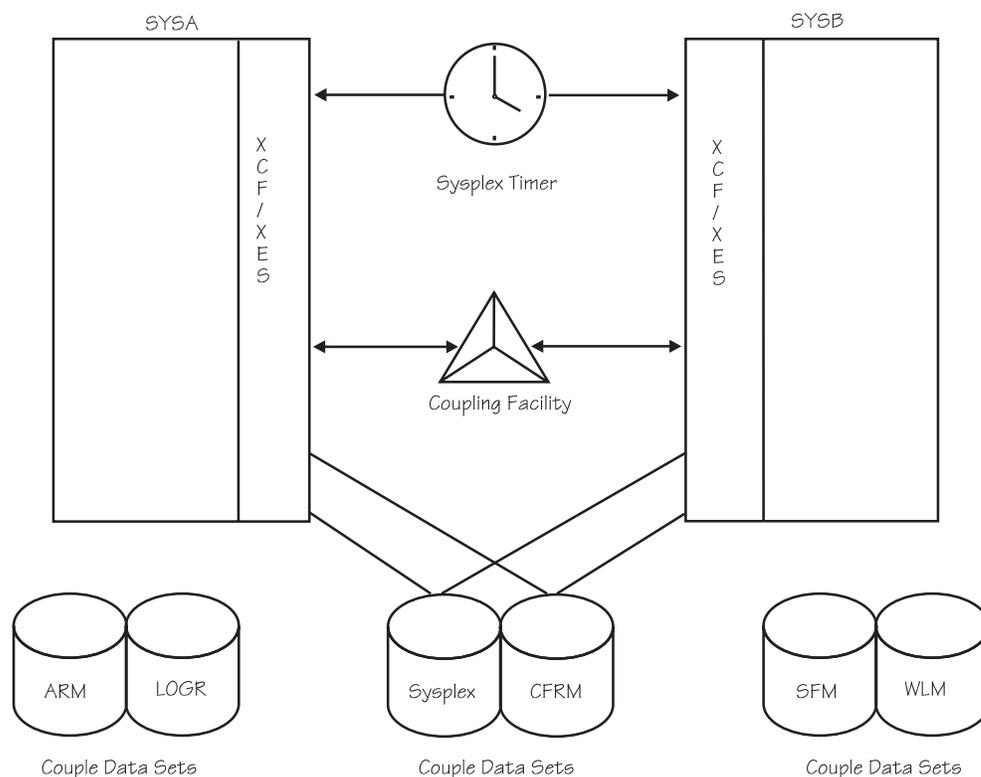


Figure 1. Sysplex with Coupling Facility and Couple Data Sets

Recommendations for Redundancy

IBM strongly recommends that you install redundant hardware in the sysplex, including the hardware for signaling paths, the Sysplex Timer, and the coupling facility. In addition, IBM recommends redundancy for the sysplex couple data set and for any other couple data sets that you use.

Coupling Facility and the Sysplex

Cross-system extended services (XES) uses one or more coupling facilities to satisfy customer requirements for:

- Data sharing across the systems in a sysplex
- Maintaining the integrity and consistency of shared data
- Maintaining the availability of a sysplex.

A coupling facility is a special logical partition on certain ES/9000 model processors and on CPCs in the S/390 CMOS processors. The coupling facility allows software on different systems in the sysplex to share data. To share data, systems must have connectivity to the coupling facility through coupling facility channels.

Systems in the sysplex that are using a coupling facility must also be able to access the coupling facility resource management (CFRM) couple data set.

“Chapter 4. Managing Coupling Facility Resources” on page 37 describes the coupling facility in detail.

Web-Based Assistants for Setting Up a Sysplex

Two web-based assistants are available to ease your migration to a Parallel Sysplex environment.

- The IBM S/390 Parallel Sysplex Configuration Assistant is for system programmers planning to migrate to a Parallel Sysplex environment for the first time, as well as for those who want to verify their Parallel Sysplex setup.
- The IBM S/390 Coupling Facility Structure Sizer simplifies the task of estimating the amount of storage required by the coupling facility structures used in your installation.

See and experiment with both new assistants on the S/390 Parallel Sysplex website at www.ibm.com/s390/psol/.

Defining Sysplex Parameters and Policies

The values you specify in SYS1.PARMLIB largely control the characteristics of systems in a sysplex.

Policies allow MVS to manage specific resources in conformance with your system and resource requirements and with little operator intervention.

SYS1.PARMLIB Parameters and Members

Many of the values that represent the fundamental decisions you make about the systems in your sysplex are in SYS1.PARMLIB.

The key parmlib members you need to consider when setting up an MVS system to run in a sysplex are:

- IEASYSxx is the system parameter list, which holds parameter values that control the initialization of MVS. Some parameters identify other members of SYS1.PARMLIB that are to be used. (For example, the GRSCNF system parameter identifies the GRSCNFxx member.)
- COUPLExx contains parameter values that reflect sysplex-related information.
- GRSCNFxx describes the global resource serialization complex for the system.
- GRSRNLxx specifies the resource name lists to be used to control the serialization of resources in the complex.
- CLOCKxx indicates how the time of day is to be set on the system.
- IEASYMxx provides a single place to define system parameters and system symbols for all systems in a sysplex.
- LOADxx optionally specifies the following:
 - The name of a sysplex in which a system participates, which is also the substitution text for the &SYSPLEX system symbol
 - One or more IEASYMxx parmlib members that specify the system symbols and system parameters to be used.
- XCFPOLxx specifications are used only if a sysplex failure management policy is not active in the sysplex. Functions provided through XCFPOLxx can provide high availability for multisystem applications on an MVS system on PR/SM. If SFM is active in the sysplex, then the SFM policy specifications can define the same PR/SM reconfiguration actions that can be defined through XCFPOLxx.

Sharing Parmlib Members

To prevent the number of parmlib definitions from increasing with each system that is added to a sysplex environment, you can share parmlib members across

systems. If systems require unique values in those parmlib members, you can code them so the different systems can supply unique values in shared parmlib definitions.

System symbols are the elements that allow systems to specify unique values in shared parmlib members. When specified in a parmlib definition that two or more system images are to share, system symbols resolve to unique values on each system.

Additional Information

Background information and considerations for selecting values for parmlib members are presented in “Chapter 2. Planning Parmlib Members for a Sysplex” on page 9 in the following topics:

- “Planning the IEASYSxx Member in SYS1.PARMLIB” on page 9
- “Planning the COUPLExx Member in SYS1.PARMLIB” on page 19
- “Planning the XCFPOLxx Member in SYS1.PARMLIB” on page 23
- “Planning the IEASYMxx Member in SYS1.PARMLIB” on page 23.

Policies

A policy is a set of rules and actions that systems in a sysplex are to follow when using certain MVS services. A policy allows MVS to manage specific resources in compliance with your system and resource requirements, but with little operator intervention. A policy can be set up to govern all systems in the sysplex or only selected systems. You might need to define more than one policy to allow for varying workloads, configurations, or other installation requirements at different times. For example, you might need to define one policy for your prime shift operations and another policy for other times. Although you can define more than one policy of each type (except for system logger) only one policy of each type can be active at a time. For system logger, there is only one LOGR policy in the sysplex.

The following policies can be used to enhance systems management in a sysplex:

- The coupling facility resource management (CFRM) policy allows you to define how MVS is to manage coupling facility resources.
- The sysplex failure management (SFM) policy, allows you to define how MVS is to manage system failures, signaling connectivity failures, and PR/SM reconfiguration actions.
- The workload management (WLM) policy allows you to define service goals for workloads.
- The automatic restart management policy allows you to define how MVS is to manage automatic restarts of started tasks and batch jobs that are registered as elements of automatic restart management.
- The system logger policy, LOGR, allows you to define, update, or delete structure or log stream definitions.

Setting Up Couple Data Sets for Policy Data

Each of these policies resides in a couple data set, which you need to format and define to MVS. The different types of policies can all reside in the same couple data set, or they can reside in separate couple data sets, either on the same volume or on different volumes. To make a policy active on an MVS system, you must define the policy and activate it.

IBM provides a couple data set format utility, IXCL1DSU in SYS1.MIGLIB, which you can use to format the couple data sets. Sample JCL to invoke this utility and an explanation of the information you need to specify when formatting a couple data

Introduction

set is included in “Appendix A. Format Utility for Couple Data Sets” on page 241. “Chapter 3. Planning the Couple Data Sets” on page 25 includes an explanation of the format utility and a discussion of what you need to consider when formatting a couple data set. (The couple data set for WLM policy data can also be formatted interactively using the administrative application. See *OS/390 MVS Planning: Workload Management*.)

Defining Policies

Once the couple data sets for a service are formatted, you can define the policies that are to reside in the data sets. These policies are called administrative policies.

For the CFRM, SFM, LOGR, and automatic restart management administrative policies, IBM supplies an administrative data utility, IXCM2APU, in SYS1.MIGLIB. You can use this utility to create new policies, delete policies that are no longer required, and provide reports describing the policies. “Appendix B. Administrative Data Utility” on page 263 describes the utility and provides examples for its use.

To work with WLM administrative policies, you can use the administrative application, which is described in *OS/390 MVS Planning: Workload Management*.

You can find information on defining and activating policies as follows:

- For the CFRM policy, see “Chapter 4. Managing Coupling Facility Resources” on page 37.
- For the SFM policy, see “Chapter 7. Planning Sysplex Availability and Recovery” on page 139.
- For the LOGR policy, see *OS/390 MVS Programming: Assembler Services Guide*.
- For the WLM policy, see *OS/390 MVS Planning: Workload Management*.
- For the automatic restart management policy, see “Controlling Job Availability and Recovery through Automatic Restart Management” on page 154.

Setting Up Couple Data Sets for OS/390 UNIX

Starting with Release 9, the OS/390 UNIX System Services element of OS/390 uses a couple data set to maintain information to support a shared Hierarchical File System (HFS) in a sysplex. There is no policy associated with the OMVS couple data set, although it must be formatted with the IXCL1DSU format utility. See “Shared HFS in a Sysplex” in *OS/390 UNIX System Services Planning* for information about the OS/390 UNIX support for a shared HFS. See “Chapter 3. Planning the Couple Data Sets” on page 25 for considerations that must be reviewed when formatting a couple data set and “OS/390 UNIX Parameters for Format Utility” on page 249 for the syntax used to format the OMVS couple data set.

Using Signaling for Sysplex Communications

Signaling is the mechanism through which XCF group members communicate in a sysplex.

In a multisystem sysplex, signaling can be achieved with:

- Channel-to channel (CTC) communication connections (which can be an ESCON channel operating in CTC mode or a channel defined through a 3088 Multisystem Channel Communication Unit)
- A coupling facility (using coupling facility list structures)
- A combination of CTC connections and list structures

The implementation of signaling is transparent to exploiters of the signaling service.

Full signaling connectivity is required between all systems in a sysplex. That is, there must be an outbound and an inbound path between each pair of systems in the sysplex.

To avoid a single point of failure in the sysplex, IBM recommends that you also establish a redundant path for each required path.

Signaling services are described in “Chapter 5. Planning Signaling Services in a Sysplex” on page 69.

Tuning the Signaling Services and Coupling Facility

Two important areas to consider when evaluating the performance of a sysplex are the performance of the signaling service and the performance of the coupling facility.

Two RMF reports provide information related to the performance of a sysplex. The XCF Activity Report is a three part report that describes messages sent to and from a system, by:

- System
- XCF group and member
- Path

The Coupling Facility Activity Report provides information about the usage of a coupling facility, structures within a coupling facility, and subchannel connections to a coupling facility in a sysplex.

“Chapter 6. Tuning a Sysplex” on page 117 briefly describes the Coupling Facility Activity Report. For more information on how to use the report, see the RMF library. “Chapter 6. Tuning a Sysplex” on page 117 also describes the factors that influence signaling performance in a sysplex and discusses how to use the XCF Activity Report to identify and analyze signaling-related constraints, performance problems, and capacity requirements.

In general, it’s a good idea to get the sysplex running using the IBM default values for parameters related to signaling. Because the signaling service attempts to optimize its use of resources, you might not need to take any additional action to ensure good signaling performance.

Once the sysplex is running, you can run RMF reports to investigate how signaling resources (signaling paths and message buffers) can be tuned to improve signaling performance. Then, you can adjust resources on each system to accommodate its message traffic.

For more information on using RMF to help analyze system performance, see the RMF library.

Installing Systems in a Sysplex

Information on installing systems in a sysplex is included in “Chapter 8. Adding MVS Systems to a Sysplex” on page 159. This chapter describes how you might build various sysplex configurations and includes diagrams for each.

Using the Year 2000 Sysplex Test Datasource LPAR Enhancement

The Year 2000 Sysplex Test Datasource LPAR enhancement allows you to do Year 2000 testing for a multisystem sysplex without having to dedicate a separate CPC and Sysplex Timer. Instead, each system in the sysplex to be tested runs as a datasource LPAR. The Change LPAR Sysplex Test Datasource configuration panel allows you to specify a date and time for the LPARs participating in the test sysplex. See *PR/SM Planning Guide* for a description of the Change Logical Partition Sysplex Test Datasource task.

When setting up the sysplex for Year 2000 testing, the following restrictions apply:

- All systems running as datasource LPARS must use a simulated Sysplex Timer (SIMETRID), which further requires that all systems must be running on the same CPC. SIMETRID is specified in the CLOCKxx parmlib member for the sysplex.
- When SIMETRID is specified in CLOCKxx, OPERATOR NOPROMPT must also be specified (or defaulted to) in CLOCKxx. If OPERATOR PROMPT is specified with SIMETRID, the system enters a non-restartable wait state at system initialization time.
- System in a given sysplex must be either all datasource LPARS or all non-datasource LPARS. Mixing of datasource and non-datasource LPARs is not allowed.

The ITSO publication, *S/390 Time Management and IBM9037 Sysplex Timer*, SG24–2070, presents additional information about the Year 2000 Sysplex Test Datasource LPAR enhancement.

Sharing a Sysplex Timer Among LPARs

Certain processor types support the definition of a logical time offset relative to the Sysplex Timer for a particular LPAR. In a sysplex environment this function allows multiple sysplexes with different time settings to run on the same CECs with the same Sysplex Timer.

The time-of-day for each LPAR is the Sysplex Timer time adjusted by the logical offset value. In a multisystem environment, XCF will allow only images with the same logical offset value to participate in the same sysplex with one another. At IPL-time, XCF detects any inconsistent logical offset values and prevents the IPL-ing system from joining the sysplex.

This support was previously available as APAR OW34644.

Chapter 2. Planning Parmlib Members for a Sysplex

SYS1.PARMLIB members contain parameter values that MVS uses as input during system initialization to define the characteristics of the system.

For an MVS system to run in a sysplex, you must specify certain values in the IEASYSxx and the COUPLExx parmlib members. Certain systems running in a PR/SM partition can specify values for the XCFPOLxx parmlib member.

And, depending on the way you set up your sysplex, you can also specify values for the IEASYMxx and the LOADxx parmlib members.

The following topics provide background information and considerations for selecting values for these parmlib members.

- “Planning the IEASYSxx Member in SYS1.PARMLIB”
- “Planning the COUPLExx Member in SYS1.PARMLIB” on page 19
- “Planning the CONSOLxx Member in SYS1.PARMLIB” on page 21
- “Planning the XCFPOLxx Member in SYS1.PARMLIB” on page 23
- “Planning the IEASYMxx Member in SYS1.PARMLIB” on page 23
- “Specifying the SPINRCVY Statement in the EXSPAT Parmlib Member” on page 24

You might find it helpful to supplement the planning information here with the corresponding reference information from *OS/390 MVS Initialization and Tuning Reference*.

Planning the IEASYSxx Member in SYS1.PARMLIB

IEASYSxx holds parameter values that control the initialization of an MVS system. Values for IEASYSxx can be part of the system parameter list or can be specified in response to the IEA101A Specify System Parameters prompt.

The following topics include planning information for IEASYSxx system parameters.

- “Considerations for the SYSNAME System Parameter”
- “Planning the PLEXCFG System Parameter”
- “Planning the GRS System Parameter” on page 12
- “Planning the GRSCNF System Parameter” on page 14
- “Planning the GRSRNL System Parameter” on page 15
- “Planning the CLOCK System Parameter” on page 15
- “Planning the COUPLE System Parameter” on page 19

Considerations for the SYSNAME System Parameter

You can specify the system name on the SYSNAME parameter in the IEASYSxx parmlib member. That name can be overridden by a SYSNAME value specified either in the IEASYMxx parmlib member or in response to the IEA101A Specify System Parameters prompt.

Planning the PLEXCFG System Parameter

The PLEXCFG system parameter restricts the type of sysplex configuration into which the system is allowed to IPL. For example, if you specify a multisystem sysplex (PLEXCFG=MULTISYSTEM), then you cannot specify GRS=NONE.

On PLEXCFG, specify one or more of the following:

Parmlib Members

PLEXCFG=MULTISYSTEM

Indicates that the system is to be part of a sysplex consisting of one or more MVS systems that reside on one or more processors. The same sysplex couple data sets must be used by all systems.

You must specify a COUPLExx parmlib member that identifies the same sysplex couple data sets for all systems in the sysplex (on the COUPLE statement) and signaling paths between systems (on the PATHIN and PATHOUT statements). You must also specify in the CLOCKxx parmlib member whether you are using a Sysplex Timer that is real (ETRMODE=YES) or simulated (SIMETRID specification).

Use MULTISYSTEM when you plan to IPL two or more MVS systems into a multisystem sysplex and exploit full XCF coupling services. GRS=NONE is not valid with PLEXCFG=MULTISYSTEM.

PLEXCFG=XCFLOCAL

Indicates that the system is to be a single, standalone MVS system that is not a member of a sysplex and cannot use couple data sets. The COUPLExx parmlib member cannot specify a sysplex couple data set, and, therefore, other couple data sets cannot be used. Thus, functions, such as WLM, that require a couple data set are not available.

In XCF-local mode, XCF does not provide signaling services between MVS systems. However, multisystem applications can create groups and members, and messages can flow between group members on this system. If signaling paths are specified, they are tested for their operational ability, but they are not used.

Use XCF-local mode for a system that is independent of other systems. In XCF-local mode, XCF services (except permanent status recording) are available on the system and you can do maintenance, such as formatting a couple data set or changing the COUPLExx parmlib member. The default COUPLE00 parmlib member is intended to be used to IPL the system in XCF-local mode, as is the specification of COUPLE=**.

PLEXCFG=MONOPLEX

Indicates that the system is to be a single-system sysplex that must use a sysplex couple data set. Additional couple data sets, such as those that contain policy information, can also be used. XCF coupling services are available on the system, and multisystem applications can create groups and members. Messages can flow between members on this system (but not between this system and other MVS systems) via XCF signaling services. If signaling paths are specified, they are not used.

You must specify a COUPLExx parmlib member that gives the system access to a sysplex couple data set to be used only by this system. When a system IPLs into a single-system sysplex, no other system is allowed to join the sysplex.

Use MONOPLEX when you want only one system in the sysplex (for example, to test multisystem applications on one system) or when you want to use a function, such as WLM, that require a couple data set.

PLEXCFG=ANY

Indicates that the system can be part of any valid sysplex configuration. Specifying ANY is logically equivalent to specifying XCFLOCAL, MONOPLEX,

or MULTISYSTEM. ANY is the default. If the system is ever to be part of a single-system sysplex or a multisystem sysplex, you must specify a COUPLExx parmlib member that gives the system access to a couple dataset. Note that the default COUPLE00 parmlib member does not include a PCOUPLE statement, and therefore is intended to be used only to bring up the system in XCF-local mode. Specifying COUPLE=** will also allow the system to be brought up in XCF-local mode. See the COUPLE parameter of IEASYSxx in *OS/390 MVS Initialization and Tuning Reference* additional information about COUPLE=**.

When PLEXCFG=ANY is specified, MVS will initialize and establish a PLEXCFG “mode” of XCFLOCAL, MONOPLEX, or MULTISYSTEM based on parmlib settings in COUPLExx and CLOCKxx as well as the availability of sysplex resources. For example:

- If you specify PLEXCFG=ANY and a sysplex couple data set is not specified in COUPLExx, or if COUPLE=** is specified, the MVS system will be initialized in XCF-local mode.
- If you specify PLEXCFG=ANY and a sysplex couple data set is specified in COUPLExx but the data set is not available, the MVS system can be initialized in XCF-local mode either by switching to a COUPLExx member that does not specify sysplex couple data sets or by specifying COUPLE=**.
- If you specify PLEXCFG=ANY and a sysplex couple data set is specified and available via COUPLExx, and a sysplex timer is specified (ETRMODE YES in CLOCKxx) but ETR signals are not present, the MVS system can be initialized in MONOPLEX mode. The system is also considered to be in ETR-LOCAL mode. If ETR signals are restored and the CPC TOD and the ETR TOD are resynchronized (thus establishing ETR mode), the MVS system will be in a sysplex that is MULTISYSTEM-capable.
- If you specify PLEXCFG=ANY and a sysplex couple data set is specified and available via COUPLExx and a sysplex time reference is specified (CLOCKxx specifies either SIMETRID or ETRMODE YES) and available, the MVS system will be in a system that is MULTISYSTEM-capable.

PLEXCFG=ANY is the least restrictive of the PLEXCFG parameters and can provide flexibility during system initialization that can be useful when a sysplex couple data set or Sysplex Timer are not available. The first goal, however, should be to provide an adequate level of resource redundancy to help prevent availability problems, and then specify the desired configuration for the PLEXCFG parameter.

When using PLEXCFG=ANY, ensure that systems requiring a multisystem configuration do not run concurrently with each other in XCF-local mode. For example, if system “A” and system “B” normally run in a multisystem sysplex using global resource serialization to serialize access to a shared common data base, you would not want to bring up “A” and “B” in XCF-local mode using two different global resource serialization complexes. To do so would introduce data integrity exposures to the shared common data base.

Programming note: To prevent the operator from overriding the PLEXCFG parameter, use OPI=NO on the PLEXCFG keyword in the IEASYSxx parmlib member. (For example, PLEXCFG=MULTISYSTEM,OPI=NO). Note, however, that IBM recommends that OPI=YES be used in case PLEXCFG=XCFLOCAL needs to be specified to IPL the system in XCF-local mode to format the sysplex couple data set.

Parmlib Members

Planning for XCF-Local Mode

In certain recovery scenarios, it might be necessary to have a system that can be initialized in XCF-local (standalone system) mode. To IPL a system in XCF-local mode, do the following:

1. Define a separate set of parmlib members to be used to IPL the systems in XCF-local mode.
 - Define an alternate LOADxx parameter that points to an IEASYSxx parmlib member that will be used to initialize a system in XCF-local mode.
 - Define an IEASYSxx parmlib member that will be used to initialize a system in XCF-local mode. Some of the parameters that might need to be specified are:
 - PLEXCFG=XCFLOCAL
 - GRS=NONE (or GRS=START or GRS=JOIN when global resource serialization CTCs are defined)
 - COUPLExx (or COUPLE=**)

You may choose to alter other parameters or parmlib members depending on your hardware and software configuration.

2. If COUPLE=** is not used, define a COUPLExx parmlib member with one of the following specified:
 - COUPLE SYSPLEX(&SYSPLEX) — If the sysplex name is contained in LOADxx.
 - COUPLE SYSPLEX(LOCAL) — If the sysplex name is not contained in LOADxx.

(Do not specify any other statements in COUPLExx.)

3. Create a separate set of system packs for the XCF-local system.
4. Create a set of IODFs that:
 - a. Prevents access to data that is in use by other systems (used to test the basic function of the XCF-local system).
 - b. Permits access to data that is used by other systems (used to access shared data when all other systems are down).
5. Create a set of operational procedures necessary to initialize and use the XCF-local system.

To preserve flexibility of configuration options during IPL, specify OPI=YES to allow the operator to change the configuration if necessary. See the OPI parameter of IEASYSxx in *OS/390 MVS Initialization and Tuning Reference* for additional information.

Take the necessary precautions to ensure that the XCF-local system does not access data concurrent with any other systems because the XCF-local system may not be serializing shared data. For example, if your sysplex processes a common workload and accesses shared data bases using mechanisms like global resource serialization to ensure serialized access to shared data, an XCF-local system should not be allowed to access the same shared data concurrent with your sysplex because the XCF-local system exists outside your sysplex and may not provide serialized access to shared data.

Planning the GRS System Parameter

The GRS system parameter indicates whether the system is to join a global resource serialization complex.

In a multisystem sysplex, every system in the sysplex must be in the same global resource serialization complex. This allows global serialization of resources in the sysplex.

To initialize each MVS system in the multisystem sysplex, you must use the global resource serialization component of MVS. You can use another product to serialize the use of shared resources, but global resource serialization is also required.

For a global resource serialization star complex, specify

GRS=STAR

The effect that STAR has depends on the system configuration as follows:

XCF-local mode (PLEXCFG=XCFLOCAL)

This configuration is not allowed in a star complex. Message ISG303D is issued, and the operator is prompted to either re-IPL the system or to continue the IPL with GRS=NONE.

Single-system sysplex (PLEXCFG=MONOPLEX)

This configuration is not allowed in a star complex. Message ISG303D is issued, and the operator is prompted to either re-IPL the system or to continue the IPL with GRS=NONE.

Default sysplex configuration (PLEXCFG=ANY)

If XCF is in local mode, this configuration is not allowed (see above). Otherwise, the system is initialized into a global resource serialization star complex and XCF protocols are used in the complex.

Multisystem sysplex (PLEXCFG=MULTISYSTEM)

The system starts or joins an existing sysplex and a global resource serialization star complex. XCF coupling services are used in the sysplex and in the complex.

For a global resource serialization ring complex, specify one of the following:

GRS=JOIN or START or TRYJOIN

The effect of JOIN, START, or TRYJOIN depends on the sysplex configuration, as follows:

XCF-local mode (PLEXCFG=XCFLOCAL)

There is only one system (in XCF-local mode) in the sysplex. The system starts or joins an existing global resource serialization complex and non-XCF protocols are used in the complex.

Single-system sysplex (PLEXCFG=MONOPLEX)

There is only one system in the sysplex. The system starts or joins an existing global resource serialization complex and non-XCF protocols are used in the complex.

Multisystem sysplex (PLEXCFG=MULTISYSTEM)

The system starts or joins an existing sysplex and a global resource serialization complex. XCF coupling services are used in the sysplex and the complex. (For systems in the complex that are not in the sysplex, non-XCF protocols are used.)

GRS=NONE

The effect of NONE depends on the intended sysplex configuration, as follows:

XCF-local mode (PLEXCFG=XCFLOCAL)

There is only one system (in XCF-local mode) in the sysplex and no global resource serialization complex.

Parmlib Members

Single-system sysplex (PLEXCFG=MONOPLEX)

There is only one system in the sysplex and no global resource serialization complex.

Multisystem sysplex (PLEXCFG=MULTISYSTEM)

This is not allowed. Global resource serialization is required in a multisystem sysplex.

Summary of PLEXCFG and GRS System Parameters

If you do not have an existing global resource serialization complex:

- Use GRS=NONE and PLEXCFG=XCFLOCAL or MONOPLEX when you want a single-system sysplex and do not want a global resource serialization complex.
- Use GRS=TRYJOIN (or GRS=START or JOIN) and PLEXCFG=MULTISYSTEM when you want a sysplex of two or more systems and want a global resource serialization ring complex that matches the sysplex and uses XCF services.
- Use GRS=STAR and PLEXCFG=MULTISYSTEM when you want a global resource serialization star complex.

If you have an existing global resource serialization complex:

- Use GRS=START or JOIN and PLEXCFG=XCFLOCAL or MONOPLEX when you want a single-system sysplex and want to maintain an existing global resource serialization complex that uses non-XCF protocols.
- Use GRS=START or JOIN and PLEXCFG=MULTISYSTEM when you want a sysplex of two or more systems and want to migrate systems in an existing global resource serialization complex. Systems that are in both the sysplex and complex use XCF services; systems that are not in the sysplex but are in the complex, use non-XCF protocols.

See *OS/390 MVS Planning: Global Resource Serialization* for additional planning information for the GRS system parameter. See “Multisystem Sysplex Configuration without an Existing Global Resource Serialization Complex” on page 161 and “Multisystem Sysplex Configuration with an Existing Global Resource Serialization Complex” on page 163 for installing information.

Planning the GRSCNF System Parameter

The GRSCNF=xx system parameter specifies the GRSCNFxx parmlib member that describes the global resource serialization complex for the initializing system.

Use GRSCNF=00 (the default) when all systems in the sysplex are in the global resource serialization ring complex and you can use the values in the default GRSCNF00 parmlib member to provide control information about the complex. With GRSCNF00, all global resource serialization communication is through XCF services.

Note: If GRSCNF00 is used to initialize a star complex, only the MATCHSYS and CTRACE parameters have meaning. The presence of other ring-related parameters is “tolerated”. An informational message is issued and the parameters are ignored other than for syntax checking.

Use GRSCNF=xx when you have a global resource serialization mixed complex (one or more systems in the complex are not part of a sysplex) and need to provide control and configuration information consistent with the complex. There is no requirement for a GRSCNFxx parmlib member to initialize a star complex, unless you are using a CTRACE parmlib member other than the default, CTIGRS00, shipped by IBM.

See *OS/390 MVS Planning: Global Resource Serialization* for planning information for the GRSCNF system parameter and the GRSCNF parmlib member.

Planning the GRSRNL System Parameter

The GRSRNL=00 system parameter specifies the GRSRNL00 parmlib member.

There are several ways to specify the GRSRNL system parameter; GRSRNL=00, GRSRNL=(xx,yy...) or GRSRNL=EXCLUDE.

Use the GRSRNL=00 system parameter when you can use the default RNLs in the GRSRNL00 parmlib member for global resource serialization in the complex.

The GRSRNL=(xx,yy...) system parameter specifies the GRSRNLxx parmlib member(s) that consist of the resource name lists (RNLs) used in the complex to control the serialization of resources.

Use GRSRNL=(xx,yy...) when you have an existing global resource serialization complex, are adding the system to a sysplex, and want to use the existing RNLs in the complex.

GRSRNL=EXCLUDE indicates that all global enqueues are to be treated as local enqueues. All ENQ macro requests are excluded from global serialization and are serialized locally; except that all resources identified on an ENQ SCOPE=SYSTEMS,RNL=NO macro are globally serialized on all systems in the complex.

Use GRSRNL=EXCLUDE when you do not have an existing global resource serialization complex, are adding the system to a sysplex, and want the serialization of most resources done with the RESERVE macro.

See *OS/390 MVS Planning: Global Resource Serialization* for planning information for the GRSRNL system parameter and the GRSRNL parmlib member.

Planning the CLOCK System Parameter

The CLOCK=xx system parameter specifies the CLOCKxx parmlib member to be used when you IPL your system. CLOCKxx indicates how the time of day (TOD) is to be set on the system.

Setting the TOD Clock

MVS bases its decisions and activities on the assumption that time is progressing forward at a constant rate. The instrument used to mark time in an MVS system is the time of day (TOD) clock. As it operates, MVS obtains time stamps from the TOD clock. MVS uses these time stamps to:

- Identify the sequence of events in the system
- Determine the duration of an activity
- Record the time on online reports or printed output
- Record the time in online logs used for recovery purposes.

To ensure that MVS makes time-related decisions as you intend, IBM recommends that you **do not reset your TOD clock**, for example, to switch to or from Daylight Savings Time. Instead, set the TOD clock to a standard time origin, such as Greenwich Mean Time (GMT), and use the TIMEZONE statement of the CLOCKxx parmlib member to adjust for local time. (The TIMEZONE statement allows you to specify how many hours east or west of Greenwich you are.) To adjust local time,

Parmlib Members

change the offset value for the TIMEZONE statement. This will not disturb MVS's assumption that time is progressing forward and will allow time stamps on printed output and displays to match local time.

Synchronizing Time Stamps in a Sysplex

Similarly, for MVS to monitor and sequence events in a sysplex as you intend, time must be synchronized across all systems. Thus, MVS requires that each processor in the sysplex share a clock that can provide synchronized time stamps. An IBM Sysplex Timer (9037) can be used for this purpose.

The Sysplex Timer provides another way to adjust local time in a sysplex without resetting the TOD clock. You can change the time zone offset on the Sysplex Timer itself.

Adjusting Local Time in a Sysplex

To adjust local time without resetting the TOD clock in the processor, you can:

- Reset the time offset in the Sysplex Timer
- Change the local time offset in the CLOCKxx member of SYS1.PARMLIB
- Issue the SET CLOCK command

The method you choose to adjust local time depends on your sysplex configuration and on the options specified on the ETRMODE, ETRZONE, and SIMETRID statements of the active CLOCKxx member.

1. Sysplex includes a Sysplex Timer
 - If the time-zone offset is taken from the Sysplex Timer (ETRMODE YES and ETRZONE YES), change the time offset from the Sysplex Timer console. See *Planning for the 9037 Sysplex Timer (GA23-0365)* for additional details.
There is no need to reIPL the system to cause the local time adjustment to take effect. However, for any of the following processors, when the change has taken effect, you must issue the SETETR command on each system:
 - 3090 model J
 - 9121-320 based model
 - 9021-340 based model
 - If the time-zone offset is taken from the CLOCKxx member (ETRMODE YES and ETRZONE NO) do either of the following:
 - Issue the SET CLOCK command to change the local time offset and change the local time offset value in the CLOCKxx member to reflect the new offset. This will preserve the change for ensuing IPLs.
 - Change the offset value in CLOCKxx and re-IPL.
2. Sysplex does not include a Sysplex Timer
 - In a single system sysplex, do either of the following:
 - Issue the SET CLOCK command to change the local time offset, and change the local time offset value in the CLOCKxx member to reflect the new offset. This will preserve the change for ensuing IPLs.
 - Change the offset value in CLOCKxx and re-IPL.
 - In a multisystem sysplex on one processor (either guest MVS systems running under a host VM system or in logical partitions (LPARs) in a PR/SM environment), do either of the following:
 - Update the CLOCKxx member for each guest MVS system with the correct time-zone value and reIPL each system.
 - Issue the SET CLOCK command to change the local time.

Systems that are not attached to the Sysplex Timer must be reIPLed in order to begin using the new time-zone value in CLOCKxx.

Understanding PLEXCFG Configurations with a Sysplex Timer

The ETRMODE parameter in your CLOCKxx member indicates whether you are using a Sysplex Timer to synchronize time. Specify ETRMODE YES if you want MVS to use the Sysplex Timer.

- In a non-sysplex (PLEXCFG=XCFLOCAL) or single-system sysplex (PLEXCFG=MONOPLEX), the use of a Sysplex Timer is optional.
- In a multisystem sysplex (PLEXCFG=MULTISYSTEM), a Sysplex Timer is required if the systems in the sysplex span hardware CPCs. If all systems in the sysplex are contained within one physical partition of a single CPC, you can choose either to use the Sysplex Timer (ETRMODE YES) or use the local TOD clock to synchronize the MVS images (SIMETRID (nn)).

A Sysplex Timer failure can affect the sysplex in several ways, depending on how the sysplex is configured with the PLEXCFG system parameter. In general, when an MVS image is running and loses Sysplex Timer signals:

- If the MVS image is in a configuration that is multisystem sysplex capable, the MVS image is placed into a non-restartable wait state.
- If the MVS image is in a configuration that is non-sysplex or single system sysplex mode, the MVS image is placed in local timer mode. This mode will not affect the operation of the MVS image in the configuration.

To avoid outages due to lost Sysplex Timer signals, you should configure redundant Sysplex Timers and Sysplex Timer links, and also eliminate single points of failure such as a single power source.

The following information summarizes the effects of a Sysplex Timer failure for each of the PLEXCFG configuration options. Two scenarios are presented: the first illustrates the effect when Sysplex Timer signals are not present at IPL; the second illustrates the effect when Sysplex Timer signals are lost while MVS is running.

- **XCF-local mode (PLEXCFG=XCFLOCAL)**
 - If no Sysplex Timer signals are available at IPL, the system is brought up in local timer mode, using the system TOD clock.
 - If Sysplex Timer signals are lost, the system enters local timer mode.
- **Single-system sysplex (PLEXCFG=MONOPLEX)**
 - If no Sysplex Timer signals are available at IPL, the system is brought up in local timer mode, using the system TOD clock.
 - If Sysplex Timer signals are lost, the system enters local timer mode.
- **Multisystem sysplex (PLEXCFG=MULTISYSTEM)**
 - If no Sysplex Timer signals are available at IPL, MVS initialization cannot complete.
 - If Sysplex Timer signals are lost, the affected MVS image is placed in a non-restartable wait state (WAIT0A2).
- **Default sysplex configuration (PLEXCFG=ANY)**
 - If a sysplex couple data set is not specified or not available via COUPLExx, the implied configuration is XCF-local mode:
 - If no Sysplex Timer signal is available at IPL, the system is brought up in local timer mode, using the system TOD clock.
 - If a Sysplex Timer signal is lost, the system enters local timer mode.

Parmlib Members

- If a sysplex couple data set is specified and available via COUPLExx, the implied configuration is MONOPLEX or MULTISYSTEM depending on the availability of Sysplex Timer signals:
 - During the IPL of the FIRST system, if Sysplex Timer signals are not available, the system is brought up in local timer mode using the system TOD clock. Additional systems cannot be brought into the sysplex until Sysplex Timer signals are restored.

If Sysplex Timer signals are restored, and the CPC TOD and the Sysplex Timer TOD are resynchronized, the sysplex becomes MULTISYSTEM-capable and will allow other systems to join the sysplex.
 - If Sysplex Timer signals are lost, the affected MVS image is placed in a non-restartable wait state (WAIT0A2).

Note: In the above scenarios, when MVS is running in local timer mode, if Sysplex Timer signals are restored, the affected MVS image will enter Sysplex Timer mode if the CPC TOD and the Sysplex Timer TOD become resynchronized. Resynchronization of CPC and Sysplex Timer TODs can occur when the difference between the CPC TOD and the Sysplex Timer TOD does not exceed the value of ETRDELTA specified in CLOCKxx.

Consequences of Resetting the TOD Clock in a Sysplex

If you choose to set the TOD clock to local time, then local time changes can be accomplished *only* by resetting the TOD clock in the processor. You should be aware of the possible consequences to XCF and the multisystem applications using XCF services when you set the TOD clock backward or forward:

- Setting the TOD Clock Backward

As it monitors system status, MVS records, in the sysplex couple data set, the time stamps obtained from the TOD clock. If you set the TOD clock back, then time stamps obtained from the clock may duplicate time stamps already recorded in the sysplex couple data set. For example, if the TOD clock is reset to 2:00 AM at 3:00 AM, then all timings between 2:00 and 3:00 will occur again. When a system attempts to IPL into the sysplex, MVS compares time stamps from the TOD clock with time stamps recorded in the sysplex couple data set. If time stamps in the couple data set are later than those specified by the incoming system, MVS assumes this is an error and does not allow the system to join the sysplex. The system is not allowed to IPL into the sysplex until the time stamp from the TOD clock is greater than time stamps recorded in the sysplex couple data set. In the example described, this condition would persist for one hour.

Furthermore, it is common practice for many MVS applications to record time stamps on external media. For example, time stamps might be written to a data set containing a database log. These logs could be used by several processes, such as database recovery, which depend critically on progressively increasing time stamps to preserve the ordering of events. Setting the TOD clock backward could make the log useless for database recovery.

- Setting the TOD Clock Forward

As part of its status monitoring, MVS periodically checks the status of each system in the sysplex. To determine system failure, MVS uses the failure detection interval value specified in the COUPLExx parmlib member. If you set the TOD clock ahead, then the difference between a time stamp previously recorded in the couple data set and a time stamp obtained from the reset TOD clock could exceed the specified failure detection interval. In this case, MVS would initiate a “status update missing” condition, which in turn could cause the system to be removed from the sysplex.

Planning the COUPLE System Parameter

The COUPLE=xx system parameter specifies the COUPLExx parmlib member used when you IPL your system. See “Planning the COUPLExx Member in SYS1.PARMLIB” for planning information for this member.

If you do not specify the COUPLE=xx parameter, COUPLE=00, which specifies the COUPLE00 parmlib member, is used. The IBM-supplied default COUPLE00 parmlib member causes the system to come up in single system sysplex mode without couple data sets, (which is also called XCF-local mode).

LOCAL is the sysplex name in the IBM-supplied default COUPLE00 parmlib member. The name LOCAL has no special meaning to XCF. In XCF-local mode, the installation can specify any sysplex name. Note, however, that the IPL will fail when using the IBM-supplied COUPLE00 parmlib member if a sysplex name other than LOCAL is specified in LOADxx.

Specifying COUPLE=**

COUPLE** is not an actual parmlib member, but a specification of COUPLE=** is understood by XCF to indicate that the system is to be IPLed in XCF-local mode. This is particularly useful when the system must be IPLed in order to rectify parmlib errors in COUPLExx that otherwise cause the IPL to fail.

Like the default COUPLE00 parmlib member, COUPLE=** uses a sysplex name of LOCAL, except when a sysplex name other than LOCAL is specified in LOADxx. In that case, COUPLE=** will use the name specified in LOADxx, thus allowing the IPL to proceed.

Planning the COUPLExx Member in SYS1.PARMLIB

The COUPLExx parmlib member contains values that reflect fundamental decisions an installation makes as it sets up a sysplex. COUPLExx statements are discussed throughout this book. For reference, the COUPLExx parmlib member is included here.

- PCOUPLE, ACOUPLE — contain the names of the primary and alternate sysplex couple data sets and are described in “Chapter 3. Planning the Couple Data Sets” on page 25.
- INTERVAL, OPNOTIFY, CLEANUP — can contain values that reflect recovery-related decisions for the sysplex. They are described in the topic, “Controlling Availability and Recovery through COUPLExx” on page 139.
- CTRACE — can name an XES or XCF component trace parmlib member that contains initialization parameters for its component trace.
In the CTRACE keyword, you can specify a CTnXCFxx, a CTnXESxx parmlib member, or both (by specifying the CTRACE keyword twice). For information about component traces, see *OS/390 MVS Diagnosis: Tools and Service Aids*.
- MAXMSG, RETRY, CLASSLEN — can contain values related to signaling services and are discussed in “Planning for Optimum Signaling Performance” on page 76. A value you specify for one of these keywords on the COUPLE statement becomes the default value if you do not specify that keyword on the CLASSDEF, PATHIN, PATHOUT, or LOCALMSG statement.
- CFRMPOL — can contain the name of a CFRM policy that is to be automatically started at IPL time, if there is no other previously-activated CFRM policy in effect. See “Starting a CFRM Policy” on page 54.

CLASSDEF, PATHIN, PATHOUT, and LOCALMSG Statements

The CLASSDEF, PATHIN, PATHOUT, and LOCALMSG statements contain values to help tailor signaling options in the sysplex. These statements are discussed in “Planning for Optimum Signaling Performance” on page 76.

The DATA Statement

A DATA statement contains the name of the couple data sets that are to hold policy information for a specified policy type. The DATA statement is discussed in “Chapter 3. Planning the Couple Data Sets” on page 25.

Modifying COUPLExx Values

To change values specified on the COUPLE statement of the COUPLExx parmlib member, use the SETXCF COUPLE command.

To start or stop an inbound or outbound signaling path or to create or delete a transport class, use the SETXCF START or SETXCF STOP command.

To modify local messages, use the SETXCF MODIFY,LOCALMSG command.

SETXCF commands do not change the values specified in the COUPLExx parmlib member. Therefore, to make permanent the changes you’ve made using SETXCF commands, ensure that you modify COUPLExx appropriately before the next IPL.

Planning the CONSOLxx Member in SYS1.PARMLIB

The CONSOLxx parmlib member contains values that define your console configuration. CONSOLxx is used to specify the attributes of multiple console support (MCS) consoles, the types of commands operators can issue, and routing information for messages. IBM recommends that you use the same CONSOLxx member to describe all MCS consoles in the sysplex. The first system initialized in a sysplex determines which MCS console is the master console. The first console in CONSOLxx for the first system initialized in the sysplex becomes the master console in the sysplex.

Parmlib Members

In a sysplex, some values defined in CONSOLxx have *sysplex scope*. When a system with those values is first IPLed into a sysplex, the values are in effect for the entire sysplex. See *OS/390 MVS Planning: Operations* for complete information about specifying CONSOLxx values in a sysplex environment.

Naming Your Consoles

A console name uniquely identifies a console to the sysplex for the life of the sysplex. IBM recommends that you specify names for all MCS consoles and subsystem-allocatable consoles in the configuration. This saves on the number of console IDs (the maximum is 99) that MVS can assign in the sysplex. Assigning console names allows you to define an alternate or backup console that is attached to another system. See the CNGRPxx parmlib member.

Understanding Message Routing

CONSOLxx defines console attributes to control the types of commands operators can issue and routing information for messages. The following recommendations apply when specifying the CONSOLxx parmlib member in a sysplex environment.

The CONSOLE statement of CONSOLxx contains the following:

- CMDSYS(*) — To direct commands issued on the console to the system on which the console is attached. Specifying CMDSYS(*) defines command association between a console and a system in the sysplex.
- MSCOPE(*ALL) — To indicate that messages issued by all systems in the sysplex can be received by the console.
- DEVNUM(SYSCONS) — To define the system console. The system console is used as a backup for recovery purposes and can also be used to IPL a system or, if necessary, to run the sysplex.

The INIT statement of CONSOLxx allows you to specify whether to activate console group definitions, used for switching a master console automatically to another console on a different system.

The HARDCOPY statement allows you to specify how routing codes and undelivered messages are to be handled.

- UD(N/Y) — To specify whether this console is to receive undelivered messages and WTOR messages.

Message Recovery in a Sysplex

To have full coverage for any combination of system failures, define at least one console with MSCOPE>(*ALL) and UD=YES for each system in the sysplex.

Have each console in the sysplex belong to an alternate group that contains at least one console on another system in the sysplex. If a target console becomes inactive, the selected alternate console will receive the target console's messages. The GROUP statement of the CNGRPxx parmlib member allows you to define an alternate group and its members.

Related SYS1.PARMLB Members to Consider

The Message Processing Facility (MPF) can control message processing. Specify the MPFLST members to use on the CONSOLxx INIT statement.

The Subsystem Definitions (IEFSSNxx) parmlib member contains parameters that define the primary and secondary subsystems in your installation, and the relevant command prefix used by the subsystem.

The Specify Alternate Console Groups (CNGRPxx) parmlib member allows you to define alternate console groups whose members can serve as alternate consoles.

Planning the CLOCKxx Member in SYS1.PARMLIB

The CLOCKxx parmlib member contains values that do the following for each system in a sysplex:

- Specify that the system is to prompt the operator to initialize the time of day (TOD) clock during system initialization.
- Specify the difference between the local time and Greenwich Mean Time (GMT).
- Control the use of the IBM Sysplex Timer (9037), which ensures accurate sequencing and serialization of events.

The values for CLOCKxx are described in *OS/390 MVS Initialization and Tuning Reference*. When assigning values for a sysplex configuration, keep in mind:

- Code ETRMODE YES for each system in the multisystem sysplex that is to use the 9037 Sysplex Timer. Configurations that support ETRMODE NO are XCF-Local mode and Monoplex mode.
- ETRMODE YES causes the system to ignore the OPERATOR parameter.
- If all members of your sysplex will run in LPARs or under VM on the same side of a single physical processor, and you are using a simulated Sysplex Timer, specify SIMETRID (nn) instead of or along with ETRMODE YES. In no case, specify SIMETRID (nn) with ETRMODE NO. The system will assume that the installation does not have any type of Sysplex Timer.

SIMETRID and OPERATOR PROMPT are mutually exclusive. Specifying both causes the system to reject the CLOCKxx member.

Planning the XCFPOLxx Member in SYS1.PARMLIB

XCFPOLxx can be used only in a multisystem sysplex on a processor with the PR/SM feature.

Functions provided through XCFPOLxx can provide high availability for multisystem applications on an MVS system on PR/SM.

The sysplex failure management (SFM) policy includes all the function available through XCFPOLxx. If a system is connected to an SFM couple data set in which the SFM policy is started, all XCFPOLxx specifications on that system are ignored, regardless of whether the SFM policy is active in the sysplex.

For more information about XCFPOLxx, see “Chapter 7. Planning Sysplex Availability and Recovery” on page 139.

Planning the IEASYMxx Member in SYS1.PARMLIB

IEASYMxx does the following for each system:

- Defines static system symbols
- Specifies the IEASYSxx parmlib members that the system is to use.

Parmlib Members

- Specifies the system name, which is also the substitution text for the &SYSNAME system symbol.

In IEASYMxx, you identify individual systems and, for each system, you can define a unique substitution text for the same system symbols. When the system symbols are specified in shared resource names, each system can substitute unique text for those system symbols.

Specifying the SPINRCVY Statement in the EXSPAT Parmlib Member

The EXSPATxx member of SYS1.PARMLIB allows you to specify actions to be taken to recover, without operator involvement, from excessive spin conditions. The SPINRCVY statement allows you to specify the actions to be taken to end a spin loop.

When adjusting the SPINRCVY specifications in a sysplex, it is important to understand its relationship to the failure detection interval (INTERVAL in the COUPLExx parmlib member). (For more information on INTERVAL, see “Planning the Failure Detection Interval and Operator Notification Interval” on page 140.) The default (and recommended) setting for INTERVAL is:

$2 * SPINTIME + 5$ (seconds).

In an excessive spin condition, MVS performs a SPIN followed by the action(s) specified on the SPINRCVY statement that it has time for. The default INTERVAL setting allows MVS to perform two recovery actions before the INTERVAL time expires.

The default SPINRCVY specifications are:

ABEND, TERM, ACR

Using the default values for INTERVAL and SPINRCVY, MVS would have time to spin and to abend before the INTERVAL time expired.

The default SPINRCVY specifications are satisfactory for most systems, even in a sysplex. However, in some sysplexes, you might want to consider specifying a more aggressive spin loop recovery setting, such as:

TERM, ACR

This specification will abend the failing code and cause its recovery routines to receive control, but will prevent the recovery routines from retrying into the failing code.

Chapter 3. Planning the Couple Data Sets

A sysplex requires:

- A sysplex couple data set to store information about its systems, the XCF groups and members running in the sysplex, and general status information.

And, depending on the policies you define to help manage resources and workload for the sysplex, you might need to define additional couple data sets to store policy-related information.

- The coupling facility resource management (CFRM) couple data set holds the CFRM policy, which allows you to define how MVS is to manage your coupling facility resources.
- The sysplex failure management (SFM) couple data set holds the SFM policy, which allows you to define how system failures, signaling connectivity failures, and PR/SM reconfiguration actions are to be managed.
- The workload management (WLM) couple data set holds the WLM policy, which allows you to define service goals for workloads.
- The automatic restart management (ARM) couple data set holds the policy that defines how MVS is to manage restarts for specific batch jobs and started tasks that are registered as elements of automatic restart management.
- The system logger (LOGR) couple data set holds the policy that allows you to define log stream or structure definitions.

Each of these policies, along with real time information about the sysplex and the resources being managed when the policy is in effect, resides in a couple data set. Policy specifications cannot reside in the sysplex couple data set, but you can combine data for different policies in the same couple data set.

Before you can define and activate a policy, you must format a couple data set to hold the policy and ensure that the data set is available to the systems in the sysplex that need it. This chapter describes what you need to consider when planning couple data sets for a sysplex.

Many of the things you need to consider when planning couple data sets are common to all couple data sets. However, requirements for connectivity, formatting, and definition differ between the sysplex couple data set and the couple data sets that hold policy information. Information about these topics is presented, as follows:

- “Considerations for All Couple Data Sets”
- “Considerations for a Sysplex Couple Data Set” on page 30
- “Considerations for Policy Couple Data Sets” on page 34

Considerations for All Couple Data Sets

The following considerations apply to sysplex couple data sets as well as to couple data sets that hold policy data.

A Couple Data Set Can Be Formatted with the Couple Data Set Format Utility

IBM provides a couple data set format utility, IXCL1DSU, to format the couple data sets. Sample JCL to invoke the format utility, along with explanations of the information you need to specify when formatting each type of couple data set is provided in “Appendix A. Format Utility for Couple Data Sets” on page 241.

Planning the Couple Data Sets

(The WLM couple data set can also be formatted interactively using the WLM administrative application. See *OS/390 MVS Planning: Workload Management*.)

An Alternate Couple Data Set Is Highly Recommended

To avoid a single point of failure in the sysplex, IBM recommends that, for all couple data sets, you create an alternate couple data set on a different device, control unit, and channel from the primary.

Information about the sysplex and the services that use couple data sets is maintained in both the primary and alternate couple data sets concurrently. If the primary data set fails, the sysplex automatically makes the alternate the primary.

Besides preventing a single point of failure, having an alternate couple data set has other advantages. You can issue the SETXCF COUPLE,PSWITCH command to dynamically make the alternate couple data set the primary. Thus, in a non-disruptive way, you can:

- Expand the size of the primary couple data set by switching to an alternate that is formatted to accommodate additional ITEMS appropriate to that type of couple data set.
- Change the couple data set to a different device, by defining the alternate couple data set on a different device from the primary. Thus, you can circumvent device errors or take advantage of the characteristics of a different device.

Then, take a formatted spare couple data set (or format a new alternate couple data set) and define it to XCF:

```
SETXCF COUPLE,ACOUPL=(alt-dsname[,alt-volume]),TYPE=dstype
```

An Alternate Couple Data Set Must Be At Least As Large As the Primary

After a primary couple data set has become active in the sysplex, the couple data set can be removed from service by adding an alternate couple data set (if one does not currently exist) to the sysplex and by issuing a “PSWITCH” to make the alternate couple data set the primary. When an alternate couple data set is added to the sysplex, it must be at least as large as the current primary data set. That is, the alternate couple data set must be formatted with ITEM values equal to or greater than the ITEM values that were used to format the primary couple data set. For example, if the primary sysplex couple data set was formatted for 50 XCF groups and 200 XCF group members, the new sysplex couple data set must be formatted for 50 or more XCF groups **and** 200 or more XCF group members. Likewise, the values specified for the MAXSYSTEM parameter and if applicable, the MAXGROUP/MAXMEMBER parameters, must be equal to or greater than the values that were used to format the primary couple data set.

When formatting couple data sets consider specifying parameter values commensurate with your hardware and software configuration. You might wish to add some contingency (for example, 20%) to individual parameter values to accommodate growth.

Avoid Overspecifying Parameter Values

Do not excessively overspecify parameter values when formatting couple data sets. Formatting couple data sets with parameters that are overspecified results in wasted space and degraded couple data set performance. Furthermore, if the current primary couple data set is formatted with parameter values that are overspecified and it becomes necessary to format and switch to a couple data set with smaller parameter values, the sysplex might need to be reinitialized in order to switch to the smaller format couple data set.

It is legitimate to add some contingency to parameter values to allow for growth. However, do not add excessive contingency. Remember that you can always increase couple data set capacity by issuing "PSWITCH" to switch to an alternate couple data set that is formatted with parameter values greater than the parameter values that were used to format the primary couple data set.

A Spare Couple Data Set Is Recommended

When an alternate couple data set replaces a primary, the original primary data set is deallocated, and there is no longer an alternate couple data set.

Because it is advisable to have an alternate couple data set always available to be switched, for each couple data set you plan to use, consider formatting three data sets before the IPL. For example, for the sysplex couple data set, you might format the following:

```
SYS1.XCF.CDS01 - specified as the primary couple data set  
SYS1.XCF.CDS02 - specified as the alternate couple data set  
SYS1.XCF.CDS03 - a spare
```

Then, if the alternate (CDS02) becomes the primary, you can issue the SETXCF COUPLE,ACOUPL command to make the spare data set (CDS03) the alternate.

The Couple Data Set Must Not Exist Prior to Formatting

The format utility cannot use an existing data set. This prevents the accidental re-formatting of an active couple data set. You must delete an existing couple data set before reformatting it.

A Multiple Extent Couple Data Set Is Not Supported

For the sysplex couple data set, the format utility determines the size of the data set based on the number of groups, members, and systems specified or defaulted, and allocates space on the volume specified for the data set. There must be enough contiguous space available on the volume for the sysplex couple data set.

For the couple data sets that hold policy information, the format utility determines the size of the data set based on parameter values within the policy.

A Couple Data Set Cannot Span Volumes

XCF does not support multiple-volume data sets.

A Couple Data Set Can Be Used by Only One Sysplex

The name of the sysplex for which a data set is intended must be specified when the data set is formatted. The data set can be used only by systems running in the sysplex whose name matches that in the couple data set.

A Couple Data Set Cannot Be Placed on a Multiple Exposure Device

XCF does not support multiple exposure devices.

Planning the Couple Data Sets

A Couple Data Set Can Share a Volume with Other Data Sets

However, if you decide to format a couple data set on a volume with other data sets,

- Avoid a volume that has the RESERVE macro issued against it.
- Avoid a volume with high use data sets, such as page or spool data sets.

Performance and Availability Considerations

The placement of couple data sets can improve performance, as well as availability. For maximum performance and availability, each couple data set should be on its own volume. While this is an expensive approach, placing some data sets on different volumes from others makes sense.

- The primary sysplex couple data set should be on a different volume from the primary CFRM couple data set.
- All other primary couple data sets can reside on one volume, and all other alternate couple data sets can reside on another volume, as shown in the following table. However, be sure to monitor these data sets, and consider placing any high-activity data set on its own volume. (For example, the LOGR couple data set might be a candidate for its own volume.)

Volume X Couple Data Sets	Volume Y Couple Data Sets
-----	-----
Sysplex (Primary)	Sysplex (Alternate)
CFRM (Alternate)	CFRM (Primary)
SFM (Primary)	SFM (Alternate)
WLM (Primary)	WLM (Alternate)
ARM (Primary)	ARM (Alternate)
...and so forth	...and so forth

- Place couple data sets on volumes that are attached to cached control units with the DASD fast write (DFW) feature. Although this recommendation applies to all couple data sets in any size sysplex, it is more critical the more systems you have in the sysplex. Those couple data sets most affected are the sysplex couple data set and the CFRM couple data set.
- Place couple data sets on volumes that are not subject to reserve/release contention or significant I/O contention from sources not related to couple data sets. This is true even if the I/O contention is sporadic.
- Do not excessively overspecify parameter values when formatting couple data sets.
- Some sysplex monitoring tools may generate a large amount of I/O to the CFRM couple data set, which may affect system performance. When using this type of tool, be aware of the performance implications. If you are designing a sysplex monitoring tool, see *OS/390 MVS Programming: Sysplex Services Reference* and *OS/390 MVS Programming: Sysplex Services Guide* for additional information about how XCF and XES macro services interact with the CFRM active policy.

If a system cannot access a couple data set for an extended period of time (for example, if the volume on which it resides is reserved by another system), MVS switches to its alternate couple data set. To minimize sysplex disruption, when you use DFSS (or another data mover) to back up a volume with a couple data set, it is recommended that you:

- Convert the SYSVTOC reserves on the volume to global enqueues by creating an entry in the RESERVE conversion RNL for QNAME(SYSVTOC) RNAME(volser). For more information, see *OS/390 MVS Planning: Global Resource Serialization*.

Planning the Couple Data Sets

- Use logical volume backup processing so that the backups are done on a data set level, rather than on a track-image level. For more information, see *DFDSS User's Guide* or *OS/390 DFSMSdss Storage Administration Guide*.

Security Considerations

It is the responsibility of the installation to provide the security environment for the couple data sets. Consider protecting the couple data sets with the same level of security as the XCF address space (XCFAS).

For OS/390 and MVS/ESA SP Version 5 systems, XCF does not need to be explicitly authorized to access couple data sets. However, if you are using RACF on a Version 4 system and you want to ensure that XCF has authorization to access RACF-protected sysplex resources, the XCF started procedure must have an associated RACF userid defined in the RACF started procedures table (ICHRIN03). The started procedure name is XCFAS; the associated userid must have universal access to any RACF-protected sysplex resources.

OS/390 SecureWay Security Server RACF System Programmer's Guide explains how to add a started procedure to the started procedures table.

The ADRDSSU Utility Can Dump the Contents of a Couple Data Set

If you experience problems with a couple data set or with the usage of a couple data set, you can use ADRDSSU to dump the data set. For example, if you are having problems with coupling facility structures, use ADRDSSU to dump the CFRM data set.

The following JCL could be used to dump a couple data set named SYS1.PRIMARY, which resides on the 3380 volume, SHR001.

```
//DUMP JOB MSGLEVEL=(1,1)
//STEP1 EXEC PGM=ADRDSSU,REGION=4M
//SYSPRINT DD SYSOUT=*
//DD1 DD DISP=SHR,VOL=SER=SHR001,UNIT=3380
//SYSIN DD *
        PRINT DATASET(SYS1.PRIMARY) INDDNAME(DD1)
/*
```

A Couple Data Set Can Be Defined on an SMS-Managed Volume

You can define a couple data set on a DASD volume that the Storage Management Subsystem (SMS) manages. When placing a couple data set on an SMS-managed volume, consider the following:

- Ensure that all systems that need to can access the couple data set. Catalog both the primary and alternate couple data sets in the master catalog. If the master catalog is not shared by all systems, ensure that the couple data set is in the master catalog of each system. Do not catalog the couple data set in a user catalog.
- To prevent a single point of failure, define the primary and alternate couple data sets to different control units and different volumes.
The couple data sets can reside on SMS-managed and non-SMS-managed volumes. For example, the primary couple data set can reside on an SMS-managed volume while the alternate can reside on a non-SMS-managed volume.
- Using the Interactive Storage Management Facility (ISMF) panels, define a storage class for the SMS-managed volume with GUARANTEED SPACE=YES. When you use the format utility to format the primary and alternate couple data sets, you can specify the appropriate storage classes (STORCLAS).

Planning the Couple Data Sets

You must ensure that the volumes you choose for the couple data set are available to all systems that require it. Specifying the guaranteed space attribute for the storage class ensures that SMS can select from these eligible volumes.

Protect the couple data set from being migrated or deleted by DFHSM. To prevent accidental expiration or migration of the volume by DFHSM, define the following management class attributes through the ISMF panels:

- EXPIRE AFTER NON-USAGE=NOLIMIT
- EXPIRE AFTER DATE/DAYS=NOLIMIT
- COMMAND or AUTO MIGRATE=NONE

You can use the format utility to format the couple data set on an SMS-managed volume by specifying the STORCLAS, MGMTCLAS, or VOLSER keywords. For a description of these keywords, see “Appendix A. Format Utility for Couple Data Sets” on page 241.

For information about SMS storage classes and management classes, see *Storage Management Subsystem Migration Planning Guide*.

Considerations for a Sysplex Couple Data Set

The following considerations apply only to the sysplex couple data set.

Formatting a Sysplex Couple Data Set

If you are migrating from an MVS/ESA SP Version 4 system, and you intend to use one or more policies or install more than eight systems in the sysplex, then you must reformat the sysplex couple data set using the OS/390 format utility.

Policy specifications cannot reside in the sysplex couple data set.

So that MVS can determine the size requirements for the sysplex couple data set, you need to specify, in the couple data set format utility, the maximum number of:

- Systems in the sysplex
- XCF groups in the sysplex
- The number of members in the XCF group that has the most members.

To specify the size requirements for your sysplex couple data set, both for your current needs and to provide for future growth, add up all the XCF groups that multisystem applications need, as well as all the XCF groups MVS components need. Then, to the total number of XCF groups, add a contingent number for future growth. The minimum number of groups that you can specify is 10, the maximum is 2045.

Then, determine which group has the largest number of members and specify this number. This maximum number includes all of the group’s members that can exist on all systems in the sysplex. The minimum number of members that you can specify is 8, the maximum is 511. The following topic includes a list of MVS components, their sysplex group names, and how to determine the number of members in each group.

Message IXC700E alerts you when the sysplex couple data set is filling up.

Group Names and Members for MVS Components and Subsystems

The following are some examples of XCF groups that might exist in a sysplex.

APPC: APPC has one group. The group has one member per system in the sysplex. The system generates the group name, which is in the format SYSATBxx. The APPC Component Control Block, ATBAPPCA, contains the system's group name.

Console Services: Console services has two groups named as follows:

- SYSMCS - has five members plus one member per system. For example, in a sysplex of four systems, console services group SYSMCS has nine members.
- SYSMCS2 - has the number of members calculated by the following:

$$\text{SYSMCS2 members} = 1 + n + m$$

Where: n = number of systems in the sysplex
 m = RMAX/256 + 1

(Round m to the nearest whole number.)

RMAX is the value specified on the RMAX keyword of the DEFAULT statement in the CONSOLxx parmlib member.

For example, in a sysplex of four systems and an RMAX value of 500, group SYSMCS2 has eight members. (SYSMCS2 = 1 + 4 + 3 = 8)

DAE: DAE has one group. The group has one member plus one member for each system sharing the DAE data set in the sysplex. For example, a three system sysplex would have four members in the SYSDAE group. The group name is SYSDAE.

DFSMS/MVS and PDSE Sharing: For PDSE sharing, DFSMS has two groups named as follows:

- SYSIGW00
- SYSIGW01

Each group has the number of members calculated by the following:

$$\text{members} = 1 + n$$

Where: n = number of systems in the sysplex with DFSMS installed

For example, in a sysplex of four systems that have two systems installed with DFSMS, SYSIGW00 and SYSIGW01 each have three members (SYSIGW00 = 1 + 2 = 3 and SYSIGW01 = 1 + 2 = 3).

ENF: ENF has one group. The group has one member for each system in the sysplex that is running OS/390 release 2 or higher. The group name is SYSENF.

Global Resource Serialization: Global resource serialization has one group. The group has one member per system in the sysplex. The group name is SYSGRS.

JES2 Multi Access Spool (MAS): A JES2 MAS must be entirely contained within one sysplex; including more than one MAS in a sysplex is not recommended.

A MAS in a sysplex must have a unique XCF group name. The default JES2 XCF group name is the local node name defined on the NAME parameter of the local NODE(nnnn) initialization statement. IBM recommends that the default name be used, unless it conflicts with an existing XCF group name. The JES2 XCF group can contain up to 32 members.

Planning the Couple Data Sets

Note: If there is more than one JES XCF group in a sysplex, automatic restarts may be affected. See “Controlling Job Availability and Recovery through Automatic Restart Management” on page 154 for more information.

JES3 Complex: A JES3 complex must be entirely contained within a sysplex; including more than one JES3 complex in a sysplex is not recommended. A JES3 complex in a sysplex must have a unique XCF group name. The default JES3 XCF group name is the node name defined by the NAME= keyword on the NJERMT initialization statement for the home node (HOME=YES) specified, if one exists. If you have not defined any NJERMT statements, the default is N1. It is recommended that you use the default name, unless it conflicts with an existing XCF group name. You can use the XCFGRPNM keyword of the OPTIONS initialization statement to override the default JES3 XCF group name.

The JES3 XCF group contains one member for each JES3 system active in the JES3 complex, plus one member for each converter/interpreter (C/I) or writer functional subsystem (FSS) active in the JES3 complex. See *OS/390 JES3 Initialization and Tuning Guide*.

VLF: VLF has one group. The group has one member for each system in the sysplex that is running MVS/ESA SP 4.3 or higher. The group name is COFVLFNO.

WLM: WLM has one group. The group has one member per system in the sysplex that is running OS/390 or MVS/ESA SP Version 5. The group name is SYSWLM.

XES: For structure connections, XES requests one group for each serialized list or lock structure that it connects to. The name of each group is IXCLOxxx, where xxx is a printable hex number. There is one member per structure connection.

Defining a Sysplex Couple Data Set to MVS

You define sysplex couple data sets to the system on the PCOUPLE and ACOUPLE keywords of the COUPLE statement of the COUPLExx parmlib member, for example,

```
COUPLE PCOUPLE(SYS1.XCF.CDS01)
      ACOUPLE(SYS1.XCF.CDS02)
```

IBM recommends that you define at least two COUPLExx members that specify different sysplex couple data sets. Then, if the primary sysplex couple data set fails before or during IPL, another COUPLExx member is available for the operator to specify.

Connectivity Requirements for a Sysplex Couple Data Set

A sysplex couple data set must have full connectivity to all systems in the sysplex.

Sysplex Couple Data Sets and Joining a Sysplex

A system can IPL into the sysplex if either of the following is true:

- Its COUPLExx parmlib member specifies, on PCOUPLE and ACOUPLE keywords, sysplex couple data sets that match those of the active sysplex.
- Its COUPLExx parmlib member specifies a sysplex couple data set that matches one previously used by at least one of the active systems in the sysplex.

MVS issues messages to a system that IPLs with COUPLExx specifications for couple data sets that do not match those currently in effect in the sysplex. To eliminate these messages, edit the COUPLExx member to reflect the current

primary and alternate couple data sets in the sysplex. If sysplex primary or alternate data sets need to be deleted, first delete their specifications from each system's COUPLExx parmlib member.

Sample Scenario: The following scenario illustrates how XCF processes COUPLExx data set definitions for systems that are joining the sysplex. Consider that the sysplex has defined four sysplex couple data sets: SYS1.CDS01, SYS1.CDS02, SYS1.CDS03, and SYS1.CDS04.

1. System SYSA initializes the sysplex with the following COUPLExx values:
PCOUPLE=SYS1.CDS01
ACOUPLE=SYS1.CDS02

The active primary sysplex couple data set for the sysplex is SYS1.CDS01. The alternate is SYS1.CDS02.

2. System SYSB tries to join the sysplex with the following COUPLExx values:
PCOUPLE=SYS1.CDS01
ACOUPLE=SYS1.CDS02

The COUPLExx definitions match those that are in effect for the sysplex. XCF allows SYSB to join the sysplex.

3. The operator issues SETXCF COUPLE,PSWITCH to make the alternate sysplex couple data set SYS1.CDS02 the new primary sysplex couple data set.

The active primary sysplex couple data set for the sysplex is now SYS1.CDS02. There is no alternate defined for the sysplex.

4. The operator issues the SETXCF COUPLE,ACOUPLE command to define SYS1.CDS03 as the new alternate sysplex couple data set for the sysplex.

The active primary sysplex couple data set for the sysplex is SYS1.CDS02. The alternate is SYS1.CDS03.

5. System SYSC tries to join the sysplex with the following COUPLExx values
PCOUPLE=SYS1.CDS01
ACOUPLE=SYS1.CDS02

The COUPLExx definitions do not match those that are in effect for the sysplex. XCF issues messages to indicate the values for the sysplex couple data sets in effect for the sysplex. XCF allows SYSC to join because at least one of its sysplex couple data sets was previously defined for the sysplex when SYSA initialized the sysplex.

6. The operator issues SETXCF COUPLE,PSWITCH to make the alternate XCF data set SYS1.CDS03 the new primary sysplex couple data set.

The active primary sysplex couple data set for the sysplex is now SYS1.CDS03. There is no alternate defined.

7. System SYSD tries to join the sysplex with the following COUPLExx values
PCOUPLE=SYS1.CDS01
ACOUPLE=SYS1.CDS02

XCF definitions do not match those that are in effect for the sysplex. XCF issues messages to indicate the values for the sysplex couple data sets in the sysplex. XCF allows SYSD to join because at least one of its XCF couple data sets was previously defined for the sysplex when SYSA initialized the sysplex.

8. The operator issues SETXCF COUPLE,ACOUPLE to define SYS1.CDS04 as the new alternate sysplex couple data set.

Planning the Couple Data Sets

The active primary sysplex couple data set for the sysplex is now SYS1.CDS03. The alternate is now SYS1.CDS04.

9. The system programmer updates COUPLExx for SYSA, SYSB, SYSC, and SYSD to reflect the current PCOUPLE and ACOUPLE values.
10. The system programmer deletes sysplex couple data sets SYS1.CDS01 and SYS1.CDS02.

Considerations for Policy Couple Data Sets

The following considerations apply to couple data sets that hold policy information.

Formatting Couple Data Sets

To use a couple data set for CFRM, SFM, WLM, LOGR, or automatic restart management in the sysplex, the couple data set must be formatted with the OS/390 couple data set format utility. See “Appendix A. Format Utility for Couple Data Sets” on page 241.

A couple data set can be formatted to hold one or more policies.

All couple data sets can be formatted using the couple data set format utility, IXCL1DSU, which resides in SYS1.MIGLIB. Sample JCL to invoke this utility and an explanation of the information you need to specify when formatting each type of data set is included in “Appendix A. Format Utility for Couple Data Sets” on page 241.

If your system supports a system-managed process (such as system-managed rebuild in OS/390 Release 8), there are additional requirements and coexistence concerns when formatting your CFRM couple data set. See “Considerations for System-Managed Process Support” on page 247.

The couple data set for the WLM policy can also be formatted interactively, using the WLM administrative application. For more information on this application, see *OS/390 MVS Planning: Workload Management*.

Defining Policy Couple Data Sets to MVS

For a policy to be in effect on a system, the couple data set that contains the policy must be defined on a DASD that is connected to the system and the couple data set must be defined to MVS.

To define couple data sets for policies (CFRM, SFM, WLM, LOGR, and automatic restart management) to MVS, you can use the DATA statement in the COUPLExx parmlib member or, after IPL, the SETXCF COUPLE command.

For example, to activate SFM primary and alternate couple data sets through parmlib, use the DATA statement, as follows:

```
DATA TYPE(SFM)
      PCOUPLE(SYS1.SFM.CDS01)
      ACOUPLE(SYS1.SFM.CDS02)
```

For example, to activate SFM primary and alternate couple data sets with commands, issue:

```
SETXCF COUPLE,PCOUPLE=SYS1.SFM.CDS01,TYPE=SFM
SETXCF COUPLE,ACOUPLE=SYS1.SFM.CDS02,TYPE=SFM
```

Defining and Activating Policies

Once the couple data sets for a policy are formatted, you can define and activate the policies that are to reside in the data sets. You can define as many policies as the data set is formatted for. You can combine administrative policies for more than one policy (for example, CFRM and SFM) in the same couple data set, as long as you have formatted the data set to support both types of data.

The administrative data utility, IXCMIAPU, which resides in SYS1.MIGLIB, allows you to define, update, and delete CFRM, SFM, system logger, and automatic restart management policies, and to run reports describing those policies. See “Appendix B. Administrative Data Utility” on page 263.

The WLM administrative application is available to define the WLM policy. See *OS/390 MVS Planning: Workload Management*.

The following topics provide information on defining and activating policies:

- The SFM policy is described in “Controlling System Availability and Recovery through the SFM Policy” on page 141.
- The CFRM policy is described in “Planning a Coupling Facility Policy” on page 41.
- The WLM policy is described in *OS/390 MVS Planning: Workload Management*.
- The automatic restart management policy is described in “Controlling Job Availability and Recovery through Automatic Restart Management” on page 154.
- The LOGR policy is described in *OS/390 MVS Programming: Assembler Services Guide*.

Updating Policies in a Sysplex

In some cases you can update or change policy information in an existing policy, while at other times you must format a new couple data set in which to define the updated policy. The difference is determined by the size of the couple data set as it was originally formatted.

A couple data set is formatted to hold not only the policy data but also status data about the resources defined in that policy. Thus, if you increase the number of resources within a policy (for example, increasing the number of structures that can be defined in a policy), it might be necessary to define a new policy in a newly formatted couple data set. In general,

- If the policy update requires increasing the size of the couple data set in which the policy resides, then you must first format a new couple data set using the IXCL1DSU utility.
- If the update does not require increasing the size of the couple data set, then you can use the IXCMIAPU administrative data utility to create your updates.

See “Format Utility for Couple Data Sets” for information on formatting a new couple data set.

To update policy information dynamically, you must run the administrative data utility, IXCMIAPU, against the active couple data set. The active couple data set contains both the active policy and a set of administrative policies. Run IXCMIAPU, omitting the DSN and VOLSER information, to update one or more administrative policies. To switch from the active policy to an updated administrative policy, issue the SETXCF command to start the new policy.

Planning the Couple Data Sets

See “Coding the Administrative Data Utility” on page 264 for information about updating policies in a sysplex using IXCMIAPU.

Also see “Updating a CFRM Policy” on page 54 for an example of updating a CFRM policy.

Connectivity Requirements for Policy Couple Data Sets

For a policy to be active in a sysplex, the couple data set containing the policy must be accessible to all systems that need to use it. If a system cannot access the couple data set in which a policy resides, then the policy does not become active on that system. (And, for an SFM policy to be active, it must be active on all systems.)

- The CFRM couple data set must have connectivity to all systems in the sysplex that use the coupling facility.

If a system using a CFRM data set loses connectivity to the data set, the system is placed in a non-restartable wait state.

- For the SFM policy to be active in a sysplex, the SFM couple data set must have full connectivity to all systems in the sysplex.
- Connectivity to the WLM couple data set is required to use WLM.
- Connectivity to the ARM couple data set is required to have automatic restart management available on a system.
- Connectivity to the LOGR couple data set is required to use system logger.

Chapter 4. Managing Coupling Facility Resources

The introduction of sysplex support for a coupling facility in MVS SP Version 5 satisfies customers' requirements for:

- Data sharing across the systems in a sysplex
- Resource sharing across the systems in a sysplex
- Maintaining the integrity and consistency of shared data
- Maintaining the availability of a sysplex.

The coupling facility makes data sharing possible by allowing data to be accessed throughout a sysplex with assurance that the data will not be corrupted and that the data will be consistent among all sharing users. The coupling facility provides the medium in which users of coupling facility structures can reconstruct shared data should a failure occur and can be used to assist in the isolation of failing systems.

The coupling facility is a shareable storage medium, but not what we usually consider a shared storage device. Rather, a coupling facility is licensed internal code (LIC) running in a special type of PR/SM logical partition (LPAR) in certain ES/9000 and S/390 processors. A coupling facility can be shared by the systems in one sysplex only; it cannot be shared simultaneously by multiple sysplexes.

This chapter describes what a coupling facility is and how to include one or more coupling facilities in your sysplex. The following topics are included:

- “Planning for a Coupling Facility” outlines the steps your installation must perform to use a coupling facility.
- “Planning a Coupling Facility Policy” on page 41 describes how to set up the policy that will govern your use of a coupling facility.
- “The Role of CFRM in a Sysplex” on page 56 explains the services that coupling facility resource management provides to manage a coupling facility in your installation.
- “Resource Allocation in the Coupling Facility” on page 59 describes how the storage in a coupling facility is allocated, based on your requirements and those of the application using a coupling facility.
- “Operating in a Coupling Facility Environment” on page 62 describes the operator intervention that might be required with a coupling facility in a sysplex and summarizes the operator commands with which you interface with a coupling facility.
- “Coupling Facility Structures for IBM Products” on page 66 provides a chart of coupling facility structures that are used by IBM products and where to find additional information about the structures.

Planning for a Coupling Facility

Using a coupling facility in your sysplex requires both hardware and software. First, you must have a processor that supports the coupling facility control code of the coupling facility. Second, you must have a processor on which one or more MVS images will run which is capable of attaching to the coupling facility with coupling facility links. Third, you must have the appropriate level of MVS that allows you to manage the coupling facility resources.

The information provided here is an overview of the points to be considered when adding a coupling facility to your sysplex.

Managing Coupling Facility Resources

Defining a Coupling Facility

A coupling facility runs as a special type of logical partition (LPAR) on certain ES/9000 and S/390 processors. See *PR/SM Planning Guide* for a list of all processors that support a coupling facility.

The coupling facility is defined through PR/SM panels. Once you have defined an LPAR to be a coupling facility logical partition, only the coupling facility control code can run in that partition. When you activate the coupling facility LPAR, the system automatically loads the coupling facility control code from the processor controller or the support element of the processor.

In the same manner, once you **deactivate** the coupling facility LPAR (for example, by doing a power-on-reset of the processor), the system does not preserve all coupling facility control code and any associated user data that might reside in the coupling facility.

Configuring a Processor and a Coupling Facility

Coupling facility channels link a processor and a coupling facility. These coupling facility channels can use high bandwidth fiber optic links (CFS or CFR channel paths), integrated cluster bus (CBS or CBR channel paths), or internal memory bus (ICS or ICR channel paths).

- Coupling facility sender channels (TYPE=CFS, CBS, or ICS) are the channels connecting the processor on which MVS is running to a coupling facility.
- Coupling facility receiver channels (TYPE=CFR, CBR, or ICR) are the channels attached to the coupling facility. These channels must be dedicated.

Both the coupling facility LPAR and the coupling facility channel paths must be defined to the I/O configuration data set (IOCDs). HCD provides the interface to accomplish these definitions and also automatically supplies the required channel control unit and I/O device definitions for the coupling facility channels.

See *OS/390 HCD Planning* for information about defining a coupling facility with HCD.

Understanding the Coupling Facility Operational Level (CFLEVEL)

The level (CFLEVEL) of the coupling facility control code (CFCC) that is loaded into the coupling facility LPAR determines what functions are available for exploiting applications. Different levels provide new functions and enhancements that an application might require for its operation. Different levels also provide model-dependent limits that might place limitations on how the coupling facility can be used. Additionally, different CFLEVELs may have implications when determining the size of a coupling facility structure. As more functionality is added to the CFCC, it might be necessary to allocate additional storage to a coupling facility structure. See “Requesting Structure Size” on page 42.

PR/SM Planning Guide provides information about the functions available with each CFLEVEL of a coupling facility, model-dependent limits, and other characteristics such as performance, storage allocation, and operations.

Base coupling facility function is available with coupling facility control code (CFCC) level 0 (CFLEVEL=0). Additional function and enhancements are provided by follow-on CFLEVEL versions. For the most accurate list of CFLEVEL functions with associated hardware and software corequisites, see “CFLEVEL Considerations” at <http://www.ibm.com/s390/ps0/> on the Library page.

Managing Coupling Facility Resources

An application might require that its structure be allocated in a coupling facility at a minimum coupling facility level. Or, an application might have a requirement for a particular CFLEVEL, but if a coupling facility of that level is not available, might allow its connectors to connect and run, bypassing the architected function provided by the higher CFLEVEL that was unavailable. To ensure that the structure allocation is successful, you must provide a preference list containing coupling facilities that meet or exceed the minimum level in your CFRM policy.

Specifying Coupling Facility Nonvolatility

An application that is using the coupling facility might require nonvolatility of the coupling facility storage. Depending on the processor on which the coupling facility is defined, you might have to provide a backup power supply to make the contents of coupling facility storage nonvolatile across utility power failures.

PR/SM Planning Guide describes the processor requirements for coupling facility nonvolatility and the coupling facility control code MODE command. The MODE command must be set so that applications using the coupling facility can monitor its nonvolatility status.

The 9674 Coupling Facility and coupling facilities configured on the 9672 Parallel Enterprise Server also have the capability of a power save state for preserving coupling facility storage. Power save state allows the coupling facility storage to be maintained during a utility power outage so that the structures in the storage remain intact. In the server, other LPARs that are not coupling facilities are automatically system-reset when the power is lost. *PR/SM Planning Guide* provides two scenarios that describe recovery actions in a parallel sysplex environment with a coupling facility enabled for power save state.

Planning for Coupling Facility Failure-Independence

An application might require that its structure be placed in a failure-independent environment. To accomplish this, you must ensure that the coupling facility is not in the same configuration as the MVS systems that access it. For example, placing the coupling facility in an LPAR in a processor with one or more additional LPARs that are running MVS to access the coupling facility would not provide a failure-independent environment.

Determining Your Coupling Facility Requirements

PR/SM Planning Guide provides the following information for use in determining the level of hardware and CFCC required for your installation:

- CPC model numbers that support a coupling facility
- CFCC EC levels
- Software corequisites
- Correspondence between CFCC levels and CFLEVELs
- Explanation of the functions provided by CFLEVEL

Managing a Coupling Facility

Storage in a coupling facility is divided up into distinct objects called structures. Structures are used by authorized programs to implement data sharing and high-speed serialization. Structure types are cache, list, and lock, each providing a specific function to the application. Some storage in the coupling facility can also be allocated as a dedicated dump space for capturing structure information for diagnostic purposes.

You manage a coupling facility through a policy, the coupling facility resource management (CFRM) policy. CFRM allows you to specify how a coupling facility

Managing Coupling Facility Resources

and its resources are to be used in your installation. In a CFRM policy, you supply information about each coupling facility and each coupling facility structure that you plan to use.

Your installation can define several CFRM policies, to be used at different times depending on the workload running on the sysplex. Each of these *administrative* policies has its own name by which you identify it when you want to make it the *active* policy. Only one policy can be active in a sysplex at a time.

You create the administrative CFRM policies with an IBM utility program, IXCMIAPU, provided in SYS1.MIGLIB. Once created, the policies reside in a CFRM couple data set. MVS and the authorized application use the definitions in the CFRM policy when managing the coupling facility storage.

There are two ways to start using a CFRM policy in your sysplex:

- With systems at OS/390 Release 10 and above (or with OW43278 installed), you can specify the name of the CFRM policy to be activated at IPL-time in the COUPLExx parmlib member that is used to initialize the sysplex. The system uses the CFRM policy name specified by the CFRMPOL keyword on the COUPLE statement to identify the policy to be started if there is no other previously-activated CFRM policy in effect. Note that the CFRM couple data set containing the policy must be accessible to all members of the sysplex.

The ability to specify a CFRM policy to be started at IPL-time allows you to initialize your sysplex in global resource serialization star mode when there is no previously-activated CFRM policy, if the CFRM policy started contains the ISGLOCK structure required for star mode.

- Specify the name of the CFRM policy to be started by issuing the
SETXCF START,POLICY,TYPE=CFRM,POLNAME=*polname*

command specifying the policy name. Note that this command cannot be issued until the sysplex has completed initialization using the COUPLExx parmlib member.

Authorizing Coupling Facility Requests

As documented by the subsystem or application using a coupling facility structure, the security administrator might have to define security profiles for certain structures. If the OS/390 Security Server, which includes RACF, or another security product is installed, the administrator can define profiles that control the use of the structure in the coupling facility.

The following steps describe how the RACF security administrator can define RACF profiles to control the use of structures:

1. Define resource profile IXLSTR.structure-name in the FACILITY class.
2. Specify the users who have access to the structure using the RACF PERMIT command.
3. Make sure the FACILITY class is active, and generic profile checking is in effect. If in-storage profiles are maintained for the FACILITY class, refresh them.

For example, if an installation wants to permit an application with an identifier of SUBSYS1 to issue the IXLCONN macro for structure-name CACHE1, the security administrator can use the following commands:

```
RDEFINE FACILITY IXLSTR.CACHE1 UACC(NONE)
PERMIT IXLSTR.CACHE1 CLASS(FACILITY) ID(SUBSYS1) ACCESS(ALTER)
SETROPTS CLASSACT(FACILITY)
```

You can specify RACF userids or RACF groupids on the ID keyword of the PERMIT command. If RACF profiles are not defined, the default allows any authorized user or program (supervisor state and PKM allowing key 0-7) to issue coupling facility macros for the structure.

For information about RACF, see *OS/390 SecureWay Security Server RACF Security Administrator's Guide*.

Planning a Coupling Facility Policy

To set up your CFRM policy, you must know which subsystems and applications in your installation are making use of the coupling facility and what their structure requirements are. Each application specifies the structure types required and provides information as to the name and size of the structure or structures it uses. Based on this input, you must determine what is an appropriate number of coupling facilities and how best to distribute the structures within the coupling facilities.

A single coupling facility can only be used by one sysplex at a time.

To define a CFRM policy, you must:

- Identify each coupling facility.
- Identify each structure.
- Specify the size of each structure, the maximum size and optionally, an initial size and a minimum size.
- Specify a percentage value that represents a percent full threshold for the structure.
- Specify whether you want the system to automatically alter the size of a structure when it reaches a user-defined or system-defined percent full threshold.
- Specify an ordered “preference list” of coupling facilities in which the structure may be allocated.
- Specify a list of structures, an “exclusion list”, that are not to be allocated in the same coupling facility as the structure.
- Specify a percent value to indicate the amount of lost connectivity to a structure that the sysplex can tolerate before MVS initiates a structure rebuild, if applicable.
- Define the amount of space within the coupling facility that you want for dumping.

The administrative data utility, IXCMIAPU, and an example of its use, is described in “Appendix B. Administrative Data Utility” on page 263.

Identifying the Coupling Facility

The coupling facility is identified to CFRM in two ways. First, each coupling facility has a unique 32-byte identifier that consists of a set of discrete variables, such as machine type, manufacturer, plant where manufactured, sequence number, CPC identifier, and configuration code. The values for each of these variables in the unique identifier are displayed on the hardware frame when you first define the coupling facility LPAR and the system loads the coupling facility control code. You use these values when defining the coupling facility in a CFRM policy.

Managing Coupling Facility Resources

Second, each coupling facility has a logical name — an eight-character name that you assign. It is this shorter, more usable coupling facility name that you use on interfaces with the coupling facility, for operator commands and recovery procedures, for example. The logical name and the unique worldwide identifier are paired in the definition of each coupling facility in a CFRM policy.

PR/SM Planning Guide describes the coupling facility and provides the information required for its installation and operation.

Identifying the Coupling Facility Structures

MVS recognizes three structure types — cache, list, and lock, and provides unique programming services for each of the structure types to allow the manipulation of data within the structure. To use these services, a system component, subsystem, or application “connects” to the structure, specifying a structure name and the type of structure it is.

OS/390 MVS Programming: Sysplex Services Guide and *OS/390 MVS Programming: Sysplex Services Reference* describe the programming services provided for coupling facility structures.

In your CFRM policy, you must provide the definitions to allow the application to use the structure. The CFRM policy defines the structures — the amount of coupling facility storage to be used for each structure, an ordered preference list of coupling facilities in which each structure should reside, an unordered exclusion list of structure names that should not be allocated in the same coupling facility as the specified structure, and other operational characteristics of the structure. Note that the CFRM policy does not identify the type of structure (cache, list, or lock). That information is provided by the application connecting to the structure. A coupling facility can support multiple coupling facility structures, but any one structure must exist entirely within a single coupling facility.

Requesting Structure Size

The size of a coupling facility structure includes the total amount of storage required by both the application and by the coupling facility itself for structure control information, and is specified in 1K blocks. Determining the correct size is based on the application’s use of the structure, characteristics of the workload environment in which the structure is to be used, and the CFLEVEL of the coupling facility in which the structure is allocated. The application provides guidance so that you can estimate an appropriate size for a structure. The S/390 Coupling Facility Structure Sizer (www.ibm.com/s390/pso/) is also available to assist in estimating the amount of storage required for a structure. You should use RMF reports to monitor the structure’s utilization. Also, with OS/390 Release 9 and higher, you can use structure full monitoring to monitor the structure utilization. With Release 10, the combined functions of structure full monitoring and automatic structure altering allow the system to tune the size of the structure when it reaches a full threshold.

The size of a structure can be specified by the application in its code or by the installation in its CFRM policy. The size that you specify in the CFRM policy with the SIZE specification is the requested maximum structure size. (The system might override this maximum size in a sysplex that supports system-managed processes. See “How MVS Allocates the Structure” on page 43.) With SP 5.2 and higher systems, you can change the structure size with the ALTER function if the application has specified that ALTER is allowed. ALTER allows you to vary the structure size between an initial size (specified by INITSIZE in the CFRM policy) and the maximum size. The coupling facility determines the minimum structure size, the minimum amount of coupling facility space required to allocate the structure with

the specified entry-to-element ratio. With OS/390 Release 10, it is possible to have the system automatically alter the size of a structure when the system determines that the structure has surpassed a “structure full” threshold. As with ALTER processing, the application must have specified that the structure is able to be altered; additionally, the installation must indicate in the CFRM policy that the system is to monitor the structure and automatically alter its size when required. The MINSIZE specification in the CFRM policy indicates the smallest size to which the structure can be automatically altered.

How MVS Allocates the Structure: The system allocates storage for a coupling facility structure based on the level of the system requesting the allocation and the level of the coupling facility in which the structure is to be allocated. It is strongly recommended that when you migrate to a higher level (CFLEVEL) coupling facility, you reassess the value specified for each structure’s size. Higher level coupling facilities provide new function, which may or may not require additional control storage to be allocated for a structure. Existing structures that might eventually be reallocated or rebuilt in a higher level coupling facility might have additional storage requirements that are not reflected in the size that was determined for a lower level coupling facility.

- Since SP 5.2, structure size allocation uses the STRSIZE defined on the IXLCONN macro, if the application specified a value, and that value is less than the maximum size specified in the CFRM policy with the SIZE parameter. Otherwise, it uses the INITSIZE specified in the CFRM active policy. If INITSIZE is not specified (it is an optional parameter), then the SIZE specified in the CFRM active policy is used.
- In OS/390 Release 10, the MINSIZE specification is added to the CFRM policy. MINSIZE allows an installation to indicate the smallest size to which a structure can be altered, as well as to provide a minimum bound on all structure allocation requests that occur at Release 10 and above. MINSIZE is an optional parameter in the CFRM policy, but if not specified, defaults to 50% of the INITSIZE value or 50% of the SIZE value if INITSIZE is not specified.

Note that a CFRM policy might be formatted on an OS/390 system, but used on a down-level system such as MVS/ESA SP 5.1. In that case, if an INITSIZE value is present, the SP 5.1 system will ignore the INITSIZE value. If, however, there are both SP 5.1 and OS/390 systems connecting to a structure, structure size allocation depends on the first system to request allocation. In that case, if an INITSIZE value is present and an OS/390 system requests to connect to the structure, structure size allocation uses the INITSIZE value (assuming that STRSIZE is not specified).

Rebuild Considerations: If you have altered the size of a structure to a size different from either SIZE or INITSIZE, and then you attempt to rebuild the structure, the system determines the target size of the structure to be rebuilt as described above. If the altered size value is to be used, it is the responsibility of either the connection, if STRSIZE is specified, or the installation, if INITSIZE or SIZE is specified, to change the value in IXLCONN or the CFRM policy.

System-Managed Processes Considerations: Systems running at OS/390 Release 8 or higher have the capability to support “system-managed processes”. When a process is system-managed, the system (as opposed to the connector) performs some, if not all, of the significant steps in the process. In OS/390 Release 8, the system-managed rebuild process is available for creating a new instance of a structure with minimal participation by connectors to the structure. As a result, the maximum size of the structure as specified by the application or installation might be affected. The system-managed process might require more coupling facility storage than specified by the SIZE parameter in order to allocate the new structure.

Managing Coupling Facility Resources

Applications that do not permit the size of a structure to be modified (IXLCONN ALLOWALTER=NO) should be aware of this effect when system-managed processes are supported by the sysplex. Systems running at OS/390 Release 10 or higher may have a MINSIZE specification in the CFRM policy that can affect structure allocation size as well. During system-managed rebuild, a structure may be allocated with a size smaller than the value specified or defaulted to for MINSIZE. This can occur when there is not enough space in any of the coupling facilities in the preference list to satisfy the MINSIZE requirement for the rebuild new structure, but there is enough space in at least one of the coupling facilities in the preference list to allocate the rebuild new structure and copy all the inuse objects from the rebuild old structure. In this case, instead of stopping the system-managed rebuild because the MINSIZE requirement could not be satisfied, XCF will allow the rebuild to continue.

The following table illustrates how the system determines the amount of storage to be allocated to a structure. The target size is chosen both when the system receives an initial request to connect to a structure and when the system receives a subsequent request to connect to a structure during rebuild processing.

Table 1 shows the size determination for an initial allocation from an OS/390 system. The STRSIZE value specified on IXLCONN might or might not be present, depending on the application using the structure. The INITSIZE value from the CFRM policy also might or might not be present, and is only available if the policy was formatted on an OS/390 system. The MINSIZE value from the CFRM policy also might or might not be present, and is only available if the policy was formatted on an OS/390 Release 10 or higher system. In all cases, the initial allocated size of the structure cannot be greater than the maximum structure size specified in the CFRM policy with the SIZE parameter.

Table 1. Structure Size Allocation in OS/390

	CFRM Policy		
	INITSIZE specified	INITSIZE not specified	MINSIZE specified
IXLCONN STRSIZE specified	Target size = STRSIZE	Target size = STRSIZE	Target size = MINSIZE if STRSIZE < MINSIZE
IXLCONN STRSIZE not specified	Target size = INITSIZE	Target size = SIZE	<ul style="list-style-type: none"> Target size = INITSIZE if specified Target size=SIZE if INITSIZE not specified

To display the actual amount of storage allocated to a structure, issue the DISPLAY XCF,STRUCTURE command.

Understanding MINSIZE/INITSIZE/SIZE Correlation: When requesting structure size, take the following into consideration.

- INITSIZE, the size at which you want to initially allocate the structure.
- SIZE, the maximum size of the structure.
- MINSIZE, the minimum bound for structure allocation for structure allocation requests at OS/390 Release 10 and higher.
- CFLEVEL, the level of CFCC in the coupling facility in which a structure is to be allocated.

Managing Coupling Facility Resources

When allocating the structure initially, whether INITSIZE is specified or not, the system attempts to build all control structures that will be required to support the maximum size of the structure. These control structures are built in the control storage allocation of the structure. For that reason, use care when specifying the maximum structure size with the SIZE parameter in the CFRM policy.

For example, if you have requested a maximum size that is very much larger than the initial size, the system will attempt to use a proportionally large amount of the allocated storage for its controls. The result could be that the allocated storage contains control structures for the future maximum size of the structure and insufficient storage might remain for the application's initial use.

The size of the control structures is affected by the CFLEVEL of the coupling facility in which the structure is to be allocated. Different CFLEVELs will have different control structure requirements, with the result that the same structure could have significantly different sizes depending on the CFLEVEL of the coupling facility. In some cases, a structure might actually become unallocatable in a coupling facility if the control structure allocation was so large that it occupied an amount of space larger than the INITSIZE value. Or, even if the initial allocation succeeds, there might remain in the structure only minimal space for the objects required by the application, thus preventing the application from successfully running. When rebuilding a structure from a coupling facility at one CFLEVEL to a coupling facility at another CFLEVEL, the amount of storage required for the control structures might increase to the point where the structure rebuild would fail because the structure is unallocatable. Increasing the INITSIZE value would be necessary in such instances.

MINSIZE is used primarily to limit the size to which a structure can be altered. However, MINSIZE also has implications at structure allocation time. MINSIZE serves as a minimum bound for the structure size on systems at OS/390 Release 10 and higher, including the structure allocations during rebuild processing. The only exception to this is during system-managed rebuild processing, when the system will allow the allocation of the rebuild new structure with a size less than MINSIZE as long as there is enough space in the new structure to copy all the inuse objects from the old structure.

Specifying SIZE to be larger than INITSIZE provides flexibility when you need to dynamically expand the size of your structure. However, overspecifying SIZE or not taking into account the CFLEVEL of the coupling facility can result in the allocation of a structure with little usable storage for application data or the inability to allocate a structure at all.

Understanding Preference and Exclusion Lists

The coupling facility in which a structure is to be allocated is influenced by preference and exclusion lists that are specified in the CFRM policy for each structure. Each list can have up to eight entries.

- The preference list is an ordered list of **coupling facility** names in which the structure can be allocated.
- The exclusion list is an unordered list of **structure** names with which the structure being defined should not share a coupling facility.

The preference list is used to designate the location of coupling facility structures for performance and capacity considerations. The system will not allocate a structure in a coupling facility that is not listed in its preference list.

Managing Coupling Facility Resources

The exclusion list can be used to request that related coupling facility structures occupy different coupling facilities if possible. This would prevent a single point of failure for the authorized applications that depend on the use of the related coupling facility structures. The system might allocate a structure in the same coupling facility as a structure in its exclusion list if no other suitable coupling facility can be found.

Developing Preference and Exclusion Lists: When determining which coupling facilities to include in the preference list and which structures to include in the exclusion list, consider various factors. For example,

- If a structure has a backup structure containing the same information, you would want to place the backup structure in a different coupling facility. Specify each structure in the other's exclusion list and specify different preference lists for each structure.
- If two structures with high activity reside in the same coupling facility, you would probably want to separate them for performance reasons. Specify each structure in the other's exclusion list and specify different preference lists for each structure.
- If a structure requires a certain operational level (CFLEVEL) of coupling facility because of the function it provides (such as the structure alter function), you would need to include coupling facilities of only that CFLEVEL in or higher in the structure's preference list.
- If a structure has been defined so that it can be rebuilt (due to coupling facility failure, structure failure, loss of coupling facility connectivity, or for planned coupling facility reconfiguration or maintenance activities), you must list more than one coupling facility in that structure's preference list to allow the structure to be rebuilt in a different coupling facility. The system will not allocate a structure in any coupling facility that is not listed in the structure's preference list.

With systems at OS/390 Release 9 or above or with OW33615 installed, you can specify when formatting your CFRM policy whether the order in which the coupling facilities are listed in the preference list is to be enforced with ENFORCEORDER (YES/NO). Specifying ENFORCEORDER(YES) prevents MVS from reordering the list when attempting to choose the coupling facility in which to allocate a structure. (See "How MVS Uses the Lists".) When listing the coupling facilities, keep in mind that even though some connectivity requirements may have been specified by the application, the system will ignore them if ENFORCEORDER(YES) is specified. For example,

- The CFLEVEL specification will not be honored. This might result in application connect failures if the structure cannot be allocated in a coupling facility of a high enough level.
- The CONNECTIVITY=BESTGLOBAL request also is not honored. This parameter, by definition, specifies that the system is to use the coupling facility selection algorithm.

With ENFORCEORDER(YES), the amount of storage requested for a structure (with the INITSIZE and SIZE CFRM policy parameters) will be honored only to the extent of the available storage in the coupling facility. The system will not choose the next coupling facility in the preference list simply because the first coupling facility did not have the requested amount of storage available. If the system was able to allocate the structure with viable object counts in the first coupling facility in the preference list, then despite the degraded size of the structure, that is where it will be allocated.

How MVS Uses the Lists: Unless the CFRM policy specifies that the order of coupling facility in the preference list is to be enforced (ENFORCEORDER(YES)),

Managing Coupling Facility Resources

MVS uses the following method of choosing a coupling facility in which to allocate a structure. The system finds the coupling facility in the preference list that best meets the following allocation criteria, listed in order of relative importance from most important to least important:

1. Has connectivity to the system trying to allocate the structure
2. Has a CFLEVEL equal to or greater than the requested CFLEVEL
3. Has available space greater than or equal to the requested structure size
4. Meets the volatility requirement requested
5. Meets the failure independent requirement requested
6. Does not contain a structure in the exclusion list.

If there is no coupling facility in the preference list that meets **all** these criteria, then the system determines the coupling facility that **most closely** meets the criteria. To do this, the system uses a weighting system for each of the coupling facilities in the structure's preference list. The weights correspond to the list of criteria — with system connectivity having the highest weight, CFLEVEL the next higher weight, and so on down the list.

Once it has ordered the coupling facilities in accordance with the relative importance of the allocation criteria, the system then attempts to allocate the structure in the coupling facility with the highest weight. If two or more coupling facilities are assigned identical weights, then the selection is made based on the order in which the coupling facilities are defined in the CFRM policy preference list. If the attempt to allocate the structure is not successful, the system attempts structure allocation in successively lower-weighted coupling facilities until allocation is successful. The system will choose the coupling facility that **most closely** meets the requirements of the connect request.

Depending on the coupling facility selected, once the structure is successfully allocated, the structure attributes might not be as requested.

Specifying a Rebuild Threshold

If applicable, you can specify, for each structure in the CFRM policy, a value that MVS will use to determine whether to initiate a structure rebuild if connectivity is lost to a coupling facility containing that structure. Loss of connectivity can occur because of a failure of a coupling facility attachment or because of certain types of failures of the coupling facility itself.

The appropriate action for MVS to take depends on the scope of the failure and the nature of the sysplex. For example, in a configuration made up of large systems, it might be appropriate to rebuild the structure in another coupling facility so that all of the systems can continue accessing the structure. On the other hand, in a configuration of many small systems, it might be better to simply let one or more of the systems discontinue accessing the structure while the other systems continue their work.

Note that at OS/390 Release 3 and higher with OW30814 installed, if any system connected to a coupling facility in which a structure resides loses connectivity to the coupling facility when an SFM policy is not active in the sysplex, MVS will automatically initiate a rebuild.

To allow MVS to initiate this structure rebuild, you must do the following:

1. Have a structure for which all active connections support structure rebuild.
2. Specify a REBUILDPERCENT value in your CFRM policy for each structure that MVS is to evaluate for rebuild, or allow it to default to 1. (Note that the default

Managing Coupling Facility Resources

- I value of 1 applies to systems at OS/390 Release 10 and higher and to systems
I at Release 3 and higher with OW41959 installed. For all other systems, the
I default rebuild percent value is 100.)
3. Optionally, have in place an active SFM policy that supports the use of system weight values for performing recovery actions in the event of loss of connectivity between systems. (An active SFM policy is required for a sysplex made up of systems at OS/390 Release 2 or lower or a sysplex without OW30814 installed, if you want MVS to initiate a structure rebuild.)

Whether MVS initiates a rebuild depends on the application using the structure.

- If the application does not support the MVS-initiated rebuild function, then even though you specify a REBUILDPERCENT value, the application would not be able to honor your request.
- If the application does support the MVS-initiated rebuild function, the application can also set its own criteria for when a rebuild is to be started (thus overriding the REBUILDPERCENT value specified in the policy).

How MVS Decides To Initiate a Rebuild: When MVS detects a loss of connectivity, MVS determines the viability of rebuilding each structure affected by the connectivity loss.

If MVS determines that there is an active SFM policy in the sysplex:

1. MVS verifies that the SFM policy data is at the same level on all systems.
2. MVS checks the active CFRM policy to see if a rebuild percent value has been specified for affected structures, or takes the default rebuild percent value of 1. (Note that the default value of 1 applies to systems at OS/390 Release 10 and higher and to systems at Release 3 and higher with OW41959 installed. For all other systems, the default rebuild percent value is 100.)
3. MVS calculates the total system weight of (A) all systems containing at least one active connection to the structure in the coupling facility that have lost connectivity, and (B) all systems containing at least one active connection to a structure in the coupling facility for which lost connectivity has been recognized. Note that if there are multiple users of a structure on one system, that system weight is counted only once.

For example, if a structure has one connection per system and all systems are of equal weight 10, then in an eight-system sysplex if one system lost connectivity the value of A (total system weight of all systems containing an active connection that have lost connectivity) is 10 and the value of B (total system weight of all systems containing an active connection) is 80.

4. MVS determines what action is to be taken and informs the connected users through event exit processing.

The determination is arrived at by dividing A by B, multiplying by 100, and then comparing the result with the rebuild percent for the structure in the CFRM policy.

- If the result is greater than or equal to REBUILDPERCENT, then MVS initiates a structure rebuild.
- If the result is less than REBUILDPERCENT, MVS does not initiate a rebuild.

In the example above, $(10/80)*100$ would be the value compared to the REBUILDPERCENT value. If the value of REBUILDPERCENT was 13 or higher, a rebuild would not be initiated.

If MVS determines that there is not an active SFM policy in the sysplex:

Managing Coupling Facility Resources

1. MVS verifies that rebuild is supported for the structure.
2. MVS initiates structure rebuild for any loss of connectivity affecting the structure, regardless of the REBUILDPERCENT specification, if structure rebuild is supported.

Monitoring Structure Utilization

Structure full monitoring adds support for the monitoring of objects within a coupling facility structure. Its objective is to determine the level of usage for objects that are monitored within a coupling facility, and issue a warning message if a structure full condition is imminent. This will allow an installation to intervene, either manually or through automation, so that the appropriate diagnostic or tuning actions can be taken to avoid a structure full condition.

Structure full monitoring, running on a given system, will periodically retrieve structure statistics for each of the active structure instances from each of the coupling facilities which it is currently monitoring. The retrieved information will indicate the in-use and total object counts for the various monitored objects. These counts will be used to calculate a percent full value. When structure full monitoring observes that a structure's percent full value is at or above a percent full threshold in terms of any of the structure objects that it contains, highlighted message IXC585E will be issued to the console and to the system message logs. You can review this message manually and take whatever action is necessary, such as adjusting the structure size or making changes in the workload which is being sent to the structure, or you can define message automation procedures to diagnose or relieve the structure full condition.

For each structure in the CFRM policy, the percent full threshold can be specified by the installation to be any percent value between 0 and 100. Specifying a threshold value of 0 means that no structure full monitoring will take place. If no threshold value is specified, the default value of 80% is used as the full threshold percent value.

When the utilization of all monitored structures comes down below the structure full threshold, message IXC586I will be issued to the console and to the system message logs to indicate that the full condition was relieved. Message IXC585E will be deleted.

Only "active" structure instances will be monitored by structure full monitoring.

Active structures include:

- Simplex allocated structures.
- Duplexed allocated structures. A duplexed allocated structure will have two allocated instances, a duplexing rebuild old instance and a duplexing rebuild new instance.
- Structures which are in the process of being rebuilt, either as a result of a user-managed or system-managed rebuild process. In this case, the structure will have two allocated instances, a rebuild old and a rebuild new instance.

Active structure instances will be eligible to be monitored by structure full monitoring regardless of whether or not they have any active, or failed-persistent, connectors at the time. Persistent structures with no connectors, or structures which have only failed-persistent connectors, will therefore be monitored.

Structure full monitoring will not monitor "inactive" structures. Inactive structure include structures which are:

- In transition during connect or disconnect

Managing Coupling Facility Resources

- Pending deallocation as a result of a FORCE structure request that cannot be processed immediately
- Pending deallocation where the structure is being kept because of an associated structure dump which has not been written to a dump data set yet.

In general, structure full monitoring will be discontinued for a structure in all cases where a structure instance transitions from an “active” to “inactive” state, or where the structure instance is deallocated. If structure full monitoring is discontinued for a structure instance for any reason while there is an outstanding IXC585E message, highlighted message IXC587I is issued to the console and to the system message logs. Message IXC585E is deleted.

In most cases, structure full monitoring will accurately represent the utilization of objects within the structure being monitored. There are a few special considerations. In the case of a cache structure, you can get a full condition that structure full monitoring is not able to detect, as in the following situations:

- Structure full monitoring does not monitor unchanged objects in a structure, so a cache structure will be monitored only in terms of the changed or locked-for-castout objects that the structure contains. However, a cache structure may be full because all objects are in-use for unchanged objects, in which case structure full monitoring will not detect the full condition. This type of “full” condition will normally cause reclaiming of resources, not a structure full condition.
- A directory only cache structure will not be monitored because it has no changed or locked-for-castout objects for structure full monitoring to count.
- Structure full monitoring will not be able to monitor cache structures allocated on a CFLEVEL=0 coupling facility. Some of the required object counts are not available from this level coupling facility, so it will appear to structure full monitoring that these counts are zero.
- A cache structure with multiple storage classes may experience a “full” condition because its available (unchanged and unlocked) objects are in a storage class where they are not available for satisfying a particular write request. Structure full monitoring will not be able to detect this kind of full condition. It is the responsibility of the structure user to reapportion resources across multiple storage classes via the use of cache reclaiming vectors.

For some structures, being over the structure full monitoring percentage may be a normal condition that does not require any diagnostic or corrective actions. Some examples of situations where structures may go over the structure full percentage without posing a problem are as follows:

- An application may “format” all of the structure objects within a structure before using them. This formatting makes it appear to structure full monitoring that 100% of the structure is in use, causing it to issue message IXC585E, even though there may be plenty of usable space in the formatted structure objects. An example of this kind of protocol is the coupling facility list structure JES2 uses for its checkpoint data. For a structure like this, we would recommend that structure full monitoring be suppressed by setting a FULLTHRESHOLD value of 0.
- Some applications define their own “thresholds” for the structures they use. When the data in these structures exceeds the application-defined thresholds, data gets offloaded, or castout, to DASD. During this process of accumulating data and offloading or casting out, these structures may exceed the structure full monitoring percentage if the installation has defined a higher application threshold for that structure. Message IXC585E will be issued even though the

Managing Coupling Facility Resources

structures have not exceeded the application offload or castout thresholds. Some examples of this kind of protocol are DB2 group buffer pools and logger structures. For a structure like this, we would recommend that the structure's FULLTHRESHOLD value be defined at some value higher than the offload or castout threshold is defined at.

The fact that these structures are over the structure full monitoring threshold does not necessarily indicate that there is a problem. You may choose to suppress or just ignore structure full monitoring messages for these structures.

Monitoring of coupling facilities by structure full monitoring is shared dynamically among systems in the sysplex. The ownership of monitoring by systems can change as systems come and go, or as losses of connectivity to the monitored coupling facilities occur. As a result, you will not be able to use automation that relies on any affinity between a system and the coupling facility being monitored.

Structure full monitoring will increase the monitoring frequency when there are structures with active connectors that are over the percent full threshold, and decrease the monitoring frequency when all monitored structures are under the threshold. Because this monitoring frequency is dynamically tuned, it cannot be relied upon by automation products.

Allowing a Structure to Be Altered Automatically

Starting with OS/390 Release 10, you can specify whether you want the system to automatically alter a structure when it reaches an installation-defined or defaulted-to percent full threshold as determined by structure full monitoring. The alter process may increase the size of the structure, reapportion the objects within the structure, or both. For a structure to be eligible to automatically be altered the following is assumed:

- The application defining the structure has specified that the structure is capable of being altered.
- The installation has specified that the structure is allowed to be altered (ALLOWAUTOALT(YES) in the active CFRM policy) and is being monitored by the structure full monitoring function.
- The structure is not in the process of being rebuilt.

Updating the CFRM Policy: For each structure that is to be eligible for system-initiated alter processing, the following parameters in the CFRM policy can be specified:

- **ALLOWAUTOALT**

Specify ALLOWAUTOALT(YES) to request that the structure be eligible for system-initiated alter processing. If ALLOWAUTOALT(NO) is specified and a value other than zero is specified or defaulted to for the FULLTHRESHOLD parameter, the system will continue to monitor the structure but will not initiate an alter.

- **FULLTHRESHOLD**

The value you assign to the FULLTHRESHOLD parameter is used by both Structure Full Monitoring and system-initiated alter processing to determine when a structure is "full". The value is specified as a percentage and represents a percent full threshold in terms of coupling facility structure objects. See "Monitoring Structure Utilization" on page 49.

If a value of zero is assigned to FULLTHRESHOLD, neither structure full monitoring nor automatic altering of the structure will occur. If a value other than zero is specified or defaulted to, the value is used as the percent full threshold

Managing Coupling Facility Resources

for structure full monitoring as well as being used by system-initiated alter processing to determine how to alter the structure.

- **MINSIZE**

The value you assign to MINSIZE specifies the smallest size to which the structure can ever be altered, either as a result of system-initiated alter or any alter process. This value also serves as a minimum bound for the structure size on all structure allocation requests that occur on OS/390 Release 10 and above (including connect and rebuild connect) with the following exception. During system-managed rebuild, new structure allocation may result in the structure being allocated with a size smaller than the value specified or defaulted to for MINSIZE. See “How MVS Allocates the Structure” on page 43 for information about structure allocation during system-managed rebuild.

The MINSIZE value must be less than or equal to the INITSIZE value, or less than or equal to the SIZE value if INITSIZE is not specified. If MINSIZE is not specified, the system will use a value of 50% of INITSIZE, or 50% of SIZE if INITSIZE is not specified.

Understanding When a Structure Is Automatically Altered: As stated previously, a structure is “eligible” to be automatically altered when the all of the following are true:

- Connectors to the structure allow alter.
- CFRM policy specifies ALLOWAUTOALT(YES).
- CFRM policy specifies a non-zero FULLTHRESHOLD value.
- The structure is not currently in the process of being rebuilt.

A system-initiated alter for an eligible structure will begin when structure full monitoring determines that a structure contains monitored objects that are at or above the structure full threshold specified by FULLTHRESHOLD. After issuing message IXC585E to externalize the structure full threshold condition, the system will then stop any alters in progress against the structure and issue a new alter request against the affected structure. The objective of the alter request will be either or both of the following:

- To expand the size of the structure to relieve the resource shortage when
 - All monitored object types of the structure are over the structure full threshold value.
 - At least one but less than all monitored object types are over the structure full threshold value, but all the objects cannot be brought below the threshold without also increasing the size of the structure.
- To reapportion the monitored structure objects to relieve the resource shortage for the object that is in short supply when at least one but less than all monitored objects are over the structure full threshold value and all of the objects can be brought below the threshold without also increasing the size of the structure.

Note: A structure that is in the process of being rebuilt, either as a result of user-managed or system-managed rebuild, will not be eligible to be altered automatically. The exception to this rule is for structures in a user-managed duplexing rebuild environment. In that case, when the structures are in the duplex established phase, the structures are eligible to be altered automatically.

Considerations for Duplexed Structures: When the system determines that either one of a pair of duplexed structures needs to be altered, the system will signal the system currently responsible for monitoring the coupling facility which contains the primary structure. That system will then issue an alter request against

the structure pair. The alter processing for user-managed duplexing will alter the primary structure first, followed by the secondary structure.

Relieving Coupling Facility Storage Constraints: There are instances when the system will automatically contract a structure when the coupling facility as a whole is at or above 80% full and coupling facility resources are not being used productively by the structure. This processing will potentially initiate alter processing against many structures, reclaiming a small amount of storage from each of them, until the coupling facility storage constraint has been relieved.

The structure must have been defined to be eligible for automatic alter (see “Understanding When a Structure Is Automatically Altered” on page 52 for the list of eligibility requirements). When structure full monitoring determines that all monitored objects for a particular structure are below the structure full threshold value, the system will start an alter request against the structure to contract the structure. The objective of the alter request is to contract a small percentage of resource at any one time, making sure that the reduction in structure size will not cause a percent full threshold in terms of any monitored objects to be exceeded. The alter processing will not allow the structure to be contracted below the MINSIZE value specified in the active CFRM policy.

Determining the Amount of Coupling Facility Dump Space

The coupling facility dump space is an area within the coupling facility that you reserve for dumping structure data. Systems in the sysplex can request that an SVC dump be taken to capture structure data. SVC dump serializes each structure to be dumped to keep the structure’s control information from changing. SVC dump uses the coupling facility dump space to quickly capture structure control information so that dump serialization is held for as brief a time as possible. The captured control information is written to an SVC dump data set after dump serialization is released. (Note: The maximum amount of time that SVC dump is allowed to hold serialization on the structure is specified initially when the application connects to the structure. The operator can override that amount of time with the DUMP or CHNGDUMP command.)

You can define only one dump space in each coupling facility; however, that dump space can contain several structure dump tables. IBM recommends that you specify 5% of the available storage in a coupling facility for dump space. If you do not specify any coupling facility dump space, none will be allocated and no structures in the coupling facility can be dumped.

An alternative method to the 5% “rule of thumb” for defining the amount of dump space is to allow all coupling facility storage that is not dedicated to structures to be used as dump space. Using this method maximizes the use of coupling facility storage while ensuring that there is enough coupling facility storage space available for all structures defined in a given CFRM policy. This method allows you to define more than 5% of coupling facility storage for structure dumping if there is excess storage available in the coupling facility. Calculate the amount of storage available for dump space as follows:

1. Determine the amount of coupling facility storage dedicated for use by structures. This includes storage for structures that might normally reside in other coupling facilities but that can be relocated, in failure scenarios, to the coupling facility for which this calculation is being performed. From your CFRM policy, add together the values of the SIZE parameters for that structures that contain the coupling facility name in their preference list (PREFLIST). The result is the total storage required by structures in 1K units.

Managing Coupling Facility Resources

2. Determine the amount of coupling facility storage that can be used for structure dumps by subtracting the total storage required for structures from the total amount of coupling facility storage. Specify this value in the DUMPSPACE parameter of your CFRM policy.

Note: You must reevaluate the DUMPSPACE parameter specification whenever you change the structure specifications in your CFRM policy.

3. If the amount of storage available for structure dumps is insufficient, consider reconfiguring your coupling facility to obtain additional storage space for structure dumps.

You can determine how the dump space in a coupling facility is being utilized by using RMF reports or the DISPLAY CF command. Information returned includes the total amount of storage in the coupling facility, the amount not being used, and information about the dump space.

You might need to change to a larger coupling facility dump space if you find that you are receiving partial SVC dumps because there is not enough available space to hold the structure information. Conversely, you might need to decrease the amount of dump space if you find that the space is not being used or that its size is causing structure allocations to fail or to be allocated in a less preferable coupling facility.

To change the size of the dump space defined in the active CFRM policy, you must redefine the amount of dump space for the coupling facility in a CFRM administrative policy and follow normal procedures for switching from one policy to another, such as with the SETXCF START,POLICY command.

Starting a CFRM Policy

To start using a CFRM policy in your sysplex, issue the SETXCF START,POLICY command, specifying the policy name and type(CFRM). The system identifies the policy, which is stored as an administrative CFRM policy on the CFRM couple data set, and makes a copy of it on the same CFRM couple data set as the “active” CFRM policy. The administrative copy of the same policy remains intact on the CFRM couple data set.

The system will add real-time data about sysplex and coupling facility operations to the area of the CFRM couple data set holding the active policy; the administrative copy of the policy will not be changed unless you use the administrative data utility (IXCMIAPU) to change it.

Updating a CFRM Policy

The CFRM policy resides on a couple data set that you have previously formatted to hold not only the policy but also status data about the resources defined in that policy. For example, if you formatted a CFRM policy that defined 64 structures, then the system would ensure that enough space was allocated on the couple data set to hold status information for 64 structures.

When updating a CFRM policy, you must determine whether you need to reformat the couple data set or whether you can update the existing policy with new information.

- If the update requires increasing the size of the couple data set in which the policy resides, then you must first format a new couple data set in which the policy is to reside using the IXCL1DSU format utility.

Managing Coupling Facility Resources

- If the update does not require increasing the size of the couple data set, then you can use the IXCMIAPU administrative data utility to create your updates.

So, for example, to increase the number of structures that can be defined in a policy from 64 to 128, you would have to format a new couple data set for the policy.

Formatting a New CFRM Couple Data Set

“Format Utility for Couple Data Sets” describes the IXCL1DSU format utility and the resources that can be defined in a CFRM policy — number of policies, number of coupling facilities within a policy, number of structures within a policy, and number of connections to a coupling facility structure. If the update to an existing policy involves an increase in any of these resources, you must format a new couple data set.

Updating an Existing CFRM Couple Data Set

To update an administrative policy that you have created (or add a new administrative policy) in the CFRM couple data set, you must run the administrative data utility (IXCMIAPU) against the CFRM couple data set that contains both the active policy and the set of administrative policies. The utility adds or changes policy data in the administrative policies **only**. The utility does not change any information in the system’s copy of the active CFRM policy.

For example, if you want to update information in your active CFRM policy, you would run IXCMIAPU to change the administrative copy of that policy. (You omit the DSN and VOLSER keywords in the IXCMIAPU utility when updating administrative policies.) To switch to the updated policy, issue the SETXCF START,POLICY command. This command causes the system to copy your updated policy to the active policy. It is possible that not all changes that you specified in the updated policy will take effect immediately. See “Transitioning to a New CFRM Policy” on page 58 for a description of how CFRM updates an active policy.

Another way to update your active policy without losing the ability to return to it later is to make a copy of the policy, apply the necessary definition changes, and rename the policy. For example, if your active policy is POLICY1, define a new POLICY2 with identical definitions except for those that you want to change. If, after you activate the newly created policy, POLICY2, you decide to return to the original policy, POLICY1, you can issue the SETXCF START,POLICY specifying POLICY1.

Summarizing CFRM Policy Setup Requirements

To set up a CFRM policy, you need to:

- Format a CFRM couple data set. Use the IXCL1DSU format utility program. Ensure that the CFRM couple data set is available to appropriate systems in the sysplex that plan to use coupling facility resources.
(“Appendix A. Format Utility for Couple Data Sets” on page 241 describes the IXCL1DSU format utility and gives an example of its use.)
- Define one or more CFRM administrative policies.
The administrative data utility, IXCMIAPU, described in “Appendix B. Administrative Data Utility” on page 263, provides the interface to associate the definitions with a policy name and to place the policy in a pre-formatted CFRM couple data set. IXCMIAPU also is used to update existing administrative policies that exist in the CFRM couple data set.
Definitions include information about each coupling facility and its space allocation and each structure and its preferred location in a particular coupling facility.

Managing Coupling Facility Resources

- Start one of the defined administrative policies, making it the “active” CFRM policy for the sysplex.

To activate a CFRM policy named POLICY1 that is defined in the CFRM couple data set, issue the following command:

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

If a policy already is active when you issue the START,POLICY command to start a (non-active) administrative policy, the system begins its transition to using the new policy. See “Transitioning to a New CFRM Policy” on page 58 for a description of this transition process.

To deactivate a CFRM policy, issue the SETXCF STOP,POLICY,TYPE=CFRM command.

The Role of CFRM in a Sysplex

Coupling facility resource management provides the services to manage coupling facility resources in a sysplex. This management includes the enforcement of CFRM policies to ensure that the coupling facility and structure requirements as defined in each policy are satisfied.

A Note about Operations

Throughout its management of coupling facility resources, CFRM issues operator messages to record the status of its processing. This topic is included so that you will understand the messages that are issued and be prepared to respond if necessary. Your operations staff should have a thorough understanding of the connections to the coupling facility and of the application’s use of one or more structures in the coupling facility.

CFRM uses the CFRM couple data set to maintain status data about the use of coupling facility resources in the sysplex and to hold its copy of the active CFRM policy. Whenever a CFRM couple data set is made available in the sysplex, either for the first time or when a switch is made to a different CFRM couple data set, CFRM becomes operational. Once operational, CFRM performs the following services for the sysplex:

- Cleans up status data
- Performs system level initialization
- Reconciles active policy data with data in the coupling facility.

Cleaning Up Status Data

Policy data is maintained on the CFRM couple data set, and therefore remains on the couple data set from one IPL to the next or from one use of the CFRM couple data set to the next. As part of its initialization, CFRM cleans up any status data in the CFRM couple data set that remains in what had previously been the active policy. The cleanup includes:

- Indicating that no system has access to any coupling facility.
- Setting the connections to coupling facility structures to a failed-persistent state (thus preventing connections from accessing those structures).

Performing System Level Initialization

CFRM performs system level initialization based on the state of the active CFRM policy. Valid CFRM policy states are:

- The CFRM couple data set contains an active policy that is empty, that is, there is no currently active CFRM policy, either because a SETXCF START,POLICY command was never issued or a SETXCF STOP,POLICY command was issued.
- The CFRM couple data set contains an active policy that was properly defined with the administrative data utility and then activated with the SETXCF START command.

If there is no currently active CFRM policy, any application requests to use coupling facility resources will be denied. It might be necessary for the system programmer to define new administrative policies with the IXCMIAPU utility before issuing the SETXCF START,POLICY command to activate a CFRM policy.

If there is a currently active CFRM policy, each system in the sysplex that has access to the CFRM couple data set determines whether it has connectivity to each of the coupling facilities listed in the active policy and records that information. During this phase of CFRM system level initialization, CFRM might need to query the operator before allowing a sysplex to gain ownership of a coupling facility. A coupling facility can be used by only one sysplex at a time, thus providing for data integrity and preventing unintentional data loss. If the sysplex previously did not own a particular coupling facility listed in the CFRM policy, the operator is prompted to respond as to whether the sysplex should be able to use the coupling facility.

Reconciling CFRM Policy Data

Once a sysplex has gained ownership of a coupling facility, CFRM continues its initialization by reconciling the data being maintained about the current coupling facility environment and, if necessary, by cleaning up the dump space in the coupling facility.

- **Reconciling the data**

The system maintains data about use of coupling facility resources in both the active policy and in the coupling facility itself. The coupling facility data is assumed to be more accurate. The system performs the reconciliation process for each coupling facility for which the sysplex has gained ownership. The process consists of comparing data about use of the coupling facility resources from the coupling facility with the data in the active policy. The reconciliation actions taken by CFRM might include deallocating structures that are not persistent and have no persistent connections. CFRM also checks that all persistent structures are listed in the active policy, and must determine how to handle structures or connections remaining in the coupling facility but not in the active policy. The operator might be required to request the deletion of these structures, or CFRM might reconstruct the active policy by adding the structures or connections.

- **Cleaning up the dump space**

If there were dumps in progress at the time of transition to a new active policy, the system dumps all captured data to a dump data set.

If the operator had issued a CHNGDUMP,STRLIST command to initiate one or more new structure dumps, the dump contains information for each structure listed on the STRLIST parameter.

Managing Coupling Facility Resources

Transitioning to a New CFRM Policy

SETXCF START,POLICY is the command used to activate an administrative policy. If no current active policy is available, the activation takes effect immediately. If a transition to the newly activated policy is required, this command causes the following to occur over time in the active policy.

- When adding a new coupling facility, the logical name of the coupling facility is updated in all the preference lists of the structure definitions in the active policy. There might be a delay while all systems complete the update before an operator can use the new logical name. Once all systems recognize the new coupling facility logical name, the operator can use the name on all commands referencing the coupling facility.
- When changing the dump space size, the size of the coupling facility dump space is updated in the CFRM active policy immediately. However, the *actual* size of the dump space at any given time is determined by:
 - The size specified in the policy
 - The amount of space assigned to a dump in the dump space
 - The total amount of space available in the coupling facility.

MVS attempts to satisfy requests to change the dump space size immediately and if not successful, continues to attempt the change. Use the DISPLAY CF command to determine the policy size and the actual size of the dump space in the coupling facility.

- When deleting or modifying a structure or coupling facility definition, the policy transition proceeds as follows. Note that the system handles a modification to a structure or coupling facility definition as an implicit “delete and add”.
 - The change takes effect immediately if coupling facility resources are not allocated for the particular structure.
 - The change remains pending if coupling facility resources are allocated either for the structure or for any structures in the coupling facility being modified. The structure’s resources need to be deallocated in order for the pending policy change to take effect. Rebuilding the structure is the preferred method of accomplishing this, if the structure allows rebuild. The operator command SETXCF START,REBUILD can be used to rebuild the structure in either the same coupling facility or in another coupling facility. In some cases it might be necessary to deallocate the structure using the SETXCF FORCE command, but that is not the preferred method.

The DISPLAY XCF command provides specific information about a structure and any pending policy changes.

- When releasing coupling facility resources, the following order should be observed:
 1. Remove all connections to an allocated structure.

The connection removal should be initiated by the application using the connection and should result in no active connections to the structure. However, operator intervention might be required to remove failed-persistent connections. Use the DISPLAY XCF,STR command to determine which connections are in a failed-persistent state and then delete each connection using the SETXCF FORCE command.
 2. Remove the allocated structure from the coupling facility.

The system deletes a non-persistent structure once all connections to the structure are removed. The operator can remove a persistent structure by using the SETXCF FORCE,STRUCTURE command. Note that if the structure has an associated structure dump, then the operator must issue SETXCF FORCE,STRDUMP before deleting the structure.

Managing Coupling Facility Resources

- The addition of a coupling facility or structure takes effect immediately, assuming that all resources from the prior active policy have been released. The CFRM couple data set is formatted to contain a fixed number of coupling facility and structure definitions in the active policy. If deletions or modifications of prior coupling facilities or deletions of structures are still pending, then additions might also become pending.

Resource Allocation in the Coupling Facility

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

You must take all of these factors into account when determining how to define your CFRM policy and how to configure your coupling facility.

Coupling Facility Storage Considerations

The storage for a coupling facility LPAR is defined in the same way as a non-coupling facility partition. However, the storage in a coupling facility LPAR cannot be dynamically reconfigured. If another partition on the same processor fails, its storage cannot be taken over by the coupling facility partition. Or, if the coupling facility partition fails, its storage cannot be taken over by another partition.

Storage in a coupling facility consists of two types — control storage and non-control storage, each of which the coupling facility control code uses for a specific purpose. Essentially, the coupling facility control code uses the control storage either for its control information or for data, and the non-control storage only for data. Depending on the particular processor on which the coupling facility is defined, the storage can be all control storage or a combination of control and non-control storage. The installation controls the amount of storage assigned to control and non-control storage when configuring the amount of central and expanded storage in the coupling facility LPAR. The amount of central storage equates to the amount of control storage; the amount of expanded storage equates to the amount of non-control storage. (In processors that do not support the concept of central and expanded storage, all coupling facility storage is considered to be control storage.)

The split of control and non-control storage becomes a consideration when the coupling facility control code allocates specific amounts of storage to a structure. The division between the two storage types must be monitored to ensure that the coupling facility storage is distributed most efficiently for the authorized application's use.

The DISPLAY CF command displays information about coupling facility storage including the total amount, total in-use, and total free control and non-control storage. The RMF Coupling Facility Usage Summary Report provides a more permanent copy of this information.

Coupling Facility Resource Allocation “Rules”

A coupling facility structure is located in a particular coupling facility and allocated at a certain size depending on:

- Values specified in the CFRM policy specification
- Minimum size as determined by the coupling facility

Managing Coupling Facility Resources

- Authorized application specification when using the XES services
- Coupling facility storage constraints
- Coupling facility storage increment.

CFRM Policy Specification

The CFRM policy contains the maximum structure size and can contain a smaller initial size also. The initial structure size defined in the CFRM policy (or the maximum size if an initial size is not specified) is used as the attempted allocation size unless it is overridden by a structure size specified on the IXLCONN macro.

Starting with OS/390 Release 10, the CFRM policy can also optionally designate a minimum structure size. The MINSIZE value specifies the minimum bound for the structure size on all allocation requests except those resulting from system-managed rebuild.

Authorized Application Specification

When requesting an XES service to connect to a structure, the application optionally can specify a size for the structure. The system uses the smaller of the two sizes (as specified in the CFRM policy or by the application), as the target allocation size for the structure.

The application also is required to specify certain structure attributes when connecting to a structure. These structure attributes are used by the coupling facility control code when determining how to most efficiently allocate the various parts of the structure in the coupling facility. Some examples of structure attributes are data element size, whether locks are used, the number of list headers, and whether an adjunct area is required. The coupling facility control code evaluates each attribute in the following sequence:

- **Available space**

The structure is allocated as large as possible based on the available storage in the requested coupling facility. The target size is derived from either the CFRM policy or the application's request to connect to the structure.

- **Fixed structure controls**

Storage for the fixed structure controls used by the coupling facility is allotted first from the total available structure storage. The amount of storage required for the structure controls is dependent on the CFLEVEL of the coupling facility and the maximum structure size defined in the CFRM policy. Therefore, it is important to consider both dependencies when specifying structure size. If too large a portion of the available storage is allotted to the structure controls, there may not be enough storage left for allocation of the remaining structure objects, and the structure in effect will be unallocatable or unusable.

- **Entry/element ratio**

Within the remaining structure storage, once the fixed structure controls have been allocated, the coupling facility control code attempts to allocate entries and elements in a way that most accurately approximates the requested entry/element ratio. If the structure has been allocated with a size less than the minimum structure size required by the specified structure attributes, then the entry/element ratio may deviate from the requested value. If the structure has been allocated with a size greater than or equal to this minimum structure size, then the entry/element ratio should be satisfied.

- **Entry and element counts**

Within the remaining structure storage, once the fixed structure controls have been allocated, the coupling facility control code attempts to maximize the actual number of entries and elements (as specified by the ratio) that can be placed in the structure.

For a description of the structure attributes that an authorized application can specify, see *OS/390 MVS Programming: Sysplex Services Guide* and *OS/390 MVS Programming: Sysplex Services Reference*.

Coupling Facility Storage Constraints

The coupling facility control code locates parts of a structure in either control or non-control storage, depending on whether the part is control information or data. Control information **MUST** reside in control storage and cannot reside in non-control storage; data may reside in either control or non-control storage.

The amount of control storage available can affect structure allocation. When there is no control storage available in the coupling facility, control information, such as list entry controls or directory entries, cannot be allocated (even though there might be ample available non-control storage in the structure).

The following summarizes the actions taken when there is no more available control storage in a coupling facility structure:

- Entries, which are control information, cannot be allocated because they must reside in control storage.
- If data elements are requested in the structure, they can continue to be allocated because they can reside in non-control storage. This action causes the actual entry/element ratio to be skewed in favor of elements.
- The coupling facility control code continues allocating data elements until either
 - The total space for the structure is equal to the requested total structure size or
 - The number of elements allocated equals the number of elements that result in an actual achieved entry/element ratio of 1/MAXELEMNUM. Note that MAXELEMNUM is specified by the authorized application when it connects to the structure and indicates the maximum number of data elements per entry that are supported for any entry in the structure.

Even though the total requested structure size may not be reached when the 1/MAXELEMNUM ratio is achieved, additional data elements are not allocated.

Coupling Facility Storage Increment

Coupling facility storage is allocated in multiples of the coupling facility operational level (CFLEVEL) storage increment size. See *PR/SM Planning Guide* for a list of the coupling facility limits for CFLEVEL.

Understanding Coupling Facility Use of Processor Resources

Both the cache and the list structure use resources in the processor to which the coupling facility is attached. These resources, in HSA, are allocated to the processor at processor initialization time and cannot be modified during sysplex processing. Specifically, the structure use of processor resources is:

- The local cache vector is a user-defined vector that provides a way for cache structure users to determine the validity of data in their local cache vectors. There is one local cache vector per user of the cache structure. Each vector is divided into separate entries with each entry corresponding to a local cache buffer. Each vector entry contains an indicator that the system sets to indicate

Managing Coupling Facility Resources

whether the data in the corresponding local cache vector is valid. Users must test the indicator to determine the validity of the data in their local cache buffers.

- The list notification vector is a user-defined vector that provides a way for list structure users to determine the state (empty or nonempty) of each list or event queue they are monitoring. Each connector to a list structure that indicates interest in list or event queue monitoring is allocated a list notification vector. The vector consists of an array of entries, each of which can be associated with a particular list header or the user's event queue. Each entry contains an indicator that the system sets to indicate the empty or nonempty state of the list or event queue.

XES provides the IXLVECTR service to examine the contents of the vectors allocated in HSA, but the service does not allow you to free up the vectors. Vectors that are allocated in HSA are freed up when connectors disconnect, but remain there in error scenarios such as when a system failure occurs. The implication of having obsolete vectors remaining in HSA is that the subsystems that require the allocation of vectors might not be able to initialize if additional HSA storage is not available.

To clear the obsolete vectors from HSA, the system must be reset with either a LOAD or SYSRESET command. On all processors, the LOAD CLEAR or SYSRESET CLEAR command can be used to clear the unused, obsolete vectors from HSA. On some processors, the LOAD NORMAL and SYSRESET NORMAL function has been enhanced to clear the vectors from HSA.

- If you are running on a processor that has not been upgraded to allow the clearing of obsolete vectors from HSA, use the LOAD CLEAR or SYSRESET CLEAR command to clear the vector space.
- If you are running on a processor that has been upgraded to allow the clearing of obsolete vectors from HSA, use the LOAD NORMAL or SYSRESET NORMAL command to clear the vector space.

See the applicable Operations Guide for your processor for a description of the LOAD and SYSRESET commands.

Operating in a Coupling Facility Environment

The coupling facility is designed to be run with little operator intervention. There are times, however, such as system-level initialization, when the operator is required to make decisions about how coupling facilities or structures are to be handled in the sysplex. A point to remember is that the coupling facility might contain important MVS or application data or structures, so precautions must always be taken to preserve that data.

Some instances when operator intervention might be required for a coupling facility in a sysplex are:

- When a coupling facility needs to be removed from a sysplex.
- When a coupling facility needs to be transferred to another sysplex.
- When a coupling facility needs to be repopulated with the most appropriate structures.

“Appendix E. Coupling Facility Guidelines” on page 321 provides specific details and examples for operating in a coupling facility environment.

Removing a Coupling Facility from a Sysplex

A coupling facility might need to be removed from a sysplex for maintenance, to have a new level of coupling facility control code installed, or to be replaced. When such a situation occurs, you must remove any structures contained in the coupling facility before it is shut down and removed from the sysplex.

“Appendix E. Coupling Facility Guidelines” on page 321 contains procedures for shutting down a coupling facility.

Transferring a Coupling Facility to Another Sysplex

You might need to transfer a coupling facility from one sysplex (sysplex A) to another (sysplex B) in order to do maintenance on sysplex A. The procedure for doing the coupling facility transfer includes:

- Move all structures out of the coupling facility owned by sysplex A.
- Remove the coupling facility from sysplex A’s CFRM policy and all preference lists in sysplex A.
- Add the coupling facility to sysplex B’s CFRM policy and one or more preference lists in sysplex B.

This procedure allows the system to grant ownership of the coupling facility to sysplex B without requiring further operator intervention after the initial structure removal.

Using the POPULATECF Function to Rebuild Coupling Facility Structures

The POPULATECF function of Structure Rebuild, introduced in OS/390 Release 6, provides a means of rebuilding multiple structures in a coupling facility with a single operator command or IXLREBLD macro invocation. The function’s use primarily is to repopulate a coupling facility with the most appropriate structures after the coupling facility has been added to the sysplex configuration or has been removed from the configuration for service and then returned. The advantage of the POPULATECF function is that it rebuilds only structures that are currently allocated into the coupling facility that the structure (in the CFRM active policy) has specified is preferable. If the structure already resides in a coupling facility that is higher in its preference list than the coupling facility being populated, the system will not rebuild the structure.

The following example outlines the steps an operator would take when having to remove a coupling facility from the configuration for maintenance and then subsequently returning it to service. In the example, coupling facility CF1 is to be removed, leaving coupling facilities CF2 and CF3 in the configuration. The structures in CF1 must be moved to one of the other coupling facilities before CF1 can be removed from the configuration.

Removing CF1 from the Configuration

The operator can move all structures in CF1 (with the exception of XCF signaling structures) to another coupling facility with the

```
SETXCF START,REBUILD,CFNAME=CF1,LOCATION=OTHER
```

command. The LOCATION=OTHER specification indicates that the structures are to be rebuilt in another coupling facility in each structure’s preference list other than the coupling facility in which the structure now resides. (If no other suitable coupling

Managing Coupling Facility Resources

facility exists in the CFRM policy for a particular structure, it might be necessary to update the CFRM policy so that a coupling facility can be found.)

XCF signaling structures must be rebuilt one at a time, using the
SETXCF START,REBUILD,STRNAME=strname,LOCATION=OTHER

command.

Once all structures have been removed from CF1, it can be removed from the configuration for maintenance.

Repopulating CF1 with Structures

After CF1 has been returned to the sysplex configuration, the operator can issue the

```
SETXCF START,REBUILD,POPULATECF=CF1
```

command to move structures back into CF1. The POPULATECF keyword ensures that only those structures that have CF1 as a higher coupling facility in their preference list than where the structure is currently allocated will be rebuilt in CF1. Thus, the entire set of structures in the CFRM active policy are eligible to be rebuilt in CF1, even if they were not part of the original CF1 evacuation.

As each structure in the CFRM policy is examined, messages to the operator indicate:

- Whether a structure is eligible to be rebuilt.
- Whether a structure is not eligible to be rebuilt because CF1 is not in the structure's preference list.
- Whether the structure is to remain in its current location because, although the structure might be eligible to be rebuilt, CF1 is not higher in the structure's preference list than that in which the structure already resides or there are other reasons why CF1 is not more suitable.
- The successful rebuild of a structure into CF1.

Note that after the population of CF1, if it is necessary to redistribute the structures for a better sysplex balance, the preference lists in the CFRM policy could be updated. Once the updated CFRM policy is activated, the POPULATECF function could again be used for CF1 or any other coupling facility to redistribute the structures.

Table 2 on page 65 provides a summary of MVS operator commands available to assist in the operation of a coupling facility. The complete syntax of each command is in *OS/390 MVS System Commands*.

Managing Coupling Facility Resources

Table 2. Summary of MVS Commands for Managing a Coupling Facility

MVS Command	Use for Coupling Facility
DISPLAY CF	<p>Displays storage and attachment information about coupling facilities attached to the system on which the command is processed. The request can be for one or more named coupling facilities or all coupling facilities. The information displayed includes:</p> <ul style="list-style-type: none"> • Coupling facility unique identifiers • Utilization of control and non-control storage • Device, subchannel, and CHPID status • Utilization of dump space. • Coupling facility level • Volatility status of the coupling facility. <p>DISPLAY CF is the command used to show how coupling facility and structure resources are being used in the sysplex, with the purpose of identifying possible adjustments or modifications to a CFRM policy.</p>
DISPLAY XCF DISPLAY XCF,COUPLE,TYPE=CFRM	<p>Displays information about CFRM couple data sets in use in the sysplex. The information includes:</p> <ul style="list-style-type: none"> • Physical attributes of the CFRM couple data set (name, volume serial number, device address, and time the data set was formatted) • Maximum number of systems that the primary CFRM couple data set can support • Names of the systems using the CFRM couple data set.
DISPLAY XCF,STRUCTURE,...	<p>Displays information about coupling facility structures in a policy. The information displayed may be summary or detailed information, depending on what options are specified on the command. The information includes:</p> <ul style="list-style-type: none"> • The status of each structure in the active CFRM policy • Whether or not the CFRM couple data set needs to be reformatted, and if so, the number of structures and connections to be used in reformatting • In which coupling facility the structure is allocated, with the identifying characteristics of the coupling facility • The structure size, both as specified in the CFRM policy and the actual size allocated • Connections to the structure, both maximum allowed and current, with specific information about each connector • Preference and exclusion list information for the structure • Pending policy information if a policy change is pending for the structure.
DISPLAY XCF,CF,...	<p>Displays information about one or more coupling facilities defined in the active CFRM policy. Summary or detailed information may be obtained. The information includes:</p> <ul style="list-style-type: none"> • Whether or not the CFRM couple data set needs to be reformatted, and if so, the number of structures and connections to be used in reformatting • Name and unique identifiers of each coupling facility as specified by the CFRM policy • Size of the dump space, as specified in the policy • Storage increment size • Names of the systems attached to each coupling facility • Names of the structures allocated in each coupling facility.
DISPLAY XCF,POLICY,TYPE=CFRM	<p>Displays information about the CFRM policy in use in the sysplex. The information includes:</p> <ul style="list-style-type: none"> • The name of the active CFRM policy • The status of the policy • The date and time the policy was started • The date and time the policy was last updated.
SETXCF	

Managing Coupling Facility Resources

Table 2. Summary of MVS Commands for Managing a Coupling Facility (continued)

MVS Command	Use for Coupling Facility
SETXCF COUPLE,TYPE=CFRM,...	Use this command to control use of the CFRM couple data set — to identify the primary CFRM couple data set (PCOUPLE=), the alternate CFRM couple data set (ACOUPLE=), or to switch from using the primary to using the alternate (PSWITCH).
SETXCF START,...	Use this command to initiate processing for coupling facility resources. SETXCF START,POLICY,TYPE=CFRM is used to activate a CFRM administrative policy in the sysplex SETXCF START,REBUILD,STRNAME or CFNAME is used to initiate the rebuild process for one or more named coupling facility structures or for all structures in a coupling facility. The system does not start rebuild for XCF signaling structures when the command specifies all structures in a named coupling facility. The PATHIN and PATHOUT options with STRNAME are used to start the use of a structure as an inbound or outbound XCF signaling path.
SETXCF STOP,...	Use this command to halt the processing that is currently being done for coupling facility resources. SETXCF STOP,POLICY,TYPE=CFRM is used to deactivate the current active policy. SETXCF STOP,REBUILD,STRNAME or CFNAME is used to discontinue the rebuild process for one or more named coupling facility structures or for all structures in a coupling facility. The PATHIN and PATHOUT options with STRNAME are used to discontinue the use of a structure as an inbound or outbound XCF signaling path.
SETXCF FORCE,...	Use this command to clean up coupling facility resources in a coupling facility. The resource can be either a structure actively in use in the sysplex, a connection to a structure, a dump associated with a structure, all failed-persistent connections to a structure, or dump serialization for a structure.
SETXCF MODIFY,...	Use this command to change structure attributes of a coupling facility structure defined for XCF signaling.
DUMP	Use this command to dump cache and list structure information, both control information and data. Control information includes directory/registration information, list controls, list entry controls, storage and castout class controls, and optionally adjunct data; data includes list or cache entry data and optionally adjunct data. Lock structures cannot be dumped.
CONFIG CHP	Use this command to change the physical status of a CHPID for a coupling facility link. The DISPLAY MATRIX command displays the CHPID information for coupling facility CHPIDs.
VARY	Use this command to change the state of the logical paths that lead to a coupling facility. The DISPLAY CF command displays the CHPIDs associated with a logical path to a coupling facility. Note that a logical path to a coupling facility is designated by a CHPID and a coupling facility name, not a CHPID and device number.

Coupling Facility Structures for IBM Products

Each application that uses a coupling facility structure must provide the information necessary to define the structure in a CFRM policy. In OS/390, information about the following structures for IBM products can be found in the associated documentation.

Managing Coupling Facility Resources

Table 3. Coupling Facility Structures in OS/390 for IBM Products

Structure Function	Structure Type	Where Documented
XCF signaling	List	See “Chapter 5. Planning Signaling Services in a Sysplex” on page 69 in this book.
Note: For communications in the sysplex, JES3 uses the XCF signaling structure if it is defined. As a result, you do not need to define JES3 signaling structures in the sysplex. For information about JES3 and the sysplex, see <i>OS/390 JES3 Initialization and Tuning Guide</i> .		
VTAM generic resources function	List	<i>OS/390 IBM Communications Server: SNA Network Implementation Guide</i>
JES2 checkpoint data set in a coupling facility	List	<i>OS/390 JES2 Initialization and Tuning Guide</i>
IMS DBCTL DL/I datasharing with IRLM 2.1 <ul style="list-style-type: none"> IRLM locking OSAM buffer invalidate VSAM buffer invalidate 	<ul style="list-style-type: none"> Lock (IRLM) Cache (OSAM) Cache (VSAM) 	<i>OS/390 Parallel Sysplex Application Migration</i>
IMS DEDB shared VSO data sharing	Cache	<i>OS/390 Parallel Sysplex Application Migration</i>
IMS shared message queues	List	<i>OS/390 Parallel Sysplex Application Migration</i> <i>IMS/ESA Common Queue Server Guide and Reference</i>
RACF data base in a large sysplex environment	Cache	<i>OS/390 SecureWay Security Server RACF System Programmer's Guide</i>
Automatic tape switching	List	See “Appendix F. Defining an IEFAUTOS Structure for Automatic Tape Switching” on page 349 in this book.
DB2 data sharing	<ul style="list-style-type: none"> Cache Lock List 	<i>DATABASE2 V4 Data Sharing: Planning and Administration</i>
Log stream data sharing	List	See “Chapter 9. Planning for System Logger Applications” on page 173 in this book.
CICS logging	List	<i>CICS Transaction Server for OS/390 Installation Guide</i>
Global resource serialization star complex	Lock	<i>OS/390 MVS Planning: Global Resource Serialization</i>
CICS temporary storage data sharing	List	<i>CICS System Definition Guide</i>
VSAM record level sharing	<ul style="list-style-type: none"> Cache Lock 	<i>OS/390 DFSMSdfp Storage Administration Reference</i>
Batch Pipes	List	<i>BatchPipes OS/390 Users Guide and Reference</i>
Enhanced Catalog Sharing	Cache	<i>OS/390 DFSMS: Managing Catalogs</i>

|
|
|
|

Managing Coupling Facility Resources

Table 3. Coupling Facility Structures in OS/390 for IBM Products (continued)

Structure Function	Structure Type	Where Documented
WLM multisystem enclaves	Cache	<i>OS/390 MVS Planning: Workload Management</i>

Chapter 5. Planning Signaling Services in a Sysplex

Signaling is the mechanism through which XCF group members communicate in a sysplex.

In a multisystem sysplex, signaling can be achieved through:

- Channel-to channel (CTC) communication connections, which can be:
 - An ESCON channel operating in CTC mode
 - The 3088 Multisystem Channel Communication Unit
- A coupling facility (using coupling facility list structures)
- A combination of CTC connections and list structures.

Full signaling connectivity is required between all systems in a sysplex; that is, there must be an outbound and an inbound path between each pair of systems in the sysplex.

To avoid a single point of signaling connectivity failure, IBM recommends redundant connections between each pair of systems, through either CTC connections or coupling facility list structures.

| This chapter is in two parts. The first part outlines the basic components of
| signaling services in a sysplex. The second part provides an IBM Washington
| Systems Center Flash, “Parallel Sysplex Performance: XCF Performance
| Considerations”. The Flash presents recommendations for signaling configurations
| and tuning options and is based on studies conducted by WSC personnel. Together,
| these two parts of the chapter should provide a basic understanding of the tasks
| required to set up and maintain XCF signaling services in a sysplex.

The first part of this chapter includes the following topics:

- “Recommended Steps for Setting Up Signaling”
- “Overview of Available Signaling Options”
- “Setting Up Outbound and Inbound Signaling Paths” on page 72
- “Planning for Optimum Signaling Performance” on page 76
- “Sample Signaling Scenarios” on page 85.

Recommended Steps for Setting Up Signaling

Following is the general recommended sequence for setting up signaling and tuning signaling performance in a sysplex.

1. Establish signaling paths using the IBM default values.
2. Examine RMF reports to determine adjustments that can be made to improve performance.
3. Adjust transport classes and message buffers, as needed.

Overview of Available Signaling Options

Figure 3 on page 70 is a diagram of a four-system sysplex in which signaling is implemented through CTC connections. The diagram shows the minimum required signaling connectivity for a four-system sysplex.

Signaling

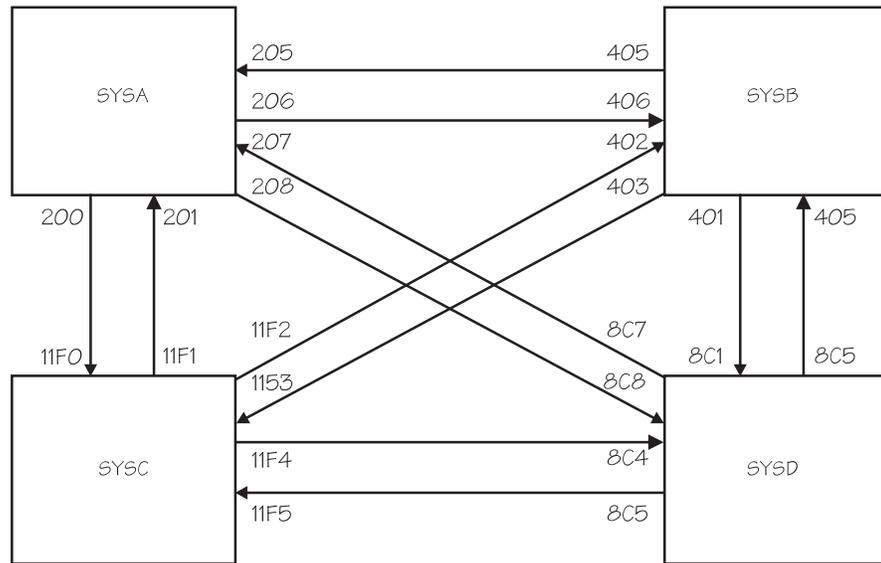


Figure 3. Signaling on a Four-System Sysplex, Implemented through CTC Connections

The following figure shows the PATHOUT and PATHIN statements for SYSA in Figure 3. Note that the address of each device is specified for each DEVICE keyword.

```

/* Device that leads to SYSB. */
PATHOUT  DEVICE(206)

/* Device that leads to SYSC. */
PATHOUT  DEVICE(200)

/* Device that leads to SYSD. */
PATHOUT  DEVICE(208)

/* Device that comes from SYSB. */
PATHIN   DEVICE(205)

/* Device that comes from SYSC. */
PATHIN   DEVICE(201)

/* Device that comes from SYSD. */
PATHIN   DEVICE(207)

```

Figure 4. Example of SYSA's PATHIN and PATHOUT Specifications in COUPLExx

Figure 5 is a diagram of a four-system sysplex in which signaling is implemented through a coupling facility list structure. The diagram shows the minimum required signaling connectivity for a four-system sysplex.

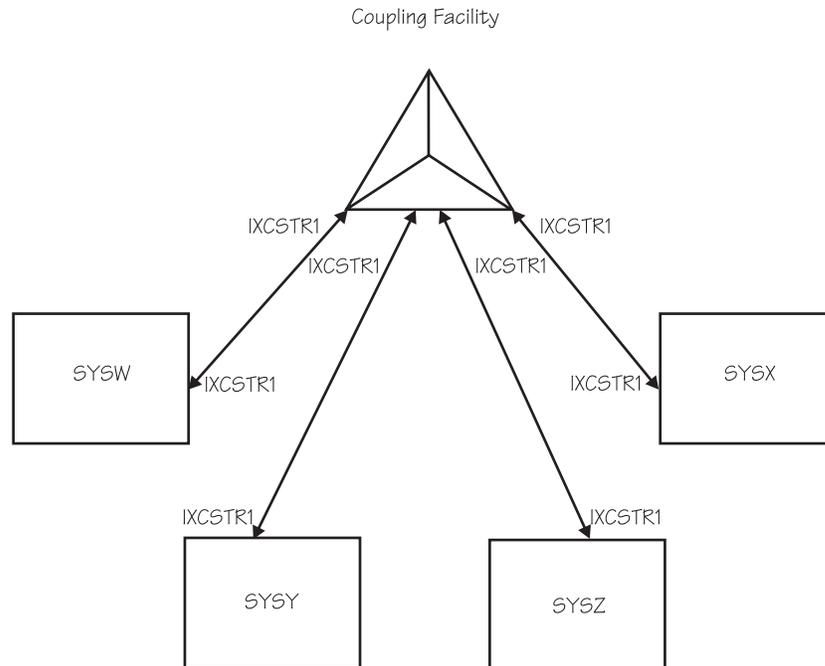


Figure 5. Signaling on a Four-System Sysplex, Implemented through a Coupling Facility List Structure

The following figure shows the PATHIN and PATHOUT specifications in COUPLExx for SYSW shown in Figure 5. Note that the same structure is specified for both PATHOUT and PATHIN and that the same specifications would be used on SYSX, SYSY, and SYSZ.

```

/* Structures that define paths to SYSX, SYSY, and SYSZ. */
PATHOUT STRNAME(IXCSTR1)

/* Structures that define paths from SYSX, SYSY, and SYSZ. */
PATHIN STRNAME(IXCSTR1)

```

Figure 6. Example of SYSW's PATHIN and PATHOUT Specifications in COUPLExx

Advantages of Implementing Signaling through List Structures

Implementing signaling through coupling facility list structures provides significant advantages in the areas of systems management and recovery and, thus, provides enhanced availability for sysplex systems.

While a signaling path defined through CTC connections must be exclusively defined as either outbound or inbound, you can define a coupling facility list structure so that it is used for both outbound and inbound signaling paths, and MVS automatically establishes the paths. For example, if you define a list structure for outbound message traffic, MVS automatically establishes a signaling path with every other system that has the structure defined for inbound message traffic. Similarly, if you define a list structure for inbound traffic, MVS automatically establishes a signaling path with every other system that has the structure defined for outbound traffic.

Because a single list structure can be defined for both inbound and outbound traffic, you can use the same COUPLExx parmlib member for each system in the sysplex.

Signaling

A CTC connection, in contrast, cannot be defined to a single system as both inbound and outbound. Therefore, with CTCs, you must specify a unique COUPLExx member for each system in the sysplex or configure your systems so that they can use the same COUPLExx member by overspecifying the number of devices for signaling paths and tolerating failure messages related to unavailable devices and other configuration errors.

Implementing signaling through coupling facility list structures also enhances the recovery capability of signaling paths and reduces the amount of operator intervention required to run the sysplex.

Setting Up Outbound and Inbound Signaling Paths

Whether you implement signaling through CTC connections or coupling facility list structures, you need to specify information in the COUPLExx parmlib member. After IPL, you can issue the SETXCF START,PATHOUT or SETXCF START,PATHIN command to specify outbound or inbound paths. To establish signaling through a coupling facility, you also need to define a coupling facility resource management (CFRM) policy.

Implementing Signaling through CTC Connections

Signaling paths implemented through CTC connections are one directional. That is, on each system, messages sent to other systems require an outbound path, and messages received from other systems require a separate inbound path.

To establish signaling paths through CTC devices, you specify their device numbers. For each system in the sysplex, you must specify all devices for outbound paths and inbound paths on the DEVICE keyword of the PATHOUT and PATHIN statements in COUPLExx.

When you issue SETXCF commands to start, stop, or modify signaling paths, you also need to specify the addresses of the devices.

Implementing Signaling Paths through Coupling Facility List Structures

When you define signaling paths through coupling facility list structures, you define to MVS which list structures to use and how to use them (as pathin or pathout) and MVS creates the logical connections between systems that are using the structure for signaling.

To implement signaling paths through list structures, you need to set up a CFRM policy and define one or more signaling list structures for PATHIN and/or PATHOUT in COUPLExx. The name of each list structure you define for XCF signaling must begin with the letters IXC. After IPL, you can use commands to define signaling list structures for PATHIN and/or PATHOUT.

Planning a CFRM Policy for Signaling

“Chapter 4. Managing Coupling Facility Resources” on page 37 provides considerations for setting up a CFRM policy. You set up your CFRM policy using the administrative data utility (IXCMIAPU), which resides in SYS1.MIGLIB. See “Appendix B. Administrative Data Utility” on page 263.

In the CFRM policy, you specify the name of the list structure, its size, and a preference list (a prioritized list of the coupling facilities on which you want the list

structure allocated). Optionally, you can also specify an exclusion list — the names of any list structures you do not want allocated on the coupling facility the specified list structure is actually allocated on.

Avoiding a Single Point of Failure: To avoid a single point of failure in the sysplex, specify the following in the CFRM policy:

- The names of at least two list structures for signaling
- For each list structure you define, the names of at least two coupling facilities in the preference list
- Exclusion lists, so that the list structures are allocated in different coupling facilities.

You also need to ensure that there is enough free space in the coupling facilities to allow MVS signaling services to recover from failures by rebuilding list structures.

If you are especially concerned about availability, you can use CTC devices, as well as coupling facilities, for signaling because they provide a different transport mechanism with different hardware and software points of failure. However, this approach increases the complexity of systems management.

Note that the REBUILDPERCENT parameter does not apply to signaling structures.

Supporting Dynamic Alteration of Signaling Structures: An operator can alter (expand or contract) a signaling structure by issuing the SETXCF START,ALTER command. This might be a response, for example, to changes in the signaling workload, to changes in the number of systems using a structure for signaling, or to performance information.

To enable such a structure change to occur without disrupting any connecting systems, CFLEVEL=1 or higher must be specified in the preference list for the structure.

To accommodate expansion of the structure, it is recommended that you define the INITSIZE (initial size) and SIZE (maximum size) keywords for signaling structures so as to allow room for future expansion of the structure,

Determining the Size of a List Structure for Signaling: When you define a list structure, you must specify its size. (If the size you specify is too small, message IXC463I describes the minimum amount of space needed to allocate the structure.)

The size of a signaling list structure depends on the number of MVS systems in the sysplex, the level of the coupling facility, the level of MVS you are running, and the size, frequency, and performance requirements for the messages (signals) that the list structure is to accommodate.

The following table shows the minimum number of 1K blocks and the recommended number of 1K blocks to be allocated for a signaling structure in the coupling facility.

The minimal structure size is calculated based on the information presented in “Appendix C. Calculating Sizes for Signaling Services” on page 309.

The recommended size is based on the following algorithm.

Recommended size = minimal size + ((number of systems*number of systems)*500)

Signaling

where 500 is the approximate size of eight 61K signals queued simultaneously on every list header used as a path (the 500 could be a range of 125-4K signals to 8-61K signals).

Table 4. Sizes for Signaling Structures

Number of systems	Minimal structure size in KB	Recommended structure size in KB
2	512	2512
3	768	5268
4	940	8,940
5	1,492	13,992
6	2,180	20,180
7	3,004	27,504
8	3,968	35,968
9	5,068	45,568
10	6,308	56,308
11	7,680	68,180
12	9,192	81,192
13	10,840	95,340
14	12,624	110,624
15	14,548	127,048
16	16,604	144,604
17	18,800	163,300
18	21,136	183,136
19	23,604	204,104
20	26,212	226,212
21	28,956	249,456
22	31,836	273,836
23	34,856	299,356
24	38,016	326,016
25	41,300	353,800
26	44,732	382,732
27	48,296	412,796
28	52,000	444,000
29	55,840	476,340
30	59,816	509,816
31	63,928	544,428
32	68,180	580,180

Detailed information for more precisely calculating the size for a signaling structure is included in “Appendix C. Calculating Sizes for Signaling Services” on page 309. For information on the message characteristics of XCF groups for some MVS components and subsystems, see “XCF Groups” on page 77.

COUPLExx Specifications and Commands

To implement signaling paths through list structures, you specify a list structure for use as PATHIN or PATHOUT and MVS automatically establishes a corresponding inbound or outbound path with all the systems that have specified that structure for pathout or pathin.

To implement signaling through a list structure, you need to specify (on the STRNAME keyword of the PATHOUT and PATHIN statements in COUPLExx) the name of the structure, defined in the CFRM policy, that holds the signaling path connections.

To implement signaling through a list structure, define the structure as both an outbound and an inbound path on every system connected to that facility. MVS creates one path in each direction for each path defined.

Handling Signaling Path Failures

If a signaling path fails, MVS cannot use the path. Whether signaling is implemented through CTC connections or through list structures, if the underlying hardware fails, all the signaling paths using that hardware stop.

Through CTC Devices

If signaling is implemented through CTC devices, the signaling paths cannot be recovered until the hardware recovers. When the hardware recovers, an operator can vary online an operative channel path or device, or the installation can reconnect an ESCON CTC device.

Through Coupling Facility List Structures

If signaling is implemented through coupling facility list structures, and if a coupling facility that holds a list structure fails or if connectivity to a coupling facility that holds a list structure is lost, MVS can rebuild the list structure in another available coupling facility and reestablish signaling paths.

If the list structure itself fails, MVS can rebuild it in the same coupling facility or in another available coupling facility and then reestablish signaling paths.

Handling Signaling Connectivity Failures

If a system loses connectivity to another, message IXC409D is issued, and the operator can try to re-establish connectivity or remove the system from the sysplex.

Or, if all systems are running OS/390 or MVS/ESA SP Version 5, the sysplex failure management policy can be used to isolate one or more disconnected systems from the sysplex. See “Handling Signaling Connectivity Failures” on page 144.

Note that if a structure used for signaling fails and there is no backup structure, signaling connectivity is lost until the structure is rebuilt. During this time, if the sysplex failure management (SFM) policy is active, SFM might take action before the structure is rebuilt, and the sysplex could end up with only one surviving system. Therefore, signaling path redundancy is strongly recommended, particularly when SFM is active in the sysplex.

Deleting Inoperative Signaling Paths

An inoperative signaling path remains defined to MVS. DISPLAY XCF lists the status of such a signaling path as INOPERATIVE. You can delete the definition of the inoperative path by issuing SETXCF STOP for the signaling path.

Communicating Signaling Path Information to Operators

It is recommended that you provide your operator with a diagram illustrating the layout of your signaling configuration. This diagram should include system names, device numbers assigned to inbound and outbound signaling paths or the names of list structures in which these paths are defined, and corresponding COUPLExx parmlib definitions. If coupling facilities are in use, the diagram should also include coupling facility-related information, such as their channel addresses. The operator will need this information if MVS is unable to establish connectivity between systems and requests operator intervention.

Planning the Retry Limit for Paths

MVS maintains a retry count for each path in the sysplex. When an error occurs on a path, MVS retries the path. If the retry is unsuccessful, MVS adds one to the retry

Signaling

count; if the retry is successful, MVS subtracts one from the retry count. When the retry count matches the specified RETRY value, MVS stops the path.

You specify a path's retry limit on the RETRY keyword of the PATHIN and PATHOUT statements (or take the default specified on the RETRY keyword of the COUPLE statement). After IPL, you can specify retry limits using various SETXCF commands.

The retry limit for a signaling path defined through a list structure becomes the retry limit for every signaling path, in that direction (outbound or inbound), on the system that uses that structure for signaling.

When selecting a retry limit, consider that:

- A large retry limit delays the removal of an inoperative or error-prone path.
- A small retry limit hastens the removal of an inoperative or error-prone path. (Infrequent errors may be tolerable.)

For most installations, the default retry limit of 10 should be satisfactory. However, in a sysplex that includes MVS/ESA SP Version 4 systems (as well as SP Version 5 or OS/390 systems), IBM recommends increasing the retry value on the SP Version 4 systems to at least 15. This increase would allow the Version 4 systems to tolerate the additional retry attempts that later systems sometimes make when attempting to establish signaling connectivity.

Planning for Optimum Signaling Performance

Optimal signaling performance for a sysplex implies that XCF groups in the sysplex have access to the signaling resources (signaling paths and message buffers) they need to communicate in a way that meets the overall communication goals for the sysplex.

IBM recommends using only defaults when initially setting up a sysplex and then changing values, assigning transport classes, and adding paths as you evaluate the operation of the sysplex.

To tune signaling performance, you need a way to separate XCF groups into different categories, as well as a way to associate these categories with specific signaling resources.

A transport class is MVS's way of enabling you to associate one or more XCF groups (based on similar signaling requirements) and then assign them signaling resources (signaling paths and message buffers).

To plan the signaling configuration, you need to:

- Identify the XCF groups in your sysplex and consider their signaling resource requirements.
- On each system, associate groups with similar signaling requirements and assign them to a transport class.
- Assign to each transport class the signaling paths and message buffers to meet these signaling requirements.

This information is presented as follows:

- "XCF Groups" on page 77
- "Planning the Transport Classes" on page 77
- "Planning the Message Buffer Space" on page 80

XCF Groups

An XCF group is a set of related members that a multisystem application defines to XCF. A member is a specific function, or instance, of the application. A member resides on one system and can communicate with other members of the same group across the sysplex.

Communication between group members on different systems occurs over the signaling paths that connect the systems; on the same system, communication between group members occurs through local signaling services.

To prevent multisystem applications from interfering with one another, each XCF group name in the sysplex must be unique.

A multisystem application can use XCF group services (a set of authorized assembler macros) to obtain information about groups and members, define its group members to the sysplex, disassociate members from XCF services, and so forth. For more information about these macros, see *OS/390 MVS Programming: Sysplex Services Guide*.

XCF Group Names

To associate an XCF group and its members with a transport class, you need to know its XCF group name.

XCF group names for some MVS components and IBM products are included in the discussion of “Group Names and Members for MVS Components and Subsystems” on page 31.

You also need to know the XCF group names for any other multisystem application programs that communicate among members on various systems. An application programmer who codes a multisystem application that uses signaling services should give you the application’s XCF group name so that you can designate transport class(es), if needed, for the group’s use. Otherwise, for a list of the XCF groups defined on the system, you can issue the DISPLAY XCF, GROUP command or run the RMF XCF Activity Report.

Message Characteristics for XCF Groups

To define suitable transport classes for an XCF group, you must know the lengths, frequency and performance requirements for messages sent by the group.

- For message information on MVS groups, see “Message Information for Some MVS Groups” on page 311.
- “Coupling Facility Structures for IBM Products” on page 66 provides reference sources for IBM product applications.
- An application programmer who codes a multisystem application that uses signaling services should provide you with the lengths, frequency, and performance requirements for messages sent by members of the group.

The RMF XCF Activity Report is another source of information about messages for the groups in the system.

Planning the Transport Classes

A transport class is MVS’s way of enabling you to associate one or more XCF groups (based on similar signaling requirements) and then assign them signaling resources (signaling paths and message buffers). A transport class allows you to segregate message traffic according to the needs of an application’s group, the lengths of its messages, or both.

Signaling

For example, you can define a transport class that dedicates signaling resources to a particular application, such as global resource serialization (the SYSGRS group). This would ensure the signaling performance of the SYSGRS group because other applications would not compete for signaling resources in that class.

Defining Transport Classes

Transport classes are defined independently for each system in the sysplex using the CLASS keyword of the CLASSDEF statement or, after IPL, using the SETXCF START,CLASSDEF command.

```
[CLASSDEF                                     ]
[      CLASS(class-name)                     ]
[      [CLASSLEN(class-length)               ]
[      [GROUP(group-name[,group-name]...)   ]
[      [MAXMSG(max-messages)                ]
[      ]                                     ]
```

Figure 7. Syntax for CLASSDEF Statement of COUPLExx Parmlib Member

On a system, each transport class must have a unique name. The class name is used in system commands and shown in display output and reports.

For some applications, the same transport class definition is suitable for all systems in the sysplex. For example, an application, such as global resource serialization, whose members in the SYSGRS XCF group on the various systems in the sysplex operate in a peer relationship, has similar message traffic patterns on all systems. In this case, defining the same transport class (such as TCGRS) on each system simplifies the task of reviewing measurement reports for the sysplex.

Assigning XCF Groups to a Transport Class

By explicitly assigning an XCF group to a transport class, you give the group priority access to the signaling resources (signaling paths and message buffer space) of the transport class. All groups assigned to a transport class have equal access to the signaling resources of that class.

Assign one or more XCF groups to a transport class using the GROUP keyword of the CLASSDEF statement or, after IPL, using the SETXCF START,CLASSDEF command or the SETXCF MODIFY,CLASSDEF command.

Undesignated Groups: If you do not specify the GROUP keyword on a CLASSDEF statement, no groups are explicitly assigned to the transport class. A group that you do not assign to a transport class is called an undesignated group. The set of all undesignated groups is collectively referred to on the CLASSDEF keyword and system commands by the pseudo-group name UNDESIG. The undesignated groups are implicitly assigned to the transport classes whose CLASSDEF statement did not specify the GROUP keyword. The undesignated groups can be explicitly assigned to a transport class by specifying GROUP(UNDESIG) on the CLASSDEF statement. After IPL, additional groups (including the UNDESIG set of groups) can be explicitly assigned to a transport class using the SETXCF START,CLASSDEF command or the SETXCF MODIFY,CLASSDEF command.

Using the Default Transport Class

The default transport class (named DEFAULT) is adequate for many installations, and IBM recommends that you use only the default class when you first set up a sysplex. When you do this, all signaling resources (message buffers and signaling paths) are assigned to the default class and are shared by all groups. Then, as you refine your signaling requirements, you can define additional transport classes and

assign groups to the classes. To determine whether to modify your transport class specifications, you need to answer questions, such as:

- Is a sufficient amount of signaling resource provided?
- Are the signaling resources well utilized?
- Will segregating message traffic by group or message length improve performance?
- What signaling resources will be needed as workloads and configurations change?

To answer these questions, you collect data using a tool such as the Resource Measurement Facility (RMF), or predict activity based on input from users of the signaling service.

Specifying the Lengths of Messages for the Transport Class

Specify the length of the messages for which the signaling service should optimize in a transport class on the CLASSLEN keyword of the CLASSDEF statement or, after IPL, on the SETXCF START,CLASSDEF or the SETXCF MODIFY,CLASSDEF command.

The class length determines the size of the XCF message buffers used to hold messages sent in that transport class. MVS selects the smallest buffer that will hold messages of the size specified on the CLASSLEN keyword. In general, specify a class length equal to the length of the messages most frequently sent by the group(s) in the transport class.

Specify the length of the messages (CLASSLEN) for a transport class based on the group(s) you assign to the transport class. Therefore, you must know the characteristics of the messages the groups send.

Selecting an appropriate class length involves making compromises between storage use and signaling performance. If messages are usually shorter than the message buffer, storage is wasted; if longer, performance is degraded.

A message whose length does not exceed the buffer length receives the best performance the signaling service can provide. Messages longer than the buffer length might require additional processing, such as the preparation of additional message buffer space or the sending of additional signals to deliver the oversized messages.

If there are many messages longer than the buffer length and the message traffic patterns warrant it, the signaling service dynamically increases the size of the buffers in the class to avoid the additional processing overhead to deliver these messages. This dynamic adjustment of buffer sizes means that specifying a small class length does not necessarily mean that larger signals will suffer performance degradation. However, if increasing the buffer size would cause the signaling service to exceed the maximum message buffer space allowed for messages in a particular transport class for a particular receiving system, these adjustments are not made. (See “Planning the Message Buffer Space” on page 80.)

Planning the Signaling Paths for the Transport Class

You assign outbound or local signaling paths to a transport class using the CLASS keyword of the PATHOUT or LOCALMSG statements or, after IPL, using the SETXCF (START or MODIFY) PATH command. (Inbound signaling paths are not segregated by transport class.)

Signaling

```
[PATHIN                                     ]
[   {DEVICE(device-number[,device-number]...)} ]
[   {STRNAME(strname[,strname]...)} ]
[   [MAXMSG(max-messages)] ]
[   [RETRY(retry-limit) ] ]
[PATHOUT                                    ]
[   {DEVICE(device-number[,device-number]...)} ]
[   {STRNAME(strname[,strname]...)} ]
[   [MAXMSG(max-messages)] ]
[   [RETRY(retry-limit) ] ]
[   [CLASS(class-name)] ]
[LOCALMSG                                  ]
[   MAXMSG(max-messages) ]
[   [CLASS(class-name)] ]
```

Figure 8. Syntax for PATHIN, PATHOUT, LOCALMSG on COUPLExx Parmlib Member

If you implement signaling through a list structure, you use the STRNAME keyword of the PATHIN or PATHOUT statement to specify the name(s) of the list structure(s) you want to use for PATHIN or PATHOUT on that system. If you assign a structure to a transport class, every outbound signaling path that MVS starts through that structure to other systems is also assigned to that transport class. (If the list structure is not explicitly assigned to a transport class, it is implicitly assigned to the default transport class.)

If you implement signaling through CTCs, you can assign each CTC to a transport class. Because a CTC connects to a specific system, you have more flexibility to add paths between systems within a class.

Assign signaling paths to a transport class based on the amount of message traffic you expect the class to handle. One signaling path may be sufficient for most transport classes; however, assign two (or more) paths if you want a redundant path available to the transport class or require additional capacity.

An XCF group usually uses only the signaling paths associated with the transport classes to which the group is assigned. However, if there are no paths available to a group in any of its assigned transport classes, messages in the group are routed to other paths.

Failure to assign a signaling path to a transport class can degrade the performance of the signaling service for XCF groups assigned to that class. In this case, the groups compete for the signaling resources of transport classes assigned to other XCF groups.

Planning the Message Buffer Space

Messages sent and received through sysplex signaling services occupy a message buffer in central storage while being processed. These buffers are allocated as needed to support the message traffic load.

At times, runaway applications, non-operational remote systems, or structure rebuilds can cause message traffic to back up to the point that the amount of storage acquired for signaling would degrade the rest of the system.

The MAXMSG parameter allows you to limit the total amount of storage (message buffer space) available to a transport class or signaling path.

Specifying Message Buffer Space

You specify the maximum message buffer space on the MAXMSG keyword of the COUPLE, CLASSDEF, PATHOUT, PATHIN, and LOCALMSG statements in the COUPLExx parmliib member. If you implement signaling through list structures, the MAXMSG value takes effect for all paths on that system that are started through that structure. You can modify these values using the SETXCF MODIFY command.

Table 5 summarizes the MAXMSG values that the system uses if MAXMSG is not specified on the statement:

Table 5. MAXMSG Values on COUPLE Statements

Statement	MAXMSG Value If Not Coded on Statement
COUPLE	750 K
CLASSDEF	MAXMSG value if coded on COUPLE statement; otherwise, 750 K
PATHOUT	MAXMSG value if coded on CLASSDEF statement for the transport class. If MAXMSG is not coded on CLASSDEF statement, then MAXMSG value for COUPLE statement. Otherwise, 750 K
PATHIN	MAXMSG value if coded on COUPLE statement; otherwise, 750 K
LOCALMSG	None; that is, zero

Kinds of Message Buffers

There are three kinds of message buffers. Outbound and inbound message buffers allow members of groups to communicate with other systems. Outbound message buffers are used to send messages to another system; inbound message buffers are used to receive messages from another system. Local message buffers are used to send and receive messages within the same system.

The various types of message buffers are segregated from one another so that the supply of one type does not affect the supply of any other type. Message buffers used to communicate with one system are segregated from those used to communicate with other systems. Thus, exhausting the supply of message buffers used to communicate with one system does not impact communication with any other system.

Outbound and Local Message Buffers

Outbound and local message buffers can be further segregated by transport class. Thus, the supply of message buffers in one transport class for a particular system does not affect the supply of message buffers available to any other system in the same transport class.

Figure 9 on page 82 shows SYSA's message buffers with the local and outbound buffers segregated by class (TC3, TC4 and TC1, TC2) and the outbound buffers segregated by system (SYSB, SYSC, SYSD). SYSA's inbound message buffers are segregated by system only (SYSB, SYSC, SYSD).

Signaling

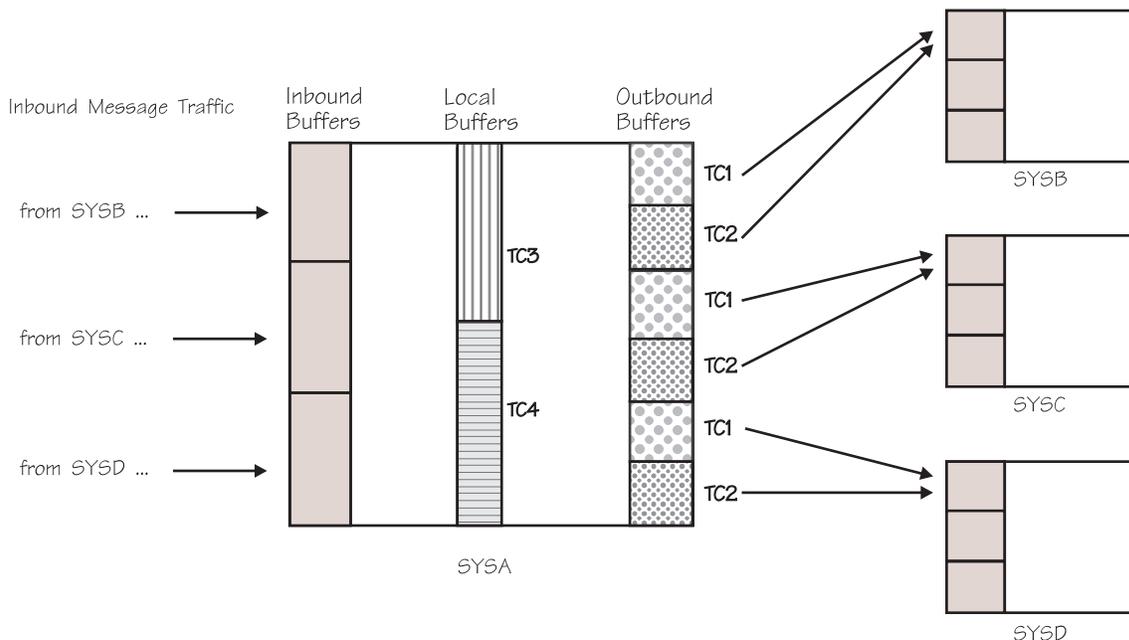


Figure 9. SYSA with Inbound, Local and Outbound Message Buffers

The message buffer space available to outbound messages in a transport class is the sum of the message buffer space contributed by the transport class plus the message buffer space contributed by each outbound signaling path in that class. If the available message buffer space is not adequate, then messages are rejected.

Because the message buffer space available for communicating with another system depends on the contribution of message buffer space by the outbound signaling paths connected to that system, the loss of a signaling path can cause a reduction in the available buffer space. Moreover, the operator can use commands to change the amount of message buffer space contributed by a signaling path or a transport class definition. Therefore, it is possible for the amount of message buffer space in use to exceed the buffer space limit. In this situation, MVS rejects message requests until the buffer space in use is consistent with the new buffer space limit.

The total message buffer space available to local messages in a transport class is the sum of the message buffer space contributed by the transport class (specified or defaulted on the MAXMSG parameter of the CLASSDEF or COUPLE statement) plus any message buffer space specified on the LOCALMSG statement.

You want to specify sufficient message buffer space to support the message traffic for each system in each transport class that is used for delivering messages to that system. When specifying buffer space for a transport class, consider the size and frequency of the messages, as well as the performance of the signaling paths and systems involved in the message transfer.

Inbound Message Buffers

Inbound message buffers are segregated by signaling path. Buffers associated with an inbound signaling path do not receive messages over any other inbound signaling path.

The message buffer space available to inbound messages is specified or defaulted on the MAXMSG parameter of the PATHIN statement. If you implement signaling with CTCs, you can specify the message buffer space for each inbound signaling path. If you implement signaling through list structures, the message buffer space you define for the structure on a system takes effect for all the paths started by that system through that list structure.

When specifying message buffer space for an inbound signaling path, consider the size and frequency of the messages to be received on that path. You want to provide enough message buffer space to allow the inbound signaling path to receive messages without causing messages to back up on the sending system. To determine the message size, note the class length of the sending system's transport class.

If the message buffer space for an inbound signaling path is too small to support a message sent from a remote system, MVS exceeds the buffer space limit to receive the message.

Calculating Message Buffer Space

In general, you should not be concerned with the buffer space required for an individual message. However, you must ensure that the message buffer space is large enough to hold a message of the length specified (on the CLASSLEN keyword) for the transport class.

However, when the message traffic in your sysplex requires you to limit the number of message buffers, you do need to know how much buffer space is required to support a particular message length. For detailed information on this topic, see "Calculating Message Buffer Space — An Example" on page 312.

"Example 3 — Handling a Range of Message Traffic Considerations" on page 90 also has information about calculating message buffer space.

When you define a list structure for signaling, more than one signaling path might be started. Therefore, to compute the message buffer space required by the system, you must anticipate the maximum number of signaling paths that might be started through the list structure.

Summary

To make decisions on maximum message buffer space, you need to understand the peak message traffic loads, distributions of messages among the systems in the sysplex, the sizes of those messages, and the consequences of having a message request rejected due to a lack of buffer space. A measurement tool such as RMF is helpful in determining message traffic patterns and loads. For information about RMF reports, see Chapter 6. Tuning a Sysplex.

Adding and Deleting Signaling Paths

Use the SETXCF START and SETXCF STOP commands to add and delete signaling paths.

Signaling

Adding a Signaling Path

To add a signaling path, issue the SETXCF START command.

Through CTC Devices

To add a signaling path defined through CTCs, the device must be unallocated before you issue the SETXCF START command.

When you add an outbound path on the sending system, also add its associated inbound path on the receiving system.

- To ensure that the device is not allocated, issue:

```
DISPLAY U,CTC,ALLOC
```

- To add an inbound or outbound signaling path, issue:

```
SETXCF START,PATHIN,DEVICE=indevnum  
SETXCF START,PATHOUT,DEVICE=outdevnum
```

Use the MAXMSG, RETRY, and CLASS keywords as needed.

Through Coupling Facility List Structures

To add a signaling path defined through a list structure, the structure must be defined in the active CFRM policy.

To add a signaling path defined in the CFRM policy, issue:

```
SETXCF START,PATHIN,STRNAME=strname  
SETXCF START,PATHOUT,STRNAME=strname
```

Deleting a Signaling Path

To delete a signaling path, issue the SETXCF STOP command.

Through CTC Devices

When you delete an outbound path on the sending system, also delete its associated inbound path on the receiving system so the path is available for other users.

- To ensure that the device for an inbound or outbound path is defined to XCF, issue:

```
DISPLAY XCF,PATHIN,DEVICE=(indevnum)  
DISPLAY XCF,PATHOUT,DEVICE=(outdevnum)
```

- To delete an inbound or outbound signaling path, issue:

```
SETXCF STOP,PATHIN,DEVICE=indevnum  
SETXCF STOP,PATHOUT,DEVICE=outdevnum
```

Use UNCOND=YES when you need to terminate an active stop request that may have failed and initiate a new stop request.

When you stop a path, MVS unallocates the path. The path remains online to the system for other users.

Through Coupling Facility List Structures

To ensure that a structure is defined to MVS for signaling, issue:

```
DISPLAY XCF,PATHIN,STRNAME=strname  
DISPLAY XCF,PATHOUT,STRNAME=strname
```

To ensure that the structure is defined in the active CFRM policy, issue:

```
DISPLAY XCF,STRNAME=strname
```

To delete an inbound or outbound signaling path defined through a list structure, issue:

```
SETXCF STOP,PATHIN,STRNAME=strname
SETXCF STOP,PATHOUT,STRNAME=strname
```

Note that these commands stop all the (inbound or outbound) paths through the structure on the system, as well as the corresponding (outbound or inbound) paths on the systems to which the paths connect.

Considerations

When deleting signaling paths, consider the following:

- The operator cannot stop the last active signaling path to or from another system in the sysplex. MVS requires full connectivity (at least one outbound and one inbound signaling path) between each of pair of systems in a sysplex.
- When an MVS system in the sysplex is running as a guest under VM, exercise care when changing paths. If a device is detached and linked via CP commands while it is online to MVS, MVS boxes the device, which makes it unusable until the guest MVS system is re-IPLed.

Sample Signaling Scenarios

The following topic contains examples of various signaling configurations.

1. “Example 1 — Using COUPLExx Defaults” on page 86

This example is a configuration of two systems that shows only the required statements and parameters and takes the system defaults for all others.

2. “Example 2 — Creating a Transport Class for Large Messages” on page 88

This example is a configuration of two systems that shows the required statements and parameters and those statements and parameters that you can use to segregate message traffic based on message length.

3. “Example 3 — Handling a Range of Message Traffic Considerations” on page 90

This example is a configuration of three systems that shows the required statements and parameters and those statements and parameters that you could use for message traffic that is not symmetric between the systems in the sysplex.

4. “Example 4 — Creating a Transport Class for a Group” on page 98

This example is a configuration of two systems that shows the required statements and parameters and those statements and parameters that you can use to segregate message traffic for a group.

5. “Example 5 — Defining Signaling Paths through the Coupling Facility” on page 100

This example is a configuration of two systems that uses a signaling configuration consisting entirely of coupling facility signaling structures.

6. “Example 6 — Signaling through a Coupling Facility and CTC Connections” on page 102

This example is a configuration of two systems that uses a signaling configuration consisting of both coupling facility signaling structures and CTC devices.

The examples contain the following topics:

- Purpose
- Scenario
- Diagram

Signaling

- Example of the COUPLExx parmlib member

To determine how to calculate the device addresses specified on the PATHIN and PATHOUT statements and shown on the example diagrams, see:

- *3088 Product Description*
- *Introducing Enterprise Systems Connection*

Example 1 — Using COUPLExx Defaults

This example illustrates a sysplex of two systems in which all defaults in the COUPLExx parmlib member are taken. Only the required statements and parameters are shown.

IBM recommends that you use only the defaults when you initially set up a sysplex. You can then examine the XCF activity reports produced by RMF to determine whether a significant number of messages are too big to fit the default buffer sizes. Only in this case should you consider defining transport classes to accommodate larger messages. See “Example 2 — Creating a Transport Class for Large Messages” on page 88.

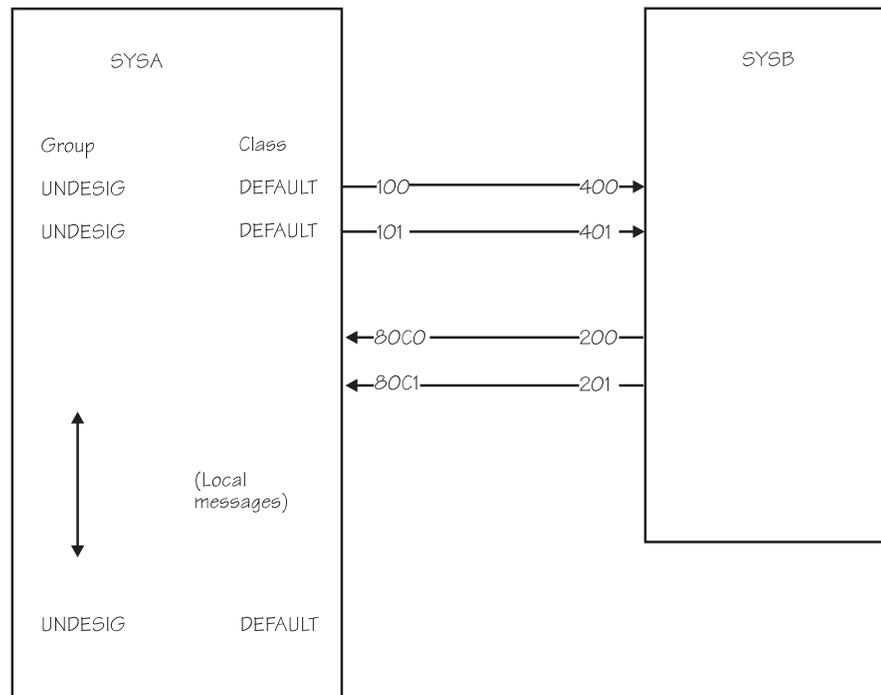
Scenario

PLEXOF2, consists of two MVS systems (SYSA and SYSB) on two processors.

In this example, all defaults are used, and no transport classes are defined. Therefore, MVS creates only the default transport class (named DEFAULT) and assigns all groups (which are collectively called UNDESIG) to the DEFAULT class. Also, MVS assigns all signaling paths to the DEFAULT class, which means that all groups share all paths equally.

Example 1 — Diagram of PLEXOF2

Figure 10 on page 87 is a logical diagram of the example sysplex, PLEXOF2.



Note: In this example, UNDESIG refers collectively to all groups not assigned to a transport class. In this case, every group on the system uses the signalling resources assigned to the transport class DEFAULT.

Figure 10. Example 1 — Diagram of PLEXOF2

Example 1 — SYSA's COUPLExx Parmlib Member

An example of the COUPLExx parmlib member for SYSA in PLEXOF2 is shown in the following figure.

```
COUPLE  SYSPLEX(PLEXOF2)
        PCOUPLE(PLEXOF2.CDS1)
        ACOUPLE(PLEXOF2.CDS2)

/* Devices that lead to SYSB. */
PATHOUT DEVICE(100,101)

/* Devices that come from SYSB. */
PATHIN  DEVICE(80C0,80C1)
```

Figure 11. Example 1 — SYSA's COUPLExx Parmlib Member

The COUPLE Statement:

- SYSPLEX(PLEXOF2) - identifies the name of the sysplex, PLEXOF2.
- PCOUPLE(PLEXOF2.CDS1) - identifies the primary sysplex couple data set.

Signaling

- ACOUPLE(PLEXOF2.CDS2) - identifies the alternate sysplex couple data set. (Although optional, an alternate sysplex couple data set is recommended.)

/* Devices that lead to SYSB. */:

- PATHOUT DEVICE(100,101) - defines outbound signaling paths 100 and 101. MVS assigns the outbound paths to the DEFAULT transport class.

/* Devices that come from SYSB. */:

- PATHIN DEVICE(80C0,80C1) - defines inbound signaling paths 80C0 and 80C1.

Example 1 — SYSB's COUPLExx Parmlib Member

The COUPLExx parmlib member for SYSB in PLEXOF2 is shown in Figure 12.

```
COUPLE  SYSPLEX(PLEXOF2)
        PCOUPLE(PLEXOF2.CDS1)
        ACOUPLE(PLEXOF2.CDS2)

/* Devices that lead to SYSA. */
PATHOUT DEVICE(200,201)

/* Devices that come from SYSA. */
PATHIN  DEVICE(400,401)
```

Figure 12. Example of SYSB's COUPLExx Parmlib Member

Example 2 — Creating a Transport Class for Large Messages

This example illustrates a sysplex of two systems in which the transport class (BIG) is defined to handle messages of up to 40,000 bytes and all other defaults in the COUPLExx parmlib member are taken.

Scenario

The sysplex, PLEXOF2B, consists of two MVS systems (SYSA and SYSB) on two processors. In the sysplex, the length of messages sent by members of various groups vary widely and, thus, messages require special consideration:

- All groups on SYSA have members that send small messages (up to 1,000 bytes) to their group members on SYSB.
- Some groups on SYSA have members that send longer messages (up to 40,000 bytes) to their group members on SYSB.

When there is a large amount of signaling traffic and message lengths vary widely, segregating messages by size, as shown in this example, can often improve signaling performance.

Example 2 — Diagram of PLEXOF2B

Figure 13 on page 89 shows a logical diagram of the example sysplex named PLEXOF2B.

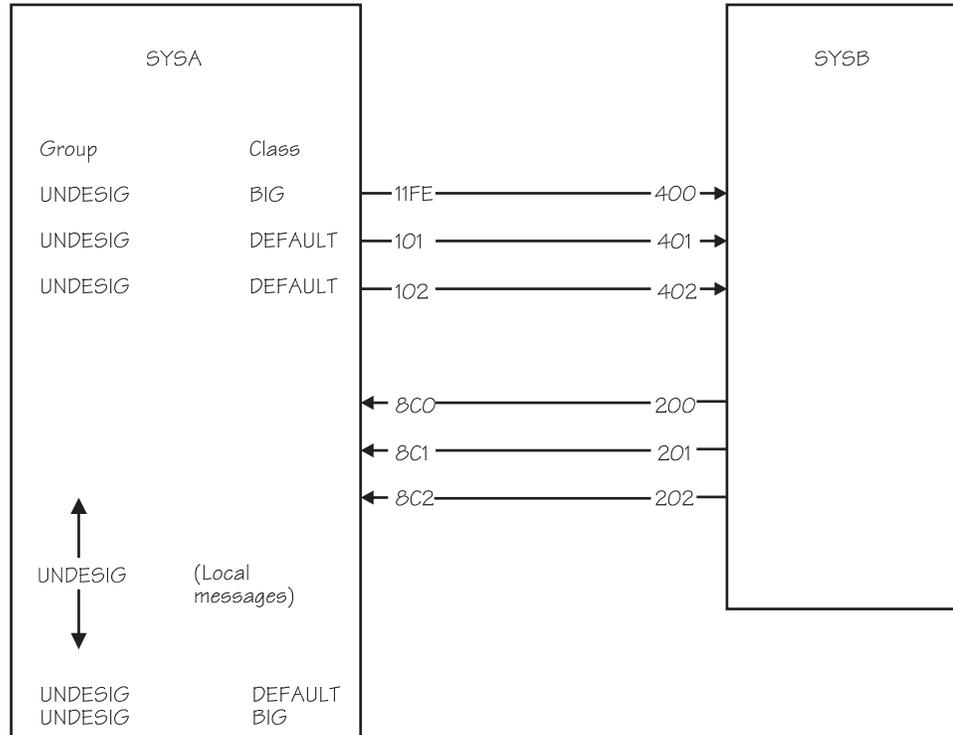


Figure 13. Example 2 — Diagram of PLEXOF2B

Example 2 — SYSA’s COUPLExx Parmlib Member

An example of a COUPLExx parmliib member for SYSA in sysplex PLEXOF2B is shown in the following figure. Assume that defaults are taken for any keywords not coded.

```
COUPLE  SYSPLEX(PLEXOF2B)
        PCOUPLE(PLEXOF2B.CDS1)
        ACOUPLE(PLEXOF2B.CDS2)

/* Transport classes. */

CLASSDEF CLASS(BIG) CLASSLEN(40000)

/* Devices that lead to SYSB. */
PATHOUT DEVICE(11FE) CLASS(BIG)
PATHOUT DEVICE(101,102)

/* Devices that come from SYSB. */
PATHIN  DEVICE(8C0,8C1,8C2)
```

Figure 14. Example 2 — SYSA’s COUPLExx Parmliib Member

The COUPLE Statement:

- SYSPLEX(PLEXOF2B) - identifies the name of the sysplex, PLEXOF2B. The sysplex name specified here must match the sysplex name used in the Format Utility job to format the couple data sets specified by ACOUPLE and PCOUPLE.
- PCOUPLE(PLEXOF2B.CDS1) - identifies the primary sysplex couple data set.
- ACOUPLE(PLEXOF2B.CDS2) - identifies the alternate couple data set.

Signaling

The CLASSDEF statement:

- CLASS(BIG) - specifies BIG as the name of the transport class.
- CLASSLEN(40000) - specifies that messages to be handled by this class are up to 40,000+ bytes long. This specification causes MVS to optimize its processing for messages of up to (slightly more than) 40,000 bytes.

MVS defines the DEFAULT transport class automatically; it is not shown. MVS uses a class length of 956 for class DEFAULT; this length is sufficient for the small messages. Because no group is specifically assigned to any transport class, UNDESIG refers to all groups.

The CLASSDEF statement (for class BIG and class DEFAULT) segregate messages into two categories. Messages that are less than or equal to 956 bytes are sent using resources assigned to the transport class, DEFAULT. Messages between 956 bytes and 40,000+ bytes are sent using resources assigned to the BIG transport class. Note that messages longer than 40,000+ bytes are subject to performance degradation.

/ Devices that lead to SYSB. */:*

- PATHOUT DEVICE(11FE) CLASS(BIG) - assigns outbound signaling path 11FE to class BIG. Outbound messages (the long 40,000-byte messages) for all groups are sent on this path.
- PATHOUT DEVICE(101,102) - defines outbound signaling paths 101 and 102. Outbound messages (the short messages) for all groups are sent on these paths.

/ Devices that come from SYSB. */:*

- PATHIN DEVICE(8C0,8C1,8C2) - defines inbound signaling paths 8C0, 8C1, and 8C2.

Example 3 — Handling a Range of Message Traffic Considerations

This example illustrates what you need to consider when the message traffic between systems in the sysplex have unique considerations. It also illustrates how to calculate message buffer space.

The example is a configuration of three systems that shows the statements and parameters that you could use to provide signaling resources for a multisystem application that has message traffic patterns that are not symmetric.

For this example, it is assumed that the system programmer has a detailed understanding of the multisystem application.

Scenario

The sysplex PLEXOF3 consists of three MVS systems (SYSA, SYSB, and SYSC) on three processors. There is one group, GROUPX, in the sysplex that requires special consideration.

A numerically intensive multisystem application joins GROUPX and has five members: XMASTER, XANALYZE, XRECORD, XCOMP1, and XCOMP2.

XMASTER resides on SYSA and directs the activities of the other members in the group by sending them messages that represent control signals.

XANALYZE also resides on SYSA and analyzes the control signals sent by XMASTER to the other members.

XRECORD resides on SYSC and records status and trace data related to the work of the group.

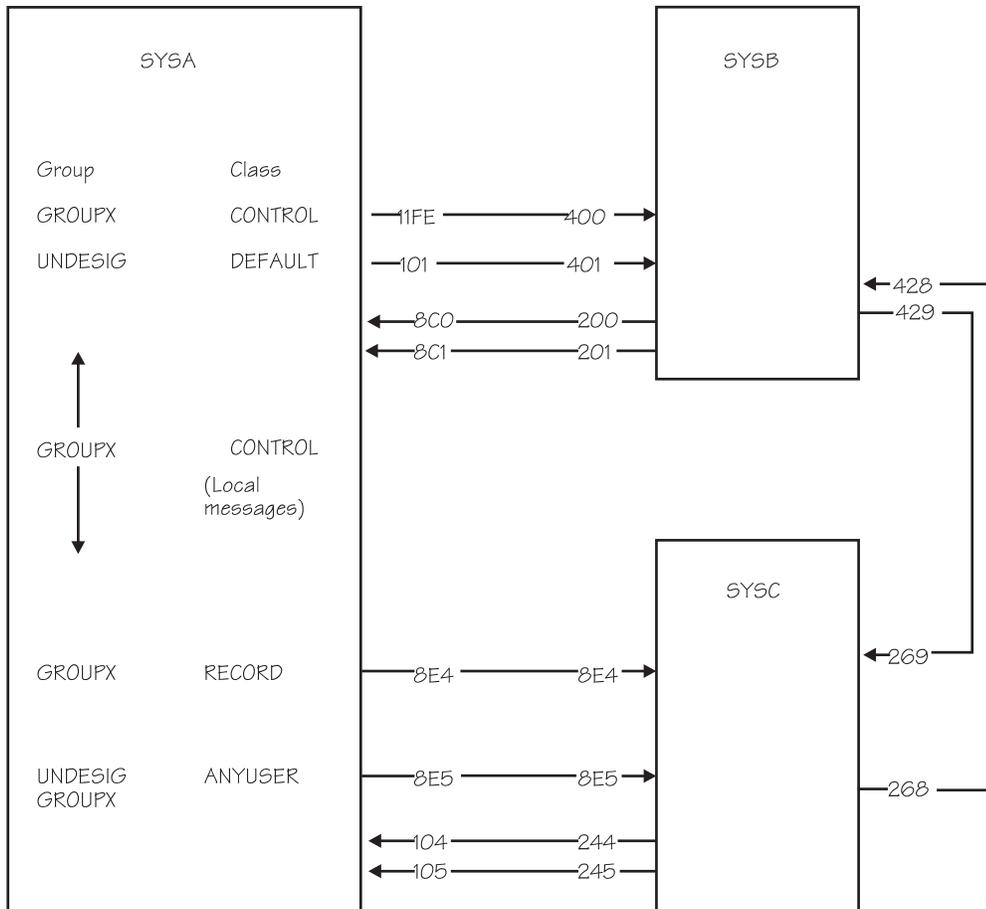
XCOMP1 and XCOMP2 perform CPU-intensive calculations and reside on SYSB to take advantage of hardware features that exist only on the processor on which SYSB is running.

- GROUPX has the following message traffic considerations:
 - XMASTER frequently sends messages (which represent control signals) of up to 300 bytes to XCOMP1 and XCOMP2 on SYSB. These messages are so frequent that they can degrade the performance of other groups that send messages between SYSA and SYSB.
 - XCOMP1 and XCOMP2 frequently reply to XMASTER with 25,000-byte messages containing the results of their calculations.
 - XMASTER periodically sends 300-byte messages (which represent control signals) to XRECORD on SYSC. The messages are not frequent enough to degrade the performance of other groups that send messages between SYSA and SYSC.
 - XMASTER periodically sends 40,000-byte messages (which represent trace records) to XRECORD on SYSC. SYSC is on a much slower processor than SYSA, which means that SYSA can process trace records more quickly than SYSC. The trace records are not critical to the GROUPX application. The sender (XMASTER) can tolerate these messages being refused by MVS and then can resend them at a later time, or even discard them.
 - XRECORD does not receive messages from XCOMP1 and XCOMP2.
 - XRECORD does not send messages to any members of GROUPX.
- Members of other groups in the sysplex have the following message traffic considerations:
 - Members of other groups on all systems periodically send and receive messages of up to 1,000 bytes. This message traffic is served by the DEFAULT transport class.

Example 3 — Diagram of PLEXOF3

The following figure shows a logical diagram of the example sysplex named PLEXOF3.

Signaling



GROUPX - Numeric-intensive application

SYSA Members - XMASTER and XANALYZE

SYSB Members - XCOMP1 and XCOMP2

SYSC Member - XRECORD

Figure 15. Example 3 — Diagram of PLEXOF3

Example 3 — SYSA's COUPLExx Parmlib Member

An example of the COUPLExx parmlib member for SYSA in PLEXOF3 is shown in the following figure. Assume that defaults are taken for any keywords not coded.

```

COUPLE  SYSPLEX(PLEXOF3)
        PCOUPLE(PLEXOF3.CDS1)
        ACOUPLE(PLEXOF3.CDS2)

/* Transport classes. */

CLASSDEF CLASS(RECORD) CLASSLEN(40000) MAXMSG(42)
        GROUP(GROUPX)

CLASSDEF CLASS(CONTROL) CLASSLEN(300)
        GROUP(GROUPX)

CLASSDEF CLASS(ANYUSER)
        GROUP(GROUPX,UNDESIG)

/* Note that MVS defines the DEFAULT class since there is no
   CLASSDEF statement to define it. */

/* Devices that lead to SYSB. */
PATHOUT DEVICE(11FE) CLASS(CONTROL)
PATHOUT DEVICE(101)

/* Devices that lead to SYSC. */
PATHOUT DEVICE(8E4) CLASS(RECORD)
PATHOUT DEVICE(8E5) CLASS(ANYUSER)

/* Devices that come from SYSB. */
PATHIN  DEVICE(8C0) MAXMSG(32)
PATHIN  DEVICE(8C1)

/* Devices that come from SYSC. */
PATHIN  DEVICE(104,105)

/* Local messages. */
LOCALMSG CLASS(CONTROL) MAXMSG(250)

```

Figure 16. Example of SYSA's COUPLExx Parmlib Member

The COUPLE Statement:

- SYSPLEX(PLEXOF3) - identifies the name of the sysplex, PLEXOF3. The sysplex name specified here must match the sysplex name used in the Format Utility job to format the couple data sets specified by ACOUPLE and PCOUPLE.
- PCOUPLE(PLEXOF3.CDS1) - identifies the primary sysplex couple data set.
- ACOUPLE(PLEXOF3.CDS2) - identifies the alternate couple data set.

First CLASSDEF statement:

- CLASS(RECORD) - specifies RECORD as the name of the transport class. This transport class is defined so that the large 40,000-byte messages (the trace records) sent from XMASTER to XRECORD are segregated from the smaller signals sent from SYSA to SYSC. It also allows the amount of message buffer space used for these messages to be finely controlled.
- CLASSLEN(40000) - specifies that the smallest message buffers provided by the signaling service for this transport class can handle 40,000-byte messages.
- MAXMSG(42) - specifies 42K bytes as the default MAXMSG value to use when the MAXMSG keyword is not specified for signaling resources assigned to the RECORD transport class (see the PATHOUT statement for device 8E4). A message buffer size of 42K bytes is sufficient to handle one 40,000-byte message. ("Calculating Message Buffer Space" on page 83 shows how 42K bytes is calculated for the 40,000-byte message.)

Signaling

Buffer Space: The actual maximum amount of message buffer space for this transport class that is made available to each possible remote system is computed by adding the message buffer space contributed by the transport class definition to the message buffer space contributed by the signaling resources assigned to the class that can be used to send messages to the remote system. For this configuration, the maximum amount of buffer space that can be used in this transport class to each possible remote system is:

SYSA - 64K
SYSB - 64K
SYSC - 106K

These values were computed as follows.

MVS always allows for a minimum of 64K bytes of message buffer space for a transport class definition for each system. Since the specified MAXMSG value of 42 is less than 64, MVS uses 64K as the contribution of message buffer space for the transport class definition. From the PATHOUT statements that follow, we see that only device 8E4 is assigned to transport class RECORD. Since the MAXMSG keyword is not specified on the PATHOUT statement that defines device 8E4, the system uses the default value for the transport class to which the device is assigned. In this case that value is 42 since the CLASSDEF statement that defined class RECORD was coded with MAXMSG(42).

Because there are no additional signaling resources assigned to class RECORD that can be used to send messages from SYSA to systems SYSB and SYSC, the contribution to the amount of message buffer space for those non-existent signaling resources is zero. Adding the contribution from the transport class definition (64) and the contribution from the signaling resources (0), we see that the maximum amount of message buffer space that would be used in transport class RECORD to send messages to systems SYSA and SYSB is 64K. In reality, none of the message buffer space allowed for systems SYSA and SYSB would be used because none of the message traffic in this class goes to SYSA or SYSB. For SYSC, the sum of the contribution from the transport class definition (64) and the contribution from the signaling resources (42 from PATHOUT device 8E4) in this class that lead to SYSC is 106. Thus, the maximum amount of message buffer space that would be used to send signals to SYSC in class RECORD is 106K.

Note that the signaling service acquires buffers on an as needed basis. Therefore, the maximum allotment will be used only if required. You normally use MAXMSG limitations to prevent runaway applications or non-operational remote systems from causing message traffic to back up to the point that the amount of storage acquired for signaling would degrade the rest of the system.

- GROUP(GROUPX) - identifies GROUPX as the group that is to use the signaling resources assigned to transport class RECORD. For this configuration, this keyword causes signaling path 8E4 to be dedicated to transferring messages sent by members of GROUPX.

Second CLASSDEF statement:

- CLASS(CONTROL) - specifies CONTROL as the name of the transport class. CONTROL is defined so that the frequent 300-byte messages (which are control signals) sent from XMASTER to XANALYZE, XCOMP1, and XCOMP2 can be isolated from other signaling requests. Since these control signals are sent frequently enough to degrade the performance of other applications, they are segregated into a discrete class so they will not interfere with other message traffic.

- CLASSLEN(300) - specifies that the smallest message buffers provided for this transport class can handle 300-byte messages. The buffers provided to support these 300-byte messages may support somewhat longer messages. The actual length supported can vary from release to release of MVS. The "BUFFER LENGTH" column in the "XCF USAGE BY SYSTEM" report produced by RMF will indicate the actual message length supported by the current release of the system. (See "Chapter 6. Tuning a Sysplex" on page 117.)
- GROUP(GROUPX) - identifies GROUPX as the group that is to use the signaling resources assigned to transport class CONTROL. All signaling resource assigned to class CONTROL will be dedicated exclusively to messages sent by members of GROUPX. GROUPX does not have to compete for signaling resources with other groups in the system. Segregating GROUPX message traffic should improve the performance of both GROUPX and other applications.

Buffer Space: For this configuration, the maximum amount of buffer space that can be used in this transport class to each possible remote system is:

SYSA - 750K
 SYSB - 1000K
 SYSC - 500K

These values were computed as follows.

Because the MAXMSG keyword was not coded on the CLASSDEF statement that defined CONTROL, the default value as determined by the MAXMSG keyword on the COUPLE statement was used. Since the MAXMSG keyword was not coded on the COUPLE statement, the IBM-supplied default of 500 was used. Since the effective MAXMSG value of 500 for class CONTROL exceeds the minimum allotment of 64K bytes of buffer storage contributed by a transport class definition, 500K is used as the contribution for the transport class definition.

Next, the contributions for the signaling resources assigned to class CONTROL that can be used to send messages locally must be added in. The LOCALMSG statement allows an additional 250K of buffer space to be used for sending messages locally (between members on the same system). The contribution from the transport class definition (500) plus the contribution from the LOCALMSG statement (250) allows 750K bytes of storage for local message traffic in class CONTROL. The PATHOUT statement for device 11FE did not specify a MAXMSG keyword. Therefore, the default value (500) for the transport class (CONTROL) to which the path is assigned is used. The contribution from the transport class definition (500) plus the contribution from the PATHOUT devices (500) allows 1000K bytes of storage for message buffers sent between SYSA and SYSB using class CONTROL. There are no additional signaling resources (that is, PATHOUT devices) assigned to class CONTROL that lead to SYSC. The contribution from the transport class definition (500) plus the contribution from the additional signaling resources (0) allows 500K bytes of storage for message buffers sent between SYSA and SYSC using class CONTROL.

Third CLASSDEF statement:

- CLASS(ANYUSER) - specifies ANYUSER as the name of the transport class. This class is defined so that members of any group in the sysplex can send short messages between SYSA and SYSC using device 8E5.
- GROUP(GROUPX,UNDESIG) - specifies that any group can use the signaling resources assigned to the ANYUSER transport class. UNDESIG is the name recognized by MVS for all groups not assigned explicitly to a transport class. Since GROUPX is the only group that has been assigned to a transport class, UNDESIG refers to every group except GROUPX. Listing both GROUPX and

Signaling

UNDESIG therefore includes every group. See also the discussion below on the PATHOUT statement that defines device 8E5.

Buffer Space: For this configuration, the maximum amount of buffer space that can be used in this transport class to each possible remote system is:

SYSA - 500K
SYSB - 500K
SYSC - 1000K

These values were computed as follows.

There are no additional signaling resources assigned to class ANYUSER for local message traffic on SYSA or for message traffic from SYSA to SYSB. Therefore, the buffer allocation is just the contribution of the transport class definition (500). Device 8E5 leads from SYSA to SYSC and is assigned to class ANYUSER. Therefore, the allocation of buffer space in class ANYUSER for communicating with SYSC is the contribution from the definition (500) plus the contribution from the other signaling resources (500) for a total of 1000K bytes of storage.

/ MVS defines the DEFAULT class */:*

- Because a CLASSDEF statement is not specified for the class DEFAULT, MVS creates a definition for the DEFAULT transport class. This definition is equivalent to having specified:

```
CLASSDEF CLASS(DEFAULT) GROUP(UNDESIG)
```

The default values for CLASSLEN and MAXMSG are used. Since the COUPLE statement did not specify the MAXMSG or CLASSLEN keywords, the IBM-supplied defaults are used: MAXMSG is 500, and CLASSLEN is 956.

Buffer Space: For this configuration, the maximum amount of buffer space that can be used in this transport class to each possible remote system is:

SYSA - 500K
SYSB - 1000K
SYSC - 500K

These values were computed as follows.

There are no additional signaling resources assigned to class DEFAULT for local message traffic on SYSA and for message traffic from SYSA to SYSC. Therefore, the buffer allocation is determined solely from the contribution from the transport class definition (500). PATHOUT device 101 is the only additional signaling resource that can be used to send messages from SYSA to SYSB in the DEFAULT transport class. Therefore, the buffer allocation is 1000K since that is the sum of the contribution from the class definition (500) and the contribution from the additional signaling resources (500) assigned to the class and leading to SYSC.

/ Devices that lead to SYSB. */:*

- PATHOUT DEVICE(11FE) CLASS(CONTROL) - defines outbound signaling path 11FE and assigns it to transport class CONTROL. Outbound messages sent from SYSA to SYSB by members of groups permitted to use resources in class CONTROL are sent on this signaling path. For this configuration, the CLASSDEF statement that defined class CONTROL allowed only GROUPX to use class CONTROL. This dedicates signaling path 11FE exclusively to the transfer of the frequent messages (control signals) sent by member XMASTER of GROUPX on SYSA to members XCOMP1 and XCOMP2 on SYSB.

Buffer Space: Because the MAXMSG keyword was not coded on the PATHOUT statement, the default value (500) for the transport class CONTROL is used. Thus, this signaling path contributes 500K of message buffer space to the total allocation of message buffer space allowed for sending messages between SYSA and SYSB using the CONTROL transport class.

- PATHOUT DEVICE(101) - defines outbound signaling path 101. Because the CLASS keyword was not coded, the device is implicitly assigned to the DEFAULT transport class. Outbound messages sent from SYSA to SYSB by members of groups permitted to use resources in class DEFAULT are sent on this signaling path. For this configuration, the implicitly defined DEFAULT transport class allows any group except GROUPX to use class DEFAULT. This prevents signaling path 101 from being used to transfer the frequent messages sent by member XMASTER of GROUPX on SYSA to members XCOMP1 and XCOMP2 on SYSB.

Buffer Space: Because the MAXMSG keyword was not coded on the PATHOUT statement, the default value (500) for the transport class DEFAULT is used. Thus this signaling path contributes 500K of message buffer space to the total allocation of message buffer space allowed for sending messages between SYSA and SYSB using the DEFAULT transport class.

/* Devices that lead to SYSC. */:

- PATHOUT DEVICE(8E4) CLASS(RECORD) - defines outbound signaling path 8E4 and assigns it to class RECORD. Outbound messages sent from SYSA to SYSC by members of groups permitted to use resources in class RECORD are sent on this signaling path. For this configuration, the CLASSDEF statement that defined class RECORD allowed only GROUPX to use class RECORD. This dedicates signaling path 8E4 exclusively to the transfer of the large messages (trace records) sent by member XMASTER of GROUPX on SYSA to member XRECORD on SYSC.

Buffer Space: Because the MAXMSG keyword was not coded on the PATHOUT statement, the default value (42) for the transport class RECORD is used. Thus this signaling path contributes 42K of message buffer space to the total allocation of message buffer space allowed for sending messages between SYSA and SYSC using the RECORD transport class.

- PATHOUT DEVICE(8E5) CLASS(ANYUSER) - defines outbound signaling path 8E5 and assigns it to class ANYUSER. Outbound messages sent from SYSA to SYSC by members of groups permitted to use resources in class ANYUSER are sent on this signaling path. For this configuration, the CLASSDEF statement that defined class ANYUSER allowed all groups to use class ANYUSER.

Buffer Space: Because the MAXMSG keyword was not coded on the PATHOUT statement, the default value (500) for the transport class ANYUSER is used. Thus this signaling path contributes 500K of message buffer space to the total allocation of message buffer space allowed for sending messages between SYSA and SYSC using the ANYUSER transport class.

If this signaling path had been assigned to the DEFAULT class, GROUPX would have been restricted to device 8E4 for its messages from SYSA to SYSC. This would cause the 300-byte messages to be sent in a transport class set up for large 40,000-byte messages. This would not be an efficient use of storage.

/* Devices that come from SYSB. */:

- PATHIN DEVICE(8C0,8C1) MAXMSG(32) - specifies 32K bytes as the maximum message buffer space for each of the inbound signaling paths 8C0 and 8C1 (64K bytes total).

/* Devices that come from SYSC. */:

Signaling

- PATHIN DEVICE(104,105) MAXMSG(500) - specifies 500K bytes as the maximum message buffer space for each of the inbound signaling paths 104 and 105 (total of 1000K bytes).

LOCALMSG Statement:

- CLASS(CONTROL) MAXMSG(250) - specifies a contribution of 250K bytes of message buffer space for sending messages in class CONTROL between members of GROUPX on SYSA. For this configuration, this statement has the effect of allowing additional message capacity for GROUPX message traffic between members XMASTER and XANALYZE, which both reside on SYSA. This need for additional capacity might have been detected by the system programmer through RMF reports. Had this statement not been coded, the maximum allocation of message buffer space for messages sent locally using class CONTROL would have consisted only of the contribution from the transport class definition itself (500K).

Example 4 — Creating a Transport Class for a Group

This example illustrates a sysplex of two systems in which the global resource serialization component joins the transport class, SYSGRS, and all other defaults in the COUPLExx parmlib member are taken. This ensures that message traffic for the global serialization of resources is sent on its own dedicated signaling path and does not compete with other multisystem applications.

Scenario

The sysplex, PLEXOF2A, consists of two MVS systems (SYSA and SYSB) on two processors.

There is one group, SYSGRS, in the sysplex that requires special consideration. SYSGRS is assigned to its own transport class (TCGRS) in the COUPLExx parmlib member on each system in the sysplex.

Members of other groups on this system share signaling resources in the DEFAULT transport class.

You might consider using this configuration when you plan to use global resource serialization to serialize resources and do not want other multisystem applications competing with it for signaling resources.

Example 4 — Diagram of PLEXOF2A

The following figure shows a logical diagram of the example sysplex named PLEXOF2A.

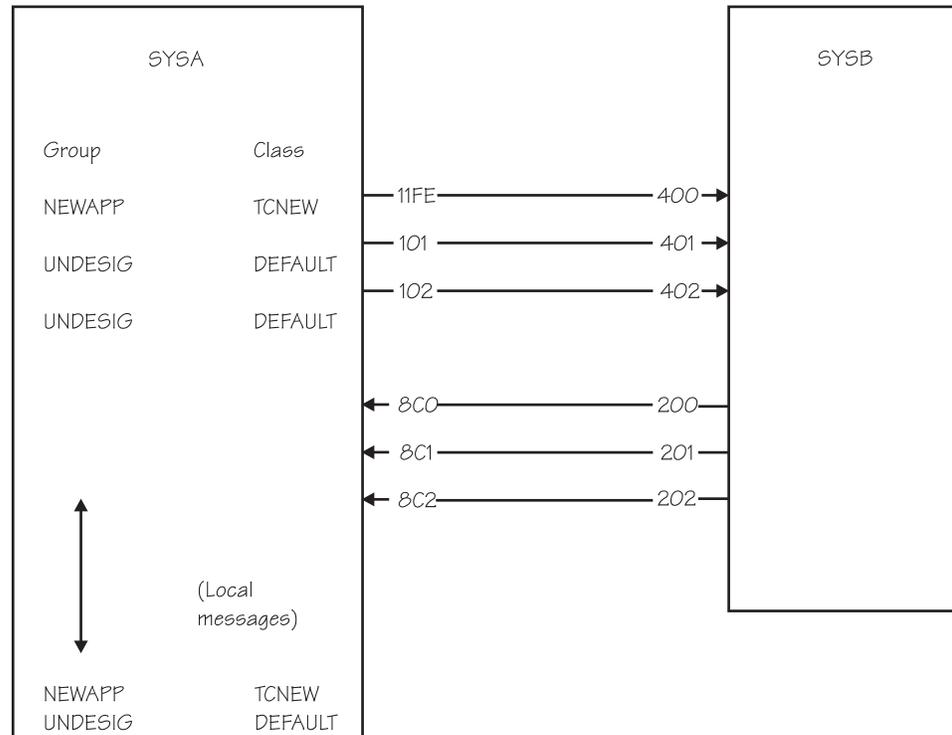


Figure 17. Example 4 — Diagram of PLEXOF2A

Example 4 — SYSA’s COUPLExx Parmlib Member

An example of a COUPLExx parmlib member for SYSA in PLEXOF2A is shown in the following figure. Assume that defaults are taken for any keywords not coded.

```
COUPLE  SYSPLEX(PLEXOF2A)
        PCOUPLE(PLEXOF2A.CDS1)
        ACOUPLE(PLEXOF2A.CDS2)

/* Transport class. */

CLASSDEF CLASS(TCGRS) GROUP(SYSGRS)

/* Devices that lead to SYSB. */
PATHOUT DEVICE(11FE) CLASS(TCGRS)
PATHOUT DEVICE(101,102)

/* Devices that come from SYSB. */
PATHIN  DEVICE(8C0,8C1,8C2)
```

Figure 18. Example of SYSA’s COUPLExx Parmlib Member

The COUPLE Statement:

- SYSPLEX(PLEXOF2A) - identifies the name of the sysplex.
- PCOUPLE(PLEXOF2A.CDS1) - identifies the primary sysplex couple data set.
- ACOUPLE(PLEXOF2A.CDS2) - identifies the alternate sysplex couple data set.

The CLASSDEF statement:

- CLASS(TCGRS) - specifies TCGRS as the name of the transport class.
- GROUP(SYSGRS) - assigns group SYSGRS to transport class TCGRS. (SYSGRS is the XCF group name for global resource serialization.)

Signaling

The defaults are taken for CLASSLEN and MAXMSG parameters.

MVS defines the DEFAULT transport class automatically. In this case, because the group SYSGRS is assigned to a transport class, UNDESIG refers to all groups except SYSGRS.

/* Devices that lead to SYSB. */:

- PATHOUT DEVICE(11FE) CLASS(TCGRS) - assigns outbound signaling path 11FE to class TCGRS. Outbound messages for group SYSGRS are sent on this path.
- PATHOUT DEVICE(101,102) - defines outbound signaling paths 101 and 102. MVS assigns the outbound paths to the DEFAULT class. Outbound messages for all groups, other than SYSGRS, are sent on these paths.

/* Devices that come from SYSB. */:

- PATHIN DEVICE(8C0,8C1,8C2) - defines inbound signaling paths 8C0, 8C1, and 8C2.

In the example, the COUPLExx parmlib member on SYSB could also define a TCGRS transport class, assign the SYSGRS group to TCGRS, and assign an outbound path (such as 200) to class TCGRS.

Example 5 — Defining Signaling Paths through the Coupling Facility

This example illustrates a sysplex of two systems whose signaling paths are defined exclusively through the coupling facility. In this example, the group, named SYSGRS, that the global resource serialization component joins is assigned to its own transport class, and all other COUPLExx defaults are taken. This ensures that message traffic for the global serialization of resources is sent on its own dedicated signaling structure and does not compete with other multisystem applications.

Scenario

The sysplex named PLEXOF2C consists of two MVS systems (SYSA and SYSB) on two processors.

One group, SYSGRS, in the sysplex requires special consideration. SYSGRS is assigned to its own transport class (TCGRS) in the COUPLExx parmlib member on each system in the sysplex.

Members of other groups on this system share signaling resources in the DEFAULT transport class.

You might consider using this configuration when you plan to use global resource serialization to serialize resources and do not want other multisystem applications competing with global resource serialization for signaling resources.

Example 5 — Diagram of PLEXOF2C

Figure 19 on page 101 shows a logical diagram of the example sysplex named PLEXOF2C.

Note that in this example, the two signaling structures are allocated in two different coupling facilities (as specified in the coupling facility resource management (CFRM) policy). This would normally be done for high availability in the event of a coupling facility failure or loss of connectivity to a coupling facility, and is the recommended configuration. However, this separation is not required; the two signaling structures can be placed in the same coupling facility.

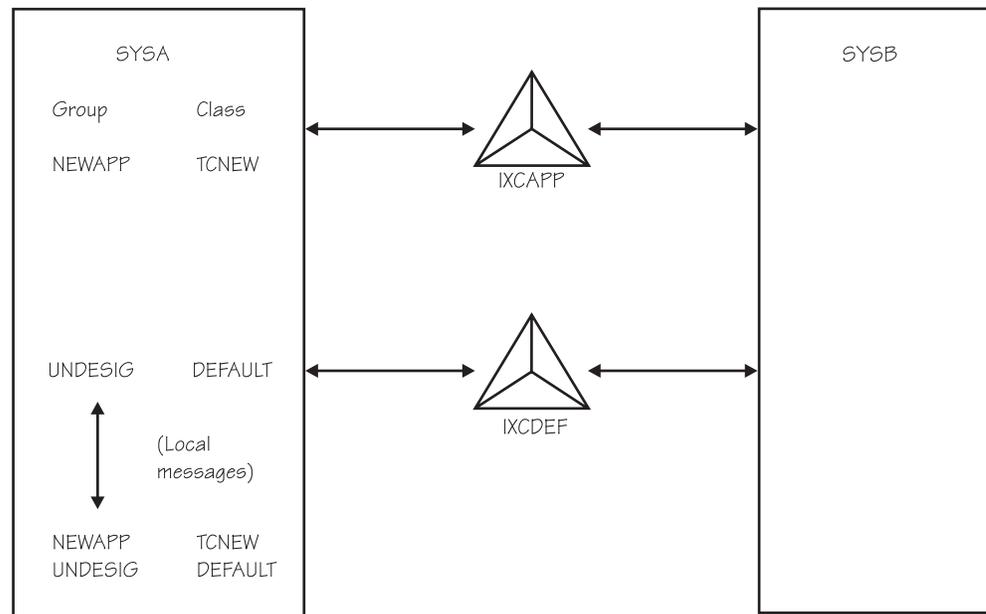


Figure 19. Example 5 — Diagram of PLEXOF2C

Example 5 — SYSA’s COUPLExx Parmlib Member

An example of the COUPLExx parmliB member for SYSA in PLEXOF2C is shown in Figure 20. (Note that the same COUPLExx parmliB member statements could be used on **any** system in the sysplex, since the PATHIN and PATHOUT structure definitions could be the same on any system.) Assume that defaults are taken for any keywords not coded.

```
COUPLE  SYSPLEX(PLEXOF2C)
        PCOUPLE(PLEXOF2C.CDS1)
        ACOUPLE(PLEXOF2C.CDS2)

/* Policy type */
DATA TYPE(CFRM) PCOUPLE(PLEXOF2C.CFRM1)
              ACOUPLE(PLEXOF2C.CFRM2)

/* Transport class */
CLASSDEF CLASS(TCGRS)
          GROUP(SYSGRS)

/* Paths out */
PATHOUT STRNAME(IXCGRS) CLASS(TCGRS)
PATHOUT STRNAME(IXCDEF)

/* Paths in */
PATHIN
STRNAME(IXCGRS,IXCDEF)
```

Figure 20. Example 5 — SYSA’s COUPLExx ParmliB Member

The COUPLE Statement:

- **SYSPLEX(PLEXOF2C)** - identifies the name of the sysplex, PLEXOF2C. The sysplex name specified here must match the sysplex name used in the Format Utility job to format the sysplex couple data sets specified by ACOUPLE and PCOUPLE.

Signaling

- PCOUPLE(PLEXOF2C.CDS1) - identifies the primary sysplex couple data set.
- ACOUPLE(PLEXOF2C.CDS2) - identifies the alternate sysplex couple data set.

The DATA Statement:

- TYPE(CFRM) - specifies that the CFRM policy is to be used.
- PCOUPLE(PLEXOF2C.CFRM1) - identifies the primary CFRM couple data set.
- ACOUPLE(PLEXOF2C.CFRM2) - identifies the alternate CFRM couple data set.

The CLASSDEF statement:

- CLASS(TCGRS) - specifies TCGRS as the name of the transport class.
- GROUP(SYSGRS) - assigns group SYSGRS to transport class TCGRS.

The defaults are taken for CLASSLEN and MAXMSG parameters.

MVS defines the transport class DEFAULT automatically. In this case, because the group SYSGRS is assigned to a transport class, UNDESIG refers to all groups except SYSGRS. (The set of all unassigned groups are collectively referred to on the CLASSDEF keyword as UNDESIG).

/* Paths out. */:

- PATHOUT STRNAME(IXCGRS) CLASS(TCGRS) - defines signaling structure IXCGRS for outbound use. Outbound signaling paths will be established with every other system that defines this structure for inbound use. These outbound signaling paths are assigned to transport class TCGRS. Outbound messages for group SYSGRS are sent via this signaling structure.
- PATHOUT STRNAME(IXCDEF) - defines signaling structure IXCDEF for outbound use. Outbound signaling paths will be established with every other system that defines this structure for inbound use. Since no transport class is explicitly specified for this structure, MVS assigns the outbound paths to the class DEFAULT. Outbound messages for all groups, other than SYSGRS, are sent via this signaling structure.

/* Paths in. */:

- PATHIN STRNAME(IXCDEF,IXCGRS) - defines signaling structures IXCDEF and IXCGRS for inbound use. Inbound signaling paths will be established with every other system that defines this structure for outbound use.

Note: Unlike a CTC device, a signaling structure can be defined for simultaneous inbound and outbound use.

In the example, the COUPLExx parmlib member on SYSB could also define a TCGRS transport class, assign the SYSGRS group to TCGRS, and assign structure IXCGRS as pathout from SYSB. In such a configuration, all SYSGRS signaling traffic in the sysplex would be preferentially sent via the IXCGRS structure, and all other signaling traffic in the sysplex would be preferentially sent via the IXCDEF structure.

Example 6 — Signaling through a Coupling Facility and CTC Connections

This example illustrates a sysplex of two systems whose signaling paths are defined both via the coupling facility **and** CTC devices. In this example, the group (named SYSGRS) that the global resource serialization component joins is assigned to its own transport class, and all other COUPLExx defaults are taken. This ensures that message traffic for the global serialization of resources is sent on its own dedicated signaling paths and does not compete with other multisystem applications.

Note that because more signaling paths are defined, the configuration in this example provides higher availability than the example shown in “Example 5 — Defining Signaling Paths through the Coupling Facility” on page 100.

Scenario

The sysplex PLEXOF2D consists of two MVS systems (SYSA and SYSB) on two processors.

There is one group, SYSGRS, in the sysplex that requires special consideration. SYSGRS is assigned to its own transport class (TCGRS) in the COUPLExx parmlib member on each system in the sysplex.

Members of other groups on this system share signaling resources in the DEFAULT transport class.

Consider using this configuration when you plan to use global resource serialization to serialize resources and do not want other multisystem applications competing with it for signaling resources.

Example 6 — Diagram of PLEXOF2D

Figure 21 on page 104 shows a logical diagram of the example sysplex named PLEXOF2D.

Note that in this example, the two signaling structures are allocated in two different coupling facilities (as determined by the CFRM policy). This is done to maintain availability in case of a coupling facility failure or loss of connectivity to a coupling facility, and is the recommended configuration. However, this separation is not required; the two signaling structures can be placed in the same coupling facility.

The example also shows three CTC connections between SYSA and SYSB.

Signaling

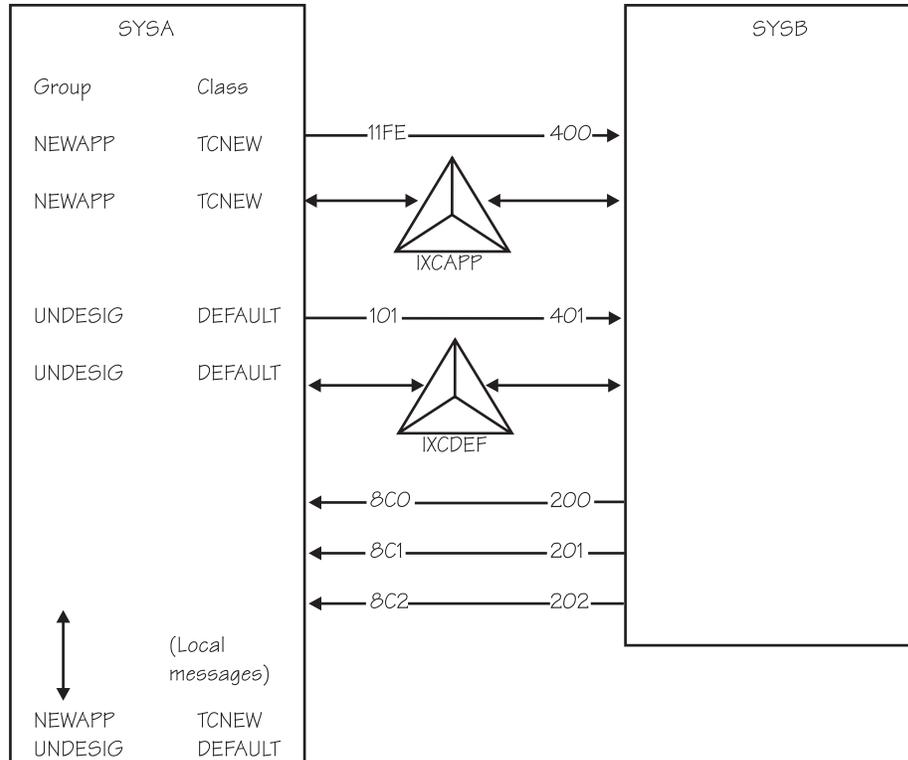


Figure 21. Example 6 — Diagram of PLEXOF2D

Example 6 — SYSA's COUPLExx Parmlib Member

An example of the COUPLExx parmlib member for SYSA in PLEXOF2D is shown in the following figure. Assume that defaults are taken for any keywords not coded.

```

COUPLE  SYSPLEX(PLEXOF2D)
        PCOUPLE(PLEXOF2D.CDS1)
        ACOUPLE(PLEXOF2D.CDS2)

/* Policy Type */
DATA    TYPE(CFRM)
        PCOUPLE(PLEXOF2D.CFRM1)
        ACOUPLE(PLEXOF2D.CFRM2)

/* Transport class */
CLASSDEF CLASS(TCGRS) GROUP(SYSGRS)

/* Paths out */
PATHOUT STRNAME(IXCGRS) CLASS(TCGRS)
PATHOUT DEVICE(11FE) CLASS(TCGRS)
PATHOUT STRNAME(IXCDEF)
PATHOUT DEVICE(101)

/* Paths in */
PATHIN  STRNAME(IXCGRS,IXCDEF)
PATHIN  DEVICE(8C0,8C1,8C2)

```

Figure 22. Example of SYSA's COUPLExx Parmlib Member

The COUPLE Statement:

- SYSPLEX(PLEXOF2D) - identifies the name of the sysplex, PLEXOF2D. The sysplex name specified here must match the sysplex name used in the Format Utility job to format the couple data sets specified by ACOUPLE and PCOUPLE.
- PCOUPLE(PLEXOF2D.CDS1) - identifies the primary sysplex couple data set.
- ACOUPLE(PLEXOF2D.CDS2) - identifies the alternate sysplex couple data set.

The DATA Statement:

- TYPE(CFRM) - specifies that the CFRM policy is to be used.
- PCOUPLE(PLEXOF2D.CFRM1) - identifies the primary CFRM couple data set.
- ACOUPLE(PLEXOF2D.CFRM2) - identifies the alternate CFRM couple data set.

The CLASSDEF statement:

- CLASS(TCGRS) - specifies TCGRS as the name of the transport class.
- GROUP(SYSGRS) - assigns group SYSGRS to transport class TCGRS.

The defaults are taken for CLASSLEN and MAXMSG parameters.

MVS defines the DEFAULT transport class automatically. In this case, because the group SYSGRS is assigned to a transport class, UNDESIG refers to all groups except SYSGRS.

/* Paths out. */:

- PATHOUT STRNAME(IXCGRS) CLASS(TCGRS) - defines signaling structure IXCGRS for outbound traffic. Outbound signaling paths will be established with every other system that defines IXCGRS for inbound traffic. These outbound signaling paths are assigned to transport class TCGRS. Outbound messages for group SYSGRS are sent via IXCGRS.
- PATHOUT DEVICE(11FE) CLASS(TCGRS) - assigns outbound signaling path 11FE to class TCGRS. Outbound messages for group SYSGRS are sent on this path.
- PATHOUT STRNAME(IXCDEF) - defines signaling structure IXCDEF for outbound use. Outbound signaling paths will be established with every other system that defines IXCDEF for inbound use. Since no transport class is explicitly specified for this structure, MVS assigns the outbound paths to the DEFAULT class. Outbound messages for all groups, other than SYSGRS, are sent via IXCDEF.
- PATHOUT DEVICE(101) - defines outbound signaling path 101. MVS assigns the outbound path to the DEFAULT class. Outbound messages for all groups, other than SYSGRS, are sent on this path.

/* Paths in. */:

- PATHIN STRNAME(IXCDEF,IXCGRS) - defines signaling structures IXCDEF and IXCGRS for inbound use. Inbound signaling paths will be established with every other system that defines these structures for outbound use.

Note: Unlike a CTC device, a signaling structure can be defined for simultaneous inbound and outbound use.

- PATHIN DEVICE(8C0,8C1,8C2) - defines inbound signaling paths 8C0, 8C1, and 8C2.

In the example, the COUPLExx parmlib member on SYSB could also define a TCGRS transport class, assign the SYSGRS group to TCGRS, and assign structure IXCGRS and/or a signaling CTC device (such as 200) as pathout from SYSB to class TCGRS.

WSC Flash — Parallel Sysplex Performance: XCF Performance Considerations

Tuning XCF

XCF signaling is used to communicate between various members of a sysplex. The user of XCF signaling, usually an MVS component or a subsystem, issue messages to members within the user's group. The content and/or use of these messages are unique to the users of the group.

As XCF messages are generated, they are assigned to a transport class based on group name and/or message size. The messages are copied into a signal buffer from the XCF buffer pool. The messages are sent over outbound paths, (PATHOUT), defined for the appropriate transport class. Messages from other systems are received by inbound paths, (PATHIN). Inbound paths are not directly assigned transport classes, although a correlation can be made about which transport class messages are received via the inbound paths based on the outbound path to which the inbound side is connected.

Transport Classes

Transport classes are used to group messages. Using the CLASSDEF parameter in the COUPLExx parmlib member, you can assign messages to a transport class based on the group name, the message size, or both.

Each transport class has its own resources that consist of a buffer pool and one or more outbound signaling paths. It is recommended you keep the number of transport classes small. In most cases, it is more efficient to pool the resources and define the transport class based on message size. Some initial product documentation recommended separate transport classes for global resource serialization or RMF. These recommendations are no longer advised. If you do have separate transport classes for specific groups based on early product recommendations, you should consider changing these recommendations.

Message Buffers

XCF message buffers are managed by correctly selecting the size of the message most frequently sent from specific buffer pools and by specifying an adequate upper limit for the size of the buffer pool.

Message Buffer Size

First let's look at the individual message buffer size definitions. Message buffer size is determined by the CLASSLEN parameter on the CLASSDEF statement in the COUPLExx parmlib member. The CLASSLEN value determines the size of the most frequent message expected in this transport class. If a message could be assigned to more than one transport class, XCF selects the one with the smallest buffer that will hold the message. If the signal is larger than the CLASSLEN for any of the assigned transport classes, XCF has to choose a transport class to expand. Since APAR OW16903, XCF assigns the message to the transport class with the largest buffer size and expands the buffer size of this transport class. Prior to this APAR, the transport class named DEFAULT was chosen to be expanded, even if it had a very small class length.

Expanding the message buffer entails some overhead. The PATHOUT on the sending side and the PATHIN on the receiving side must be cleared out and expanded to handle the larger buffer size. A new, larger buffer must be obtained on the PATHIN side. If no additional messages of this size are received in a short time

period, XCF then contracts the PATHIN, PATHOUT, and buffer sizes. In both of these cases extra XCF internal signals are generated to communicate these changes.

The best way to eliminate the overhead of expanding and contracting the message buffers is to define transport classes based solely on the size of the message buffers. One class with the default length of 956 should handle most of the traffic. A second class can be defined to handle larger messages.

An example of this specification in the COUPLExx parmlib member is: CLASSDEF CLASS(DEFSMALL) CLASSLEN(956) GROUP(UNDESIG) CLASSDEF CLASS(DEFAULT) CLASSLEN(16316) GROUP(UNDESIG)

The parameter GROUP(UNDESIG) specifies the messages should be assigned to the transport class based solely on message size. This definition makes all the resources available to all users and provides everyone with peak capacity.

There may be times when you want a separate transport class for a specific group. For instance, if you have a particular XCF user which is consuming a disproportionate amount of XCF resources, you may want to isolate this user to a separate transport class to investigate the user's behavior and protect the other XCF users. Hopefully, after you have diagnosed the problem, you can reassign this user to a transport class based on the length of the messages.

You can use an RMF XCF report to determine how well the messages fit:

```

XCF USAGE BY SYSTEM
-----
                                REMOTE SYSTEMS
-----
                                OUTBOUND FROM JB0
-----

```

TO SYSTEM	TRANSPORT CLASS	BUFFER LENGTH	REQ OUT	----- BUFFER -----			
				% SML	% FIT	% BIG	% OVR
JA0	DEFAULT	16,316	189	98	1	1	100
	DEFSMALL	956	55,794	0	100	0	0
JB0	DEFAULT	16,316	176	100	0	0	0
	DEFSMALL	956	44,156	0	100	0	0
JC0	DEFAULT	16,316	176	100	0	0	0
	DEFSMALL	956	34,477	0	100	0	0
TOTAL			134,968				

%SML is the % of messages smaller than the buffer length
 %FIT is the % of messages which fit the buffer length
 %BIG is the % of messages larger than the buffer length

In this example, the majority of the messages fit in the DEFSMALL class. A few exceeded the size of the DEFAULT class, but not enough to justify the definition of a new transport class.

Note: XCF has internal buffers of fixed size: 1K, 4K, 8K, ...64K. XCF uses 68 bytes for internal control blocks. So if you specify a length that doesn't fit one of these sizes, XCF will round up to the next largest size. For example, if you specify 1024, it will not fit into the 1K block (1024-68=956), and XCF will round up to the next largest block. The command, D XCF,CLASSDEF, will

Signaling

list the CLASSLEN specified in the PARMLIB member, in this example, 1024. The RMF XCF report will show the actual buffer length, in this case, 4028.

Message Buffer Pools

Having determined the optimal size for the individual message buffer, the next thing to do is select an upper limit for the amount of virtual storage to be allocated to the message buffer pool. The message buffer space is virtual storage used by XCF to store the message buffers which are being processed, sent or received.

Most of the virtual storage used for this purpose is backed by fixed central and expanded storage. The storage to hold LOCAL buffers (for communication within the processor) is DREF storage which is backed by central storage. LOCAL buffers are used for messages within groups which are on the same MVS image. Currently APPC and JES3 are the only known IBM exploiters of local messages but OEM applications can choose to take advantage of LOCAL message processing.

XCF only uses the amount of storage it needs; but to insure there are no surprises, the installation can use the MAXMSG parameter to place an upper limit on the amount of storage which can be used for this purpose.

Storage is associated with the transport class, the outgoing paths, and the incoming paths, so MAXMSG can be specified on the CLASSDEF, PATHIN and PATHOUT definitions, or more generally on the COUPLE definition. MAXMSG is specified in 1K units. The default values are determined in the following hierarchy:

OUTBOUND	INBOUND
-----	-----
PATHOUT - not specified, use	PATHIN - not specified, use
CLASSDEF - not specified, use	COUPLE
COUPLE	

The default for MAXMSG is 500 in OS/390 R1 and prior releases. In OS/390 R2 and higher, the MAXMSG default is 750. By not specifying the default parameter, you will automatically get the most current default size as you migrate to newer releases. If you do want a larger value than the default, specify it at the lowest level of the hierarchy as appropriate.

The total amount of storage used by XCF on a single system is the sum of:

- Sum of MAXMSG for all classes multiplied by the number of systems in the sysplex
- Sum of MAXMSG for all PATHOUTs
- Sum of MAXMSG for all PATHINs

In this example:

XCF PATH STATISTICS					
OUTBOUND FROM JC0			INBOUND TO JC0		

TO	T FROM/TO	TRANSPORT	FROM	T FROM/TO	
SYSTEM	Y DEVICE, OR	CLASS	SYSTEM	Y DEVICE, OR	
	P STRUCTURE			P STRUCTURE	
JA0	S IXCPLEX_PATH1	DEFAULT	JA0	S IXCPLEX_PATH1	
	C C600 TO C614	DEFSMALL		C C600 TO C614	
	C C601 TO C615	DEFSMALL		C C601 TO C615	
	C C602 TO C616	DEFSMALL		C C602 TO C616	
JB0	S IXCPLEX_PATH1	DEFAULT	JB0	S IXCPLEX_PATH1	

```

C C600 TO C614   DEFSMALL           C C600 TO C614
C C601 TO C615   DEFSMALL           C C601 TO C615
C C602 TO C616   DEFSMALL           C C602 TO C616
    
```

If a MAXMSG of 1000 was specified on the CLASSDEF parameter and MAXMSG was not specified on the other parameters, the maximum storage which could be used by XCF is 22M:

- 2 classes * 3 systems * 1M = 6M
- 8 PATHOUTs * 1M = 8M
- 8 PATHINs * 1M = 8M

Note: This implies if you add additional transport classes, signaling paths or systems, you will be increasing the upper limit on the size of the message buffer pool.

Outbound Messages

For the outbound messages to a particular system if the sum of the storage for the CLASSDEF and the PATHOUTs is insufficient, the signal will be **rejected**. This is reported on the RMF XCF report as REQ REJECT for OUTBOUND requests. In general, any non-zero value in this field suggests some further investigation. The problem is generally resolved by increasing MAXMSG on the CLASSDEF or PATHOUT definition.

XCF USAGE BY SYSTEM						

REMOTE SYSTEMS						

OUTBOUND FROM SYSC						

TO SYSTEM	TRANSPORT CLASS	BUFFER LENGTH	REQ OUT	...	ALL PATHS UNAVAIL	REQ REJECT
K004	DEFAULT	956	126,255	...	0	1,391
	DEF16K	16,316	28		0	0
SYSA	DEFAULT	956	97,834		0	0
	DEF16K	16,316	3,467		0	0

TOTAL			227,584			

Inbound Messages

For the inbound messages from a particular system, if the storage for the PATHINs is insufficient, the signal will be **delayed**. This is reported on the RMF XCF report as REQ REJECT for INBOUND requests. If the delay causes signals to back up on the outbound side, eventually an outbound signal could get rejected for lack of buffer space. In this case, you may wish to increase the MAXMSG on the PATHIN definition.

XCF USAGE BY SYSTEM						

REMOTE SYSTEMS				LOCAL		

INBOUND TO SYSC				SYSC		

.....	FROM SYSTEM	REQ IN	REQ REJECT	TRANSPORT CLASS	REQ REJECT	
	K004	117,613	1,373	DEFAULT		0
	SYSA	101,490	0	DEF16K		0

TOTAL		219,103				

Signaling

Another indicator that the storage for PATHINs is insufficient is the BUFFERS UNAVAIL count on the XCF PATH STATISTICS report. If this is high, check the AVAIL and BUSY counts: AVAIL counts should be high relative to BUSY counts. High BUSY counts can be caused by an insufficient number of paths or a lack of inbound space. First look at the inbound side and see if there are any REQ REJECTs. If so, increase the PATHIN MAXMSG. Otherwise, it is important to review the capacity of the signaling paths. The methodology for determining this is described in "Capacity Planning" on page 113.

Note: The RMF Communications Device report cannot be used to determine if the CTC devices are too busy. XCF CTCs will typically always report high device utilization because of the suspend / resume protocol used by XCF.

Local Messages

Local messages are signals within the same image, so no signaling paths are required. In this case, the message buffer storage used is the CLASSDEF storage plus any storage specified on the LOCALMSG definition. If MAXMSG is not coded on the LOCALMSG statement the additional message buffer storage contributed is none, or 0 buffers.

Signaling Paths

XCF signals from each transport class are sent out on the PATHOUT path and received into the system on the PATHIN paths. Tuning is achieved by altering the number or type of paths, or both. To review the XCF path configuration use the RMF XCF Path Statistics report. Two different issues commonly reported to IBM regarding signaling paths are reviewed in this flash: no paths defined, and an insufficient number of paths defined.

Number of Paths

No Paths

In the worst case, there may be NO operational paths for a transport class. This is not fatal. XCF routes the requests to another transport class but there is additional overhead associated with this operation. To determine if this condition exists, look at the RMF XCF Usage by System report. ALL PATHS UNAVAIL should be low or 0. In many cases, this is caused by an error in the path definition; in other cases, there may be a problem with the physical path.

XCF USAGE BY SYSTEM						

REMOTE SYSTEMS						

OUTBOUND FROM SDO						

TO SYSTEM	TRANSPORT CLASS	BUFFER LENGTH	REQ OUT	...	ALL PATHS UNAVAIL	REQ REJECT
JA0	DEFAULT	16,316	189	...	0	0
	DEFSMALL	956	55,794		55,794	0
JB0	DEFAULT	16,316	176		0	0
	DEFSMALL	956	44,156		0	0
JC0	DEFAULT	16,316	176		0	0
	DEFSMALL	956	34,477		0	0

TOTAL			134,968			

In this example, the CTC links to system JA0 had been disconnected.

In the **next** example from the same system, notice for system JA0 there were no paths for the transport class DEFSMALL, so all the requests were re-driven through the DEFAULT class. This caused some queuing (see AVG Q LENGTH of 0.16).

```

XCF PATH STATISTICS
-----
OUTBOUND FROM SD0
-----

```

TO SYSTEM	T FROM/TO Y DEVICE, OR P STRUCTURE	TRANSPORT CLASS	REQ OUT	AVG Q LENGTH	T FROM/TO AVAIL	BUSY	RETRY
JA0	S IXCPLEX_PATH1	DEFAULT	56,011	0.16	55,894	117	0
JB0	S IXCPLEX_PATH1	DEFAULT	176	0.00	176	0	0
	C C600 TO C614	DEFSMALL	16,314	0.01	16,297	17	0
	C C601 TO C615	DEFSMALL	15,053	0.01	15,037	16	0
	C C602 TO C616	DEFSMALL	15,136	0.01	15,136	20	0
JC0	S IXCPLEX_PATH1	DEFAULT	176	0.00	176	0	0
	C C600 TO C614	DEFSMALL	11,621	0.01	11,515	106	0
	C C601 TO C615	DEFSMALL	13,086	0.01	12,962	124	0
	C C602 TO C616	DEFSMALL	11,626	0.00	11,526	100	0

Is it necessary to correct the 'ALL PATHS UNAVAIL' condition? In most cases it is. In the example above, DEFSMALL was defined to hold small messages (956). Because there is no path, they are being re-driven through the **DEFAULT** class. The DEFAULT class is sending data in large buffers (16,316 bytes). This is certainly not an efficient use of message buffer storage to transfer a 956 byte message in a 16,316 byte buffer. Re-driving large messages through a transport class defined with small messages causes more problems. It causes the buffers in this class to expand and contract with all the extra signaling explained previously. Defining separate classes is done for a purpose. If you don't provide paths for these classes, it negates this purpose.

Insufficient Number of Paths

Signaling paths can be CTC links or Coupling Facility structures. In the example above, the TYP field indicates the connection is a coupling facility structure (S) or a CTC link (C). Since these two types of paths operate in unique ways, different methods are used to evaluate their performance.

- Coupling facility structures

For coupling facility structures, an insufficient number of PATHOUT links could result in an increase in the AVG Q LENGTH, and BUSY counts high relative to AVAIL counts. Additional paths are obtained by defining more XCF signaling structures in the CFRM policy and making them available for use as PATHOUTs (and/or PATHINs).

Note: RETRY counts should be low relative to REQ OUT for a transport class. A non zero count indicates a message has failed and was resent. This is usually indicative of a hardware problem.

- CTCs

CTCs can be configured in a number of ways. The installation can define CTC's as unidirectional (one PATHOUT or one PATHIN per physical CTC) or bi-directional (one or more PATHOUTs and PATHINs on a physical CTC). Due to the nature of XCF channel programs, a unidirectional path definition can achieve the most efficient use of a CTC thus providing the best XCF response time and message throughput capacity. However, a unidirectional definition will also require using **at least four physical CTCs** to configure for availability. As will be noted in the capacity planning section below, two paths are sufficient for most

Signaling

systems, thus only those customers with very high XCF activity, (requiring ≥ 4 paths), should consider using the unidirectional definition.

What indicators should be used to determine if there are enough CTCs for a particular transport class? First of all, the AVG Q LEN on the RMF XCF report is **not a good indicator**. In the case of CTCs, queued requests are added to the CCW chain which can increase efficiency. A better indicator to use instead is the Display XCF command. This command was updated by XCF APAR OW38138 to provide the path response time (as seen by XCF).

```
D XCF,PI,DEVICE=ALL,STATUS=WORKING
IXC356I 12.02.12 DISPLAY XCF 901
LOCAL DEVICE  REMOTE  PATHIN  REMOTE
PATHIN        SYSTEM  STATUS  PATHOUT  RETRY  MAXMSG  LAST    MXFER
C200          JA0    WORKING C200     10     500    3496   339
C220          JA0    WORKING C220     10     500    3640   419
```

The MXFER TIME is the mean transfer time in microseconds for up to the last 64 signals received within the last minute. If the MXFER TIME is acceptable, less than 2 milliseconds, (or 2000 microseconds), there is probably enough CTC capacity. To insure capacity for heavier or peak workloads, also check the channel utilization for the CTCs, as reported on an RMF Channel Activity report. In laboratory testing, acceptable XCF message response times were observed even at channel utilization of 70% (or 90% when there were multiple CTCs per transport class). Beyond this threshold, response time degenerated rapidly.

RMF, with APAR OW41317 installed, will store the MXFER TIME as observed in the last minute before the end of the RMF interval in the RMF SMF 74 subtype 2 record.

Type of Signaling Path

A CTC provides a direct path between two systems, while sending a message through a coupling facility is a two step, push-pull process. Thus, depending on message size and the type of coupling facility link, CTCs are sometimes faster than using coupling facility structures.

These are examples of XCF response time, (MXFER TIME), from controlled experiments in a test environment. The unidirectional CTCs have a single PATHIN or PATHOUT per physical CTC. The bi-directional CTCs have a pair of PATHIN AND PATHOUT defined for physical CTC. The 4 bi-directional CTCs have 4 pairs of PATHIN and PATHOUT per physical CTC.

A comparison of these examples shows unidirectional CTCs are the fastest option for 1K messages, although ICBs are close behind. The bi-directional CTCs are **somewhat** slower, but perfectly adequate for most installations. For larger messages, ICBs are the faster option. This results from the higher bandwidth associated with ICB, (and ISC), coupling links compared to CTCs, (ESCON).

XCF internally times the various signals and gives preference to the faster paths. In the following example, compare the number of requests for DEFSSMALL which were sent through the structure to the number which were sent through the CTCs. It should be noted XCF does not attempt to balance the workload across paths; once it finds a fast path, it continues to use it. APAR OW38138 describes changes which improves the path distribution.

```
-----
XCF PATH STATISTICS
-----
OUTBOUND FROM JA0
-----
T FROM/TO                                T FROM/TO
```

TO SYSTEM	Y DEVICE, OR P STRUCTURE	TRANSPORT CLASS	REQ OUT	AVG Q LENGTH	AVAIL	BUSY	RETRY
JC0	S IXCPLEX_PATH1	DEFAULT	1,744	0.00	1,176	0	0
	S IXCPLEX_PATH2	DEFSMALL	8,582	0.01	8,362	220	0
	C C600 TO C614	DEFSMALL	20,223	0.01	20,160	63	0
	C C601 TO C615	DEFSMALL	23,248	0.01	23,229	19	0
	C C602 TO C616	DEFSMALL	23,582	0.01	23,568	14	0

In many environments, the difference in response time between CTCs and coupling facility structures is indiscernible and using coupling facility structures certainly simplifies management of the configuration.

Capacity Planning

For availability, a minimum of two physical paths must be provided between any two systems. This can be accomplished with two physical CTCs, structures in each of two different CFs, or a combination of CTCs and coupling facility structures.

Most environments will find the rate of XCF traffic can be handled by the two paths which were configured for availability. Only for environments with very high rates of XCF traffic would additional paths be required.

The XCF message rate capacity of a path is affected by many factors:

1. The size of the message
2. How the paths are defined
3. If the path is also used for other (non-XCF) functions

Based on these factors, message rates (XCF IN+OUT), have been observed from 1000/sec to 5000/sec on a CTC, up to 9000/sec via an ICB and up to 4000/sec per HiPerLink. The adage "Your mileage may vary" is certainly true here.

When using coupling facility structures for XCF messaging, there is also a cost in coupling facility CPU utilization to plan for. As an example, running 1000 XCF messages/sec through an R06 coupling facility would utilize approximately 10% of one coupling facility processor. Additionally, if you use coupling facility structures as XCF paths, make sure the structure size is adequate. You can use the CF Sizer available on the Parallel Sysplex website, www.s390.ibm.com/products/pso to obtain an initial estimate for the structure size. If the structure is too small, you will see an increase in the number of REQ REJECT and AVG Q LENGTH, and these events will definitely affect response time.

CTC Configuration Planning

When configuring CTCs for large volumes of XCF traffic some additional configuration planning needs to be done. CTC I/O will use SAP capacity, and large XCF environments can generate I/O rates much higher than traditional DASD and Tape workloads.

The SAP acts as an offload engine for the CPUs. Different processor models have different numbers of SAPs, and a spare 9672 PU can be configured as an additional SAP processor. SAP functions include:

- Execution of ESA/390 I/O operations. The SAP (or SAPs) are part of the I/O subsystem of the CPC and act as Integrated Offload Processor (IOP) engines for the other processors.
- Machine check handling and reset control
- Support functions for Service Call Logical Processor (SCLP)

Signaling

In high volume XCF environments planning should be done to ensure the CTC configuration is defined so the CTC I/O load is spread across all available SAPs. Information on channel to SAP relationships can be found in the *IOCP User's Guide and ESCON CTC Reference*, GC38-0401-11. Additional Information on 9672 SAP performance and tuning can be found in WSC Flash 9646E at www.ibm.com/support/techdocs.

Case Study

This is a case study that illustrates some of the items discussed.

An application was invoked that was changed to use coupling facility signaling. When the workload was increased XCF delays increased. This was evident from messages like ERB463I which indicated the RMF Sysplex Data Server was not able to communicate with another system because the XCF signaling function was busy.

Looking at RMF Monitor III it showed:

```

RMF 1.3.0 XCF Delays
Samples: 120 System: J90 Date: 02/07/97 Time: 13.03.00

Jobname  Service  DLY  -----Main Delay Path(s)
C Class  %    % Path  % Path  % Path
WLM      S  SYSTEM  87  87  -CF-
*MASTER* S  SYSTEM  10  10  -CF-
RMFGAT   S  SYSTEM   3   3  -CF-
JESXCF   S  SYSTEM   1   1  C601
  
```

Comparing the RMF XCF reports to some earlier reports, it was noticed the amount of XCF traffic had quadrupled and the increase was in the class with the larger CLASSLEN (DEFAULT on this system).

In order to protect other XCF users and to investigate what was happening, a decision was made to separate these messages into their own transport class. A new transport class, NEWXCF, was defined using the GROUP keyword to specifically assign messages from the new application to this class. Since it was known the messages were bigger than the transport class with the smaller CLASSLEN (DEFSMALL), using guess work it was decided the messages might fit into a 4K(-68) buffer. This report was generated:

```

----- BUFFER -----
TO      TRANSPORT  BUFFER  REQ  %  %  %  %  ALL  REQ
SYSTEM  CLASS      LENGTH  OUT  SML FIT BIG OVR  PATHS  REJECT
JA0     DEFAULT   20,412  2,167  92  8  <1  100  0  0
        DEFSMALL  956    29,730  0  100  0  0  0  0
        NEWXCF   4,028  106,018  0  0  100  0  0  0
JB0     DEFAULT   20,412  6,132  97  3  <1  100  0  0
        DEFSMALL  956    82,687  0  100  0  0  0  0
        NEWXCF   4,028  18,085  0  0  100  0  0  0
  
```

Since all the NEWXCF messages were too big, the CLASSLEN was increased.

```

----- BUFFER -----
TO      TRANSPORT  BUFFER  REQ  %  %  %  %  ALL  REQ
SYSTEM  CLASS      LENGTH  OUT  SML FIT BIG OVR  PATHS  REJECT
JA0     DEFAULT   20,412  1,715  90  10  0  0  0  0
        DEFSMALL  956    37,687  0  100  0  0  0  0
        NEWXCF   8,124  103,063  0  100  0  0  0  3,460
JB0     DEFAULT   20,412  2,075  92  8  0  0  0  0
        DEFSMALL  956    38,985  0  100  0  0  0  0
        NEWXCF   8,124  117,727  0  100  0  0  0  195
  
```

Now all the messages fit, but some are being REJECTED. This suggests message buffer space for the outbound path is no longer large enough. The XCF path statistics confirm outbound messages are queuing up.

TO SYSTEM	Y DEVICE, OR P STRUCTURE	TRANSPORT CLASS	REQ OUT	AVG Q LENGTH	AVAIL	BUSY
JA0	S IXCPLEX_PATH1	DEFAULT	1,715	0.00	1,715	0
	S IXCPLEX_PATH2	DEFSMALL	486	0.00	486	0
	S IXCPLEX_PATH3	NEWXCF	103,063	1.42	102,818	245
	C C600 TO C584	DEFSMALL	13,684	0.00	13,644	0
	C C601 TO C585	DEFSMALL	13,603	0.00	13,603	0
JB0	C C602 TO C586	DEFSMALL	12,610	0.00	12,610	0
	S IXCPLEX_PATH1	DEFAULT	2,075	0.00	2,075	0
	S IXCPLEX_PATH2	DEFSMALL	737	0.00	737	0
	S IXCPLEX_PATH3	NEWXCF	117,727	1.26	117,445	282
	C C610 TO C584	DEFSMALL	16,391	0.00	16,391	0
	C C611 TO C585	DEFSMALL	12,131	0.00	12,131	0
	C C612 TO C586	DEFSMALL	12,294	0.00	12,294	0

Increasing the MAXMSG on the PATHOUT for the NEWXCF transport class from 1000 to 2000 clears up the queuing delays.

TO SYSTEM	TRANSPORT CLASS	BUFFER LENGTH	REQ OUT	% SML	% FIT	% BIG	% OVR	ALL PATHS UNAVAIL	REQ REJECT
JA0	DEFAULT	20,412	2,420	93	7	0	0	0	0
	DEFSMALL	956	41,215	0	100	0	0	0	0
	NEWXCF	8,124	133,289	0	100	0	0	0	0
JB0	DEFAULT	20,412	2,362	93	7	0	0	0	0
	DEFSMALL	956	39,302	0	100	0	0	0	0
	NEWXCF	8,124	143,382	0	100	0	0	0	195

The BUSY conditions are reduced, and more importantly the AVG Q LENGTH has been greatly reduced. Since the pathout with the contention is a coupling facility structure AVG Q LENGTH is an appropriate metric to use when tuning.

TO SYSTEM	Y DEVICE, OR P STRUCTURE	TRANSPORT CLASS	REQ OUT	AVG Q LENGTH	AVAIL	BUSY
JA0	S IXCPLEX_PATH1	DEFAULT	2,420	0.00	2,420	0
	S IXCPLEX_PATH2	DEFSMALL	361	0.00	361	0
	S IXCPLEX_PATH3	NEWXCF	133,289	0.08	133,117	2
	C C600 TO C584	DEFSMALL	12,700	0.00	12,700	0
	C C601 TO C585	DEFSMALL	16,421	0.00	16,421	0
JB0	C C602 TO C586	DEFSMALL	14,173	0.00	14,173	0
	S IXCPLEX_PATH1	DEFAULT	2,362	0.00	2,362	0
	S IXCPLEX_PATH2	DEFSMALL	1,035	0.00	1,033	2
	S IXCPLEX_PATH3	NEWXCF	143,383	0.09	143,087	296
	C C610 TO C584	DEFSMALL	12,647	0.00	12,646	1
	C C611 TO C585	DEFSMALL	15,944	0.00	15,944	0
	C C612 TO C586	DEFSMALL	12,183	0.00	12,182	1

When determining how to tune the application to limit the number of XCF messages, a DEF8K transport class for UNDESIG messages was created and the NEWXCF class assigned to this application was eliminated.

Note: In this case study, the messages were being queued because the message buffer space was too small. If, instead of REJECTS, there was a high percentage of messages marked as BUSY, then increasing the number of signaling paths would have been appropriate.

Incidentally the path associated with the NEWXCF was a coupling facility structure that used the new HiPerLinks available on the G3 server. The structure was chosen

Signaling

| since it was quicker and easier to implement. Since the structure was receiving over
| 500 req/sec, it was unclear if the structure could handle the traffic. As can be seen
| from the queue lengths, it was capable of handling this rate.

Chapter 6. Tuning a Sysplex

The performance of systems in a sysplex is closely related to the performance of the signaling service. The throughput and responsiveness of the systems participating in the sysplex are dependent on the signaling service. It is therefore important to be able to identify and correct performance problems related to signaling.

It is also important to perform capacity planning so that the signaling service is able to adequately meet the needs of its users, and thereby provide good signaling performance as workloads change over time.

The performance of the coupling facility in the sysplex also affects overall sysplex performance, especially if you are using the coupling facility for signaling.

RMF Reports and the Sysplex

The Resource Measurement Facility (RMF) provides an XCF Activity Report that contains useful measurement data for analyzing the performance of the XCF signaling service and doing capacity planning for an MVS system in a sysplex.

RMF also provides a Coupling Facility Activity Report. It provides information about the usage of a coupling facility, structures within a coupling facility and subchannel connections to a coupling facility in a sysplex.

Additional Information About RMF Report Data

Additional information about the data in the RMF reports can be found in the following publications:

- *RMF Report Analysis* gives a detailed description of each field on the report as well as tuning suggestions.
- *RMF Performance Management Guide* is a top-down discussion of key fields used in tuning.

XCF Activity Report

The XCF Activity Report shows the data collected on a system during sysplex processing. Each system collects its own data, and RMF on each system produces reports only about that system's data. It might be necessary to run the RMF reports on two or more systems to get data for corresponding outbound and inbound signaling paths in order to better understand the message traffic patterns.

The XCF Activity Report includes data about:

- Transport classes
- Message buffers and message traffic
- Groups and members
- Signaling paths

This chapter describes considerations for tuning the signaling service and illustrates how you can use RMF reports to do tuning and capacity planning for the signaling service.

Tuning a Sysplex

Coupling Facility Activity Report

The coupling facility activity report provides summary information about the use of each coupling facility in the sysplex, detailed information about each active structure in the coupling facility, and summary information about subchannels connecting the coupling facility to the systems in the sysplex.

Tuning the Signaling Service

The key to ensuring good performance for the signaling service is to provide sufficient signaling resources, which are the message buffers and signaling paths, and to control access to those resources by defining transport classes.

Because the signaling service attempts to optimize its use of its resources, you might not need to take any action to ensure good signaling performance. For example, the signaling service dynamically adjusts message buffer sizes to accommodate the actual message traffic patterns so you may not need to modify any signaling parameters.

However, if signaling resources are constrained, the signaling service might not be able to optimize its use of resources. Also, if the resources provided are inadequate for the message traffic, performance may be degraded.

Signaling service resources can be adjusted on a system by system basis to accommodate various message traffic loads.

The following topics discuss the factors that influence the performance of the signaling service and the tradeoffs to consider when adjusting signaling parameters, as well as how to use RMF reports to identify and analyze signaling constraints, performance problems, and capacity needs.

- “Tuning the Message Buffer Size (Class Length)”
- “Tuning the Maximum Message Buffer Space” on page 121
- “Tuning the Signaling Paths” on page 124
- “Tuning the Transport Classes” on page 127
- “Capacity Planning” on page 128.

Tuning the Message Buffer Size (Class Length)

The class length selected for a transport class definition determines the size of the smallest message buffer that XCF provides in a transport class. (Specify class length on the CLASSLEN keyword of the CLASSDEF statement in the COUPLExx parmlib member.)

If a message buffer is too small for a particular message request, the request incurs additional overhead because XCF must format a larger message buffer. If a message buffer is too large for a particular message request, central storage is wasted. XCF adjusts the size of the buffers in a particular transport class on a system by system basis based on the actual message traffic.

The “XCF Activity Report — Usage by System” on page 130 provides data that can be useful for evaluating message buffer sizes. This report provides data about the message buffer size for each remote system, by transport class. The columns in the report that are relevant to message buffer sizes are:

TO SYSTEM

Indicates the name of the system to which the messages were sent.

TRANSPORT CLASS

Indicates the name of the transport class used to send the messages.

BUFFER LENGTH

Indicates the defined message buffer size used for the transport class.

The defined message buffer size is the length of the buffer that best accommodates the class length chosen for the transport class definition.

REQ OUT

Indicates the number of message requests accepted for delivery to the indicated system in the indicated transport class.

The number of message requests accepted for delivery can help you understand the impact of the transport class on the message traffic. If few message requests use a class, consider eliminating the class definition or reducing the resources assigned to the class.

The columns labeled SML, FIT, and BIG under the heading "BUFFER" indicate the distribution of message buffer sizes for messages that were delivered. This distribution is reported in relationship to the defined buffer size, that is, it indicates whether the buffer size needed to accommodate the message is smaller than, equal to, or larger than the defined buffer size.

SML

Indicates the percentage of messages sent that could have used a message buffer smaller than the defined buffer size.

This column is zero if the defined class length causes XCF to use the smallest buffer size that it supports. When this column is not zero, it indicates that some of the messages delivered caused storage to be wasted because the defined buffer size was larger than needed. In this case, consider decreasing the class length or defining a new transport class with a smaller class length. You normally want SML to be close to zero.

FIT

Indicates the percentage of messages sent that fit the defined buffer size.

This means that the message would not have fit in the next smaller buffer size supported by XCF, and it would have wasted storage if the next larger buffer size had been used. You want FIT to be as large as possible.

BIG

Indicates the percentage of messages sent that needed a message buffer larger than the defined buffer size.

When this column is not zero, it indicates that XCF had to prepare message buffers that were larger than the defined size. Such messages incur additional overhead. XCF may increase the buffer size it provides in order to eliminate this overhead.

If BIG is not zero, consider increasing the class length for the transport class. Or, consider defining a new transport class with a larger class length that can be used to segregate the messages by size. You normally want BIG to be close to zero.

OVR

Indicates the percentage of BIG messages sent that suffered performance degradation.

Tuning a Sysplex

It provides a measure of how successful XCF was in its attempt to satisfy oversized message requests without additional overhead.

Note: Be careful to interpret OVR correctly. It provides the percentage of BIG signals that suffered performance degradation. It is **not** the percentage of all signals accepted for delivery. For example, if 100 messages were accepted for delivery and one of those messages required a buffer larger than the defined buffer size, then BIG would be 1% and OVR would be 100%. In this situation, OVR being 100% is most likely not significant because only one message had a problem.

Segregating Message Traffic

In general, if you segregate message traffic according to size, signaling performance is improved. This would be true even if there were no additional overhead incurred when XCF had to format larger buffers. If large and small messages are intermixed, the small messages tend to be delayed simply because the longer messages take longer to process.

In the “XCF Activity Report — Usage by System” on page 130, the SML and BIG columns indicate when it may be useful to segregate messages by size. The reports can indicate that such a change might be useful, but not what the appropriate size should be, only that it should be smaller or larger.

Information about message sizes can be determined by creating some temporary transport classes (use the SETXCF operator command) that use different class lengths. Then use the RMF reports to investigate the appropriateness of that class length. By assigning a unique group to those temporary classes, you could determine the range of message sizes used by that group. Alternatively, the message sizes used by the group may have been documented and you could use that information. This information might indicate that a particular group is sending the messages that don't match the defined buffer size. That group can be assigned to its own transport class in order to segregate it from the other users of the message service.

Choosing a Class Length

If you choose a class length equal to the length of the smallest message sent in a transport class, XCF has the maximum flexibility to dynamically make adjustments to larger buffer sizes.

If you choose a class length equal to the length of the largest message sent in the transport class, XCF does not adjust buffer sizes because XCF never provides a message buffer smaller than defined message buffer size.

The smallest message buffer that XCF uses is large enough to accommodate the defined class length. If the class length is equal to the length of the largest message sent in the class, then the smallest buffer XCF uses can always accommodate the largest message. The advantage is that no message ever needs a larger message buffer and therefore never incurs the extra overhead necessary for XCF to prepare larger buffers.

It is possible that the algorithm used to adjust buffer size does not respond to the particular message traffic pattern at your installation. In this case, you may need to choose a larger class length. The disadvantage of choosing a larger class length is that storage might be wasted by messages that don't need such a large buffer.

Summary

Use the report shown in “XCF Activity Report — Usage by System” on page 130 to evaluate the class lengths chosen for your transport class definitions. SML and BIG can be used to identify mismatches between the defined message buffer size and the size of the messages actually sent. Use OVR to determine if a significant number of BIG messages suffered performance degradation. REQ OUT is helpful in determining whether these problems are occurring for a significant number of message requests.

Some of the ways to fix the problems highlighted by these numbers are:

- Create one separate class for large messages and assign all groups to that class, using UNDESIG.
- Change the class length for the transport class
- Define a new transport class to segregate message traffic by size
- Reassign groups so that the senders of the messages use a different transport class.

Tuning the Maximum Message Buffer Space

XCF acquires message buffers on an as needed basis, that is, XCF acquires only those buffers needed to support the current message traffic load. Therefore it is generally not a problem to specify excess message buffer capacity because XCF does not acquire it if it is not needed. You generally want sufficient capacity to satisfy each message request. If there is not sufficient message buffer space to satisfy a message request, XCF rejects the request. This could negatively impact the performance or the processing of exploiters of the XCF signaling services.

Because message buffers can occupy central storage, there can be a detrimental impact to the rest of the system if too many message requests cause XCF to acquire an excessive amount of message buffer space. Therefore you might wish to limit the total amount of storage that the signaling service uses for message buffers.

To limit the buffer space that XCF can acquire, specify maximum message buffer space on the MAXMSG keywords in the COUPLExx parmlib member or on SETXCF operator commands. See also “Planning the Message Buffer Space” on page 80 for planning information for message buffers.

Message buffer space is segregated by XCF by remote system and by transport class. Thus using up all the message buffer space for one particular remote system does not impact the message buffer space for any other remote system.

Message buffer space for a remote system is further segregated by transport class. Therefore, using up all the message buffer space for one particular transport class does not impact the message buffer space available to other transport classes for that remote system.

Note: Statements about limits on message buffer space apply to virtual storage. If there is insufficient central storage to back the buffers, the accumulation of message buffers for one system or transport class might affect the message buffers available to other systems or classes. The inability to obtain central storage for a message buffer is one reason for XCF to issue ABEND X'00C' - Reason X'06C'.

Adjusting Message Buffer Space

Assuming there is not an error condition, you can take the following actions to prevent message requests from being rejected:

Tuning a Sysplex

1. Increase the message buffer space that XCF is allowed to acquire.
Message buffer space can be increased with the SETXCF command.
To increase the message buffer space for message senders:
 - Increase the MAXMSG value for the transport class definition to provide additional default message buffer space for each possible remote system.
 - Selectively increase the MAXMSG value for the transport class definition to provide additional default message buffer space for communicating with each possible system.
 - Selectively increase the message buffer space for the local system in a particular transport class by increasing the MAXMSG value for LOCALMSG traffic in that class.

To increase the amount of message buffer space available for inbound messages, increase the MAXMSG value for one or more inbound signaling paths connected to the sending remote system.

Adding signaling paths also increases the amount of message buffer space available.

2. Modify or redistribute the workload so that less message traffic is generated.
One way to reduce message traffic is to change the characteristics of the workload. However, this might not be possible.

The report shown in Figure 24 on page 130 provides a summary of each group's impact on message traffic and can help determine which workloads (that is, groups) would be good candidates to shed or move in order to reduce the message traffic.

3. Tolerate the condition.
You may be able to tolerate the condition if it occurs infrequently or does not degrade performance and if the multisystem applications can tolerate the condition.

Using RMF Reports to Analyze Message Buffer Space

To determine whether the signaling service was unable to satisfy a message request, the Figure 23 on page 130 provides the following columns:

REQ REJECT

Indicates the number of message buffer requests not satisfied due to message buffer constraints.

This column occurs under several headings in the report:

OUTBOUND FROM sysname

Where it indicates the number of requests to send a message to a particular remote system that were rejected in a particular transport class.

INBOUND TO sysname

Where it indicates the number of times that XCF was not able to obtain an inbound message buffer in anticipation of receiving a new message.

LOCAL

Where it indicates the number of requests to send a message within the local system that were rejected in a particular transport class.

In the XCF Activity Report - Path Statistics shown in Figure 25 on page 131, the following columns under the heading "INBOUND TO sysname" are useful:

FROM/TO DEVICE OR STRUCTURE

Which identifies the device number or structure of the inbound/outbound signaling paths that could not obtain message buffers.

FROM SYSTEM

Which identifies the name of the remote system that may have suffered performance degradation because the inbound signaling path could not obtain message buffers.

REQ IN

Which identifies the number of requests.

BUFFERS UNAVAIL

Which indicates the number of times that XCF was not able to obtain an inbound message buffer for the signaling path in anticipation of receiving a new message.

When requests for message buffers are rejected, determine whether the maximum amount of space that XCF is allowed to use for message buffers was too small to support the load (and therefore degrades system performance), or whether there was some unusual/failure condition that caused the limits on buffer space to be reached (in which case the limits performed their intended function because they prevented large amounts of storage from being obtained and further impacting the system).

Analyzing a Lack of Outbound Message Buffer Space

In the XCF Activity Report shown in Figure 23 on page 130, the “REQ REJECT” columns under “OUTBOUND FROM sysname” and under “LOCAL” indicate the number of times a request to send a message was rejected because XCF could not obtain a message buffer. This count is presented for each remote system, by transport class.

Note that “REQ REJECT” can be incremented more than once for a single message. Some multisystem applications might retry to send a message one or more times before taking some other action (such as setting a timer to retry later). Thus, REQ REJECT reflects the number of attempts to send one or more messages without success.

If outbound messages are being rejected,

- Check the corresponding inbound path at the receiving system. If that system is periodically non-operational, it can cause messages to back up on the outbound path of the sending system. To resolve the problem, you can reduce these outages on the receiving system or increase the message buffer space to allow the messages to be accepted for delivery during these non-operational periods.
- Use the XCF Activity Report shown in Figure 24 on page 130, to see if any of the members in the groups using the transport class have had an unusually high number of send message requests. This might indicate that an error condition is causing the members of that group to generate too many messages. Or perhaps the workload that the group must process simply requires a large number of messages. Resolving the error condition, changing the workload, or providing additional message buffer space are possible solutions.
- Look at RMF reports for the remote system to determine if that system was experiencing inbound message buffer constraints. Such constraints could cause delays in receiving messages and therefore cause outbound messages to back up.

Tuning a Sysplex

- For local message traffic, message request rejections are most likely due to a runaway sender, delays in receiving messages (such as scheduling delays), or simply not enough space.

Analyzing a Lack of Inbound Message Buffer Space

In the XCF Activity Report shown in Figure 23 on page 130, the “REQ REJECT” column under the heading “INBOUND TO sysname” indicates the number of times XCF was not able to obtain an inbound message buffer in anticipation of receiving a message over an inbound signaling path. This count is the sum of the number of times this condition occurred for each inbound signaling path and is presented for each possible remote sending system.

The inability to obtain an inbound message buffer does not necessarily indicate a problem. Signaling performance is degraded only if a message is pending on the outbound side of the signaling path. If an inbound message buffer is available before the next outbound message is accepted for delivery, the first message is not delayed. Therefore, use “REQ REJECT” under “INBOUND TO sysname” as an indicator that messages **might** have been delayed.

In the XCF Activity Report - Path Statistics shown in Figure 25 on page 131, the column labeled “BUFFERS UNAVAIL” can be used to identify the individual inbound signaling paths whose request for a new message buffer may have been rejected. By correlating the individual inbound signaling path back to the device on the outbound side, you can determine if the inability to obtain an inbound message buffer had a detrimental impact on the outbound side. Knowing the device number of the outbound path, you can identify the transport class to which the outbound signaling path is assigned. If the XCF Activity Report for the outbound system indicates that message requests targeted to this inbound system were rejected, there likely was an impact.

Even if messages on the outbound path were not rejected outright, there may still have been performance degradation. Use the report the system on the outbound signaling path to determine whether this inability to obtain an inbound buffer delayed the transfer of outbound messages over this signaling path.

Tuning the Signaling Paths

To achieve the best signaling performance you want a signaling path available to transfer a message when a send message request is made.

The RMF reports provide counts of the number of messages sent and received by the system that collected the data. If a system tends to send more messages than it receives, consider providing more outbound signaling paths than inbound paths.

Because the XCF Activity Report provides a breakdown of messages sent in each transport class, you can determine whether a transport class is used to send messages to a particular remote system. If a transport class is not used to send messages to a particular system, signaling paths are not needed in that class for the system. For transport classes that are used, determine if the number of signaling paths provided is sufficient to handle the message traffic. If there are too few paths, signaling performance is degraded; if there are too many, system resources are wasted.

Signaling Path Capacity

For transport classes with assigned signaling paths, you need to determine whether the number of outbound paths leading to a particular remote system is adequate to

support the message traffic. To decide whether too few or too many signaling paths are provided, use the XCF Activity Report.

In the XCF Activity Report - Path Statistics shown in Figure 25, a summary of the messages sent on each signaling path is provided. Under “OUTBOUND FROM sysname”, the column labeled “FROM/TO DEVICE OR STRUCTURE:”, indicates the inbound/outbound devices or structure that handled the data; the column “TO SYSTEM” indicates the system to which the path is connected; and “TRANSPORT CLASS” indicates the transport class to which the path is assigned. You can use the inbound device number to associate the data for the outbound signaling path with data presented for the inbound side of the signaling path in an RMF report using data collected on the remote system. The column labeled “REQ OUT” indicates the total number of attempts made to send a message over the indicated outbound signaling path.

If there is more than one outbound signaling path assigned to a transport class and leading to the same system, the “REQ OUT” column can be used to determine the distribution of messages among the paths. XCF attempts to use a signaling path that can deliver a message with the least amount of delay. One potential sign of excess signaling path capacity is for “REQ OUT” on one path to greatly exceed the “REQ OUT” counts for the other paths in the class that lead to the same system. Consider assigning a less-used signaling paths to a transport class that needs additional capacity, or removing the signaling paths from service so that they do not tie up resources.

Note: It is recommended that at least two outbound and two inbound signaling paths be kept in service (even though they may not be needed for capacity) in order to avoid a single point of failure. It is not necessary that each transport class have at least two outbound signaling paths, only that there be at least two outbound (and two inbound) signaling paths between systems.

Signaling Path Performance

In the XCF Activity Report shown in Figure 25 on page 131, the columns “AVAIL” and “BUSY” provide a breakdown of the number of times the path was selected while available to immediately transfer a message (AVAIL), and the number of times the signaling path was selected while a message was being transferred (BUSY). XCF prefers to use signaling paths that can immediately transfer a message because those paths should provide the least amount of delay in delivering a message. Therefore, interpret the “AVAIL” counts that are high relative to “BUSY” counts as an indication that the signaling service is performing well.

As “BUSY” approaches “AVAIL”, consider adding signaling paths to increase capacity, thereby increasing the possibility that XCF is able to select an idle path for message transfer. Or, consider changing the transport class definition so that a different set of groups can use the path. Signaling performance can be enhanced by isolating the groups that send messages frequently into their own class.

It is also possible for the “BUSY” count of an outbound path to be relatively high if the inbound signaling path on the remote system does not have sufficient message buffer space to accommodate the inbound message traffic. This can cause signals on the outbound path of the sending system to back up, thereby causing the “BUSY” counts to rise. If the “BUSY” counts are high, run an RMF XCF Activity Report - Path Statistics on the remote system. Look for the column labeled “BUFFER UNAVAIL” under the heading “INBOUND TO sysname”. If the “BUFFER UNAVAIL” count for the inbound signaling path connected to the subject outbound path is not zero, it may indicate that the outbound path’s “BUSY” count is high

Tuning a Sysplex

because the inbound path does not have sufficient message buffer capacity to handle the message traffic. Consider increasing the amount of message buffer space that the inbound path can acquire.

Analyzing Message Counts

In the XCF Activity Report - Path Statistics shown in Figure 25 on page 131, the column labeled “REQ OUT” indicates the total number of attempts made to send a message over the indicated outbound signaling path. Note that summing the “REQ OUT” for all paths leading to a system may not equal the total number of messages sent to that system as presented in the RMF XCF Activity Report - Usage by System, nor will it necessarily equal the sum of the messages sent to that system as presented in the RMF XCF Activity Report - Usage by Member. Often these differences occur because the data for the reports is collected at slightly different times. But it is also the case that there is a subtle difference between the counts in the path statistics report for outbound paths and the counts in the other reports. The path statistics report counts the number of attempts to send a message over a path. The other reports count the number of messages accepted by the signaling service for delivery.

When a signaling path suffers a failure (such as an I/O error), any messages that were not successfully transferred are re-sent on some alternate signaling path. The second attempt to send the message is counted in the path statistics report, but does not show up in the XCF usage reports because the message is accepted for delivery just once. Except for a delay in delivering the message, the re-send of a message is transparent to the message sender. Also, if signaling paths are started and stopped during the monitoring intervals, the path statistics report may not show those devices, hence the messages sent over those paths would not show up in the path statistics report. The XCF usage reports would, however, show those messages.

Impact of Signaling Path Restarts

In the XCF Activity Report - Path Statistics, shown in Figure 25 on page 131, the column labeled “RETRY” counts the number of times the signaling path was restarted and indicates the number of failures the signaling path has had. When this count is nonzero, it indicates that several attempts were made to send one or more messages. A message that cannot be sent on the first attempt suffers performance degradation. Even if all the messages were successfully sent before the path restart occurs (that is, all messages were sent on the first attempt), there may still be degraded performance for the signaling service because a path is not available for message transfer while it is being restarted. Thus, a path restart represents a loss of signaling capacity that may cause the message traffic to be distributed among fewer signaling paths, thereby degrading signaling performance.

Restarts can occur as systems enter and leave the sysplex, or as signaling paths in the sysplex are stopped and started. Barring these kinds of configuration changes, a high restart count suggests that signaling performance suffered because the path was not always operational. XCF automatically stops using a signaling path if path restarts occur so frequently that the device appears to be unusable.

Impact of Having No Paths in a Class

In the XCF Activity Report - Usage by System, shown in Figure 23 on page 130, the column labeled “ALL PATHS UNAVAIL” indicates the number of outbound message requests that were migrated to a signaling path in another transport class because there was no operational signaling path connected to the intended remote system and assigned to the selected transport class. A particular transport class could fail to have an operational signaling path for a particular remote system because:

- There are no signaling paths assigned to that class that lead to the remote system.
- All the signaling paths assigned to that class that lead to the remote system have temporarily or permanently failed.

When “ALL PATHS UNAVAIL” is not zero, there is interference between the transport classes. This means that the desired segregation of message traffic is not obtained. Generally, you want to avoid this condition by assigning signaling paths to the transport class that lead to the intended remote system. Messages migrated to an alternate signaling path incur additional overhead for XCF to select an alternate transport class. Furthermore, these messages can degrade the performance of messages sent in the transport classes that are chosen as alternates; particularly if the buffer size used by the alternate class is smaller than the buffer size required by the migrated message.

Thus “ALL PATHS UNAVAIL” can be an indicator of severe problems with the transport class definitions and/or the resources assigned to the class. However, the severity of the interference between classes must be considered in light of the number of message requests in the transport class as listed under “REQ OUT”. If these messages occur relatively infrequently, it may not be worth providing a signaling path for their exclusive use.

Tuning the Transport Classes

An integral part of making a transport class definition is determining which groups should be assigned to the transport class. You might want to assign a group to a particular transport class in order to dedicate signaling resources (such as a signaling path) to that group for its exclusive use. Exclusive access to signaling resources can enhance a group’s signaling performance. Or you might want to assign a group to a transport class so that it does not make use of signaling resources used by other groups, thereby preventing the group from degrading the performance of the other groups by using all the signaling resources.

The XCF Activity Report provides data that can be used to determine each group’s utilization of signaling resources. This report can identify which groups send and receive the most messages. If the number of messages sent by a group is relatively high compared to the other groups, it may enhance the performance of the other groups if the group that sends the most messages is isolated into its own transport class. This would reduce contention for the signaling resources for the groups that don’t send messages so often.

There might be cases where a group needs to be assured of excellent signaling performance even though it does not send many messages.

The RMF reports cannot make this kind of judgment call for you. Use your knowledge of what kinds of performance requirements the group requires, or what kind of impact it has on the rest of the system if that group has to compete with other groups for signaling resources. Again, defining a transport class for exclusive use of such a group and providing signaling resources for the class, would ensure that the group was isolated from other potential users of those resources.

Hints

If exploiters of the XCF signaling services have not provided their group name, message size, and message frequency requirements, there are several techniques you can use to determine some of this information. For example, use the XCF Activity Report - Usage by Member to identify group names and the frequency of

Tuning a Sysplex

messages sent and received by each group. Define a transport class and assign to that class just the group under investigation. Use the report to see the distribution of message sizes employed by that group. Adjusting the class length for the transport class definition and examining the report for this class may prove useful to the investigation.

Capacity Planning

For capacity planning, you try to predict how changes in workload will change the requirements for signaling resources. This might be done in order to predict future requirements as additional workload is brought into the system. Or it might be done to predict the impact to signaling resources if the workload is redistributed around the sysplex in a different configuration.

The RMF reports provide counts of the number of messages sent and provide indications of when performance problems occur. For capacity planning purposes, collecting this data over extended periods of time will help you evaluate the impact of changing workloads on the amount of message traffic generated in the sysplex. By periodically running these reports, you will have a history of data with which you can assess the impact of your changing workload. This will be most beneficial if you understand what the workload changes were to be able to associate those workloads with the RMF reports that provide you your signaling data.

Using the RMF Reports

The XCF Activity Report - Usage by Member provides counts of the messages sent to and received from each system, by group, member, and system. It also summarizes local messages sent and received, by local group and member.

You can use this report to understand the message traffic loads associated with the various groups and members that communicate with the system. With this understanding, you can do capacity planning.

A group is most likely to be either directly responsible for a workload, or to be performing a service function on behalf of a variety of workloads. Even if you are not able to associate a particular workload with a particular group, you can still use the reports to collect data that describes the history of your installation's use of the signaling resources. This data can be used to predict future needs. If a workload can be directly tied to a particular group, it may be possible to understand the impacts to signaling resources when that workload is moved to some other set of systems in the sysplex.

Remote Message Traffic: In the XCF Activity Report - Usage by Member under the heading labeled "MEMBERS COMMUNICATING WITH sysname", are counts of messages sent to and received from remote systems in the sysplex. The name of the group and member and the remote system on which the member resides are identified, respectively, under the columns GROUP, MEMBER, and SYSTEM. The column "REQ FROM sysname" provides the count of messages sent by one or more members on system "sysname" (the system on which the data was collected) to the indicated member (that resides on the remote system). The column "REQ TO sysname" provides the count of messages sent to one or more members on system "sysname" (the system on which the data was collected) by the indicated member (that resides on the remote system).

Use your historical data to predict what signaling resources will be needed as changing workloads impact the remote signaling performed by these groups. For example, you might predict that additional signaling paths will be required at some point in the future to provide sufficient signaling capacity. Or you might determine

that fewer signaling paths would be needed if the members of the remote system were moved to the local system or if some workload that uses services provided by the group were moved to the local system. You might not always be able to move a group member because the multisystem application may not have been structured to permit this kind of placement of members.

You can use the XCF Activity Report - Path Statistics to watch the “BUSY” and “AVAIL” counts on a signaling path. If “BUSY” grows relative to “AVAIL” over time, it suggests that the message traffic workload is increasing. You might want to add additional signaling paths to maintain signaling performance.

In the XCF Activity Report - Usage by Member under the heading labeled “MEMBERS ON sysname”, are counts of messages sent and received by each member on system “sysname”, the local system that collected the data. The name of the group and member is identified respectively under the columns GROUP and MEMBER. The column “REQ OUT” indicates the number of messages sent by the member. The column “REQ IN” indicates the number of message received by the member. Use this data to determine which members on the local system are responsible for most of the message traffic.

If there is only one member per group on each system, you might be able to determine the precise message traffic patterns between particular group members. However, this is not possible to do in the general case.

Local Message Traffic: In this report, the “REQ OUT” count includes all messages sent by the indicated member, including any messages sent to members (possibly itself) on the local system. Similarly, “REQ IN” includes all messages received by the indicated member, including those sent by members (possibly itself) on the local system.

To determine how many messages were sent within the local system by a particular group, take “REQ OUT” and “REQ IN” and discard those signals that were sent to or received from another system. For example, to determine how many signals were sent by members of a local group, take the total for the “REQ OUT” values for all the members of that group as reported under the heading “MEMBERS ON sysname”. Then take the total “REQ FROM sysname” value for that group from under the heading “MEMBERS COMMUNICATING WITH sysname”. Subtract the total “REQ FROM” value from the total “REQ OUT” value and you will have the total number of signals sent locally by that group. With this data you can predict what signaling resources will be needed to support local message traffic in the future.

RMF XCF Activity Report — Sample

This section includes a sample XCF Activity Report.

For a description of RMF reports, see *RMF User's Guide*.

Also, see *OS/390 MVS Programming: Sysplex Services Guide* and *OS/390 MVS Programming: Sysplex Services Reference* for descriptions of the IXCMG and IXLMG macros, which can also be used to obtain tuning and capacity planning information.

Tuning a Sysplex

XCF Activity Report — Usage by System

X C F A C T I V I T Y														
MVS/ESA SP5.1.0		SYSTEM ID SYS1 RPT VERSION 5.1.0				DATE 03/14/94 TIME 16.30.00			INTERVAL 15.00.000 CYCLE 1.000 SECONDS			PAGE 1		
XCF USAGE BY SYSTEM														
REMOTE SYSTEMS										LOCAL				
OUTBOUND FROM SYS1					INBOUND TO SYS1					SYS1				
TO SYSTEM	TRANSPORT CLASS	BUFFER LENGTH	REQ OUT	---- % SML	---- % FIT	---- % BIG	---- % OVR	ALL PATHS UNAVAIL	REQ REJECT	FROM SYSTEM	REQ IN	REQ REJECT	TRANSPORT CLASS	REQ REJECT
SYS3	DEFAULT	956	334,956	0	100	0	0	0	0	SYS3	335,259	0	DEFAULT	0
TOTAL			334,956				TOTAL			335,259				

Figure 23. XCF Activity Report - Usage by System

XCF Activity Report — Usage by Member

X C F A C T I V I T Y														
MVS/ESA SP5.1.0		SYSTEM ID SYS1 RPT VERSION 5.1.0				DATE 03/14/94 TIME 16.30.00			INTERVAL 15.00.000 CYCLE 1.000 SECONDS			PAGE 2		
XCF USAGE BY MEMBER														
MEMBERS COMMUNICATING WITH SYS1										MEMBERS ON SYS1				
GROUP	MEMBER	SYSTEM	REQ FROM SYS1	REQ TO SYS1	GROUP	MEMBER	REQ OUT	REQ IN						
COFVLFNO	SYS3	SYS3	52	3	COFVLFNO	SYS1	52	3						
TOTAL			52	3	TOTAL			52	3					
					SYSATB01	M222 M223	0 0	0 0						
					TOTAL			0	0					
SYSATB02	M95 M96	SYS3 SYS3	0 0	0 0										
TOTAL			0	0										
SYSDAE	SYS3	SYS3	0	0	SYSDAE	SYS1	0	0						
TOTAL			0	0	TOTAL			0	0					
SYSGRS	SYS3	SYS3	335,177	335,177	SYSGRS	SYS1	335,177	335,177						
TOTAL			335,177	335,177	TOTAL			335,177	335,177					
SYSMCS	SYS3	SYS3	39	65	SYSMCS	SYS1	39	65						
TOTAL			39	65	TOTAL			39	65					
SYSMCS2	SYS3	SYS3	0	0	SYSMCS2	SYS1	0	0						
TOTAL			0	0	TOTAL			0	0					
SYSWLM	SYS3	SYS3	14	14	SYSWLM	SYS1	14	14						
TOTAL			14	14	TOTAL			14	14					

Figure 24. XCF Activity Report - Usage by Member

XCF Activity Report — Path Statistics

X C F A C T I V I T Y												
MVS/ESA SP5.1.0		SYSTEM ID SYS1 RPT VERSION 5.1.0			DATE 03/14/94 TIME 16.30.00			INTERVAL 15.00.000 CYCLE 1.000 SECONDS				
TOTAL SAMPLES = 900				XCF PATH STATISTICS								
OUTBOUND FROM SYS1								INBOUND TO SYS1				
TO SYSTEM	T FROM/TO Y DEVICE, OR P STRUCTURE	TRANSPORT CLASS	REQ OUT	AVG Q LNTH	AVAIL	BUSY	RETRY	FROM SYSTEM	T FROM/TO Y DEVICE, OR P STRUCTURE	REQ BUFFERS IN UNAVAIL		
SYS3	C 0CA0 TO 0CA0	DEFAULT	91,864	0.04	90,789	1,075	0	SYS3	C 0CA1 TO 0CA1	84,966	0	
	C 0C80 TO 0C80	DEFAULT	86,081	0.02	84,990	1,091	0		C 0C81 TO 0C81	78,088	0	
	C 0EA0 TO 0EA0	DEFAULT	87,490	0.03	86,423	1,067	0		C 0EA1 TO 0EA1	82,012	0	
	C 0E80 TO 0E80	DEFAULT	77,021	0.02	75,955	1,066	0		C 0E81 TO 0E81	94,293	0	
TOTAL			342,456					TOTAL		339,359		

Figure 25. XCF Activity Report - Path Statistics

Tuning OS/390 UNIX Performance in a Sysplex

Starting with OS/390 Release 9, a sysplex can include a shared HFS for use by OS/390 UNIX. The use of a shared HFS may affect response time and throughput on file systems being shared in a sysplex because of the increase in XCF message traffic. Therefore, it is important to closely monitor XCF signaling reports in order to benefit from the additional availability and recoverability that shared HFS offers.

For example, SY1, a client system, requests a read on a file system mounted on SY2. Using shared HFS support, SY1 sends a message requesting this read to SY2 via an XCF messaging function:

```
SY1 ==> (XCF messaging function) ==> SY2
```

After SY2 gets this message, and the file becomes available for a read, SY2 gathers the requested data from the file. If the file is frequently accessed, SY2 may be able to simply retrieve the data from its cache. SY2 then returns the data via the same route the request message took:

```
SY2 ==> (XCF messaging function) ==> SY1
```

The increase in message traffic requires that you use the appropriate RMF reports to monitor both message buffer and signaling path utilization. It may be necessary to increase the number of message buffers and to define additional XCF signaling paths.

Tuning Coupling Facilities

The Coupling Facility (CF) provides hardware assistance to the management of global resources and workloads in a Parallel Sysplex environment. Requests to the CF can be SYNChronous (the sending processor waits for the completion of the request) or ASYNChronous (the sending processor continues with other work until it recognizes that the request has completed). In some cases, when SYNC requests encounter delays, they are changed to ASYNC requests.

There are two key indicators that can be used to determine if the coupling facility is performing well: the response time for a coupling facility request (service time), and the number of SYNC requests that were changed to ASYNC requests (CHNGD requests). The SYNC and ASYNC service times, as well as the CHNGD requests,

Tuning a Sysplex

are reported on the RMF Coupling Facility Structure Activity report, as seen in Figure 26.

```

                                COUPLING FACILITY ACTIVITY
                                PAGE 1
OS/390                          SYSPLX UTCPXJ8          DATE 05/01/1997      INTERVAL 030.00.000
REL. 01.03.00                    RPT VERSION 1.3.0      TIME 16.00.00       CYCLE 01.000 SECONDS
-----
COUPLING FACILITY NAME = CF1
-----
                                COUPLING FACILITY STRUCTURE ACTIVITY
-----
STRUCTURE NAME = COUPLE_CKPT1    TYPE = LIST
# REQ  -----  REQUESTS  -----  DELAYED REQUESTS  -----
SYSTEM TOTAL  #  % OF  -SERV TIME(MIC)-  REASON  #  % OF  --- AVG TIME(MIC)  ---  EXTERNAL
NAME  AVG/SEC  REQ  ALL  AVG  STD_DEV  #  REQ  REQ  /DEL  STD_DEV  /ALL  CONTENTIONS
-----
J90   9114  SYNC  3127  3.4%  296.6  118.3  NO SCH 1831  30.6%  997.4  2362  305.0  REQ TOTAL  897
      5.06  ASYNC 5949  6.5%  1524.9  2108.8
      CHNGD 38  0.0%  INCLUDED IN ASYNC
  
```

Figure 26. Example of RMF CF Structure Report

In this example, during a 30 minute interval, the JES2 Checkpoint structure had 3127 SYNC requests that completed in an average service time of 296.6 microseconds. There were 5949 ASYNC requests and 38 SYNC requests that were changed (CHNGD) to ASYNC requests. The ASYNC and CHNGD requests completed in an average of 1524.9 microseconds.

Service Time

Service time is accumulated from the time MVS issues a CF command to the CF until the return from the command is recognized by MVS. It includes the time spent on the CF links, the processing within the CF and time for MVS to recognize that the command has completed. Service time in microseconds is recorded for each structure used by each system.

What are the best service times you can expect using the latest technology? The following chart shows ranges for SYNC and ASYNC times for the earliest IBM CMOS CECs to the earliest IBM CMOS CFs and for the latest ones. The lower number in each range applies to requests with little or no data to transfer (like lock requests) and the higher number to the requests with larger amounts of data (like DB2 cache updates).

	9672-R61 to C01	9672-RX6 to R06
SYNC	250 - 350	25 - 50
ASYNC	1500 - 5000	250 - 750

CHNGD Requests

There are two varieties of SYNC requests: Immediate SYNC requests (like locks) which must be completed as soon as possible. Non-immediate SYNC requests can be delayed, if necessary. If MVS determines that a non-immediate SYNC request will be delayed (probably because the subchannels are busy), it will change the request to an ASYNC request. These requests are counted in the CHNGD field of the RMF CF Structure Activity report and their service times are included in the ASYNC service times.

The total number of requests CHNGD should be less than 10% of the total requests to a given structure.

Using the previous guidelines, let's look at a sample RMF CF Structure Activity report. In this example, system J90 is a 9672-R54 connected to CF1, a 9674-C01, so we would expect service times between the upper and lower guidelines.

COUPLING FACILITY ACTIVITY													PAGE 1
OS/390 REL. 01.03.00		SYSPLEX UTCPLXJ8 RPT VERSION 1.3.0			DATE 05/01/1997 TIME 16.00.00			INTERVAL 030.00.000 CYCLE 01.000 SECONDS					

COUPLING FACILITY NAME = CF1													

COUPLING FACILITY STRUCTURE ACTIVITY													

STRUCTURE	NAME =	IXCPLEX_PATH2		TYPE = LIST			REQUESTS			DELAYED REQUESTS			
SYSTEM	# REQ			% OF	-SERV	TIME(MIC)	REASON	#	% OF	AVG	TIME(MIC)		
NAME	TOTAL		#	ALL	AVG	STD_DEV		REQ	REQ	/DEL	STD_DEV	/ALL	
AVG/SEC			REQ										
J90	155K	SYNC	0	0.0%	0.0	0.0							
	86.21	ASYNC	155K	13.1%	1418.4	3637.9	NO SCH	20K	13.1%	2894	4209	377.7	
		CHNGD	0	0.0%	INCLUDED	IN ASYNC							

STRUCTURE	NAME =	COUPLE_CKPT1		TYPE = LIST			REQUESTS			DELAYED REQUESTS			
SYSTEM	# REQ			% OF	-SERV	TIME(MIC)	REASON	#	% OF	AVG	TIME(MIC)		EXTERNAL
NAME	TOTAL		#	ALL	AVG	STD_DEV		REQ	REQ	/DEL	STD_DEV	/ALL	CONTENTIONS
AVG/SEC			REQ										
J90	9114	SYNC	3127	3.4%	296.6	118.3							
	5.06	ASYNC	5949	6.5%	1524.9	2108.8	NO SCH	1831	30.6%	997.4	2362	305.0	REQ TOTAL 897
		CHNGD	38	0.0%	INCLUDED	IN ASYNC							REQ DEFER

STRUCTURE	NAME =	IRLMLOCK1		TYPE = LOCK			REQUESTS			DELAYED REQUESTS			
SYSTEM	# REQ			% OF	-SERV	TIME(MIC)	REASON	#	% OF	AVG	TIME(MIC)		EXTERNAL
NAME	TOTAL		#	ALL	AVG	STD_DEV		REQ	REQ	/DEL	STD_DEV	/ALL	CONTENTIONS
AVG/SEC			REQ										
J90	191K	SYNC	191K	8.8%	212.2	78.8							
	106.0	ASYNC	0	0.0%	0.0	0.0	NO SCH	0	0.0%	0.0	0.0	0.0	REQ TOTAL 21
		CHNGD	0	0.0%	INCLUDED	IN ASYNC	DUMP	0	0.0%	0.0	0.0	0.0	REQ DEFER 90
													-CONT 90
													-FALSE CONT 13

STRUCTURE	NAME =	RLSCACHE01		TYPE = CACHE			REQUESTS			DELAYED REQUESTS			
SYSTEM	# REQ			% OF	-SERV	TIME(MIC)	REASON	#	% OF	AVG	TIME(MIC)		
NAME	TOTAL		#	ALL	AVG	STD_DEV		REQ	REQ	/DEL	STD_DEV	/ALL	
AVG/SEC			REQ										
J90	85288	SYNC	85K	9.9%	328.0	126.3							
	47.38	ASYNC	0	0.0%	1847.3	4383.3	NO SCH	455	102%	3476	4908	3538	
		CHNGD	447	0.1%	INCLUDED	IN ASYNC							

TOTAL	859K	SYNC	837K	97.4%	308.5	147.1							-- DATA ACCESS -
	477.0	ASYNC	20K	2.3%	741.7	1355.4	NO SCH	2095	9.5%	4341	8261	414.5	READS
		CHNGD	2027	0.2%			DUMP	0	0.0%	0.0	0.0	0.0	WRITES 37095
													CASTOUTS 9116
													XI'S 86446

Figure 27. Example of RMF CF Structure Report

Figure 27 shows one of each type of CF structure: List, Serialized List, Lock and Cache.

- IXCPLEX_PATH2 is a LIST structure used for XCF signaling. All requests to this structure will be asynchronous requests. These requests transfer 1K XCF signals (the size of the buffers in the transport class using this structure as a path) so we would expect their service times to be near the low ASYNC guideline (1500) and 1418 is.
- COUPLE_CKPT1 is a serialized list structure used by the JES2 Checkpoint. The SYNC requests read changes made by other systems. Since they transfer data, we would expect them to be near the top of the SYNC range (350) and 296 is. The ASYNC requests are 4K blocks of checkpoint updates which occurred throughout the HOLD period. Since these are smaller ASYNC signals we would expect them to be near the low end of the ASYNC range (1500) and 1524 is. Some of the SYNC requests were changed to ASYNC because the subchannels were busy. Less than 10% (0.0 in this case) were CHNGD
- IRLMLOCK1 is a LOCK structure used to serialize the sharing of the IMS database. All requests to this structure will be synchronous immediate requests

Tuning a Sysplex

with very little data. None of these should be CHNGED and the average SYNC service time should be below the low end of the SYNC range (250). 212 fits nicely.

- RLSCACHE01 is a CACHE structures used to cache VSAM RLS data. All the request are SYNC requests which transfer data to the cache so we would expect them to be slightly below the high SYNC range (350). 328 is reasonable. Notice that some of the SYNC requests were changed to ASYNC because the subchannel was busy. They completed within the ASYNC range (1500-5000) and less than 10% (.01) were changed.

In this example, all the service times fall within the guidelines.

Potential Sources of Delay

If the service times or the percentage of requests changed exceed the recommended values, there are four potential sources of delay:

- Insufficient CF capacity
- Contention for CF paths
- IOP contention
- Shortage of CF subchannels

For each of these, we'll look at the indicators which point to the problem and discuss some options for correcting it.

Insufficient CF Capacity

If both the ASYNC and SYNC service times for the various structures exceed the recommended values, the first thing to check is the CF utilization. The RMF CF Usage Summary Reports shows the AVERAGE CF UTILIZATION of the processors in the CF.

COUPLING FACILITY ACTIVITY											PAGE 1
OS/390		SYSPLEX UTCPLXJ8		DATE 05/01/1997		INTERVAL 030.00.000					
REL. 01.03.00		RPT VERSION 1.3.0		TIME 16.00.00		CYCLE 01.000 SECONDS					

COUPLING FACILITY NAME = CF1											

COUPLING FACILITY USAGE SUMMARY											

STRUCTURE SUMMARY											

TYPE	STRUCTURE NAME	STATUS CHG	ALLOC SIZE	% OF CF STORAGE	# REQ	% OF ALL REQ	AVG REQ/ SEC	LST/DIR ENTRIES TOT/CUR	DATA ELEMENTS TOT/CUR	LOCK ENTRIES TOT/CUR	RECLAIMS
LIST	IXCPLX_PATH2	ACTIVE	31M	1.5%	996232	16.3%	553.46	7611	7596	N/A	
	LG2_DFHLOG_001	ACTIVE	26M	1.3%	44159	0.7%	24.53	61 21K 15K	86 85K 42K	N/A N/A N/A	
LIST	COUPLE_CKPT1	ACTIVE	13M	0.6%	71760	1.2%	39.87	3192 3007	14K 3182 3007	3 2 1	0
LOCK	IRLMLOCK1	ACTIVE	31M	1.5%	1627K	26.7%	904.00	114K 27	0 0	4194K 384	
CACHE	RLSCACHE01	ACTIVE	244M	12.1%	714874	11.7%	397.15	323K 323K	92K 92K	N/A N/A	
	VSAMCACHE1	ACTIVE	34M	1.7%	657723	10.8%	365.40	183K 101K	0 0	N/A N/A	

PROCESSOR SUMMARY											

AVERAGE CF UTILIZATION (% BUSY)			17.3	LOGICAL PROCESSORS:			DEFINED 6	EFFECTIVE 6.0			

Figure 28. Example of RMF CF Usage Report

Generally, best performance from the CF is obtained when the average CF utilization is less than 50%. If the utilization exceeds this, there are various actions that can be taken, depending on the environment.

- Verify that the number of logical processors defined is correct. There have been instances when the CF was configured to one logical processor during service and was not reset.
- Verify that the processor resource allocated to the CF is what you expected. If this is an LPAR environment and the CF partition is sharing CPs, the processor resource will be limited by the definition of the WEIGHT and CAP parameters and contention from the other partitions. If this CF resides on a processor with MVS images, information about the WEIGHT and CAP definitions can be found on the CPU Partition Data Report. The amount of processor resource available to the CF is reported as LOGICAL PROCESSORS EFFECTIVE. You will get the best response time by dedicating CP(s) to the CF partition.
- If this is a stand-alone CF check the Customize Operating Environment Panel to verify that the CF MODE has been specified (rather than I/O INTENSIVE). This will optimize the set of instructions used most frequently by the CF microcode.
- If you have multiple stand-alone CFs in your configurations, check the distribution of structures. If one CF has much higher utilization than the other, redistribute the structures based on ALLOC SIZE and # REQ.
- If all else fails, you can increase the capacity of the CF by adding CPs or obtaining an additional CF.

Contention for CF Paths

When a CF request obtains a subchannel, in most cases, it will proceed down the path to the CF and complete the processing with no further delays. However, if this is a PR/SM environment with multiple MVS images sharing CF links, the request could encounter a busy path. If this is a SYNC request, the request is immediately retried until it obtains a path. The time spent retrying the request is accumulated in the SYNC service time. If this is an ASYNC request, the request is returned and goes back through the process of obtaining another subchannel, which may include requeuing for a subchannel.

The number of times a 'Path Busy' condition is encountered is reported on the RMF CF Subchannel Activity report in the BUSY COUNTS - PTH. To limit the additional overhead occurred in processing requests deferred for 'Path Busy', the percentage of requests encountering this delay should be less than 10%. For example, in Figure 29 on page 136, system J90 has 852192 total requests; using the guideline of 10%, you'd want less than 85219 PTH BUSYs. In this example there are only 5895.

If the percentage of PTH BUSYs exceeds 10%, you should consider dedicating the CF links to each MVS image or adding additional CF links.

Tuning a Sysplex

COUPLING FACILITY ACTIVITY														
OS/390		SYSPLEX UTCPLXJ8				DATE 05/01/1997			INTERVAL 030.00.000					PAGE 3 0
REL. 01.03.00		RPT VERSION 1.3.0				TIME 16.00.00			CYCLE 01.000 SECONDS					

COUPLING FACILITY NAME = CF1														

SUBCHANNEL ACTIVITY														
SYSTEM NAME	# REQ TOTAL	-- CONFIG --	--BUSY--	--COUNTS--	REQUESTS				DELAYED REQUESTS					
					# REQ	-SERVICE AVG	TIME(MIC)-	STD_DEV	# REQ	% OF REQ	-----	AVG TIME(MIC)	----	
J00	450326	SCH GEN	4 PTH 0	0	SYNC	256010	310.4	111.3	SYNC	1718	0.7%	569.3	816.7	3.8
	250.2	SCH USE	4 SCH 1718	1718	ASYNC	189413	2514.1	2538	ASYNC	27K	14.2%	6208	5679	879.4
		SCH MAX	4		CHANGED	1419	INCLUDED	IN ASYNC	TOTAL	29K	6.4%			
		PTH	2		UNSUCC	0	0.0	0.0						
...														
J90	852192	SCH GEN	8 PTH 5895	5895	SYNC	681178	245.8	148.7	SYNC	1930	0.3%	461.2	10831	1.3
	4733.4	SCH USE	4 SCH 1930	1930	ASYNC	164352	1457.3	3604	ASYNC	23K	14.1%	2753	4129	388.2
		SCH MAX	4		CHANGED	1107	INCLUDED	IN ASYNC	TOTAL	25K	3.0%			
		PTH	2		UNSUCC	0	0.0	0.0						
...														

Figure 29. Example of RMF CF Subchannel Activity Report

IOP Contention

If the ASYNC service times exceed the guidelines, one possible cause could be a delay in the IOP microcode. The IOP (or SAP) handles I/O to DASD, CTC traffic and ASYNC CF requests prior to HiPerLinks. Any one of these could encounter a delay in the IOP, which in turn would delay the others.

To determine if this delay is occurring, look at the RMF I/O Queuing Activity report. The AVG Q LNGTH records only I/O and CTC requests, but a queue length greater than 1.0 suggests delays are occurring and should be investigated.

In Figure 30, the large AVG Q LNGTH of 7.58 was due to a Director Port busy condition.

I/O QUEUING ACTIVITY							
MVS/ESA				SYSTEM ID J80			
SP5.1.0				RPT VERSION 5.1.0			
TOTAL SAMPLES = 7200				IOP	ACTIVITY RATE	AVG Q LNGTH	
				00	406.534	7.58	
LCU	CONTENTION	DELAY	% ALL	CH PATH	CHAN	CHPID	% DP
	RATE	Q	CH PATH	BUSY	PATHS	TAKEN	BUSY
		LNGTH	BUSY				BUSY
000C	4.176	0.02	0.06	29	1.895	73.33	0.32
				35	1.870	73.75	0.15
000F	65.181	23.10	0.06	29	12.203	67.32	2.49
				35	12.240	60.11	3.87

Figure 30. Example of RMF CF Subchannel Activity Report

Shortage of CF subchannels

Another possible cause for higher service times and excessive queuing may be a shortage in the number of CF links from the sending CECs. For each CF link, there are two subchannels for each MVS image. The data to be sent to the CF is loaded into these subchannels. If no subchannels are available, ASYNC requests are queued. Non-immediate SYNC requests are changed to ASYNC requests (which are then also queued). Immediate SYNC requests (like locks) 'spin' waiting for the next available subchannel.

Data about the delayed requests is reported on the RMF CF Subchannel Activity report as SYNC and ASYNC DELAYED REQUESTS and summarized on the TOTAL line. For those requests which experience a delay, the duration of the delay is reported separately (/DEL). (Some of these fields were added or changed by RMF APAR OW13536 which was included in OS/390 R2. WSC FLASH 9609 describes the changes and provides a detailed information about the data on the reports).

You can assess the impact of the subchannel delay, by looking at the /ALL field. This takes the delay time for the requests that were delayed and amortizes it over all the requests. So, for example, in Figure 29 on page 136, for system J90, 1930 SYNC requests encountered a subchannel busy. Each of these was delayed for an average of 461.2 microseconds. Since the guideline for SYNC request service times is 250-350 microseconds, this looks like a problem. However, when this delay is amortized over all the SYNC requests as shown under /ALL, it is only 3.8 microseconds.

As a guideline, the sum of the SYNC and ASYNC requests delayed (TOTAL - % OF REQ) should be less than 10% of all requests. If the percentage of requests queued is greater than this, you should:

- Verify the configuration is what you expected. This can be done by looking at the RMF CF Subchannel activity report. The number of subchannels configured (SCH GEN) tells you how many subchannels were defined in the IOCDs. (HCD automatically defines 2 subchannels for each CF link). RMF also reports the number of subchannels which are currently being used (SCH USE) and how many CF links are currently connected (PTH). If the number of subchannels currently being used are less than the number of subchannels defined, verify that you have not lost connectivity on some CF links.
- If all the CF links are connected, you may want to consider additional CF links or the new HiPerLinks. If the CF is totally configured for CF paths, you may want to consider moving a structure to another CF facility or adding another CF.

Tuning a Sysplex

Chapter 7. Planning Sysplex Availability and Recovery

When work on one system in a sysplex cannot complete because the system fails, other systems in the sysplex remain available to recover the work and continue processing the workload. The goals of failure management in a sysplex are to minimize the impact that a failing system might have on the sysplex workload so that work can continue, and to do this with little or no operator intervention.

The actions MVS is to take in failure situations is determined by the information specified through the COUPLExx parmlib member, the SFM policy, the XCFPOLxx parmlib member, and the automatic restart management policy.

- **COUPLExx Parmlib Member**

From COUPLExx, MVS obtains basic failure-related information, such as when to consider a system to have failed and when to notify the operator of the failure. (COUPLExx parmlib specifications might be different for each system, depending on its workload, processor capacity, or other factors.)

- **SFM Policy**

If all systems in a sysplex are running OS/390 or MVS/ESA SP Version 5, you can use the sysplex failure management (SFM) policy to define how MVS is to handle system failures, signaling connectivity failures, or PR/SM reconfiguration actions. Although you can use SFM in a sysplex without a coupling facility, to take advantage of the full range of failure management capabilities that SFM offers, a coupling facility must be configured in the sysplex.

SFM makes use of some information specified in COUPLExx and includes all the function available through XCFPOLxx.

The SFM policy also can be used in conjunction with the REBUILDPERCENT specification in the CFRM policy to determine whether MVS should initiate a structure rebuild when loss of connectivity to a coupling facility occurs.

- **XCFPOLxx Parmlib Member**

In a multisystem sysplex on a processor with the PR/SM feature, XCFPOLxx functions can provide some of the same capabilities as those provided by the SFM policy. XCFPOLxx functions are also referred to as the XCF PR/SM policy.

- **Automatic Restart Management Policy**

Use the automatic restart management policy to specify how batch jobs and started tasks that are registered as elements of automatic restart management should be restarted. The policy can specify different actions to be taken when a system fails, and when an element fails. Automatic restart management uses the IXC_WORK_RESTART exit, the IXC_ELEM_RESTART exit, the event exit, and the IXCARM macro parameters, in conjunction with the automatic restart management policy (the specified values and the defaults) when determining how to restart elements.

| Other sysplex and couple data set failures, such as those caused by a power
| failure, might require operator intervention. See “Handling Concurrent System and
| Couple Data Set Failures” on page 157.

Controlling Availability and Recovery through COUPLExx

The following topics describe the recovery-related parameters in COUPLExx.

Planning Availability and Recovery

Planning the Failure Detection Interval and Operator Notification Interval

On each system in the sysplex, MVS periodically updates its own status and monitors the status of other systems in the sysplex. A *status update missing condition* occurs when a system in the sysplex does not update its status information within a certain time interval. This time interval is the failure detection interval. You specify this on the INTERVAL keyword in COUPLExx.

The operator notification interval is the amount of time between when a system no longer updates its status and when another system issues message IXC402D to inform the operator of the condition. You specify the operator notification interval on the OPNOTIFY keyword in COUPLExx.

After IPL, you can modify the failure detection and operator notification intervals using the SETXCF COUPLE command.

Considerations

The operator notification interval must be equal to or greater than the failure detection interval. For many installations, the default values for the failure detection interval and the operator notification interval are reasonable.

If you are not using the default failure detection interval, you need to evaluate the tradeoff between removing a failing system immediately to prevent disruption to your sysplex and tolerating a period of non-productivity so that the system has a chance to recover.

If you specify these intervals, consider the following:

- If SFM is active in the sysplex, and ISOLATETIME is specified, then the operator is not prompted.

If SFM or XCFPOLxx is active and RESETTIME or DEACTTIME is specified, then the operator notification time runs in parallel to the RESETTIME or the DEACTTIME.

Otherwise, to ensure that the operator receives timely notification of a failure, specify an operator notification interval that is the same as or only slightly greater than the failure detection interval.

- To avoid an unwanted status update missing condition because of an excessive spin condition, use the default failure detection interval. The default failure detection interval is two times the default spin loop timeout interval (SPINTIME in the EXSPATxx parmlib member) plus 5 seconds.
- For a system running under VM or in a logical partition that is sharing central processing resources, if you set the INTERVAL value to less than the default value in order to expedite the removal of an unresponsive system, it is recommended that you also adjust the SPINTIME parameter in the EXSPAT parmlib member to expedite MVS spin loop recovery actions. Making a corresponding adjustment to SPINTIME can ensure that MVS has time to perform spin loop recovery actions before the failure detection interval expires. For more information, see “Specifying the SPINRCVY Statement in the EXSPAT Parmlib Member” on page 24.
- XCF signaling uses the failure detection interval of the inbound system to determine when an outbound signaling path is inoperative. If signals are queued for transfer longer than the receiving system’s failure detection interval, the path is restarted.

Planning the Cleanup Interval

When MVS is removing a system from a sysplex, the cleanup interval is the maximum number of seconds that MVS waits for members of a group to clean up their processing before putting the system on which they reside in a wait state. (Note that this is not a required protocol — some groups, such as the global resource serialization group and the console services groups, delay cleanup until the expiration of the cleanup interval. When all members finish their cleanup or when the cleanup interval elapses, MVS puts the system in a non-restartable wait state.

For example, there are two systems in a sysplex (SYS1 and SYS2). On SYS1 the VARY XCF,SYS2,OFFLINE command is issued. MVS on SYS2 notifies the group members on SYS2 that SYS2 is going to be removed from the sysplex and then waits until the members have performed cleanup, or until the cleanup interval elapses, whichever comes first, before putting the system in a wait state.

You specify the cleanup interval on the CLEANUP keyword of the COUPLE statement of the COUPLExx parmlib member or, after IPL, using the SETXCF COUPLE command.

Controlling System Availability and Recovery through the SFM Policy

Sysplex failure management (SFM) allows you to define a sysplex-wide policy that specifies the actions that MVS is to take when certain failures occur in the sysplex. A number of situations might occur during the operation of a sysplex when one or more systems need to be removed so that the remaining sysplex members can continue to do work. The goal of SFM is to allow these reconfiguration decisions to be made and carried out with little or no operator involvement.

Information on the SFM policy is presented in the following topics:

- “Overview and Requirements of SFM Policy”
- “Planning for a Status Update Missing Condition” on page 142
- “Handling Signaling Connectivity Failures” on page 144
- “Planning PR/SM Reconfigurations” on page 146
- “Setting Up an SFM Policy” on page 147.

Overview and Requirements of SFM Policy

An SFM policy includes the following statements:

- Policy statement
- System statement(s)
- Reconfiguration statement(s)

SFM allows you to define responses for:

- Signaling connectivity failures in the sysplex
- System failures, indicated by a status update missing condition
- Reconfiguring systems in a PR/SM environment.

If the sysplex includes a coupling facility, the full range of failure management capabilities that SFM offers is available to the sysplex. For SFM to handle signaling connectivity failures without operator intervention or to isolate a failing system, a coupling facility must be configured in the sysplex.

Requirements for Using the SFM Policy

For a sysplex to take advantage of an SFM policy, the policy must be active on all systems in the sysplex. That is:

Planning Availability and Recovery

- All systems must be running OS/390 or MVS/ESA SP Version 5.
- An SFM policy must be started in the SFM couple data set.
- All systems must have connectivity to the SFM couple data set.

If any system loses access to the SFM couple data set, the policy becomes inactive in the sysplex. If that system regains access to the SFM couple data set, SFM automatically becomes active again in the sysplex.

Similarly, if an MVS/ESA SP Version 4 system joins a sysplex where an SFM policy is active, the policy is disabled for the entire sysplex. When that system is removed from the sysplex, SFM automatically becomes active again in the sysplex.

Relationship between SFM Policy and XCF PR/SM Policy

If a system with an active XCF PR/SM policy is connected to a couple data set with a started SFM policy, the XCF PR/SM policy is deactivated, regardless of whether the SFM policy is active in the sysplex.

The following table summarizes what policy would be in effect for a system with an XCF PR/SM policy that is connected to the SFM couple data set, based on whether the SFM policy is started in the SFM couple data set and whether the other systems in the sysplex are connected to the SFM couple data set. Assume that all systems running OS/390 or MVS/ESA SP Version 5.

SFM Policy Started in Couple Data Set	All Systems Connected to Couple Data Set	Policy in Effect
No	Doesn't matter	XCF PR/SM Policy
Yes	No	Neither policy
Yes	Yes	SFM Policy

Planning for a Status Update Missing Condition

A status update missing condition occurs when a system does not update its status information within the time interval specified on the INTERVAL keyword in COUPLExx. SFM allows you to specify how a system is to respond to this condition. Depending on your sysplex configuration, you can specify one of the following responses on the system statement of the SFM policy: PROMPT, ISOLATETIME, RESETTIME, DEACTTIME.

Prompting the Operator

If PROMPT (the default) is specified, the system waits for the length of the operator notification interval (the OPNOTIFY value in COUPLExx) and then notifies the operator with message IXC402D.

Isolating a Failing System

System isolation allows a system to be removed from the sysplex without operator intervention, while ensuring that data integrity in the sysplex is preserved. Specifically, system isolation terminates I/O and coupling facility accesses, resets channel paths, and loads a non-restartable wait state on the failing system, thus ensuring that the system is unable to corrupt shared resources.

System isolation requires that a coupling facility be configured in the sysplex and that the system being isolated and at least one active system have connectivity to the same coupling facility.

Also note that a system that is manually reset or re-IPLed cannot be isolated and will therefore require manual intervention to be removed from the sysplex.

Planning Availability and Recovery

Therefore, to remove a system from the sysplex, it is recommended that you use the `VARY XCF,sysname,OFFLINE` command. If SFM is active, it will then attempt to isolate the system.

The `ISOLATETIME` parameter indicates how long SFM will wait after detecting a status update missing condition before starting to isolate the failing system.

- With or without `OW30926`, if a system has not updated its status within the failure detection interval and is producing no XCF signaling traffic, SFM will start to isolate the failing system at the expiration of the `ISOLATETIME` interval.
- With `OW30926`, if a system has not updated its status within the failure detection interval, but is continuing to produce XCF signaling traffic, the operator is prompted to optionally force the removal of the system. (The fact that XCF signaling continues indicates that the system is functional but may be experiencing a temporary condition that does not allow the system to update its status.) If the operator deems that removal of the system is necessary, message `IXC426D` provides the prompt to isolate the system and remove it from the sysplex. In this case, the `ISOLATETIME` interval specified in the SFM policy is ignored.

If XCF signaling also stops, SFM will start to isolate the failing system at the expiration of the `ISOLATETIME` interval.

If an isolation attempt is not successful (for example, if the failing system is not connected through a coupling facility to another system in the sysplex), message `IXC102A` prompts the operator to reset the system manually so that the removal can continue.

Figure 31 on page 144 shows a three-system sysplex with an active SFM policy. `SYSB` and `SYSC` are connected to a coupling facility. If either `SYSB` or `SYSC` enters a status update missing condition, the system can be isolated by the other. However, because `SYSA` is not connected to the coupling facility, it cannot participate in isolation in case of failure.

Planning Availability and Recovery

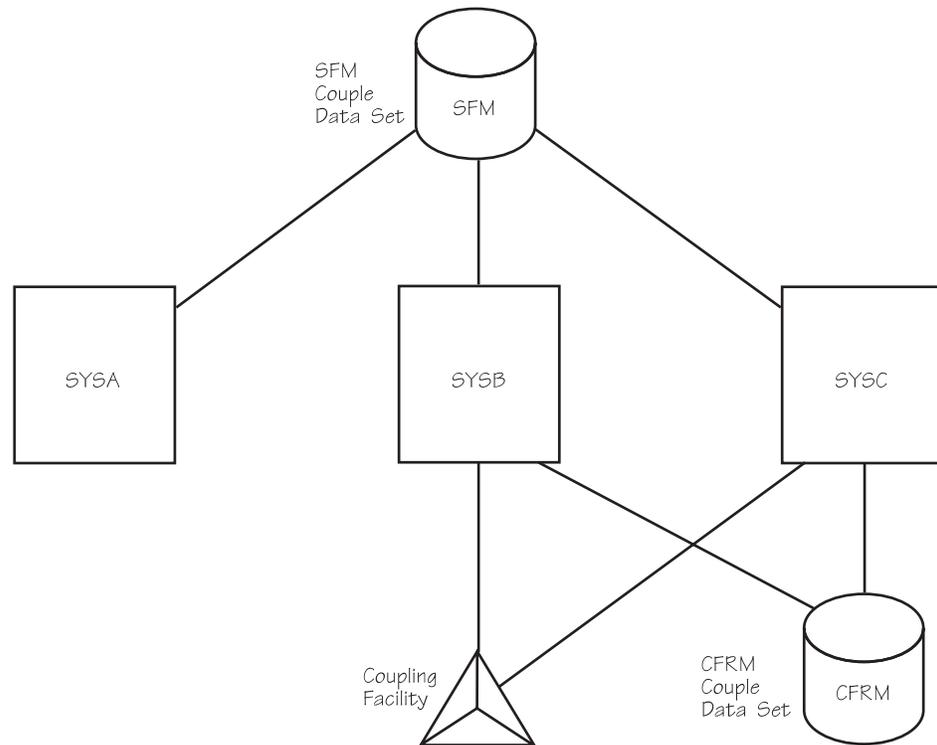


Figure 31. Three-System Sysplex with Active SFM Policy

Resetting or Deactivating a Failed System on PR/SM

When a system running in a PR/SM partition fails, another system in the sysplex that is running in a partition on the same processor can reset or deactivate the failed system and remove it from the sysplex without operator intervention.

The `RESETTIME` and `DEACTTIME` parameters allow you to specify how much time should elapse after the status update missing condition is detected before the action is taken.

These automatic actions have the same effect as the actions the operator would take if `PROMPT` were specified. Because the `RESETTIME` and `DEACTTIME` parameters do not terminate I/O in-progress, in cases where system isolation is possible, specifying `ISOLATETIME` is preferable.

Handling Signaling Connectivity Failures

All systems in the sysplex must have signaling paths to and from every other system at all times. Loss of signaling connectivity between sysplex systems can result in one or more systems being removed from the sysplex so that the systems that remain in the sysplex retain full signaling connectivity to one another. SFM can eliminate operator intervention when signaling connectivity between two or more systems is lost.

The `CONNFAIL` parameter indicates whether SFM is to handle signaling connectivity failures for the sysplex. You specify `CONNFAIL` on the `DEFINE POLICY` statement of the SFM policy.

Planning Availability and Recovery

If you specify CONNFAL(YES) (the default), and there is a signaling connectivity failure, SFM automatically determines which systems to keep and which to remove from the sysplex and then attempts to implement that decision by system isolation.

In handling a system connectivity failure, SFM attempts to maximize the aggregate value of the surviving sysplex to the installation. The WEIGHT parameter of the system statement allows you to indicate the relative value of each system in the sysplex so that SFM can base its decision on installation-specified values. Ensure that you consider carefully the relative importance of each system when assigning the weight. For example, if there is a system that should never be partitioned out of the sysplex when a connectivity failure occurs, its weight should be greater than the combined weights of all other systems in the sysplex. In cases where the weights of possible surviving sysplex configurations are equal, SFM's decision on which configuration to keep is arbitrary. (If SFM is unable to determine all potential reconfigurations in a timely manner, SFM chooses the best fully-connected configuration identified to that point.)

Figure 32, illustrates a sysplex made up of three systems, SYSA, SYSB, SYSC.

Assume that connectivity is lost between SYSA and SYSB.

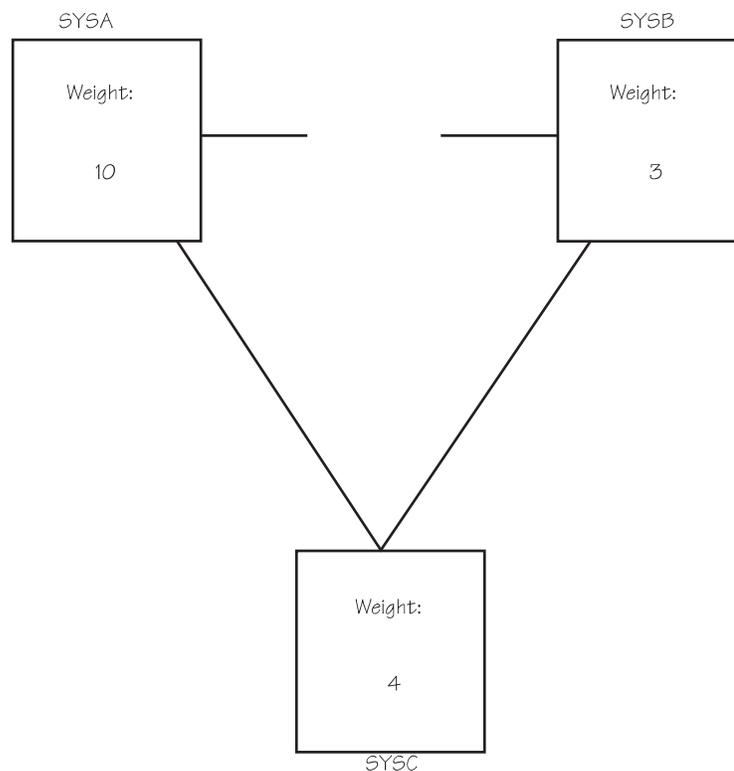


Figure 32. Signaling Connectivity Failure Between SYSA and SYSB

SFM determines that the sysplex can be reconfigured as SYSA and SYSC or as SYSB and SYSC. The new sysplex made up of SYSA and SYSC has a weight of 14, whereas the sysplex made up of SYSB and SYSC has a weight of 7. SFM chooses the sysplex with the higher weight and removes SYSB from the sysplex.

Planning Availability and Recovery

If a system that is being removed is connected to a coupling facility that is also connected to an active system, SFM can reconfigure the sysplex by isolating and removing the system without operator intervention. Without a coupling facility, a system cannot be isolated; in this case, message IXC102A prompts the operator to reset the system manually before it is removed from the sysplex.

CONNFAIL(NO) indicates that in a connectivity failure situation, MVS is to prompt the operator with message IXC409D to retry or remove the affected system(s).

Handling Coupling Facility Connectivity Failures

Loss of connectivity to a coupling facility can occur because of a failure of the coupling facility attachment or because of certain types of failures of the coupling facility itself. MVS provides the capability to initiate a rebuild of one or more structures in the coupling facility to which connectivity has been lost, using the CFRM policy and optionally, the SFM policy.

For each structure defined in the CFRM policy, you can specify a value that MVS will use to determine whether to initiate a structure rebuild when a loss of connectivity occurs. When an active SFM policy is in effect in the sysplex and a coupling facility loss of connectivity occurs, MVS uses the system weights defined in the SFM policy in conjunction with the REBUILDPERCENT value defined in the CFRM policy to determine whether to initiate a structure rebuild. Using the SFM policy system weights, MVS calculates the weighted percentage of lost connectivity and compares the result with the user-specified REBUILDPERCENT value. If the calculated percentage value is greater than or equal to that specified by REBUILDPERCENT, MVS will initiate the structure rebuild.

With OS/390 R3 and higher, with OW30814 installed, MVS will initiate a rebuild when loss of connectivity to a structure occurs and there is no active SFM policy in the sysplex.

See “Specifying a Rebuild Threshold” on page 47 for information about MVS-initiated structure rebuild.

Planning PR/SM Reconfigurations

After a system running in a PR/SM partition is removed from the sysplex, SFM allows a remaining system in the sysplex to reconfigure processor storage for use by the remaining systems.

You specify PR/SM reconfiguration statements on the RECONFIG statement of the SFM policy. PR/SM reconfiguration definitions include the name of the failing system (FAILSYS), the name of the system that is to perform the reconfiguration actions (ACTSYS), and whether PR/SM is to deactivate a specific system or all logical partitions in the addressing range of the acting system (TARGETSYS). You can also specify whether the system performing the reconfiguration is to take over the target system's storage and/or expanded storage. To carry out a PR/SM reconfiguration, ACTSYS and TARGETSYS must be on the same processor. FAILSYS can be the same system as TARGETSYS, or it can be another system, on the same processor or on another processor.

When the TARGETSYS is deactivated, its CPU resources are added to the pool of shared CPU resources, thereby increasing the CPU cycles available to all shared logical partitions. The distribution of the additional CPU cycles is proportional to the weights of the active shared logical partitions. When the TARGETSYS is reactivated, its CPU resources are removed from the shared CPU resource pool.

Figure 33 shows a three-system sysplex, including two MVS images (SYSA and SYSB) running in logical partitions on a single processor and one MVS (SYSC) on a single processor. Because an SFM policy is active, XCFPOLxx specifications are not active. SYSB can be configured to take over the resources associated with the logical partition of SYSA if either SYSA or SYSC is removed from the sysplex. This would give SYSB the extra resources needed to handle the work of either SYSA or SYSC.

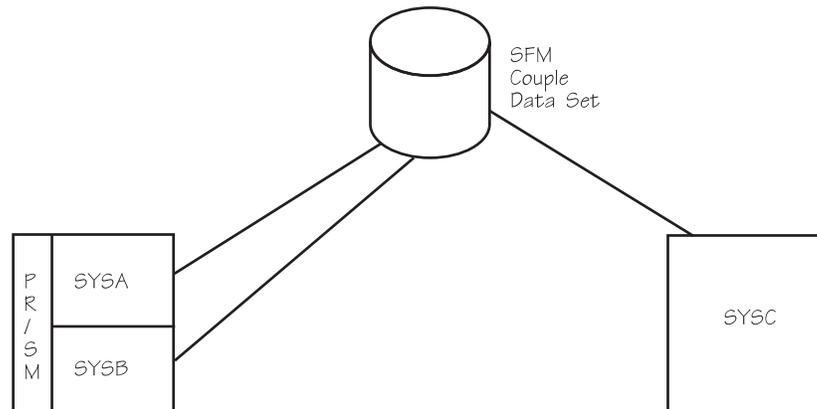


Figure 33. Three-System Sysplex on Two Processors with Active SFM Policy

Note: PR/SM reconfiguration actions can take place only if XCF can do a system reset of the MVS system and its LPAR and deactivate the LPAR.

For information about the hardware PR/SM feature and the assignment of LPARs, central processors (CPs), and storage elements, see *PR/SM Planning Guide*.

Setting Up an SFM Policy

To implement an SFM policy, you need to:

- Format an SFM couple data set and ensure that it is available to all systems in the sysplex
- Define the SFM policy
- Start the SFM policy in the SFM couple data set.

Information on formatting and defining a couple data set is included in “Considerations for Policy Couple Data Sets” on page 34.

The following topics describe how to define the SFM policy in the couple data set and how to start the policy.

Defining SFM Policies

The administrative data utility, IXCMIAPU, described in “Appendix B. Administrative Data Utility” on page 263 allows you to associate the definitions with a policy name and to place the policy in a pre-formatted SFM couple data set.

Starting an SFM Policy

To start an SFM policy (POLICY1, for example) that is defined in the SFM couple data set, issue:

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

Planning Availability and Recovery

Updating an SFM Policy

For information on updating the SFM policy dynamically, see “Updating Policies in a Sysplex” on page 35.

Controlling Availability and Recovery through XCFPOLxx (PR/SM Only)

The SFM policy includes all the function available through XCFPOLxx. If a system is connected to a couple data set with a started SFM policy, all XCFPOLxx specifications on that system are deactivated, regardless of whether the SFM policy is active in the sysplex.

Because the SFM policy provides function beyond that provided by the XCF PR/SM policy, it is generally recommended that you use the SFM policy to manage failures in your sysplex. However, in cases where you cannot activate an SFM policy (for example, if the sysplex includes an SP Version 4 system), activating an XCF PR/SM policy can be useful.

For a description of the functions provided through XCFPOLxx, see “Resetting or Deactivating a Failed System on PR/SM” on page 144 and “Planning PR/SM Reconfigurations” on page 146.

Implementing XCFPOLxx

You specify XCFPOLxx values in parmlib, activate (or deactivate) the policy with the SETXCF PRSMPOLICY command, and display the name of the active policy with the DISPLAY XCF,PRSMPOLICY command.

To change an existing XCF PR/SM policy or activate a new policy, issue:

```
SETXCF PRSMPOLICY,ACTIVATE=memname
```

where memname is the parmlib member that contains the new XCF PR/SM policy.

To deactivate the XCF PR/SM policy, issue:

```
SETXCF PRSMPOLICY,DEACTIVATE
```

When command processing completes, message IXC322I or IXC321I is issued.

The following shows the corresponding statements and keywords, in the SFM policy and the XCF PR/SM policy, for initiating PR/SM reconfigurations. For example, the FAILSYS keyword in the SFM policy corresponds to SYSGONE in XCFPOLxx.

SFM Policy:	RECONFIG	FAILSYS	ACTSYS	TARGETSYS
XCFPOLxx:	No statement	SYSGONE	SYSTEM	DEACTIVATE

Examples

The following examples illustrate the use of the SFM policy to handle failure conditions in a sysplex. Where possible, corresponding XCFPOLxx specifications are also provided.

Example 1: Failure Management in a PR/SM Environment — One Processor

Figure 34 shows two MVS systems, PRIMARY and BACKUP1. The two systems are the only systems in LPARs on the same PR/SM processor and in the same sysplex. In the example, the group members of the multisystem applications on PRIMARY are using more processor storage resources than their group members on BACKUP1.

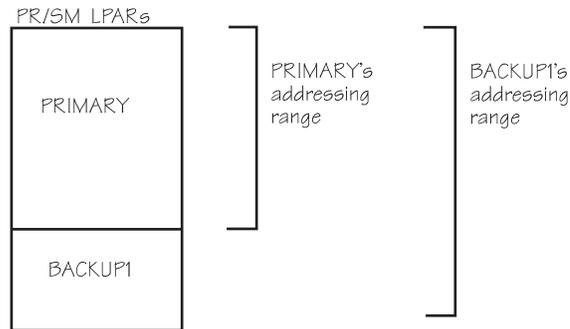


Figure 34. Failure Management in a PR/SM Environment — One Processor

Objective

If PRIMARY fails, enable an application's group members on BACKUP1 to take over the workload of their group members on PRIMARY without operator intervention. This requires more processor storage resources than is available to BACKUP1, so MVS must obtain more storage.

Note that you must define the processor storage to the LPARs in such a way that the BACKUP1 LPAR can obtain the storage freed from PRIMARY. For information on how to do this, see *PR/SM Planning Guide*.

SFM Policy Specifications

```
DEFINE POLICY NAME(POL1) CONNFAIL(NO)

SYSTEM NAME(PRIMARY)
  RESETTIME(5)

RECONFIG
  FAILSYS(PRIMARY) ACTSYS(BACKUP1) TARGETSYS(ALL)
  STORE(YES) ESTORE(YES)
```

Scenario

If PRIMARY stops updating its status, the following events occur, based on the policy:

When the failure detection interval specified on the INTERVAL keyword in PRIMARY's COUPLExx parmlib member elapses, MVS on BACKUP1 detects a status update missing condition for PRIMARY.

BACKUP1 waits 5 seconds. If the status update missing condition on PRIMARY is not resolved, BACKUP1 performs a system reset of PRIMARY.

BACKUP1 notifies group members on remaining systems in the sysplex (only BACKUP1 in the example) that PRIMARY has failed and no longer has access to shared resources.

Planning Availability and Recovery

Group members of the multisystem applications on BACKUP1 can now take over the workload of their group members on PRIMARY.

BACKUP1 deactivates all other LPARs in its addressing range (only PRIMARY in this example) and then acquires both the central and expanded storage assigned to those LPARs (only PRIMARY in this example).

BACKUP1 brings all processor storage in its addressing range online. The storage previously used by PRIMARY is now available to BACKUP1.

Also, the shared central processors used by PRIMARY are available to BACKUP1.

As a result of the SFM policy, there is enough processor storage available on BACKUP1 for group members of the multisystem applications on BACKUP1 to take over the workload of their group members on PRIMARY.

Note that CONNFAL(NO) is specified on the policy statement because a coupling facility is not available.

Corresponding XCFPOLxx Specifications

You can use the following XCFPOLxx specifications to achieve the same function:

```
NOSTATUS(PRIMARY) RESETTIME(5)
```

```
SYSGONE(PRIMARY) SYSTEM(BACKUP1) DEACTIVATE(ALL)  
STORE(YES) ESTORE(YES)
```

Example 2: Failure Management in a PR/SM Environment — Two Processors and a Coupling Facility

Figure 35 shows three MVS systems called PRIMARY, TEST, and BACKUP2. PRIMARY is on one processor and TEST and BACKUP2 are in LPARs on a PR/SM processor. PRIMARY and BACKUP2 are in the same sysplex and are connected through the same coupling facility. TEST is not in the sysplex. In the example, the group members of the multisystem applications on PRIMARY need more processor storage resources than their group members on BACKUP2.

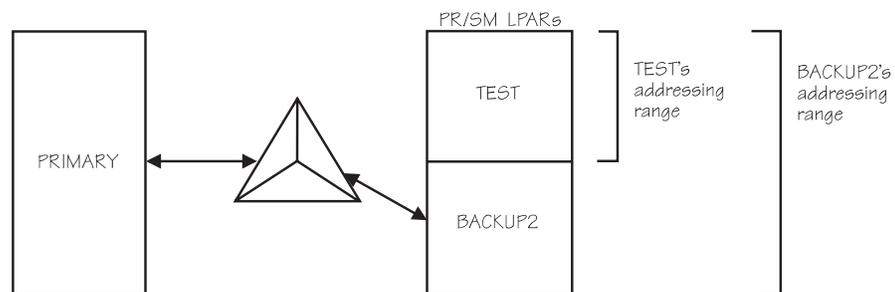


Figure 35. Failure Management in a PR/SM Environment — Two Processors and a Coupling Facility

Objective

If PRIMARY fails, enable an application's group members on BACKUP2 to take over the workload of their group members on PRIMARY. This requires more

Planning Availability and Recovery

processor storage than is available to BACKUP2, so MVS obtains more storage by taking TEST down and configuring its storage online to BACKUP2.

Note that you must define the processor storage to the LPARs in such a way that the BACKUP2 LPAR can obtain the storage freed from TEST. For information on how to do this, see *PR/SM Planning Guide*.

SFM Specifications

```
DEFINE POLICY NAME(POL2)

SYSTEM NAME(PRIMARY) ISOLATETIME(5)

RECONFIG
  FAILSYS(PRIMARY) ACTSYS(BACKUP2) TARGETSYS(ALL)
  STORE(YES) ESTORE(YES)
```

Scenario

If PRIMARY stops updating its status, the following events occur, based on the SFM policy.

BACKUP2 isolates PRIMARY and removes it from the sysplex. Because PRIMARY and BACKUP2 are connected through the same coupling facility, this can be done without operator intervention.

BACKUP2 then deactivates all other LPARs in its addressing range, which includes TEST.

BACKUP2 brings online all processor storage in its addressing range. The storage previously used by TEST is now available to BACKUP2.

Also, the shared central processors used by TEST are available to BACKUP2.

Because BACKUP2 has sufficient processor storage, group members of the multisystem applications on BACKUP2 can take over the workload of their group members on PRIMARY.

Corresponding XCFPOLxx Specifications

You can use the following XCFPOLxx specifications to achieve the same function:

```
SYSGONE(PRIMARY) SYSTEM(BACKUP2) DEACTIVATE(ALL)
STORE(YES) ESTORE(YES)
```

Using the XCF PR/SM policy, the operator must respond to a prompt to remove PRIMARY before reconfiguration can begin.

Example 3: Failure Management in a PR/SM Environment — One Processor

Figure 36 on page 152 shows two MVS systems, SYSA and SYSB. Both systems are in LPARs on the same PR/SM processor and in the same sysplex. The group members of the multisystem applications on SYSA share resources with their group members on SYSB. Group members on either system can take over the workload of the other system without additional processor storage

Planning Availability and Recovery



Figure 36. Failure Management in a PR/SM Environment — One Processor

Objective

If either system fails, without operator intervention, system reset the failing system and notify the application's group members on the other system of the failure.

SFM Specifications

```
SYSTEM NAME(SYSA)
  RESETTIME(20)
SYSTEM NAME(SYSB)
  RESETTIME(10)
```

Scenario

If SYSA stops updating its status, the following events occur, based on the policy:

When the failure detection interval specified on the INTERVAL keyword in SYSA's COUPLExx parmlib member elapses, SYSB detects a status update missing condition for SYSA.

SYSB waits 20 seconds. If the status update missing condition on SYSA is not resolved, SYSB performs a system reset of SYSA.

Then, SYSB notifies the group members of the multisystem applications on SYSB that SYSA has failed. Thus, group members on SYSB know that their group members on SYSA are no longer using shared resources and that they must take over their workload.

Corresponding XCFPOLxx Specifications

You can use the following XCFPOLxx specifications to achieve the same function:

```
NOSTATUS(SYSA) RESETTIME(20)
NOSTATUS(SYSB) RESETTIME(10)
```

Example 4: Failure Management with SFM Policy and a Coupling Facility

Figure 37 on page 153 shows five MVS systems, SYSA, SYSB, SYSC, SYSD, and SYSE in the same sysplex. All systems have connectivity to the same coupling facility. SYSD and SYSE are running in logical partitions on the same processor.

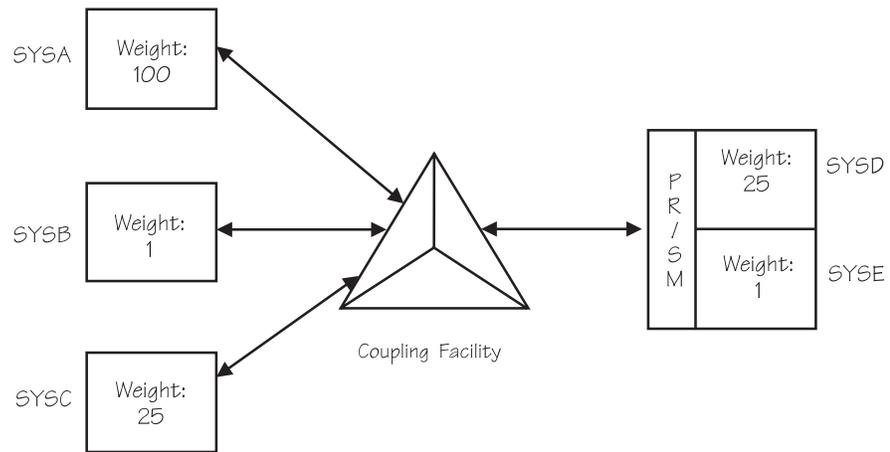


Figure 37. Failure Management with SFM Policy and a Coupling Facility

Objective

This policy defines actions to be taken for the following conditions:

- System failures
- Signaling connectivity failures
- PR/SM reconfiguration actions when a system is removed from the sysplex.

SFM Policy Specifications:

```
DEFINE POLICY NAME(POL4) CONNFAIL(YES)

SYSTEM NAME(*) /* Policy default statement */
ISOLATETIME(0)

SYSTEM NAME(SYSA)
PROMPT
WEIGHT(100)

SYSTEM NAME(SYSC)
WEIGHT(25)

SYSTEM NAME(SYSD)
WEIGHT(25)

RECONFIG
FAILSYS(SYSE)
ACTSYS(SYSD)
TARGETSYS(SYSE)
STORE(YES) ESTORE(YES)
```

Scenario

- Connectivity failure

The policy specifies one heavily weighted system, SYSA. The policy also specifies two lesser weighted systems, SYSC and SYSD. Two systems, SYSB and SYSE, will use the system default weight of 1, since they do not have a system statement and weight is not specified in the policy default statement. (For

Planning Availability and Recovery

a description of how SFM uses weights to make isolation decisions, see “Handling Signaling Connectivity Failures” on page 144.)

- System failure
If SYSA stops updating its status, the operator will be prompted, and no automated action will take place. Because SYSB and SYSE do not have a system statement and the system statements for SYSC and SYSD do not specify an action for a status update missing condition, if SYSB, SYSC, SYSD, or SYSE enters a status update missing condition, the policy default action will be used, and a system isolation will be attempted with no delay.
- Reconfiguration
If SYSE is removed from the sysplex for any reason, and SYSD is active, then SYSD will acquire both the central and expanded storage assigned to SYSE.

Controlling Job Availability and Recovery through Automatic Restart Management

Automatic restart management is an MVS recovery function that can improve the availability of specific batch jobs or started tasks. When a job or task fails, or the system on which it is running fails, automatic restart management can restart the job or task without operator intervention.

The goals of SFM and automatic restart management are complementary. While SFM keeps the sysplex running, automatic restart management keeps specific work in the sysplex running. If a job or task fails, automatic restart management restarts it on the same system it was running on at the time of the failure. If a system fails, automatic restart management restarts the work on other systems in the sysplex; this is called a **cross-system restart**.

A program cannot use both automatic restart management and checkpoint/restart. If a program using checkpoint/restart tries to register with automatic restart management, the request is rejected.

Requirements for Participating in Automatic Restart Management

To participate in automatic restart management:

- A system must be running OS/390 or MVS/ESA SP 5.2.
Full automatic restart management function is available for systems running OS/390, JES2 SP 5.2.0, or JES3 SP 5.2.1 (or subsequent levels).
Limited automatic restart management function is available for systems running JES2 SP 4.1 or JES3 SP 4.2.1 (or subsequent levels):
 - Single system restarts are possible for started tasks running under the master subsystem.
 - Restart groups running on an OS/390 system or a JES2 SP 5.2.0 or JES3 SP 5.2.1 (or subsequent level) system, in which all elements are started tasks running under the master subsystem, can be restarted on a JES2 SP 4.1 or JES3 SP 4.2.1 (or subsequent level) system.
- A system must be connected to an ARM couple data set with an active automatic restart management policy.
- A batch job or started task must register with automatic restart management. (That is, it must issue the IXCARM macro. See *OS/390 MVS Programming: Sysplex Services Guide* for more information on the IXCARM macro.)

Note: A batch job or started task registered with automatic restart management can only be restarted within the same JES XCF group. (That is, JES2 MAS or JES3 complex.)

- The XCF address space must have adequate authorization through the OS/390 Security Server (RACF) or an equivalent security product.

In order to perform restarts, ARM must issue operator commands from the XCF address space (XCFAS). You must be certain that the XCFAS has enough authority to issue any commands required to restart a failed element. If you are using ARM, IBM recommends that your RACF support personnel define the XCFAS in the RACF STARTED class or the started procedures table with the trusted attribute. See *OS/390 SecureWay Security Server RACF Security Administrator's Guide* for information about using the STARTED class. See *OS/390 SecureWay Security Server RACF System Programmer's Guide* for information about using the started procedures table.

Using Automatic Restart Management

IBM provides policy defaults for automatic restart management, as shown in Table 6 on page 156. You can use these defaults or override them for your installation.

Automatic restarts do not need to be enabled at all times. For example, you might not want automatic restart management enabled for restarts unless certain jobs are running, or during off-shift hours. However, even while automatic restarts are disabled, elements can register with automatic restart management as long as the system is connected to an ARM couple data set.

To implement the policy, you need to:

- Format an ARM couple data set and ensure that it is available to all systems in the sysplex that are to participate in automatic restart management. (That is, all systems running batch jobs or started tasks that might be restarted by automatic restart management and all systems on which batch jobs or started tasks might be restarted if the system on which they are running fails.)

Information on formatting and defining a couple data set is included in “Considerations for Policy Couple Data Sets” on page 34.

- Start the automatic restart management policy in the ARM couple data set. Once the policy is started, all systems connected to the ARM couple data set will use the same active policy.

To start automatic restart management with the policy defaults, issue:

```
SETXCF START,POLICY,TYPE=ARM
```

To start your own automatic restart management policy (mypol, for example), issue:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME=myspo1
```

Use the SETXCF STOP command to disable automatic restarts.

- For information on updating the automatic restart management policy dynamically, see “Updating Policies in a Sysplex” on page 35.

Customizing the Automatic Restart Management Policy

You can override one or more of the policy defaults through the administrative data utility (IXCMIAPU), described in “Appendix B. Administrative Data Utility” on page 263, through the IXCARM macro, or through the exits.

Planning Availability and Recovery

For example, your installation might want to define a policy to group together elements (such as CICS and DB2) that are related or perform interdependent processing and, therefore, must run on the same system. These groups are referred to as **restart groups**.

To determine whether you need to override any policy defaults, you must decide how you want to manage automatic restarts and whether the defaults satisfy those goals. Use the following table and the considerations that follow. The table provides a list of the automatic restart management parameters of IXCMIAPU, their default values (which you can override in the policy), and, in the column to the far right, the IXCARM macro parameters that can override some of the defaults.

Table 6. Automatic Restart Management Parameters for IXCMIAPU and IXCARM Parameter Overrides

IXCMIAPU Parameter	Optional/Required	Default	IXCARM Parameter that Overrides Default
RESTART_ORDER	Optional	LEVEL(2),SYSLVL2	None
LEVEL	Optional	(2)	None
ELEMENT_NAME	Optional	None	None
ELEMENT_TYPE	Optional	(SYSLVL2)	ELEMTYPE
RESTART_GROUP	Required	DEFAULT	RESTART_GROUP
TARGET_SYSTEM	Optional	(*)	TARGET_SYSTEM
FREE_CSA	Optional	(0,0)	FREE_CSA
RESTART_PACING	Optional	(0)	RESTART_PACING
ELEMENT	Required	None	None
RESTART_ATTEMPTS	Optional	(3,300)	None
RESTART_TIMEOUT	Optional	(300)	None
READY_TIMEOUT	Optional	(300)	None
TERMTYPE	Optional	(ALLTERM)	TERMTYPE
RESTART_METHOD	Optional	(BOTH,PERSIST)	STARTTXT

Considerations

The following considerations will help you determine the policy information you need for each batch job or started task that will register as an element of automatic restart management. The policy parameters you would use to override the defaults are shown in parentheses. For more information on these parameters, see “Automatic Restart Management Parameters for Administrative Data Utility” on page 267.

1. Determine which batch jobs and started tasks will be using automatic restart management for recovery purposes. For the IBM products that use automatic restart management, read their documentation for any policy considerations.
2. Determine if any of these elements is interdependent — that is, need to run on the same system. (RESTART_GROUP).

Note that any elements that are not explicitly assigned to a restart group become part of the restart group named DEFAULT. Thus, if these elements are restarted, they are restarted on the same system.

3. Determine if there is an order in which MVS should restart these elements — if any elements in the restart group are dependent upon other elements being restarted and ready first (RESTART_ORDER).
4. Determine if the elements in a restart group need to be restarted at specific intervals (RESTART_PACING).

Planning Availability and Recovery

5. Determine if the element should be restarted when only the element fails, or when either the element or the system fails (TERMTYPE).
6. Determine whether specific JCL or command text is required to restart an element (RESTART_METHOD).
7. Determine if a minimum amount of CSA/ECSA is needed on the system where the elements in a restart group are to be restarted (FREE_CSA).
8. Determine if you want the elements in the restart group to be restarted on a specific system (TARGET_SYSTEM).
Consider requirements like:
 - Which systems have access to the ARM couple data sets.
 - Which systems run within the same JES XCF group.
 - The workloads of the systems in the sysplex, and how they might be affected if these jobs were restarted.
 - CPU capacity needed.
 - DASD requirements.
 - Which systems have the class of initiator required by batch jobs that might need to be restarted on another system.
9. Determine if an element should be restarted and, if so, how many times it should be restarted within a given interval (RESTART_ATTEMPTS). If an element should not be restarted, set RESTART_ATTEMPTS to 0.
10. Determine how long MVS should wait for an element to re-register once it has been restarted (RESTART_TIMEOUT).
11. Determine how long MVS should wait for an element to indicate it is ready to work after it has been restarted (READY_TIMEOUT).

Using Exits to Customize Automatic Restart Management Processing

Several exits can influence how an element will be restarted:

- To coordinate restarts with other automation packages, use the element restart exit (IXC_ELEM_RESTART) described in *OS/390 MVS Installation Exits*.
- To prepare a system that will be receiving additional workload from a failing system, use the workload restart exit (IXC_WORK_RESTART), described in *OS/390 MVS Installation Exits*.
- To prepare a system for the restart of an element or to prevent the restart of an element, use the event exit, described in *OS/390 MVS Programming: Sysplex Services Guide*.

Handling Concurrent System and Couple Data Set Failures

It is possible to have concurrent loss of couple data sets, systems, and other sysplex resources if a disaster, such as a power failure, occurs. Concurrent failure of one or more couple data sets and one or more systems requires a special recovery procedure. In the event that a concurrent failure occurs, XCF issues message IXC256A. If the systems identified in message IXC256A are in an unrecoverable state (such as a disabled wait state or a loss of power), those systems must be removed from the sysplex to allow couple data set recovery to proceed.

Remove each of the systems identified in message IXC256A as follows:

1. Perform a system reset on each system identified in IXC256A.
2. From an active system in the sysplex, issue the command

Planning Availability and Recovery

| VARY XCF,sysname,OFFLINE

| where sysname is the name of the system identified in IXC256A.

| 3. From the same active system used in the preceding step, issue the command

| VARY XCF,sysname,OFFLINE,FORCE

| where sysname is the name of the sysname used in the preceding step.

Chapter 8. Adding MVS Systems to a Sysplex

This chapter describes how various sysplex configurations are set up. It includes the purpose and required parameters for each configuration and diagrams for many of the configurations.

The following configurations are covered:

- “XCF-Local Mode Configuration”
- “Single-System Sysplex Configuration” on page 160
- “Multisystem Sysplex Configuration without an Existing Global Resource Serialization Complex” on page 161
- “Multisystem Sysplex Configuration with an Existing Global Resource Serialization Complex” on page 163
- “Multisystem Sysplex Configuration on One Processor under VM” on page 168
- “Multisystem Sysplex Configuration on PR/SM” on page 168
- “Multisystem Sysplex Configuration with a Coupling Facility” on page 169
- “Multisystem Sysplex Configuration of Greater than Eight Systems” on page 170.

Because there are times when you need to remove a system from a sysplex, you can find information on how to do that in:

- “Removing a System from the Sysplex” on page 171.

XCF-Local Mode Configuration

Purpose

Use XCF-local mode when you:

- Want to run a single system without sysplex services
- Want to IPL the system to make changes, for example to change the names of the primary and alternate couple data sets specified in the COUPLExx parmlib member.

Considerations

Multisystem applications can join XCF groups on an XCF-local system. Members of an XCF group can communicate with each other on the XCF-local system (via local message traffic). Note that on the IXCJOIN macro, LASTING=YES is not supported in XCF-local mode, which means that permanent status recording is not available to group members. Functions, such as WLM, that require a couple data set are not available because the sysplex couple data set required to support such couple data sets is not available.

Required System Parameters

Use the following system parameters to initialize each MVS system:

- COUPLE=00 (the default) specifies the default COUPLE00 parmlib member that indicates XCF-local mode for the initializing system. A couple data set is not specified for XCF-local mode.
- PLEXCFG=XCFLOCAL indicates that the system is to be in XCF-local mode.

Figure 38 shows the multisystem environment with the MVS systems in XCF-local mode.

Adding MVS Systems to a Sysplex

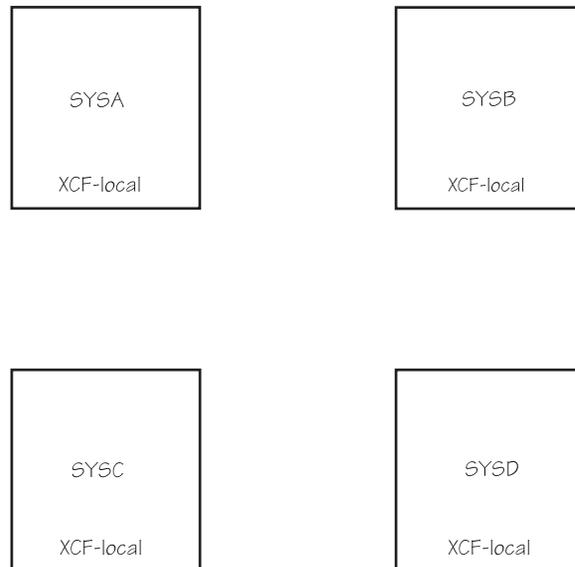


Figure 38. XCF-Local Mode Configuration

Single-System Sysplex Configuration

In single-system sysplex mode, you need a formatted primary sysplex couple data set on each system before you IPL the system.

Purpose

Use single-system sysplex mode when you want XCF services on the system but do not need XCF or XES coupling services between systems. For example, you might want to implement a WLM policy on a single system, which requires a sysplex couple data set and a WLM couple data set, or test multisystem applications.

Considerations

Multisystem applications can define XCF groups on the system and members of the group can communicate with their group members on the same system, but they cannot use signaling services to communicate with members on other systems. Note that, because a sysplex couple data set is in use, permanent status recording (LASTING=YES on the IXCJOIN macro) is available for group members on the single-system sysplex.

Consider this a temporary configuration until a multisystem sysplex is established.

Required System Parameters

Use the following parameters to IPL the system:

- COUPLE=xx specifies the COUPLExx parmlib member that describes the single-system sysplex for the system. COUPLExx must specify at least a sysplex name and a primary sysplex couple data set name. The sysplex couple data set must be unique for that system. COUPLExx must also specify the couple data set for any function, such as WLM, that you plan to use.

Adding MVS Systems to a Sysplex

- PLEXCFG=MONOPLEX indicates that the system is to be initialized as a single-system sysplex. Note that the system cannot migrate to a multisystem sysplex unless you specify PLEXCFG=MULTISYSTEM.

Figure 39 shows a single-system sysplex (SYSA) connected to a sysplex couple data set and three independent systems. To migrate SYSB, SYSC, and SYSD into the sysplex, you must specify PLEXCFG=MULTISYSTEM for each system.

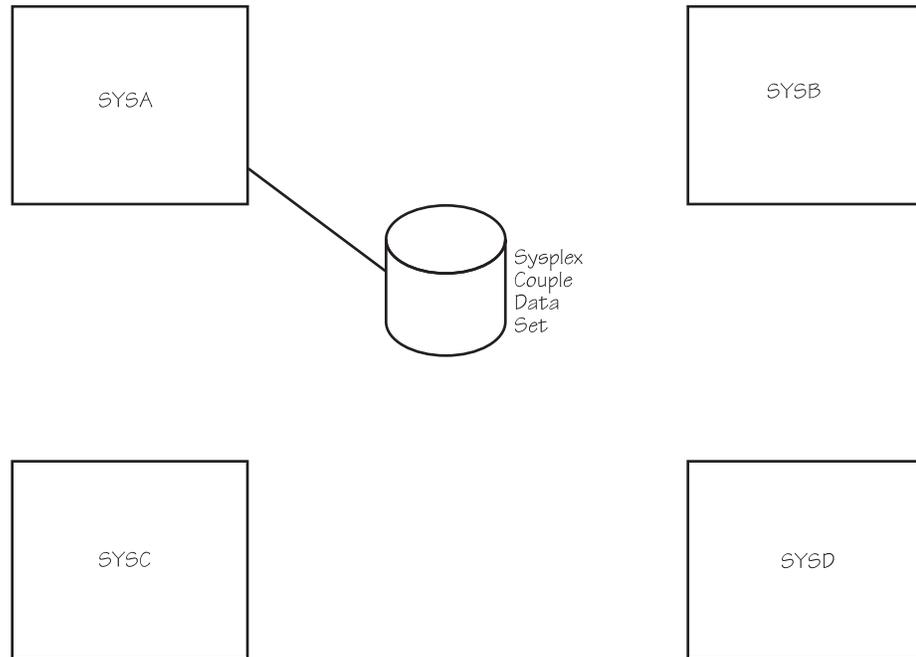


Figure 39. Configuration with a Single-System Sysplex and Three Independent Systems

Multisystem Sysplex Configuration without an Existing Global Resource Serialization Complex

A sysplex requires a global resource serialization complex. The scope of the complex must encompass the scope of the sysplex. For information on creating a global resource serialization ring, see *OS/390 MVS Planning: Global Resource Serialization*.

This topic assumes that you do not have an existing global resource serialization complex and want to IPL four systems into a sysplex.

In multisystem sysplex mode, you need:

- A formatted primary sysplex couple data set shared by all systems in the sysplex.
- Signaling connectivity between all systems in the sysplex.
- The same Sysplex Timer shared by all systems in a sysplex that includes more than one CPC.

As the systems are IPLed and join the sysplex, a global resource serialization complex is created for the systems in the sysplex. Thus, the sysplex and complex match as you IPL the systems.

Adding MVS Systems to a Sysplex

Purpose

Use multisystem sysplex mode when you:

- Want XCF group services on the systems and signaling services between systems in the sysplex.
- Want to stop using the RESERVE macro to serialize global resources and start using the functions in a global resource serialization complex.

Required System Parameters

Use the following system parameters to initialize each MVS system.

- COUPLE=xx specifies the COUPLExx parmlib member that describes the sysplex environment for the initializing system. COUPLExx must specify at least the sysplex name, the primary couple data set name (which must be shared by all systems), and sufficient outbound and inbound signaling paths to obtain connectivity with all other systems in the sysplex. Defaults can be taken for other parameters.
- PLEXCFG=MULTISYSTEM indicates that the system is to initialize as part of a multisystem sysplex.
- GRS=TRYJOIN specifies that the system is to join an existing global resource serialization complex or start one.
- GRSCNF=00 (the default) specifies the GRSCNF00 parmlib member. GRSCNF00 contains global resource serialization parameters that can be used in the sysplex. You can specify GRSCNF=xx for a GRSCNFxx parmlib member if GRSCNF00 does not meet your installation's needs.
- GRSRNL=(xx,yy...) specifies the GRSRNLxx parmlib member(s) that contain the resource name lists (RNLs) that you require for the global resource serialization complex to meet your installation needs. (The GRSRNL00 parmlib member contains IBM defaults.)
- CLOCK=xx specifies the CLOCKxx parmlib member that requests the Sysplex Timer to be used. MVS requires the Sysplex Timer to synchronize the coupling services.

As each of the MVS systems is IPLed, the sysplex and global resource serialization complex match, and consist of one, two, three, and then four systems. XCF coupling services and global resource serialization are available on the systems as they complete initialization.

Figure 40 on page 163 shows the multisystem environment after the four systems are IPLed.

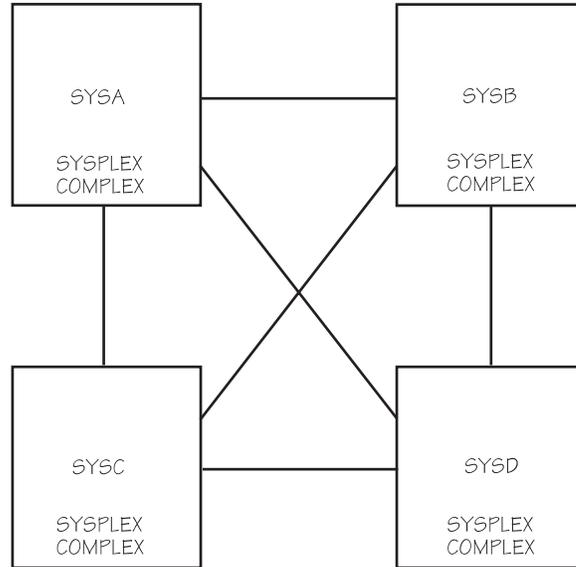


Figure 40. Multisystem Sysplex Configuration

After the four systems are IPLed, the multisystem sysplex configuration has:

- XCF coupling services available on all systems, including signaling services between all of the systems in the sysplex.
- The same primary sysplex couple data set shared by all systems in the sysplex (not shown).
- MVS component exploiters (such as global resource serialization) that exploit the XCF coupling services.
- Multisystem applications that can exploit the XCF coupling services in the sysplex.
- The same Sysplex Timer on all systems, which synchronizes time stamps for all systems (not shown).

Multisystem Sysplex Configuration with an Existing Global Resource Serialization Complex

This topic assumes that you have an existing global resource serialization complex of four systems and want to initialize them into a sysplex that matches the complex.

Until all systems are IPLed and join the sysplex, a mixed complex exists; that is, one or more of the systems in the global resource serialization complex are not part of the sysplex.

In multisystem sysplex mode, you need:

- A formatted primary sysplex couple data set shared by all systems in the sysplex.
- Signaling connectivity between all systems in the sysplex
- The same Sysplex Timer for all systems in a sysplex that includes more than one CPC.

Purpose

Use multisystem sysplex mode with an existing global resource serialization complex when you want XCF services on the initializing systems and signaling services between the systems in the sysplex.

Adding MVS Systems to a Sysplex

Required System Parameters

Use the following system parameters to initialize each MVS system:

- COUPLE=xx specifies the COUPLExx parmlib member that describes the sysplex environment for the system. COUPLExx must specify at least the sysplex name, the primary couple data set name (which must be shared by all systems in the sysplex), and sufficient outbound and inbound paths to obtain full signaling connectivity with all other systems in the sysplex. Defaults can be taken for the other parameters.
- PLEXCFG=MULTISYSTEM indicates that the system is to initialize as part of a multisystem sysplex.
- GRS=TRYJOIN specifies that the system is to join an existing global resource serialization complex or start one.
- GRSCNF=xx specifies the GRSCNFxx parmlib member for the existing global resource serialization complex. This includes the definitions of the necessary CTC links for the complex as systems are migrated into the sysplex. When the sysplex matches the complex, global resource serialization uses signaling services and no longer needs its own CTC links between systems.
- GRSRNL=(xx,yy...) specifies the GRSRNLxx parmlib member(s) that contain the resource name lists (RNLs) that you require for the complex to meet your installation needs.
- CLOCK=xx specifies the CLOCKxx parmlib member that requests the Sysplex Timer to be used. MVS requires the Sysplex Timer to synchronize coupling services in the sysplex.

The following topics describe the multisystem environment as each of the four systems are initialized in order (SYSA, SYSB, SYSC, and then SYSD).

The First System IPL

Figure 41 shows the multisystem environment after SYSA is initialized.

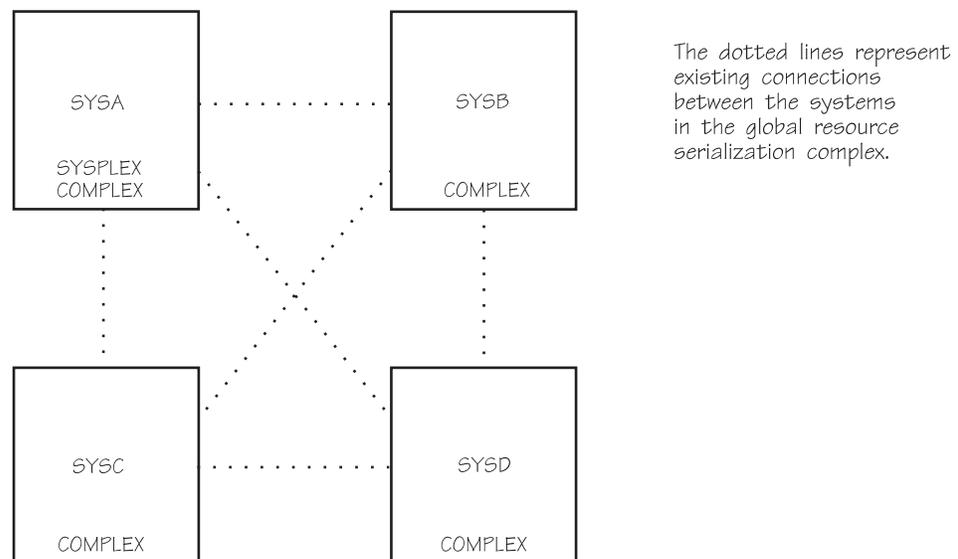


Figure 41. One System in the Multisystem Sysplex

Adding MVS Systems to a Sysplex

When SYSA is initialized and joins the sysplex and the global resource serialization complex:

- XCF coupling services are available on SYSA, but signaling services are not available between any of the systems.
- A primary sysplex couple data set is used for SYSA (not shown).
- MVS component exploiters can exploit XCF coupling services on SYSA but must use non-XCF connections to communicate with other systems.
- Multisystem applications can exploit XCF coupling services on SYSA.

Consider this mixed complex a temporary configuration during migration until the sysplex matches the complex.

The Second System IPL

Figure 42 shows the multisystem environment after SYSB is initialized.

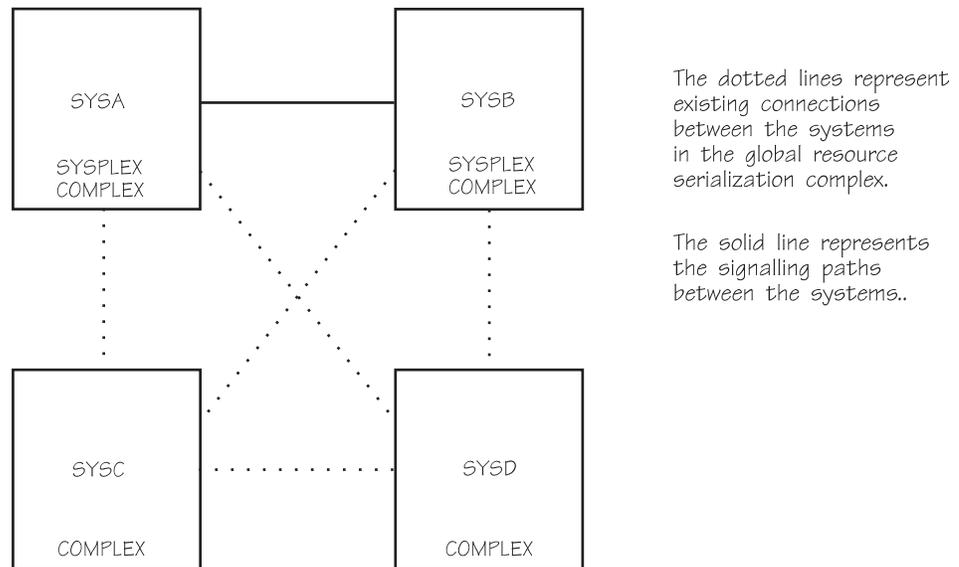


Figure 42. Two Systems in the Multisystem Sysplex

When SYSB is initialized and joins the sysplex and the global resource serialization complex:

- XCF coupling services are available on SYSA and SYSB, including signaling services between SYSA and SYSB.
- The same primary sysplex couple data set is shared by SYSA and SYSB (not shown).
- MVS component exploiters (such as global resource serialization) use signaling services between SYSA and SYSB but must use non-XCF connections to communicate with the other systems.
- Multisystem applications can exploit XCF coupling services and signaling services on SYSA and SYSB.
- SYSA and SYSB must share a Sysplex Timer if they are running on separate CPCs (not shown).

Adding MVS Systems to a Sysplex

Consider this mixed complex a temporary configuration during migration until the sysplex matches the complex.

The Third System IPL

Figure 43 shows the multisystem environment after SYSC is initialized.

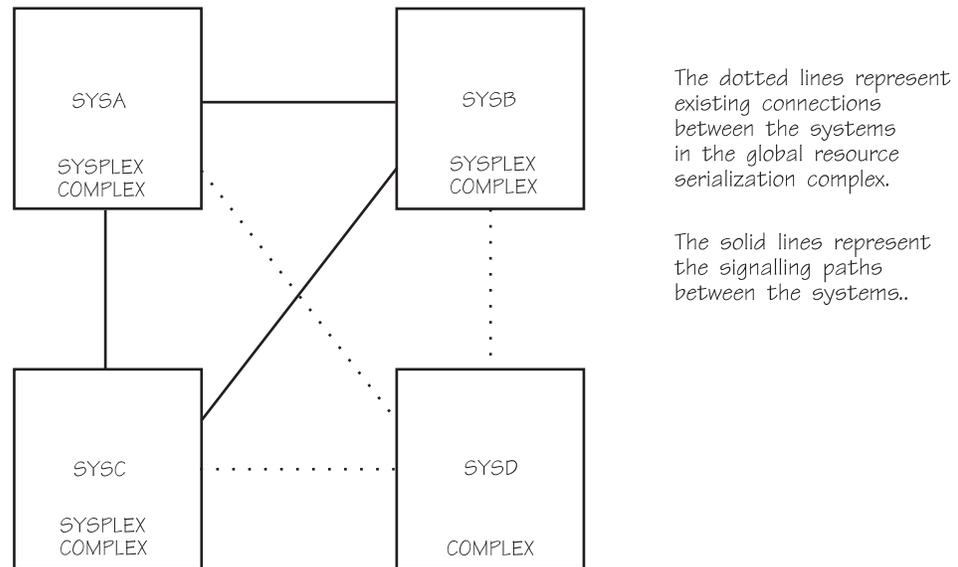


Figure 43. Three Systems in the Multisystem Sysplex

When SYSC is initialized and joins the sysplex and the global resource serialization complex:

- XCF coupling services are available on SYSA, SYSB, and SYSC, including signaling services between the three systems.
- The same primary sysplex couple data set (not shown) is shared by SYSA, SYSB, and SYSC.
- MVS component exploiters (such as global resource serialization) use signaling services between SYSA, SYSB, and SYSC but must use non-XCF connections to communicate with SYSD.
- Multisystem applications can exploit XCF coupling services and signaling services on SYSA, SYSB, and SYSC.
- SYSA, SYSB, and SYSC must share a Sysplex Timer if they are running on separate CPCs (not shown).

Consider this mixed complex a temporary configuration during migration until the sysplex matches the complex.

The Fourth System IPL

Figure 44 on page 167 shows the multisystem environment after SYSD is initialized. The sysplex now matches the global resource serialization complex.

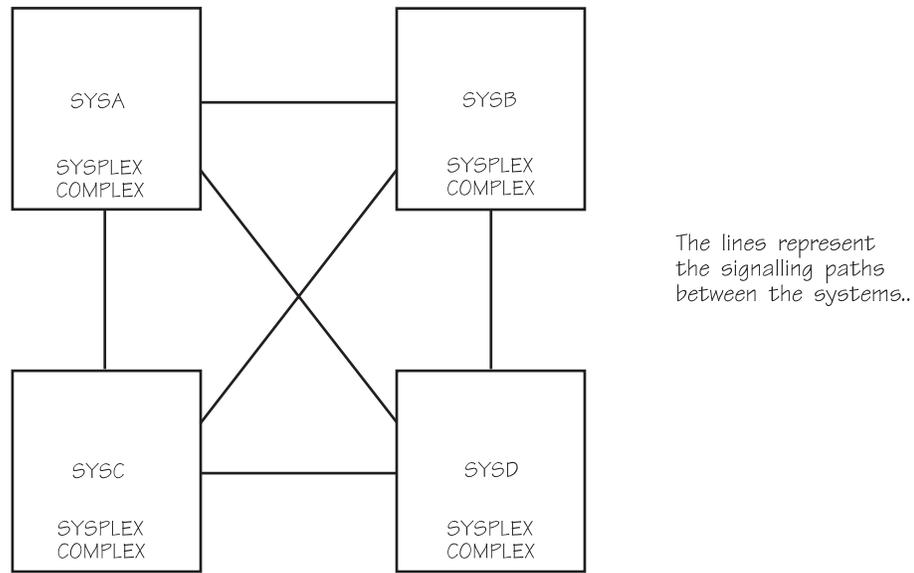


Figure 44. Four Systems in the Multisystem Sysplex

When SYSD is initialized and joins the sysplex, the sysplex matches the global resource serialization complex.

- XCF coupling services are available on all systems, including signaling services between all systems.
- The same primary sysplex couple data set is shared by all systems in the sysplex (not shown).
- MVS component exploiters (such as global resource serialization) use signaling services between all systems.
- Multisystem applications can use XCF coupling services and signaling services on all systems.
- The systems must share a Sysplex Timer if they are running on separate CPCs (not shown).

Advantages

The following are some additional advantages when the multisystem sysplex matches the global resource serialization complex.

- The CTC links assigned in the GRSCNFxx parmlib member for the global resource serialization complex are no longer needed because global resource serialization uses signaling services to communicate between systems.

See *OS/390 MVS Planning: Global Resource Serialization* for the commands that you can use to vary the global resource serialization CTC links between SYSA and SYSB offline and then dynamically reassign them to XCF.

- There is reduced operator intervention when there is a ring disruption because global resource serialization can rebuild a global resource serialization ring without operator intervention.
- The resource name lists (RNLs) can be dynamically changed with the SET GRSRNL command while maintaining serialization integrity.

Multisystem Sysplex Configuration on One Processor under VM

This topic describes a multisystem sysplex on one processor under VM. The VM system must be ESA capable.

Note that an XCF-local mode or single-system sysplex can be established under VM similar to those shown in “XCF-Local Mode Configuration” on page 159 and “Single-System Sysplex Configuration” on page 160. However, a sysplex running under VM cannot support a Sysplex Timer or a coupling facility.

Purpose

Use multisystem sysplex mode under VM for the reasons listed in “Multisystem Sysplex Configuration without an Existing Global Resource Serialization Complex” on page 161 and “Multisystem Sysplex Configuration with an Existing Global Resource Serialization Complex” on page 163. In a multisystem sysplex on one processor under VM, a simulated Sysplex Timer must be used.

Each of the MVS systems in the sysplex must be under a single host VM on one processor.

Considerations

A multisystem sysplex on one processor under VM should have APAR VM58844 installed. This APAR supports the CP SET command that you must issue for any DASD that contains any type of XCF couple data set to avoid integrity exposures.

- SET WRKALLEG (for VM/ESA Release 1.2.2 and above)
- SET SHARED (for VM/ESA releases prior to Release 1.2.2)

XCF will remove data sets that appear to have suffered an integrity exposure. If XCF removes both the primary and alternate data set for any type of XCF couple data, the function using that data set might cause one or more systems in the sysplex to enter a disabled wait state or might lose some of its function due to its inability to access its sysplex shared data.

Required System Parameters

Use the following system parameters to initialize each MVS system:

- CLOCK=xx specifies the CLOCKxx parmlib member that requests the same simulated Sysplex Timer (SIMETRID xx) on all systems.
- Other system parameters for a multisystem sysplex as described in the preceding topics.

As each of the MVS systems are IPLed, the sysplex consists of one, two, three, and then four systems. XCF coupling services and global resource serialization are available on the systems as they complete initialization.

Multisystem Sysplex Configuration on PR/SM

This topic describes a multisystem sysplex on PR/SM.

Note that an XCF-local mode or single-system sysplex can be established on PR/SM similar to those shown in “XCF-Local Mode Configuration” on page 159 and “Single-System Sysplex Configuration” on page 160.

Purpose

Use multisystem sysplex mode on PR/SM for the reasons listed in “Multisystem Sysplex Configuration without an Existing Global Resource Serialization Complex” on page 161 and “Multisystem Sysplex Configuration with an Existing

Global Resource Serialization Complex” on page 163. A multisystem sysplex on one processor with PR/SM can use a Sysplex Timer or a simulated Sysplex Timer.

Required System Parameters

Use the following system parameters to IPL each MVS system on one processor on PR/SM:

- CLOCK=xx specifies the CLOCKxx parmlib member that requests either:
 - The same simulated Sysplex Timer (SIMETRID xx) on all systems, or
 - The same Sysplex Timer (ETRMODE YES) to be used on all systems.If the sysplex spans more than one processor, ETRMODE YES is required for all systems.
- Other system parameters for a multisystem sysplex as described in the preceding topics.

As each of the MVS systems is IPLed, the sysplex consists of one, two, three, and then four systems. XCF coupling services and global resource serialization are available on the systems as they complete initialization.

See “On PR/SM - Physical Configuration of a Sysplex on LPARs” on page 317 and “Controlling Availability and Recovery through XCFPOLxx (PR/SM Only)” on page 148 for additional information about PR/SM configurations.

Multisystem Sysplex Configuration with a Coupling Facility

This topic assumes that you have an existing multisystem sysplex and want to add a coupling facility to the configuration.

In addition to the requirements for multisystem sysplex mode (see “Multisystem Sysplex Configuration with an Existing Global Resource Serialization Complex” on page 163), you need:

- A sysplex couple data set formatted with the OS/390 format utility (IXCL1DSU) (not shown).
- A formatted primary CFRM couple data set shared by systems in the sysplex that use the coupling facility. The CFRM couple data set must contain at least one CFRM administrative policy.
- Connectivity between the coupling facility and the systems in the sysplex requiring its use.

Purpose

Use a coupling facility to provide data sharing capability, the ability to isolate MVS systems through the SFM policy, or as an alternative to CTC connections for signaling.

Required System Parameters

Use the following system parameter to IPL each system in the sysplex that is to use the coupling facility:

- COUPLE=xx specifies the COUPLExx parmlib member that is updated to identify the primary CFRM couple data set.

Alternatively, you can use the SETXCF COUPLE,TYPE=CFRM,PCOUPLE command to bring the CFRM couple data set into use.

Adding MVS Systems to a Sysplex

To start a policy in the CFRM data set, issue the SETXCF START,POLICY,TYPE=CFRM command, specifying the name of a CFRM policy that resides in the CFRM couple data set.

In Figure 45, SYSA and SYSB are connected to the coupling facility. (Required connections for the sysplex couple data set, Sysplex Timer, and signaling are not shown.)

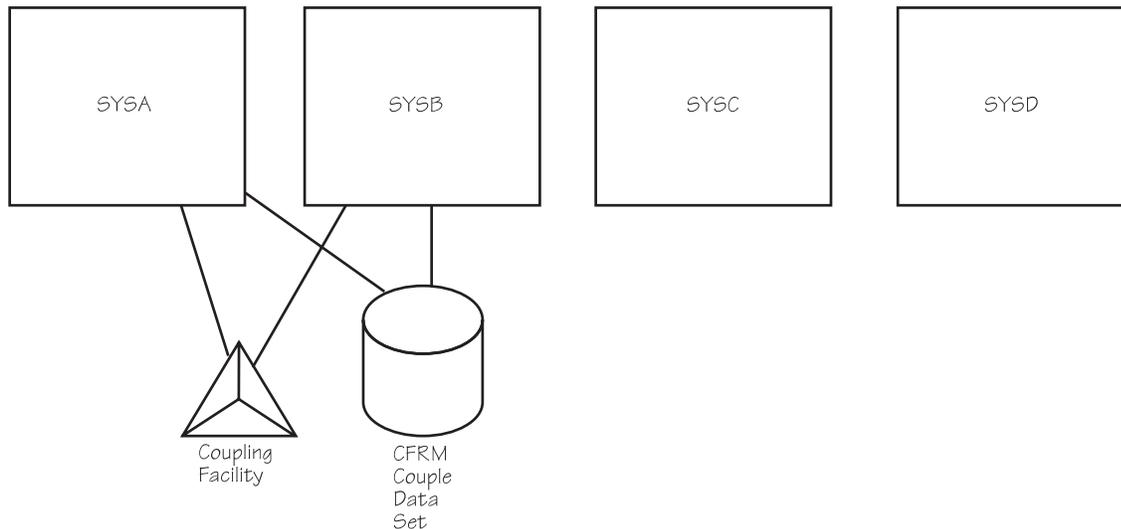


Figure 45. Multisystem Sysplex Configuration with a Coupling Facility

Multisystem Sysplex Configuration of Greater than Eight Systems

This topic assumes that you have an existing multisystem sysplex and want to upgrade it to support more than eight systems. To operate a sysplex of more than eight systems, all systems must be at the OS/390 (or the MVS/ESA SP Version 5) level. This topic assumes that you are migrating your Version 4 systems to OS/390.

To operate a sysplex of more than eight systems, the primary sysplex couple data set must be formatted with the OS/390 format utility.

Migration Scenario

Use the following migration scenario to upgrade your existing sysplex to support more than eight systems.

1. One at a time, bring down the Version 4 systems in your sysplex and upgrade them to OS/390. Then, re-IPL each system into the sysplex. The system shares the same sysplex couple data set that it used prior to the upgrade. Continue in this way until each system in the sysplex is at the OS/390 level.
2. In the meantime, use the OS/390 format utility to format a new sysplex couple data set to support more than eight systems. When all systems in the sysplex are at the OS/390 (or the MVS/ESA SP Version 5 level), introduce the new sysplex couple data set as the alternate, using the SETXCF COUPLE,ACOUPLE command. You can then use the SETXCF COUPLE,PSWITCH command to make the alternate sysplex couple data set the primary.

Adding MVS Systems to a Sysplex

At this point, the sysplex can make use of the ARM, CFRM, SFM, and WLM couple data sets.

3. New OS/390 systems can now join the sysplex, up to the limit for which the sysplex couple data set is formatted. The limit is specified in the format utility on the MAXSYSTEM keyword of the DEFINEDS statement.

Note the following restrictions:

- Once the sysplex couple data set formatted with the OS/390 format utility becomes the primary sysplex couple data set, you cannot use as an alternate sysplex couple data set one that has been formatted with the SP Version 4 format utility.
- You can no longer introduce an MVS/ESA SP Version 4 system to the sysplex.

Removing a System from the Sysplex

Removing a system from the sysplex means that:

- All XCF group members in the sysplex know that the system is being removed, so they can perform any necessary cleanup processing.
- All I/O to sysplex-shared resources is completed, to ensure data integrity.
- All signaling paths are configured to a desired state (retained or not).

The following are the basic steps for removing a system from a sysplex. For detailed instructions for the necessary steps, see “Removing an MVS System from a Sysplex” on the IBM Parallel Sysplex home page — <http://www.ibm.com/s390/psa/>.

Note: Steps 3 and 4 can be automated using the SFM policy. See “Controlling System Availability and Recovery through the SFM Policy” on page 141.

1. Use the VARY XCF command to remove a system from the sysplex. You can remove the system temporarily or permanently.

To temporarily remove a system from the sysplex, issue:

```
VARY XCF,system-name,OFFLINE,RETAIN=YES
```

With RETAIN=YES (the default), MVS on each remaining system in the sysplex retains the definition of the devices for the signaling paths that connected to the removed system. Therefore, the removed system can be re-IPLed into the sysplex or another system can be added in its place, and MVS automatically starts the signaling paths. Note that the last system removed from the sysplex, remains defined to the sysplex couple data set.

To permanently remove a system from the sysplex, issue:

```
VARY XCF,system-name,OFFLINE,RETAIN=NO
```

With RETAIN=NO, MVS on each remaining system stops the signaling paths connected to the removed system and discards their definition. Therefore, you cannot re-IPL the system into the sysplex.

2. Reply to message IXC371D, which requests confirmation of the VARY XCF command.

Message IXC101I then informs you that the system is being removed.

Adding MVS Systems to a Sysplex

3. When the target system is in a wait state (issues message IXC220W with a wait state code 0A2), issue a system reset. The reset must be done after the wait state, to ensure the integrity of I/O to sysplex shared I/O resources.
4. Reply DOWN to message IXC102A to continue removing the system from the sysplex.

After you reply, when the removal of the system is complete, message IXC105I is issued.

Multisystem Application Considerations

When you issue the VARY XCF,system-name,OFFLINE command, MVS notifies all groups, through their member group routines, that the system is to be removed from the sysplex. Multisystem applications that are not members of a group can also be notified of a system being removed from the sysplex through ENF event code 35. It is an application program's responsibility to ensure that its remaining members on the other systems in the sysplex can continue processing when a system is removed from the sysplex.

If the system is re-IPLed into the sysplex, MVS notifies all groups, through their member group routines, that the system is added to the sysplex. Multisystem applications that are not members of an XCF group can be notified of a system joining the sysplex through ENF event code 35.

Rejoining a Sysplex

There are instances when a system remains defined in the couple data set as "active" even though measures have been taken to remove the system from the sysplex. For example, the last system removed from a sysplex with a VARY command remains defined in the couple data set as an "active" system. Before that system can rejoin the sysplex, it must be determined whether the sysplex should be reinitialized (implying the cleanup of residual system information). The following applies:

- In native MVS mode, when the system attempts to rejoin the sysplex, XCF is able to determine that the system is a new instance of the "active" system that is defined in the couple data set. As such, it can reinitialize the couple data set without requiring operator intervention.
- In a VM guest environment, with OW31481 or OW32876 installed, when the VM guest system attempts to rejoin the sysplex, XCF determines whether the system is a new instance of the "active" system defined in the couple data set by verifying that:
 - Both systems are VM guests.
 - Both systems have the same system name.
 - The system status of the system identified in the couple data set has not been updated in a reasonable time frame prior to the attempt to rejoin the sysplex.

If these conditions exist, XCF will initialize the couple data set without requiring operator intervention.

When the system attempting to rejoin the sysplex has a different system name from the "active" system defined in the couple data set and the status of the "active" system has not been updated in a reasonable time frame, the operator is prompted to specify whether the new system should join or initialize the sysplex.

Chapter 9. Planning for System Logger Applications

This chapter covers planning steps for any system logger applications. It includes the following topics:

- “What is System Logger?”
- “The System Logger Configuration” on page 176
- “Finding Information for System Logger Applications” on page 184
 - “Finding Information for CICS Log Manager” on page 185
 - “Finding Information for OPERLOG Log Stream” on page 186
 - “Finding Information for Logrec Log Stream” on page 186
 - “Finding Information for APPC/MVS” on page 187
 - “Finding Information for IMS Common Queue Server Log Manager” on page 188
- “Preparing to Use System Logger Applications” on page 188
 - “Understand the Requirements for System Logger” on page 190
 - “Plan the System Logger Configuration” on page 191
 - “Determine the Size of Each Coupling Facility Structure” on page 197
 - “Develop a Naming Convention for System Logger Resources” on page 203
 - “Plan DASD Space for System Logger” on page 206
 - “Managing Log Data: How Much? For How Long?” on page 213
 - “Define Authorization to System Logger Resources” on page 215
 - “Format the LOGR Couple Data Set and Make it Available to the Sysplex” on page 216
 - “Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy” on page 217
 - “Deleting Log Streams From the LOGR Policy” on page 224
 - “Define the Coupling Facility Structures Attributes in the CFRM Policy Couple Data Set” on page 225
 - “Activate the LOGR Subsystem” on page 226
- “Deleting Log Data and Log Data Sets” on page 227
- “Upgrading a Log Stream from DASD-Only to Coupling Facility” on page 229
- “System Logger Recovery” on page 230

What is System Logger?

System logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex. A system logger application can be supplied by:

- IBM. For example, CICS log manager and the operations log stream (OPERLOG) are IBM-supplied system logger applications.
- Independent software vendors.
- Your installation.

A system logger application can write log data into a **log stream**, which is simply a collection of data. Data in a log stream spans two kinds of storage:

- **Interim storage**, where data can be accessed quickly without incurring DASD I/O.

System Logger

- **DASD log data set storage**, where data is hardened for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage for log data is in coupling facility list structures. See “Coupling Facility Log Stream”.
- In a DASD-only log stream, interim storage for log data is contained in local storage buffers on the system. **Local storage buffers** are data space areas associated with the system logger address space, IXGLOGR. See “DASD-Only Log Stream” on page 175.

Your installation can use just coupling facility log streams, just DASD-only log streams, or a combination of both types of log streams. The requirements and preparation steps for the two types of log streams are somewhat different; see “Preparing to Use System Logger Applications” on page 188.

Coupling Facility Log Stream

Figure 46 shows how a coupling facility log stream spans two levels of storage: the coupling facility for interim storage and DASD log data sets for more permanent storage. When the coupling facility space for the log stream fills, the data is offloaded to DASD log data sets. A coupling facility log stream can contain data from multiple systems, allowing a system logger application to merge data from systems across the sysplex.

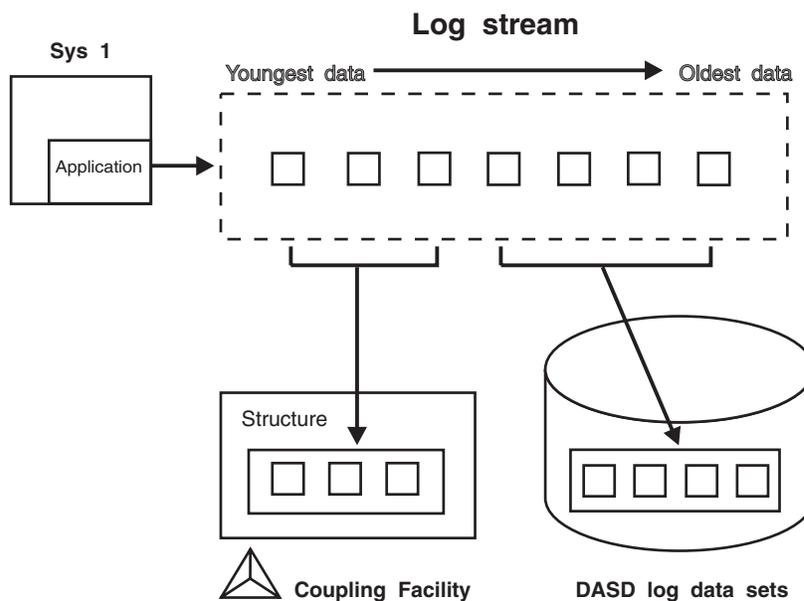


Figure 46. Log Stream Data on the Coupling Facility and DASD

When a system logger application writes a log block to a coupling facility log stream, system logger writes it first to a coupling facility list structure. System logger requires that a coupling facility list structure be associated with each log stream. When the coupling facility structure space allocated for the log stream reaches the installation-defined threshold, system logger moves (**offloads**) the log blocks from the coupling facility structure to VSAM linear DASD data sets, so that the coupling facility space for the log stream can be used to hold new log blocks. From a user's point of view, the actual location of the log data in the log stream is transparent.

DASD-Only Log Stream

Figure 47 shows a DASD-only log stream spanning two levels of storage: local storage buffers for interim storage, which is then offloaded to DASD log data sets for more permanent storage.

A DASD-only log stream has a single-system scope; only one system at a time can connect to DASD-only log stream. Multiple applications from the same system can, however, simultaneously connect to a DASD-only log stream.

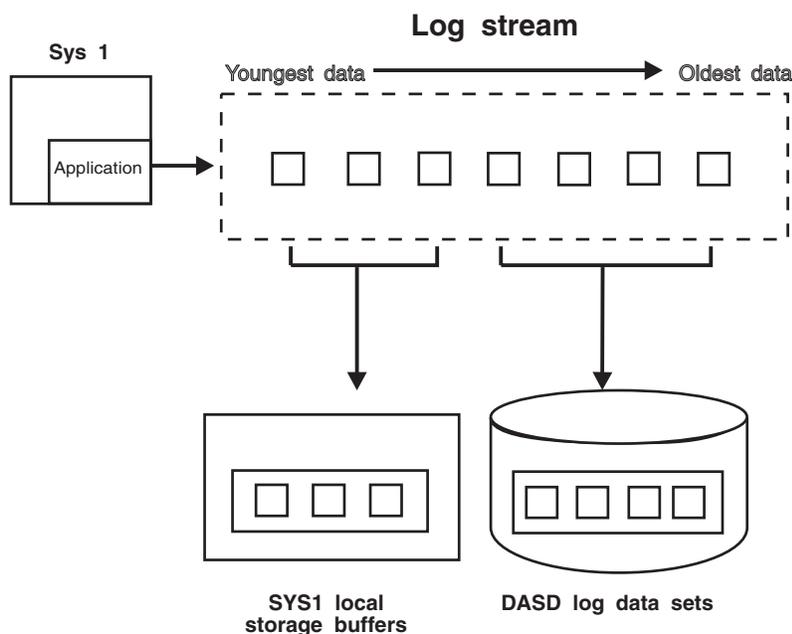


Figure 47. Log Stream Data in Local Storage Buffers and DASD Log Data Sets

When a system logger application writes a log block to a DASD-only log stream, system logger writes it first to the local storage buffers for the system and duplexes it to a DASD staging data set associated with the log stream. When the staging data set space allocated for the log stream reaches the installation-defined threshold, system logger offloads the log blocks from local storage buffers to VSAM linear DASD data sets. From a user's point of view, the actual location of the log data in the log stream is transparent.

Both a DASD-only log stream and a coupling facility log stream can have data in multiple DASD log data sets; as a log stream fills log data sets on DASD, system logger automatically allocates new ones for the log stream.

The System Logger Configuration

The system logger configuration you use depends on whether or not you use a coupling facility.

Coupling Facility Log Stream Configuration: Figure 48 shows all the parts involved when a system logger application writes to a coupling facility log stream. In this example, a system logger application runs on two systems in a sysplex. Both instances of the application write data to the same log stream, TRANSLOG. Each system contains a system logger address space. A system logger application uses system logger services to access the system logger capabilities.

When a system logger application writes data to a coupling facility log stream, system logger writes the data to a coupling facility list structure associated with the log stream. Then, when the coupling facility structure fills with data, system logger offloads the data to DASD log data sets.

You can optionally elect to have coupling facility data duplexed to DASD staging data sets for a coupling facility log stream.

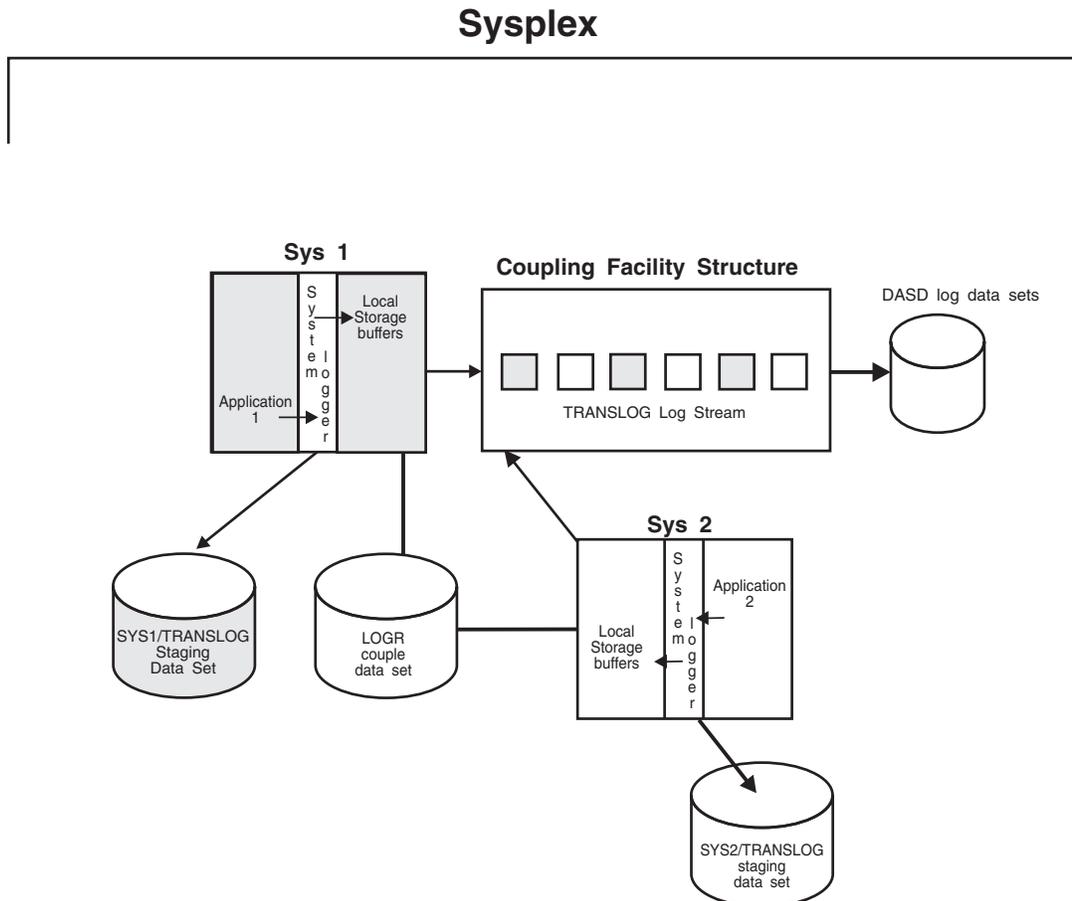


Figure 48. A Complete Coupling Facility Log Stream Configuration

DASD-Only Log Stream Configuration: Figure 49 on page 177 shows all the parts involved when a system logger application writes to a DASD-only log stream. System logger writes the data to the local storage buffers on the system, duplexing

it at the same time to the DASD staging data sets associated with the log stream. Then, when the staging data set fills with data, system logger offloads the data to DASD log data sets. Note that where duplexing to DASD staging data sets is an option for a coupling facility log stream, it is a required automatic part of a DASD-only log stream.

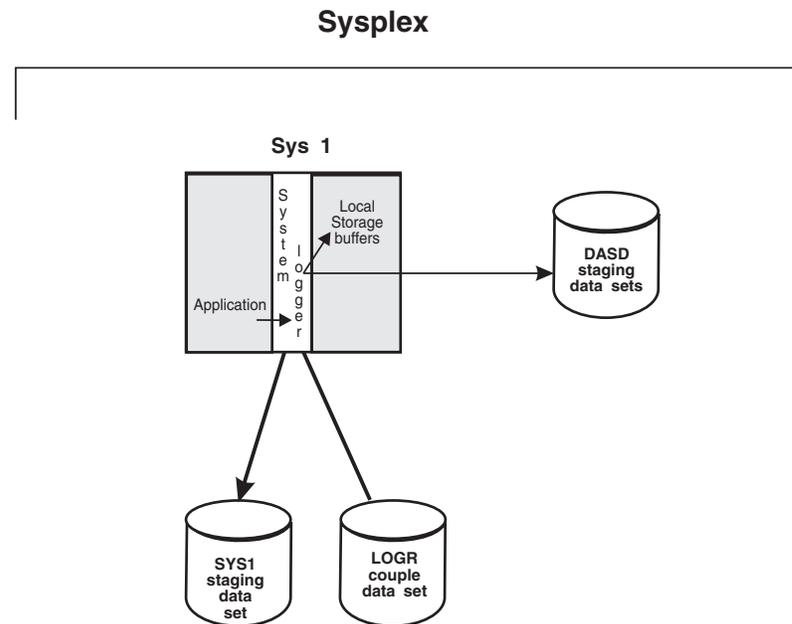


Figure 49. A DASD-Only Configuration

The System Logger Component

The system logger component resides in its own address space, IXGLOGR, on each system in a sysplex. Some of the component processing will differ, depending on whether a given log stream is a coupling facility log stream or a DASD-only log stream. The system logger component does the following:

- Provides a set of system services that allows a system logger **application** to use the system logger component. See *OS/390 MVS Programming: Assembler Services Guide* and *OS/390 MVS Programming: Assembler Services Reference* for information on writing a system logger application.
- Maintains information in the LOGR policy about the current use of log streams and if used, coupling facility list structures.
- For coupling facility log streams, system logger interacts with cross-system extended services (XES) to connect to and use the coupling facility for system logger applications.
- Obtains local storage buffer space. For a coupling facility log stream, local storage buffers can be used for duplexing log data. For a DASD-only log stream, local storage buffers are used as interim storage for log data before it is offloaded to DASD log data sets.
- Offloads data to DASD log data sets as follows:

For coupling facility log streams, system logger offloads log data from the coupling facility to DASD log data sets as the coupling facility structure space associated with the log stream reaches the installation-defined thresholds.

For DASD-only log streams, system logger offloads log data from the local storage buffers to DASD log data sets as the DASD staging data set space reaches the installation-defined thresholds.

System Logger

- Automatically allocates new DASD log data sets for log streams.
- Maintains a backup copy of (duplexes) log data that is in interim storage for recovery. Log data in interim storage is vulnerable to loss due to system or sysplex failure because it has not yet been hardened to DASD log data sets. System logger duplexes interim storage log data for both coupling facility and DASD-only log streams.
- Produces SMF record type 88 for system logger accounting on a single system. Record type 88 focuses on the usage of interim storage (coupling facility or local storage buffers) and log stream data for a system in the sysplex. Using the record can help an installation avoid the STRUCTURE FULL or STAGING DATA SET FULL exceptions, and perform other tuning and/or capacity planning analysis.

See *OS/390 MVS System Management Facilities (SMF)* for more information on record type 88 and system logger accounting. Sample program IXGRPT1 in SYS1.SAMPLIB shows an example of producing a report from SMF record type 88.

- Ensures that:
 - When the last connection from a system disconnects from the log stream, all log data written by that system to the log stream is offloaded to DASD log data sets.
System logger also deletes any staging data sets in use for a system at this time.
 - When the last connection to a coupling facility log stream in the sysplex disconnects, all coupling facility log data is offloaded to DASD log data sets and the coupling facility space is returned to XES for reallocation.
- Provides recovery support in the event of application, system, sysplex, or coupling facility structure failure for coupling facility log streams. (See “System Logger Recovery” on page 230 for information about recovery for DASD-only log streams.)

The LOGR Couple Data Set (LOGR Policy)

The system logger component manages log streams based on the policy information that installations place in the LOGR couple data set. The LOGR couple data set must:

- Be accessible by all the systems in the sysplex.
- Be formatted using the IXCL1DSU utility.
- Contain policy information, which is defined using the IXCMIAPU utility or the IXGINVNT service. (See *OS/390 MVS Programming: Assembler Services Guide* for information about the IXGINVNT service.)

You can have only one LOGR policy for the sysplex, even if you have primary and alternate LOGR couple data sets. System logger will only use the policy information contained in the primary LOGR couple data set. Because there is only one LOGR policy allowed for a sysplex, you cannot specify the DEFINE POLICY NAME parameter in the LOGR policy, nor can you issue the SETXCF START,POLICY command to activate a policy. Updates made to the LOGR policy are made in the primary LOGR couple data set. If you switch to your alternate LOGR couple data set (making it now the primary LOGR couple data set), system logger copies the LOGR policy from the old primary couple data set into the new one.

The LOGR policy includes the following:

- Log stream definitions.

- Coupling facility list structure definitions, if applicable.
- Data containing the current state of a log stream (for example, whether a log stream is currently connected to the coupling facility structure).

Formatting a LOGR couple data set and defining the LOGR policy are steps an installation must take to prepare for using a system logger application. See “Preparing to Use System Logger Applications” on page 188.

Log Data on The Coupling Facility

For a coupling facility log stream, system logger requires that a coupling facility list structure be assigned for each log stream as interim storage. When an application writes data to a coupling facility log stream, it is first written to the coupling facility list structure for that log stream. System logger manages connections to the coupling facility, but the installation must define coupling facility structures dedicated to log streams in the LOGR policy as part of the planning process for system logger applications. More than one log stream can share space in a single structure.

System logger will dynamically manage the entry-to-element ratio for log stream coupling facility structures if you are using a LOGR primary couple data set formatted at the OS/390 Release 3 level or above. The installation defines an initial average buffer size for a structure on the AVGBUFSIZE parameter in the LOGR policy using the IXGINVNT service or the IXCMIAPU utility. System logger uses this value to determine the initial entry-to-element ratio for the structure. After that, system logger will dynamically manage the entry-to-element ratio for the structure based on actual structure usage. The most current entry-to-element ratio is used on any subsequent allocations of the structure.

Log Data On DASD Log Data Sets

System logger stores log data on DASD log data sets when interim storage fills. Both a DASD-only log stream and a coupling facility log stream can have data in multiple DASD-log data sets; as a log stream fills log data sets on DASD, system logger automatically allocates new ones for the log stream.

IBM recommends that you use System Managed Storage (SMS) to manage DASD log data sets. System logger can use DFHSM or another product to migrate DASD log data sets.

Duplexing Log Data

Every time a system logger application writes a log block to a log stream, system logger automatically makes a duplicate copy of the data as insurance against data loss due to coupling facility, system failure, or system logger failure. System logger keeps a duplex copy of data in log stream interim storage only. The duplicate copy is kept until the data is offloaded from interim storage to DASD log data sets.

The way system logger duplexes log data depends on whether the log stream is a coupling facility or a DASD-only log stream.

Duplexing Log Data on a Coupling Facility Log Stream

System logger allows the installation to choose between two methods of duplexing coupling facility log data for a coupling facility log stream:

- Maintain a copy of coupling facility resident log data in local storage buffers on each system.

System Logger

This option is the default, and occurs automatically if you do not specifically request duplexing on staging data sets in the log stream definition in the LOGR policy.

- Maintain a copy of coupling facility resident log data in DASD **staging data sets**, one per log stream on a system.

This option provides hardening of the data on failure independent, persistent media.

You can request the duplexing options for a coupling facility log stream in the log stream definition in the LOGR policy.

You can request duplexing to staging data sets either conditionally, so that a connection will have a staging data set when it contains a single point of failure, or unconditionally, so that every connection to a log stream duplexes to staging data sets. This means that the number of staging data sets active at any given time will depend on the status of each connection and the duplexing method selected for each log stream.

If you request duplexing using staging data sets for a coupling facility log stream in the LOGR policy, system logger uses VSAM linear DASD staging data sets to hold a back-up copy of coupling facility log data for log streams that specify it.

IBM recommends that you plan for DASD staging data sets even if you do not request duplexing to staging data sets. If the primary LOGR couple data set for the sysplex is at OS/390 Release 3 or above, system logger will automatically allocate a staging data set for a log stream under specific error conditions where the log data exists only in a volatile configuration. System logger does this to help safeguard log data against data loss.

Duplexing for a DASD-Only Log Stream

For a DASD-only log stream, system logger automatically duplexes log data from the system's local storage buffers to DASD staging data sets. System logger uses VSAM linear DASD data sets for staging data sets. Note that the staging data set naming convention is different for DASD-only and coupling facility log streams.

Offloading Log Data to DASD

As applications write data to a log stream, the interim storage defined for log data begins to fill, eventually reaching or exceeding its high threshold.

You specify (in the LOGR policy) high and low thresholds for each log stream to define when system logger begins and ends the process of moving, or offloading, log data to DASD log data sets. It is important for you to understand the way system logger offloading works so that:

- You can set the high and low thresholds that control offloading.
- You understand when, why, and how often offloading will occur for your installation.
- You understand when log data is physically deleted from the log stream.
- You can plan coupling facility, log data set, and staging data set sizes to control how often offloading occurs.

The **high threshold** and offloading work differently for coupling facility versus DASD-only log streams:

- **For a coupling facility log stream**, system logger offloads data from the coupling facility to DASD log data sets when coupling facility usage for the log

stream reaches the high threshold. The high threshold is the point, in percent, of coupling facility structure usage for a log stream, where system logger will begin offload processing.

- **For a DASD-only log stream**, system logger offloads data from local storage buffers to DASD log data sets. The high threshold, however, is the point, in percent, of staging data set usage. Thus, for a DASD-only log stream, offload processing moves log data from local storage buffers to DASD log data sets, but the offloading is actually triggered by staging data set usage.

For either type of log stream, when a log stream reaches or exceeds its high threshold, system logger begins processing to offload enough of the oldest log stream data to get to the low offload threshold point specified in the log stream definition. Note that a log stream might exceed the high threshold before offloading starts, because applications might keep writing data before system logger can begin offload processing.

The **low threshold** is the target where offloading is to stop, leaving roughly the specified percentage of log data for that log stream in the coupling facility or staging data set.

For a **coupling facility log stream**, the high threshold you define for the coupling facility structure also applies to the staging data sets for the log stream; offloading is performed when either a staging data set or the coupling facility structure for a log stream hits its high threshold.

While offload processing is occurring, system logger applications can still write data to the log stream; system logger accepts and completes write requests while offloading is in progress. Since system logger calculates the amount of data to offload at the time offload processing starts, by the time offloading has finished, the amount of data in the coupling facility or staging data set exceeds the target low threshold.

When the offload completes, system logger issues an ENF 48 signal.

Other Events That Trigger Offloading

In general, normal offloading is triggered when the high offload threshold for coupling facility or staging data set usage for a log stream is reached. However, there are other system events that cause system logger to begin offload processing. In these cases, **all** the log stream is offloaded. Events triggering offload include:

- Recovery processing for a log stream. When recovery for a log stream is complete, system logger flushes all log data to DASD. See the recovery section in “System Logger Recovery” on page 230 for details.
- Structure rebuild, for coupling facility log streams.
- When the last connector to a log stream disconnects.

Using the High and Low Thresholds to Control Offloading

Applications can influence the frequency of offloads by choosing high and low offload thresholds on the log stream definition in the LOGR policy on the HIGHOFFLOAD and LOWOFFLOAD parameters.

In general, the greater the difference between the two thresholds the less often offloading occurs. If you choose threshold values that are close together, it means that you can keep more data on the coupling facility or local storage buffers for high-performance, low-overhead reads of the log data, but it also means that offloading and offloading overhead will occur more often.

System Logger

The default thresholds are 80% for HIGHOFFLOAD and 0% for LOWOFFLOAD. This is a good setting for applications that primarily write to the log stream and do not retrieve data often, such as logrec log stream and OPERLOG. This setting has the advantage of moving larger chunks of log data to DASD at one time, reducing the number of times that offloading runs.

On the other hand, applications that want a relatively high amount of their log data to reside in the coupling facility or local storage buffers for quick access, transaction logs, for example, might find a HIGHOFFLOAD threshold of 80% and a low offload threshold of 60% more desirable.

IBM recommends that you do not define your HIGHOFFLOAD value to greater than the default of 80%. If you define a higher HIGHOFFLOAD value, you will be more vulnerable to filling your coupling facility or staging data set space for the log stream during sudden bursts of system activity that push the log data above the high threshold. Once coupling facility or staging data set space for a log stream is 100% filled, system logger rejects all write requests until the log data can be offloaded to DASD log data sets.

In any case, it is important to consult the documentation on the log manager application you are running for its application-specific recommendations.

You can tune your threshold settings by monitoring each log stream through SMF data, which shows:

- The number of times offloading has run.
- The number of coupling facility structure full conditions (for coupling facility log streams).
- The number of staging data set full conditions.
- The number of bytes written to and deleted from interim storage (coupling facility for coupling facility log streams or staging data set for DASD-only log streams).

Offloading and Log Data Deletion

Physical deletion of log data that is marked for deletion is tied to offloading; system logger does not physically delete the log data or log data set until an offload occurs. For more information, see “When Is My Log Data or Log Data Set Physically Deleted?” on page 228.

Coupling Facility Size and Offloading

For a coupling facility log stream, applications can influence the frequency of offloads and the processing overhead offloading entails by sizing their coupling facility structure carefully. The larger the coupling facility structure, the less often offloading processing will occur and the larger the chunk of data that will be offloaded. This minimizes offloading overhead and keeps more log data available on the coupling facility structure for high performance reads of log data.

For example, if you have a log stream with a high volume of data being written to it, and you have allocated a small coupling facility structure for this log stream, the coupling facility structure will fill quickly and meet its high threshold, initiating frequent offloading of log data to DASD. This increases system usage, but does not interfere with processing of write requests to the log stream, because write requests can run at the same time as offload processing.

However, the coupling facility structure for a log stream should not be larger than the application really requires and should reflect the needs of the rest of the installation for coupling facility space. See “Determine the Size of Each Coupling

Facility Structure” on page 197 for information about optimal sizing of coupling facility structures for response, throughput, and availability.

If the coupling facility space allocated for a log stream reaches 100% utilization, all write requests against that log stream are rejected until offloading can complete.

Staging Data Set Size and Offloading

Coupling Facility Log Stream: For a coupling facility log stream, the high offload threshold defined by an installation for the coupling facility space allocated to a log stream applies also to the staging data sets in use by connections to that log stream. For example, if you define a high threshold of 80% for a log stream, system logger will begin offload processing when either the coupling facility structure allocated for a log stream or the staging data set for any connection to the log stream reaches or exceeds 80% usage.

This means that when a staging data set reaches the high threshold, system logger immediately offloads data from the coupling facility to log data sets, even if the coupling facility usage for the log stream is below the high threshold.

Thus, if your staging data sets are small in comparison to the coupling facility structure size for a log stream, the staging data sets will keep filling up and system logger will offload coupling facility data to DASD frequently. This means that your installation experiences frequent offloading overhead that could affect performance.

If your staging data sets are too small, you also run the risk of filling them up completely. If this occurs, system logger immediately begins offloading the coupling facility log data in DASD log data sets to harden it. System logger applications will be unable to log data until system logger can free up staging data set space. See the recovery section in the system logger chapter of *OS/390 MVS Programming: Assembler Services Guide*.

In general, the sizing guideline for staging data sets is to make them large enough to hold all the log data in the coupling facility. See “Plan Space for Staging Data Sets” on page 208 for complete information on sizing staging data sets.

DASD-only Log Streams: For a DASD-only log stream, offloading of log data to DASD log data sets is always triggered by staging data set usage. System logger offloads data from local storage buffers to DASD log data sets when it hits the high threshold defined for the staging data sets. For example, if you define a high threshold of 80% for a DASD-only log stream, system logger will begin offload processing when the staging data set space defined for a log stream reaches or exceeds 80% usage.

This means that if you size your staging data sets too small, system logger will offload log data from local storage buffers to DASD log data sets frequently, incurring overhead. You might also run the risk of filling the staging data sets up completely. If this occurs, system logger will immediately begin offloading log data to DASD log data sets.

Finding Information for System Logger Applications

System logger applications can be supplied by IBM, independent software vendors, or written at your installation. For each of these applications, some information is necessary for set-up and use.

Some of the IBM-supplied system logger applications are:

- **CICS log manager** is a CICS system logger application that replaces the journal control management function. It provides a focal point for all CICS system log, forward recovery log, and user journal output within a sysplex and flexible mapping of CICS journals onto log streams. CICS log manager also enables faster CICS restart, dual logging, and flexible log and journal archiving.
- **Logrec log stream** is an MVS system logger application that records hardware failures, selected software errors, and selected system conditions across the sysplex. Using a logrec log stream rather than a logrec data set for each system can streamline logrec error recording.
- **Operations log (OPERLOG)** is an MVS system logger application that records and merges messages about programs and system functions (the hardcopy message set) from each system in a sysplex that activates OPERLOG. Use OPERLOG rather than the system log (SYSLOG) as your hardcopy medium when you need a permanent log about operating conditions and maintenance for all systems in a sysplex.
- **RRS (resource recovery services)** is an MVS system logger application that records events related to protected resources. RRS records these events in five log streams that are shared by systems in a sysplex. Use the information in *OS/390 MVS Programming: Resource Recovery* about defining the log streams with the information in “Preparing to Use System Logger Applications” on page 188 to set up and use the RRS system logger application.
- **APPC/MVS** is an MVS system logger application that records events related to protected conversations. An installation-defined log stream is required for APPC/MVS to protect conversations, through its participation in resource recovery.
- **IMS Common Queue Server (CQS) log manager** is an IMS system logger application that records the information necessary for CQS to recover structures and restart. CQS writes log records into a separate log stream for each coupling facility list structure pair that it uses. IMS CQS log manager uses coupling facility based log streams.

Use the following tables to find the information you'll need to set up and use these system logger applications:

- “Finding Information for CICS Log Manager” on page 185
- “Finding Information for OPERLOG Log Stream” on page 186
- “Finding Information for Logrec Log Stream” on page 186
- “Finding Information for APPC/MVS” on page 187
- “Finding Information for IMS Common Queue Server Log Manager” on page 188.

For information on writing your own system logger application using the system logger services, see

- “Preparing to Use System Logger Applications” on page 188
- *OS/390 MVS Programming: Assembler Services Guide*
- *OS/390 MVS Programming: Assembler Services Reference*

To prepare to use a system logger application supplied by a independent software vendor see “Preparing to Use System Logger Applications” on page 188.

Finding Information for CICS Log Manager

Use this table to find information to help you set up and use the CICS log manager. Almost all of the information you will need is in the CICS library, but there are a few steps needed to define CICS log streams to MVS, and these steps are documented in this chapter. Note that it is important that you follow the CICS recommendations when using the CICS log manager.

Table 7. Finding CICS Log Manager Information

What is your goal?	You should read:
Learn about CICS log manager.	<i>CICS Transaction Server for OS/390 Release Guide</i> <i>CICS Transaction Server for OS/390 Installation Guide</i> <i>CICS System Definition Guide</i>
Plan for and set up the CICS log manager log streams.	<i>CICS Transaction Server for OS/390 Installation Guide</i> <i>CICS System Definition Guide</i>
Develop a naming convention for system logger resources.	<i>CICS Transaction Server for OS/390 Installation Guide</i> for log stream names. <i>CICS System Definition Guide</i>
Size the coupling facility structure space needed for the CICS log manager.	<i>CICS Transaction Server for OS/390 Installation Guide</i>
Plan and set up DASD space needed for CICS log manager log data sets and staging data sets.	“Plan DASD Space for System Logger” on page 206 <i>CICS Transaction Server for OS/390 Installation Guide</i>
Define authorization to system logger resources for CICS log manager.	“Define Authorization to System Logger Resources” on page 215
Define the CICS log manager DASD log and staging data sets to global resource serialization in the GRSRNLxx parmlib member.	“Add the DASD Data Sets to the GRSRNL Inclusion List” on page 212
Format the LOGR couple data set, identify it to the sysplex, and make it available.	“Format the LOGR Couple Data Set and Make it Available to the Sysplex” on page 216
Add, update, or delete CICS policy data (CICS log streams and coupling facility structures) in the LOGR policy using the IXCMIAPU utility.	“LOGR Parameters for Administrative Data Utility” on page 281 See <i>CICS Transaction Server for OS/390 Installation Guide</i> for details on recommendations for structure and log stream attributes.
Update the CFRM sysplex couple data set with information on CICS log manager coupling facility structures.	“CFRM Parameters for Administrative Data Utility” on page 275 See <i>CICS Transaction Server for OS/390 Installation Guide</i> for recommendations for structure and log stream attributes and sample jobs. See <i>CICS Transaction Server for OS/390 Release Guide</i> for sample jobs.
Activate the LOGR subsystem.	“Activate the LOGR Subsystem” on page 226.

System Logger

Finding Information for OPERLOG Log Stream

Use this table to find information to help you set up and use the OPERLOG system logger application.

Table 8. Finding OPERLOG Information

What is your goal?	You should read:
Learn about the operations log stream (OPERLOG).	<i>OS/390 MVS Planning: Operations</i>
Decide whether to use OPERLOG or SYSLOG as your hardcopy medium.	
Plan to use OPERLOG as your hardcopy medium.	
Learn requirements for system logger applications.	"Understand the Requirements for System Logger" on page 190.
Develop a naming convention for system logger resources.	"Develop a Naming Convention for System Logger Resources" on page 203
Do set-up for the OPERLOG log stream.	See "Preparing to Use System Logger Applications" on page 188.
Size the coupling facility structure space needed for the OPERLOG log stream.	See the appendix on coupling facility structures in <i>System/390 MVS Parallel Sysplex Configuration Cookbook</i> , SG24-4706.
Tune the OPERLOG log stream definition using SMF record type 88.	<i>OS/390 MVS System Management Facilities (SMF)</i>
Delete OPERLOG log data.	See: <ul style="list-style-type: none">• "Managing Log Data: How Much? For How Long?" on page 213• "Deleting Log Data and Log Data Sets" on page 227
At IBM's request, gather system logger component trace data.	<i>OS/390 MVS Diagnosis: Tools and Service Aids</i>

Finding Information for Logrec Log Stream

Use this table to find information to help you set up and use the logrec log stream.

Table 9. Finding Logrec Log Stream Information

What is your goal?	You should read:
Learn about logrec error records.	<i>OS/390 MVS Diagnosis: Tools and Service Aids</i>
Decide whether to use the logrec data set or logrec log stream as your logrec recording medium.	
Plan to use the logrec log stream.	
Learn requirements for system logger applications.	"Understand the Requirements for System Logger" on page 190.
Develop a naming convention for system logger resources.	"Develop a Naming Convention for System Logger Resources" on page 203
Do set-up for the logrec log stream.	See "Preparing to Use System Logger Applications" on page 188.
Size the coupling facility structure space needed for the logrec log stream.	See the appendix on coupling facility structures in <i>System/390 MVS Parallel Sysplex Configuration Cookbook</i> , SG24-4706.
Activate the LOGR subsystem.	"Activate the LOGR Subsystem" on page 226.

Table 9. Finding Logrec Log Stream Information (continued)

What is your goal?	You should read:
Use the LOGR subsystem to obtain logrec log stream error records.	<i>OS/390 MVS Programming: Assembler Services Guide</i> See <i>OS/390 MVS Diagnosis: Tools and Service Aids</i> for logrec specific information.
Delete logrec log data.	See: <ul style="list-style-type: none"> • “Managing Log Data: How Much? For How Long?” on page 213 • “Deleting Log Data and Log Data Sets” on page 227
Tune the logrec log stream definition using SMF record type 88.	<i>OS/390 MVS System Management Facilities (SMF)</i>
Determine your current and alternate logrec medium names: <ul style="list-style-type: none"> • To find your current logrec medium, issue the DISPLAY LOGREC command. • To find the current and alternate logrec mediums, use the IEANTRT service. 	For the DISPLAY LOGREC command, see <i>OS/390 MVS System Commands Summary</i> . For the IEANTRT service in an authorized environment, see <i>OS/390 MVS Programming: Authorized Assembler Services Reference ENF-IXG</i> . For the IEANTRT service in an unauthorized environment, see <i>OS/390 MVS Programming: Assembler Services Reference</i> .
Issue the SETLOGRC command to change your logrec medium.	<i>OS/390 MVS System Commands Summary</i>
At IBM’s request, gather system logger component trace data.	<i>OS/390 MVS Diagnosis: Tools and Service Aids</i>

Finding Information for APPC/MVS

Use this table to find information to help you set up the log stream for APPC/MVS to use during resource recovery. An APPC/MVS log stream is required for protected conversations support.

Table 10. Finding APPC/MVS Information

What is your goal?	You should read:
Plan for and set up the APPC/MVS log stream.	“Preparing to Use System Logger Applications” on page 188 contains most of the information you need to plan and set up a log stream for APPC/MVS. For information that is specific to APPC/MVS, however, refer to the topic on protected conversations support in <i>OS/390 MVS Planning: APPC/MVS Management</i> , which contains: <ul style="list-style-type: none"> • The required name to use for the APPC/MVS log stream • Requirements and suggestions for the associated coupling facility structure.
Plan and set up DASD space needed for APPC/MVS log data sets and staging data sets.	“Plan DASD Space for System Logger” on page 206
Define the APPC/MVS log and staging data sets to global resource serialization in the GRSRNLxx parmlib member.	“Add the DASD Data Sets to the GRSRNL Inclusion List” on page 212
Define authorization to system logger resources for APPC/MVS.	“Define Authorization to System Logger Resources” on page 215
Format the LOGR couple data set, identify it to the sysplex, and make it available.	“Format the LOGR Couple Data Set and Make it Available to the Sysplex” on page 216

System Logger

Table 10. Finding APPC/MVS Information (continued)

What is your goal?	You should read:
Add, update, or delete APPC/MVS policy data (the APPC/MVS log stream and its associated coupling facility structure) in the LOGR policy using the IXCMIAPU utility.	"LOGR Parameters for Administrative Data Utility" on page 281
Update the CFRM sysplex couple data set with information on the coupling facility structure associated with the APPC/MVS log stream.	"CFRM Parameters for Administrative Data Utility" on page 275
Activate the LOGR subsystem.	"Activate the LOGR Subsystem" on page 226

Finding Information for IMS Common Queue Server Log Manager

Use this table to find information to help you set up and use the IMS subsystem common queue server (CQS) log manager. Almost all of the information you will need is in the IMS library, but there are a few steps needed to define IMS log streams to MVS, and these steps are documented in this chapter.

Table 11. Finding IMS (CQS) Log Manager Information

What is your goal?	You should read:
Learn about the IMS subsystem uses the CQS log.	<i>IMS/ESA Common Queue Server Guide and Reference</i>
Plan for and set up the IMS (CQS) log manager log streams.	<i>IMS/ESA Common Queue Server Guide and Reference</i>
Develop a naming convention for system logger resources.	"Develop a Naming Convention for System Logger Resources" on page 203
Size the coupling facility structure space needed for the logrec log stream.	"Determine the Size of Each Coupling Facility Structure" on page 197
Plan and set up DASD space needed for IMS (CQS) log manager log data sets and staging data sets.	"Plan DASD Space for System Logger" on page 206
Define authorization to system logger resources for IMS (CQS) log manager.	"Define Authorization to System Logger Resources" on page 215
Format the LOGR couple data set, identify it to the sysplex, and make it available.	"Format the LOGR Couple Data Set and Make it Available to the Sysplex" on page 216
Add, update, or delete IMS CQS log data (IMS CQS log streams and coupling facility structures) in the LOGR policy using the IXCMIAPU utility.	"LOGR Parameters for Administrative Data Utility" on page 281 <i>See IMS/ESA Common Queue Server Guide and Reference for details on recommendations for structure and log stream attributes.</i>
Update the CFRM sysplex couple data set with information on IMS (CQS) log manager coupling facility structures.	"CFRM Parameters for Administrative Data Utility" on page 275
Activate the LOGR subsystem.	"Activate the LOGR Subsystem" on page 226

Preparing to Use System Logger Applications

Before using the system logger services or system logger applications an installation must do the following:

- "Understand the Requirements for System Logger" on page 190
- "Plan the System Logger Configuration" on page 191
- "Determine the Size of Each Coupling Facility Structure" on page 197

- “Develop a Naming Convention for System Logger Resources” on page 203
- “Plan DASD Space for System Logger” on page 206
- “Managing Log Data: How Much? For How Long?” on page 213
- “Define Authorization to System Logger Resources” on page 215
- “Format the LOGR Couple Data Set and Make it Available to the Sysplex” on page 216
- “Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy” on page 217
- “Define the Coupling Facility Structures Attributes in the CFRM Policy Couple Data Set” on page 225
- “Activate the LOGR Subsystem” on page 226.

These planning steps apply for both coupling facility and DASD-only log streams. Each planning step following indicates how it applies to coupling facility versus DASD-only log streams. When there is a difference in how a step should be performed based on the type of log stream, this is indicated as well.

The following chart illustrates the steps required for setting up both types of log streams.

Table 12. Reviewing System Logger Setup Activities

Common Setup Activities	Additional CF Log Stream Steps
Sysplex Environment	
<ul style="list-style-type: none"> • Need to be in a sysplex environment (PLEXCFG) • Set up sysplex, LOGR couple data sets and LOGR policy • Parmlib members — IEASYSxx, COUPLExx 	<ul style="list-style-type: none"> • Need to be in a Parallel Sysplex environment • Add coupling facility • Set up CFRM couple data sets and policy
Log Stream Configuration	
Plan <ul style="list-style-type: none"> • Types of log streams • How many log streams • Log stream recovery 	Map log streams to CF structures <ul style="list-style-type: none"> • Determine structure sizes • Plan structure placement • Plan peer connector recovery
Naming Conventions	
<ul style="list-style-type: none"> • Log stream names • Log stream DASD data set names • Add log stream DASD data set names to GRSRNLxx inclusion list 	<ul style="list-style-type: none"> • CF structure names
DASD Space	
<ul style="list-style-type: none"> • Allow for DASD expansion, DSEXTENTs • Plan use of retention period for DASD log stream data • Plan log stream staging data set size and access • Plan SMS data set characteristics for log stream data sets • If multisystem sysplex: <ul style="list-style-type: none"> – Need shared catalog – Access to DASD volumes – Serialization mechanism – SHAREOPTIONS(3,3) 	<ul style="list-style-type: none"> • Select duplexing method for log stream data
Security Authorization	

System Logger

Table 12. Reviewing System Logger Setup Activities (continued)

Common Setup Activities	Additional CF Log Stream Steps
<ul style="list-style-type: none">• IXGLOGR address space• Administrator of LOGR couple data set• Exploiters of log streams	<p>CFRM couple data set and structures</p> <ul style="list-style-type: none">• For IXGLOGR address space• Administrator of CFRM couple data set• Exploiters of log stream structures
Activate SMS	
Activate LOGR	
Establish SMF88 Reporting	

Understand the Requirements for System Logger

Coupling Facility Requirement for System Logger

If your installation plans to use any coupling facility log streams, system logger requires a coupling facility running coupling facility control code at one of the following levels:

- CFLEVEL 1 with service level 4.03 or higher.
- CFLEVEL 2 or higher.

This requirement applies to coupling facility log streams only. See “Determine the Right Kind of Log Stream for Each Application” on page 191 for determining whether you should use coupling facility versus DASD-only log streams.

If you are testing coupling facility function using the Integrated Coupling Migration Facility (ICMF), then system logger requires that ICMF be at coupling facility control code at the same level as above.

Sysplex Requirement

To use system logger applications with either a coupling facility or DASD-only log stream, your sysplex must be in monoplex or multisystem mode. You must specify MONOPLEX or MULTISYSTEM on the PLEXCFG parameter in the IEASYSxx parmlib member.

System Managed Storage (SMS) Requirement for System Logger

For either a coupling facility or a DASD-only log stream, system logger requires that you have SMS installed and active at your installation. This is true **even if you do not** use SMS to manage the DASD log data sets and staging data sets, because system logger uses SMS to allocate data sets, such as the DASD log stream and staging data sets.

If SMS is not active when you attempt to use a system logger application, system logger issues system messages that indicate allocation errors.

VSAM Share Options for System Logger

If you have multiple systems in the sysplex, it is typical for system logger to require access to log stream data sets and staging data sets from multiple systems. For this reason, you must specify VSAM SHAREOPTIONS(3,3) for log stream data sets and staging data sets.

If you use SMS to manage DASD data sets, See “Set Up the SMS Environment for DASD Data Sets” on page 212.

If you do not use SMS to manage DASD data sets, your installation must do one of the following:

- Provide an allocation exit that will add a DATACLAS text unit that corresponds to a specification of SHAREOPTIONS(3,3) for log stream related data sets. See *OS/390 MVS Programming: Authorized Assembler Services Guide*, Chapters 25 and 26 for more information.
- Provide an ACS DATACLAS routine where the DATACLAS specifies SHAREOPTIONS(3,3) for log stream related data sets. See *OS/390 DFSMSdfp Storage Administration Reference*, Chapter 7, for more information.

If you are using a single system sysplex, there is no requirement for any particular SHAREOPTIONS.

Authorization Requirements for System Logger

To use system logger applications, installations must define authorization to system logger resources, such as log streams or coupling facility structures associated with log streams. For more about the authorization requirements for system logger, see “Define Authorization to System Logger Resources” on page 215.

Plan the System Logger Configuration

When you set up your logging configuration for a system logger application, you should do some basic planning. See the following topics:

- “Determine the Number of Log Streams Required”. This step is required for both coupling facility and DASD-only log streams.
- “Determine the Right Kind of Log Stream for Each Application”.
- “Determine Which Log Streams Map to Which Coupling Facility Structures” on page 192 . Do this for coupling facility log streams only.
- “Plan Your Configuration for Log Data Recovery” on page 195. Do this for coupling facility log streams only.

Determine the Number of Log Streams Required

For both coupling facility and DASD-only log streams, you must plan the number of log streams required by system logger applications for your installation based on the kinds of logger applications you plan to have and the type of log stream, coupling facility or DASD-only. For example, one application might run on multiple systems in the sysplex and require a single sysplex-wide log stream. For another application running on multiple systems, you might want a separate log stream for each system.

For OPERLOG and logrec log streams, if you want to merge data from across the sysplex, you define one coupling facility log stream for the sysplex, written to by all the systems.

Determine the Right Kind of Log Stream for Each Application

Now that you know how many log streams you need and what system logger applications your installation will use, you must determine what kind of log stream is right for each application - coupling facility or DASD-only log stream. You will define the log stream as either coupling facility or DASD-only in the log stream definition in the LOGR policy in a later step in the setup process (see “Specifying Whether the Log Stream is DASD-Only” on page 217).

To decide which type of log stream is right for you, consult the application documentation or programmer for the system logger applications your installation uses and keep in mind the following:

- Does the application support DASD-only log streams? DASD-only log streams are supported for the following IBM applications:
 - CICS log manager

System Logger

- Operations log stream (OPERLOG)
- Logrec log stream
- Resource Recovery Services

You cannot use a DASD-only log stream for APPC/MVS log streams.

For other system logger applications, consult the programmer or the application documentation.

- If an application needs to merge data from multiple systems, you must use a coupling facility log stream, which requires, of course, a coupling facility.
- If you do not have access to a coupling facility, a DASD-only log stream is the only kind you can use.
- DASD-only log streams can only be single system in scope. Only one system can connect to a DASD-only log stream. Multiple instances of an application may connect to a DASD-only log stream, but they must all be on the same system. Use a DASD-only log stream when you do not have access to or do not want to use a coupling facility for logging.
- You can use a coupling facility log stream for single system scope applications, but a coupling facility is still required.

Here are some application examples:

- An application runs on multiple systems in the sysplex and requires a single sysplex-wide log stream. For this application, you would have to use a coupling facility log stream, because data is merged from multiple systems to a single log stream. This requires a coupling facility.
- Another application runs on multiple systems and requires a separate log stream for each system. For this application you can use either a DASD-only or a coupling facility log stream. However, if you use a coupling facility log stream you must have a coupling facility.

Note that an installation can consist of coupling facility log streams only, DASD-only log streams only, or a combination of both, as long as all requirements are met.

Determine Which Log Streams Map to Which Coupling Facility Structures

Each installation planning for coupling facility log streams needs to plan how to assign log streams to coupling facility structures. This can be done in a variety of ways: multiple log streams sharing one structure, one structure for each log stream, related log streams using a structure, and so forth. The important point is that the approach be systematic, easy for your installation to track, and easy to recover data for, in the event of a system, sysplex, or coupling facility failure.

When you plan how log streams will map to coupling facility structures, there are some limitations to consider, such as the maximum allowable numbers of:

- Structures you can define to a coupling facility.
- Connectors you can define to a coupling facility structure.
- Structures you can define in a CFRM policy.

For coupling facility limitations, see the *PR/SM Planning Guide*. Structure limitations are documented with the IXCL1DSU utility for the CFRM and LOGR policies. See “Appendix A. Format Utility for Couple Data Sets” on page 241 for the IXCL1DSU utility.

When you have more than one log stream using a single coupling facility structure, system logger divides the structure storage evenly among the log streams that have at least one connected system logger application. For example, if an installation assigns three log streams to a single structure, but only one log stream has a connected system logger application, then that one log stream can use the entire coupling facility structure. When an application connects to the second log stream, system logger dynamically divides the structure evenly between the two log streams. When an application connects to the third log stream, system logger allocates each log stream a third of the coupling facility space. For this reason, it is important that the log streams that share a coupling facility structure be equal in size and usage rates. Knowing this can help you understand and plan how much coupling facility space you have for each log stream and predict when log stream data will be written to DASD as the coupling facility space becomes filled.

Start planning your configuration by determining the number of coupling facility structures available for logging, then determine the number of log streams needed for all system logger applications. Group log streams with the same characteristics together in coupling facility structures. **IBM recommends** that you limit the number of log streams per structure to 20 or fewer. This is because system logger connect and log stream recovery processing (which affect the restart time of system logger applications) have been optimized to provide parallelism at the coupling facility structure level. Therefore, the more structures you use, the greater parallelism you'll get during log stream connect and rebuild processing.

For example, if you have 10 log streams, each one defined to its own unique coupling facility structure, system logger can process connect requests to each log stream in parallel. On the other hand, if you define all 10 log streams to the same coupling facility structure, system logger must sequentially process log stream connect requests for the log streams.

The LOGSNUM Parameter: You will define the maximum allowable number of log streams that map to a coupling facility structure on the LOGSNUM parameter for each structure in the LOGR policy in a later step, (“Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy” on page 217).

Because system logger divides the structure storage evenly among each log stream that writes to it, it is important to plan carefully what value you want to specify for LOGSNUM and how many of the log streams defined to a shared coupling facility structure will be active at any one time. The more connected log streams that share a coupling facility structure, the less coupling facility space each one has. In addition, when a structure is allocated for system logger, some of the structure space is taken up for control information for each log stream that could be connected to this structure. The higher the number for LOGSNUM, the more coupling facility control space is consumed.

In general, the value you define for LOGSNUM should be as small as possible. IBM recommends that you configure 20 or fewer log streams into one structure.

Examples of Coupling Facility Log Stream Configurations

The following examples show different ways to associate log streams and coupling facility structures for system logger applications. There are many other ways to set up a configuration for your system logger application.

Example - Merging Log Data from Across the Sysplex: The scenario in Figure 50 on page 194 shows a configuration where a transaction manager system logger application runs on two systems in a sysplex. The applications write and

System Logger

merge log data into one log stream, named TRANSLOG in this example. This configuration is the one you would set up for the OPERLOG or logrec log stream applications.

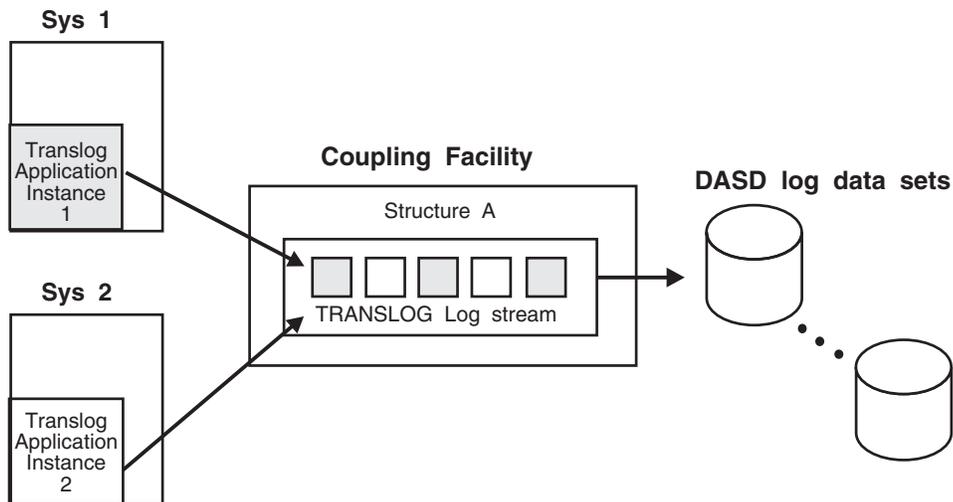


Figure 50. Example: Merging Log Data from Systems Across a Sysplex

Some of the key points this example shows are:

- Log data is written to the TRANSLOG log stream on a coupling facility list structure.
- Multiple instances of a system logger application running on different systems to share data.
- The TRANSLOG log stream is the only log stream occupying the coupling facility list structure.
- System logger automatically moves the oldest log data from the coupling facility to DASD when the data hits the installation-defined thresholds for the coupling facility list structure for this log stream. From the point of view of the system logger application, it does not matter whether the log data resides on the coupling facility structure or on DASD.

Example - Maintaining Multiple Log Streams in the Same Coupling Facility

Structure: Figure 51 on page 195 shows a configuration where you can maintain individual logs for each system logger application in separate, independent log streams that map to the same coupling facility list structure. You might, for example, want to have your logrec log stream and OPERLOG log stream map to the same coupling facility structure.

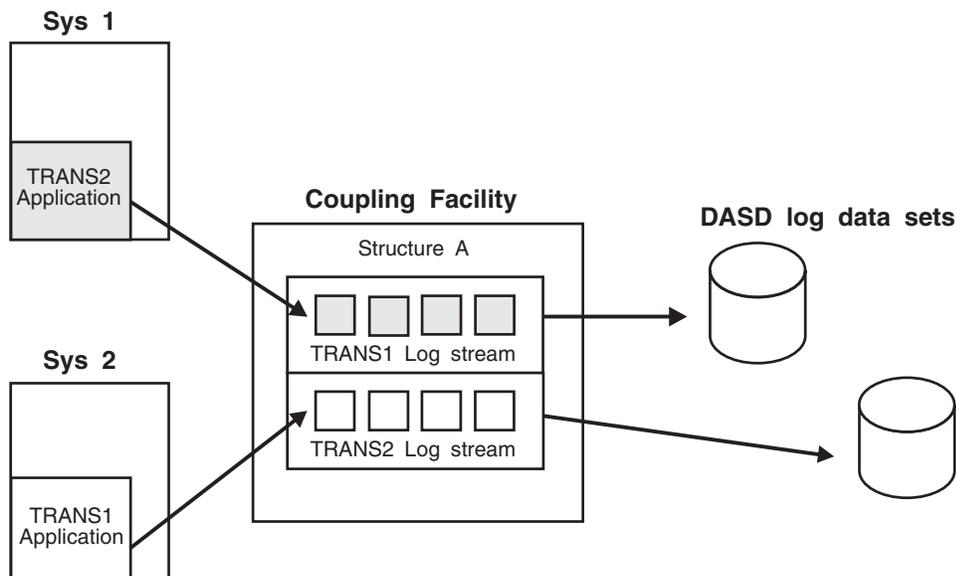


Figure 51. Example: Maintaining Multiple Log Streams in the Same Coupling Facility Structure

Some of the key points this example shows are:

- Although they reside in the same coupling facility structure, the two log streams are separate. The data from the two log streams is **not** merged.
- System logger automatically moves the oldest log data from the coupling facility to DASD when the data hits the installation-defined thresholds for the coupling facility list structure for a log stream.

Plan Your Configuration for Log Data Recovery

When you are deciding which coupling facility structures to associate with each coupling facility log stream, you can plan it so that system logger can easily recover coupling facility data in the case of a system failure. This step applies to coupling facility log streams only.

When a system that has a system logger application fails, system logger tries to safeguard all the coupling facility log data for the failed system by offloading it to DASD log data sets so that it is on a persistent media. This recovery processing is done by a **peer connector**, which is another system in the sysplex with a connection to a coupling facility structure that the failing system was also connected to. In general, when you set up your logging configuration, make sure that each log stream structure is accessed by more than one system. Note that a peer connector need only be connected to the same coupling facility structure, not the same log stream.

If there is no peer connection available to perform recovery for a failed system, recovery is delayed until either the failing system re-IPLs or another system connects to a log stream in the same coupling facility structure that the failing system was connected to. In general, however, the log stream data you need will be available when you need it: when a system reconnects or connects to a log stream that the failed system was connected to in order to write or retrieve data. Recovery will be performed when that peer connector connects to the log stream.

Examples - Peer Connectors Recovering Coupling Facility Log Data For a Failing System: In Figure 52, system SYS1 fails. System logger recovery processing for SYS1 consists of moving all the coupling facility resident log data for

System Logger

the system to DASD log data sets. Because SYS1 is connected to log streams on coupling facility structure A, it needs a peer connector with a connection to structure A to perform recovery. SYS2 can perform recovery for SYS1 because it has a system logger application connected to a log stream that maps to coupling facility structure A. SYS3 is not a peer connector for SYS1 and cannot perform recovery because it does not have a connection to coupling facility structure A.

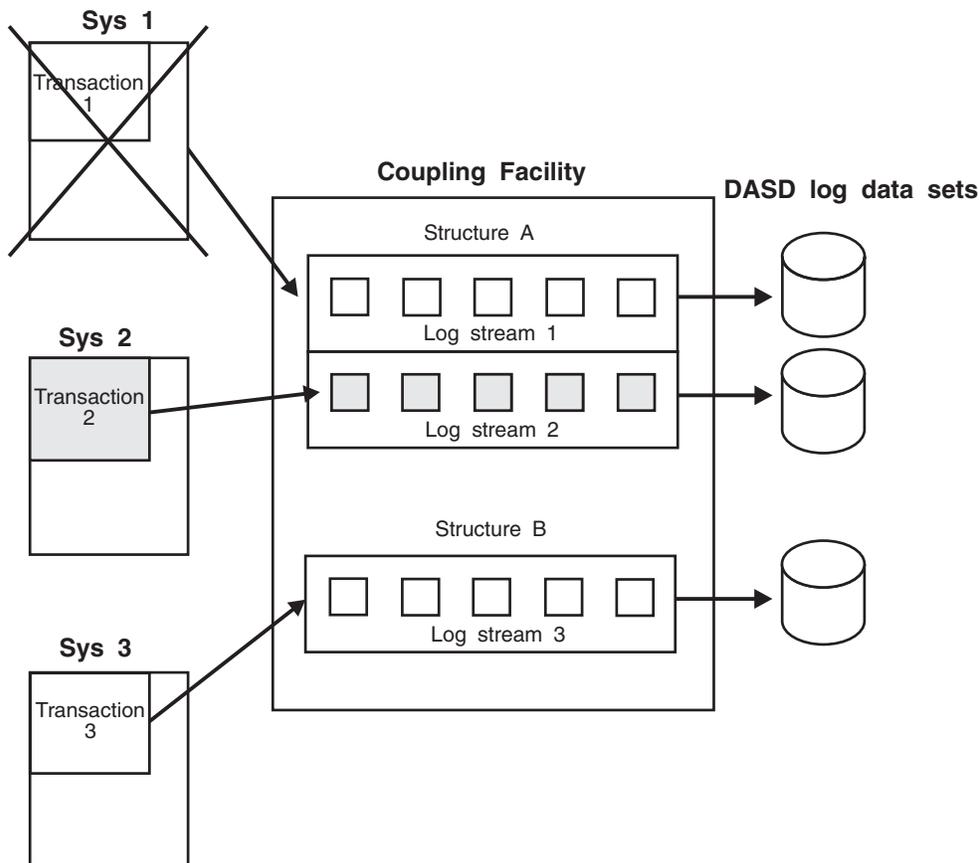


Figure 52. Example: Recovery for a Failing System - All Data Recovered

In Figure 53 on page 197, system SYS1 also fails. SYS1 has connections to coupling facilities structures A and B and so needs peer connectors to both structures to recover all its coupling facility resident log data. SYS2 can perform recovery for SYS1 log data on coupling facility structure B, but there is no peer connector to perform recovery for SYS1's log data on coupling facility structure A.

Recovery for SYS1's coupling facility structure A log data is delayed until either SYS1 re-IPLs or another system connects to a log stream on coupling facility structure A.

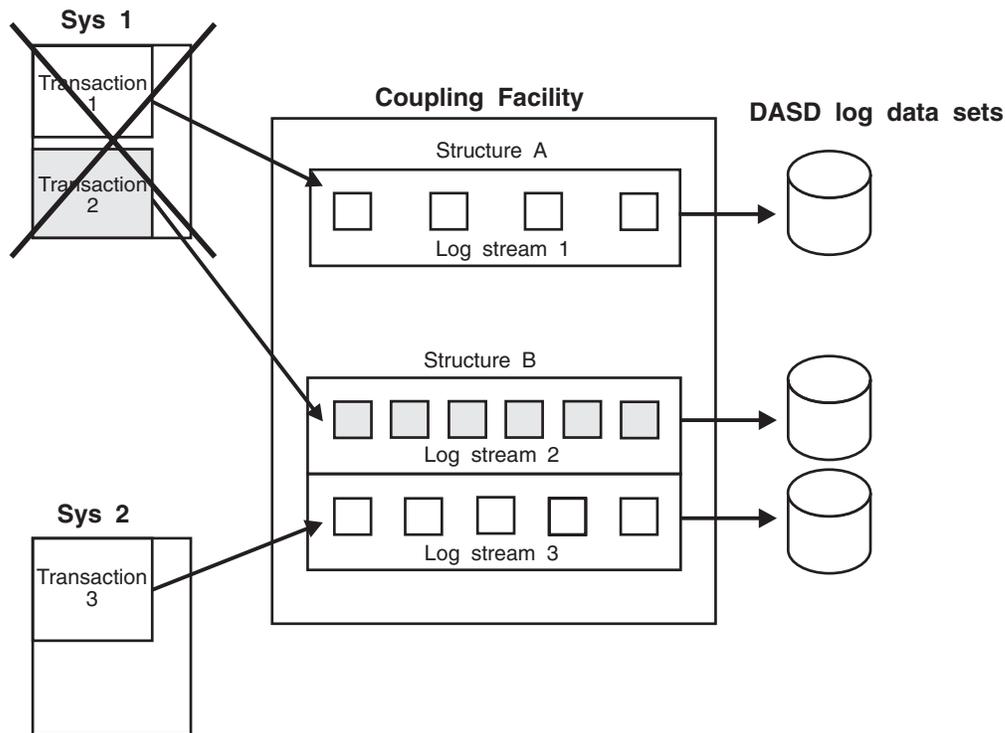


Figure 53. Example: Recovery for a Failing System - Recovery Delayed for Some Data

Determine the Size of Each Coupling Facility Structure

Use the following procedure to calculate how large you should make each coupling facility list structure allocated for coupling facility log streams. This procedure applies only to coupling facility log streams. You can use this procedure for any system logger application, including applications provided by IBM, independent software vendors, or your own installation.

For logrec log stream and operations log log streams, specific sizing recommendations are provided in the *System/390 MVS Parallel Sysplex Configuration Volume 2: Cookbook*, SG24-2076.

For other IBM-provided system logger applications, such as CICS log manager, you should get recommendations for the size of your coupling facility structure or structures from documentation on that particular application.

The procedure will help you determine the minimum amount of coupling facility space you should allocate for a coupling facility structure, based on the amount of space needed for all the log streams mapping to the structure.

This procedure requires the use of the *PR/SM Planning Guide*, order number GA22-7123-13 or higher. Versions at a lower dash level than -13 are not valid with

System Logger

this sizing procedure. The example provided at the end of this procedure applies only for a coupling facility at the following levels:

- CFLEVEL 1 with a service level 4.03 or higher.
- CFLEVEL 2 or higher.

Go through the procedure once for every coupling facility structure.

1. **Gather information about the coupling facility structure.**

For each structure, gather the following information common for all the log streams that map to the structure. You will use these values in sizing the coupling facility space you need. Most of this information corresponds to parameters in the structure definition of the LOGR policy. You can specify the values you gather here when you add information to the LOGR policy (see “Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy” on page 217).

AVGBUFSIZE Average size of log blocks written by applications to log streams associated with this coupling facility structure. This value corresponds to the AVGBUFSIZE parameter specified for each structure in the LOGR policy. For this procedure, pick the value that represents the average size of log blocks generated by all the log streams writing to this coupling facility structure. For information on specifying the AVGBUFSIZE parameter in the LOGR policy, see “Specifying the Average Log Block Size” on page 218.

For a logrec log stream, **IBM recommends** an AVGBUFSIZE of 4068.

For an OPERLOG log stream, **IBM recommends** an AVGBUFSIZE of 512.

For an RRS log stream, see *OS/390 MVS Programming: Resource Recovery*.

Installation or vendor supplied applications should base AVGBUFSIZE on the average size of the log blocks they want to write to the log stream.

MAXBUFSIZE Maximum size of a log block corresponding to the MAXBUFSIZE parameter in the LOGR policy.

For a logrec or OPERLOG log stream, **IBM recommends** a MAXBUFSIZE of 4096.

For an RRS log stream, see *OS/390 MVS Programming: Resource Recovery*.

Installation or vendor supplied applications should base MAXBUFSIZE on the maximum size of the log blocks they want to write to the log stream.

Note that once you have defined the MAXBUFSIZE for a coupling facility structure, you cannot update the value. To change the MAXBUFSIZE, you must delete the log streams associated with the structure, delete the structure, and then re-define the structure with the new MAXBUFSIZE value.

LOGSNUM Maximum number of log streams in the structure corresponding to the LOGSNUM parameter in the LOGR policy. See “The LOGSNUM Parameter” on page 193 for more information.

MXSS-to-TSS The ratio of the maximum structure size to the target structure size. For example, if you want your maximum structure size to be twice as big as your target structure size, then MXSS-to-TSS would be 2. See Appendix B in the *PR/SM Planning Guide*.

2. Gather information about each log stream

For each log stream that maps to a coupling facility structure, gather the following information.

Most of these values are projected or desired values for your log stream, and a few correspond to parameters you will specify in the LOGR policy. You will use these values in sizing the coupling facility space you need.

LOWOFFLOAD

LOWOFFLOAD specifies the point, in percent value of space consumed, where system logger will stop offloading coupling facility log data to the DASD log data sets for this log stream. It corresponds to the LOWOFFLOAD parameter in the LOGR policy. For OPERLOG, logrec log stream, or the CICS log manager, **IBM recommends** that you use the LOWOFFLOAD default value of zero.

See “Appendix B. Administrative Data Utility” on page 263 for more information on the LOWOFFLOAD parameter.

HIGHOFFLOAD

HIGHOFFLOAD specifies the point, in percent value of space consumed, where system logger will begin offloading coupling facility log data to the DASD log data sets for this log stream. It corresponds to the HIGHOFFLOAD parameter in the LOGR policy.

For OPERLOG, logrec log stream, or the CICS log manager, **IBM recommends** that you use the HIGHOFFLOAD default value of 80.

See “Appendix B. Administrative Data Utility” on page 263 for more information on the HIGHOFFLOAD parameter.

ResidencyTime

Desired residency time for log data in the coupling facility structure, in seconds. Residency time is the amount of time you want a log block to stay in the coupling facility between the time it is written to the coupling facility and being offloaded to DASD.

For a logrec or OPERLOG log stream, **IBM recommends** a residency of 10 seconds. For other system logger applications, get the desired residency time from the documentation for the application.

WritePerSec

Projected write requests per second against the log stream. This value is the total IXGWRITE requests per second issued by all system logger applications or systems connected to a log stream.

For an OPERLOG log stream, you can calculate projected writes per second using the current SYSLOGs. For each system in the sysplex, do the following calculation:

- a. Choose a spot in SYSLOG and note the timestamp at that spot. You might want to choose a peak or high usage time of day.

System Logger

- b. Page down 2000 lines by issuing, for example, DOWN 2000 to SDSF or TSO/E browse.
- c. Note the time stamp in this spot.
- d. Calculate the number of seconds between the two time stamps. This is the number of seconds it took to write 2000 lines of SYSLOG. (This calculation is based on most messages being one line).
- e. Divide 2000 by the number of seconds it took to write 2000 lines to get the lines per second.
- f. Add the results for each system's SYSLOG to get the total writes per second for the OPERLOG log stream.

For a logrec log stream, calculate the projected writes per second using the current logrec data sets. For each system that will write to the logrec log stream, do the following calculation:

- a. Request or obtain an EREP report for a particular time span. It must be a report that includes all records. For example, take an EREP daily report that processed all records.
 - b. Near the top of the report, message IFC120I will tell how many records were written in the time span chosen for the report. Divide this number by the number of seconds in the time span for the average writes per second for this logrec data set.
 - c. You can also look at timestamps in the output to analyze how many records were written in a particular second. You can do this to check for peak usage.
 - d. Add the results for each logrec data set that will write to the log stream to get the total writes per second for a logrec log stream.
3. **Calculate coupling facility structure list entries required per log stream.**

For each log stream that maps to the coupling facility structure, use the following formula to calculate the number of list structure entries required to represent a log stream's data. The number of entries a log stream needs in the coupling facility list structure depends on the coupling facility residency time desired for a given log stream. Use the values for WritePerSec and ResidencyTime gathered in the prior step:

$$\text{Entries} = ((100/(\text{HighOffload}-\text{LowOffload})) * \text{WritePerSec} * \text{ResidencyTime}) + 4$$

4. **Calculate the total coupling facility structure size.**

This step calculates the size needed for each coupling facility structure to contain all the log data for the log streams that map to it. Perform this step for each coupling facility structure planned for system logger applications.

This step requires that you use the values from the previous steps and refer to Appendix B in *PR/SM Planning Guide* for formulas for the total structure size (TSS) and maximum structure size (MSS) for a coupling facility level 1 list structure at service level 4.01. This formula applies to the following coupling facility levels:

- CFLEVEL 1 with service level 4.03 or higher.
- CFLEVEL 2 or higher.

In order to pick the correct formula, you'll need to know that the system logger uses keys and adjunct data, but no names for its list structures.

For each structure, use the following data as input to the TSS and MXSS formulas:

MLSEC

MLSEC is the number of coupling facility list entries needed for all the log streams that map to a coupling facility structure. Calculate this value by taking the value for **entries** calculated in step 3 on page 200 for the log stream with the largest **entries** value and do the following:

$$\text{MLSEC} = \text{entries} * \text{maximum number concurrently active log streams for the structure}$$

You take the **entries** value for the largest log stream because when there are multiple log streams assigned to one coupling facility structure, system logger divides the structure space equally between each log stream connected to the structure.

If you are not sure how many of the log streams mapped to the structure will be concurrently active at any given time, multiply the entries times the LOGSNUM value for the structure to calculate the value for MLSEC. That way, you can be certain your coupling facility structure will be large enough for all logging activity.

MLSELC

Not used.

LELX If MAXBUFSIZE is greater than 65,280, LELX = 1
Otherwise, LELX = 0.

MDLES

To calculate MDLES

$$(\text{Greater of MAXBUFSIZE or 8192}) / (256 * (2 ** \text{LELX})).$$

LC Do the following to calculate LC:

$$\text{LOGSNUM} * 4 + 1$$

LTEC LTEC = 0

LTEX LTEX = 0

R_le R_le=1

R_data

Do the following to calculate R_data:

$$\begin{aligned} \text{R_data} = & (\text{AVGBUFSIZE} + 4 \text{ Bytes} \\ & \text{rounded up to the next } (256 * (2 ** \text{LELX})) \text{ boundary}) \\ & / (256 * (2 ** \text{LELX})) \end{aligned}$$

5. **Calculate SIZE and INITSIZE parameter values for the structure definition in the CFRM policy.**

Take the TSS and MXSS values for a level 1 coupling facility at service level 4.01 or higher and do the following to calculate SIZE and INITSIZE:

$$\text{INITSIZE} = \text{TSS} * 4$$

$$\text{SIZE} = \text{MXSS} * 4$$
Example of determining the size of a coupling facility structure.

The following example shows how to calculate the size of a coupling facility structure that has two log streams mapping to it. Note that this example was calculated for a level 1 coupling facility at service level 4.01 or higher.

a. Gather information about the coupling facility structure.

AVGBUFSIZE AVGBUFSIZE = 4068

System Logger

MAXBUFSIZE AVGBUFSIZE = 65,532

LOGSNUM LOGSNUM = 2

MXSS-to-TSS MXSS-to-TSS = 2

This means that the maximum structure size should be twice the initial structure size.

b. Gather information about each log stream:

- Log stream information for log stream 1:

LOWOFFLOAD LOWOFFLOAD = 0

HIGHOFFLOAD HIGHOFFLOAD = 80

ResidencyTime ResidencyTime = 10

WritePerSec WritePerSec = 5

- Log stream information for log stream 2:

LOWOFFLOAD LOWOFFLOAD = 0

HIGHOFFLOAD HIGHOFFLOAD = 80

ResidencyTime ResidencyTime = 10

WritePerSec WritePerSec = 20

c. Calculate coupling facility structure list entries required per log stream.

For log stream 1:

Entries = $((100 / (\text{HighOffload} - \text{LowOffload})) * \text{WritePerSec} * \text{ResidencyTime}) + 4$
Entries = $((100 / (80 - 0)) * 5 * 10) + 4$
Entries = 67

For log stream 2:

Entries = $((100 / (\text{HighOffload} - \text{LowOffload})) * \text{WritePerSec} * \text{ResidencyTime}) + 4$
Entries = $((100 / (80 - 0)) * 20 * 10) + 4$
Entries = 254

d. Calculate the total coupling facility structure size. Use the following data as input to the TSS and MXSS formulas in the *PR/SM Planning Guide*.

MLSEC

MLSEC = $(254 * 2)$

MLSEC = 508

Note that 254, the **entries** value for log stream 2 is used to calculate MLSEC because it is the larger of the entries values for the two log streams.

MLSELC

Not used.

LELX LELX = 1

MDLES

MDLES = 128

LC LC = 9

LTEC LTEC = 0

LTEX LTEX = 0

R_le R_le = 1

R_data

R_data = 8

Calculate SIZE and INITSIZE parameter values for the structure definition in the CFRM policy from the TSS and MXSS values:

MXSS MXSS = 1472**TSS** TSS = 704

SIZE = MXSS * 4

SIZE = 1472 * 4

SIZE = 5888

INITSIZE = TSS *4

INITSIZE = 704 * 4

INITSIZE = 2816

6. **If you need to change the coupling facility structure size later**, there are two ways you can do that:
- Dynamically alter the coupling facility structure size by entering the SETXCF START,ALTER command or issuing the IXLALTER service.
 - Update the CFRM policy with the new coupling facility size, activate the policy, and then have the operator initiate a rebuild of the structure. See “Appendix B. Administrative Data Utility” on page 263 for updating the CFRM policy using the IXCMIAPU utility.

Note, however, that these options should not be substituted for advance planning, which can optimize both capacity and performance. See the following books for more information:

- *OS/390 MVS System Commands Summary* for the SETXCF command.
- *OS/390 MVS Programming: Sysplex Services Reference* for reference information about the IXLALTER service.
- *OS/390 MVS Programming: Sysplex Services Guide* for guidance on using the IXLALTER service.

Develop a Naming Convention for System Logger Resources

If you are going to have multiple system logger applications, you will need to define sysplex-wide consistent naming conventions for coupling facility structures, DASD log data sets, and DASD staging data sets in order to conveniently track them. These names will subsequently be specified in the LOGR policy.

Naming Conventions for the Log Stream and DASD Data Sets

For each coupling facility and DASD-only log stream you plan, do the following:

1. Determine the log stream name.

A log stream name should be a unique descriptive identifier, made up of one or more qualifiers (each 1 to 8 characters in length) separated by periods, up to the maximum length of 26 characters. **IBM recommends** that the log stream name match the installation’s data set naming conventions, with userid or system logger application name as the first qualifier. This will simplify reporting, defining RACF generic profiles, and writing SMS ACS routines.

It is also recommended that, when possible, one qualifier in the log stream name should be variable, to distinguish between instances used by the application writing to the logstream. To use this naming method effectively, specify a different variable name to define a different instance of a logstream for use by the application. For example, a valid log stream name for one instance of a system logger application named LOG1 might be LOG1.*instance1*.RECOVERY. Another instance of that system logger application

System Logger

may have a log stream named LOG1.*instance2*.RECOVERY. Make sure that each instance of the application has a mapping of which log stream resource name it should be using. The variable names and the log streams they represent need to be provided to the application as part of the application's initialization or setup.

Use the following rules in naming a log stream:

- Each qualifier can contain up to eight numeric, alphabetic, or national (\$, #, or @) characters.
- The first character of each qualifier must be an alphabetic or national character.
- Each qualifier must be separated by periods, which you must count as characters.

2. Determine the high-level qualifier for log stream data sets.

Define a high-level qualifier. The high-level qualifier will be used in the DASD log data sets and DASD staging data sets for each log stream.

For example, depending on the needs of your installation, you could set the high-level qualifier to one of the following:

- Sysplex name
- Subsystem group name
- System logger application type name

System logger does no SAF authorization checking for the high level qualifier you select for the DASD log data sets. This means that while you can select any high level qualifier you like, you should also plan carefully the name you choose. For example, system logger will allow you to choose SYS1 as your high level qualifier, but your DASD log stream data sets will end up in your master catalog! **IBM recommends** that you plan your high level qualifier carefully and add an alias for each high level qualifier in the master catalog that points to a user catalog.

If you do not specify a high level qualifier, system logger uses the default, IXGLOGR.

System logger uses the log stream name to generate the DASD log data sets and staging data sets:

- **To form the log data set name**, system logger combines the log stream name and the high level qualifier, adding an automatically generated sequence number to form the log data set name. This combination of high level qualifier and log stream name must be unique within the sysplex.

The sequence number generated by the system consists of an 8-character string of numbers, letters, or both. You should not try to predict or depend on the format in locating log stream data sets.

The log data set name is the name you use with SMS to manage the log data sets on DASD.

The following is an example of the log data set that is allocated when log stream name LOG1.AFORWARD.RECOVERY is defined without a high level qualifier:

```
IXGLOGR.LOG1.AFORWARD.RECOVERY.A0000001
```

Note that system logger has added the default high level qualifier, IXGLOGR.

The following is an example of a log data set that is allocated when log stream name LOG1.AFORWARD.RECOVERY is defined with a high level qualifier of PROD.

```
PROD.LOG1.AFORWARD.RECOVERY.A0000002
```

- **To form the staging data set name** for a connection, system logger does the following:

For a **coupling facility log stream**, system logger combines the high level qualifier defined for the log stream plus the log stream name and the **system name** derived from the IEASYSxx parmlib member. See *OS/390 MVS Initialization and Tuning Reference* for a description of IEASYSxx.

For example, a staging data set defined on system MVS1 with high level qualifier IXGLOGR and associated with log stream LOG1.AFORWARD.RECOVERY will have the following name:

```
IXGLOGR.LOG1.AFORWARD.RECOVERY.MVS1
```

System logger cannot use a system name that starts with a digit as a qualifier in a staging data set name. Therefore, if your system name begins with a digit, system logger uses 'STG' plus the last 5 significant characters of the system name to form the low level qualifier for the staging data set name. If your system name begins with a digit, you must **make sure** that the last five significant characters will identify the system uniquely within the sysplex. The following examples show how system logger converts a system name starting with a digit to a low level qualifier for a staging data set name:

- System name 1SYS starts with a digit and has less than five characters. System logger converts the digit to 'STG' and adds the four characters to come up with STG1SYS for the low level qualifier for the staging data set.
- System name 1SYSTEM starts with a digit and has more than five characters. System logger converts the digit to 'STG' and uses the last five significant characters to come up with a staging data set low level qualifier of STGSYSTEM.
- Systems 1SYSTEM and 2SYSTEM both start with a digit and have more than 5 characters. Unfortunately, this results in identical low level qualifiers of STGSYSTEM for both system's staging data set names. You may need to change the system name specified on IEASYSxx.

For a **DASD-only log stream**, system logger combines the high level qualifier defined for the log stream, plus the log stream name and the **sysplex name** taken from the COUPLExx parmlib member. See *OS/390 MVS Initialization and Tuning Reference* for a description of COUPLExx.

For example, a staging data set defined on system MVS1 in sysplex PLEX1 with high level qualifier IXGLOGR, and associated with log stream LOG1.AFORWARD,RECOVERY will have the following name:

```
IXGLOGR.LOG1.AFORWARD.RECOVERY.PLEX1
```

System logger cannot use a sysplex name that starts with a digit as a qualifier in a staging data set name. Therefore, if the sysplex name specified in the COUPLExx parmlib member begins with a digit, system logger uses 'STG' plus the last 5 significant characters of the sysplex name to form the low level qualifier for the staging data set name. If your sysplex name begins with a digit, you must **make sure** that the last five significant characters will identify the sysplex uniquely. The following examples show how system logger converts a sysplex name starting with a digit to a low level qualifier for a staging data set name:

- Sysplex name 1PLX starts with a digit and has less than five characters. System logger converts the digit to 'STG' and adds the four characters to come up with STG1PLX for the low level qualifier for the staging data set.

System Logger

- Sysplex name 1SYSPLEX starts with a digit and has more than five characters. System logger converts the digit to 'STG' and uses the last five significant characters to come up with a staging data set low level qualifier of STGSPLEX.
- Sysplexes 1SYSPLEX and 2SYSPLEX both start with a digit and have more than 5 characters. Unfortunately, this results in identical low level qualifiers of STGSPLEX for both systems' staging data set names. You may need to change the sysplex name specified on COUPLExx.

Naming Conventions for System Logger Coupling Facility Structures

The coupling facility structures associated with coupling facility log streams must follow structure naming conventions. The name must be:

- 1 - 8 characters long.
- Uppercase alphabetic characters (A-Z), numeric characters (0-9), national characters (\$,@,#), or an underscore (_).
- Started with an alphabetic character (A-Z).

IBM recommends that you define your coupling facility structure name to match the name of the LPAR in which the coupling facility is to run.

See “Appendix B. Administrative Data Utility” on page 263 for specifying the name in the CFRM policy using the IXCMIAPU utility.

Plan DASD Space for System Logger

Both coupling facility and DASD-only log streams use DASD space for log data sets and staging data sets. All of the planning steps in this section must be performed for either coupling facility or DASD-only log streams, unless otherwise noted.

System logger allocates VSAM linear data sets for the DASD log data sets and DASD staging data sets.

IBM strongly recommends that these DASD data sets be SMS-managed.

Do not use a JCL data set retention period or expiration date to manage log data sets. See “Managing Log Data: How Much? For How Long?” on page 213 for information on managing your log data sets.

This topic includes the following steps for preparing your DASD log data sets and staging data sets:

- “Plan Space for DASD Log Data Sets”
- “Plan Space for Staging Data Sets” on page 208
 - “Plan Staging Data Sets for Coupling Facility Log Streams” on page 208
 - “Plan Staging Data Sets for DASD-Only Log Streams” on page 210
 - “Monitoring Staging Data Set Usage Log Streams” on page 211
- “Set Up the SMS Environment for DASD Data Sets” on page 212
- “Define Share Options for DASD Data Sets” on page 212
- “Add the DASD Data Sets to the GRSRNL Inclusion List” on page 212

Plan Space for DASD Log Data Sets

You can specify the size for DASD log data sets for each log stream on the LS_SIZE parameter in the log stream definition in the LOGR policy.

IBM recommends that you size the data sets as large as your installation can afford to make them. This will minimize the number of log data sets required to represent a log stream. It will also minimize the number of times that system logger must reallocate and switch to using a new log data set when an old one becomes full. Because allocating and switching to a new log data set incurs overhead, it should be done as little as possible.

For logrec log stream and OPERLOG, use your current logrec and SYSLOG output data sets as a guide for log data set size. To calculate the log data set size for logrec, add together the logrec data set sizes for all the systems writing to the logrec log stream. For OPERLOG, add the sizes of the SYSLOG data sets for each system in the sysplex as your log data set size.

By default, each log stream is limited to a maximum of 168 log data sets unless you define data set directory extent records in an OS/390 Release 3 or higher LOGR couple data set and make it the active primary LOGR couple data set. By defining data set directory extent records, a log stream is no longer limited to 168 log data sets.

Increasing the Space Available for DASD Log Data Sets

You can, if necessary, increase the number of log data sets available for log streams using the DSEXTENT parameter in a OS/390 Release 3 or higher LOGR couple data set. If you have log streams that will exceed 168 DASD log data sets, perhaps because you retain data in your log stream for a long period of time, increase the number of directory extents available for the sysplex on the DSEXTENT parameter in a OS/390 Release 3 or higher LOGR couple data set using the format utility (see “Appendix A. Format Utility for Couple Data Sets” on page 241). Each additional directory extent specified goes into a common pool available to any log stream in the sysplex. System logger allocates these directory extents as needed. Each directory extent allows a log stream to extend beyond 168 log data sets. Whenever all the log data sets in a data set directory extent have been physically deleted, system logger returns the directory extent record to the available pool of directory extents for the sysplex. Each log stream has one data set directory extent which is part of the log stream record, not part of the data set directory extent pool. This permanent directory extent is always used first for a log stream, before retrieving a directory extent from the common data set directory pool. This permanent extent does not revert to the common pool when unused and is not available for use by any other log stream - it is held for the owning log stream.

To specify additional extents, format a OS/390 Release 3 level or higher LOGR couple data set with additional directory extents on the DSEXTENT parameter. Then make that LOGR couple data set available as the active primary (see “Format the LOGR Couple Data Set and Make it Available to the Sysplex” on page 216).

If you decide to specify DSEXTENT, specify a moderate number of additional directory extents, at least 15% more than should be in use at any given time, and then monitor usage by:

- Periodically running a LOGR couple data set report using the IXCMIAPU utility. This will show the number of directory extent records in use. Try to keep the number of directory extent records in use below 85% of the total formatted.
- Watching for system messages IXG261E and IXG262A, which indicate that usage of log data set directory extents is over 85% and 95% respectively. Watch also for message IXG301I with a return code of X'08' and a reason code of X'085C'. This indicates that an offload has failed due to a log data set directory full condition. If an offload fails because of a lack of directory extents, system

System Logger

logger will retry the offload after additional directory extents have been made available for the sysplex. When the offload completes and the log stream is again available, system logger issues an ENF 48 signal.

If you are not using DSEXTENT in the LOGR couple data set, the data set directory limit for a log stream is 168 data sets. (Note that your DASD staging data sets **do not** count toward the 168 limit.) If you reach the data set directory limit for a log stream, system logger will not be able to offload data.

You may be able to make additional directory space available for a log stream by deleting log data from the log stream. (See “Deleting Log Data and Log Data Sets” on page 227.) Whenever the number of log data sets in use gets above 90% of the total number allowed (168), the system logger issues message, IXG257I. This message will be deleted when either the number of log data sets for the log stream drops below 85% of the total allowed or the last connection to the log stream in the sysplex disconnects. In the latter case, the message will be reissued if an application reconnects to the log stream.

Whether or not you use DSEXTENT to increase the number of log data set directory extents for a log stream, you can also set up a retention period and automatic deletion policy for a log stream to ensure that your log streams will not run out of directory space. See “Managing Log Data: How Much? For How Long?” on page 213.

Plan Space for Staging Data Sets

DASD staging data sets are used for duplexing log data. The way you plan for staging data sets depends on whether you are planning for coupling facility or DASD-only log streams. See:

- “Plan Staging Data Sets for Coupling Facility Log Streams”
- “Plan Staging Data Sets for DASD-Only Log Streams” on page 210
- “Monitoring Staging Data Set Usage Log Streams” on page 211

Plan Staging Data Sets for Coupling Facility Log Streams

For coupling facility log streams, staging data sets are optional. However, **IBM recommends** that you plan for staging data sets for each system even if your installation does not intend to use them for duplexing and does not define duplexing to staging data sets in the log stream definition. This is recommended because system logger might allocate temporary staging data sets to safeguard log data when there is an error that leaves the only copy of log data in a volatile configuration. Specifically, system logger automatically allocates a staging data set when a structure failure followed by a rebuild failure leaves the only copy of log data for a log stream in local storage buffers. System logger automatically allocates staging data sets for each system not already using them under these conditions, even if you do not specify duplexing to staging data sets. System logger automatically allocates temporary staging data sets during rebuild processing only for installations with a primary active LOGR couple data set formatted at OS/390 Release 3 or higher. System logger does the following to allocate staging data sets during rebuild processing:

- Allocates a staging data set on each system.
- Copies the log data into the staging data set.
- Disconnects user connections to the log stream and places the system connection to the log stream into a failed state.

For installations using a pre-OS/390 Release 3 LOGR couple data set, system logger will issue error messages and wait for the operator to resolve the problem and re-initiate rebuild.

For coupling facility log streams, **IBM recommends** that all the staging data sets reside on devices that all systems in the sysplex have connectivity to through channel paths. This is important because other systems might need to access the backed up log data in case of system or coupling facility failure. If peer systems do not have connectivity to staging data sets, system logger might not be able to recover all data in case of a failure.

Sizing Staging Data Sets for Coupling Facility Log Streams: To specify a size for staging data sets for a coupling facility log stream, you can either take the system logger defined default size, or define your own size on the STG_SIZE parameter in the log stream definition in the LOGR policy (for this log stream or the one specified on the LIKE parameter).

Note that whether you take the default staging data size or define your own, the definition applies for all staging data sets for a log stream.

Staging data sets on a system are deleted when the last connector to the log stream for that system disconnects.

- **Using the system logger default staging data set size:**

If you do not define a staging data set size on the STG_SIZE parameter in the LOGR policy, system logger determines the default staging data set for a system from the maximum coupling facility structure size allocated for a log stream. This value is defined on the SIZE parameter in the CFRM policy couple data set. This makes a default staging data set as large as if each system was actually going to write the amount of data estimated for the entire log stream.

- **Defining your own staging data set size:**

To define your own staging data set size for a log stream, specify it on the STG_SIZE parameter in the log stream definition in the LOGR policy. The value specified on STG_SIZE overrides the system logger default staging data set size. Use the following guidelines to determine the staging data set size for a log stream:

- For log streams that are used by only one system, size the staging data set to be large enough to hold all the log data written by the system logger application. Use the maximum coupling facility structure space allocated for the log stream, specified on the SIZE parameter in the CFRM policy couple data set.
- For log streams written to by multiple systems, **IBM still recommends** that you play it safe by sizing the staging data sets for the log stream the same size as the coupling facility space allocated for that log stream on the SIZE parameter in the CFRM policy couple data set.

If you need to save DASD consumption by sizing your staging data sets smaller than the recommended size, make sure you at least specify a size large enough for the system writing the most data. For example, if one system writes about 40% of the data to the log stream, while another writes 60%, you must specify a staging data set size for the log stream that will accommodate **at least** 60% of the data. To make this calculation and depend on it, you must know exactly how much data each system writes to the log stream.

Remember also, that if your staging data set is too small, you might find that offloading, with its overhead cost, is kicked off by the high offload threshold in

System Logger

the staging data set, rather than the coupling facility threshold. See “Staging Data Set Size and Offloading” on page 183.

Change the Staging Data Set Size for Coupling Facility Log Streams: If you need to change the staging data set size for a log stream because of a change in conditions or in response to monitoring SMF record 88, note that the staging data sets must actually be deleted and then reallocated in order to change. This is because the staging data set size and other log stream attributes are defined at allocation and remain in effect as follows:

- For a log stream defined as STG_DUPLEX(YES) and DUPLEXMODE(UNCOND), until the last connection to the log stream disconnects.
- For a log stream defined with STG_DUPLEX(YES) and DUPLEXMODE(COND), while the configuration contains a single point of failure or until the last connection to the log stream disconnects.

To change staging data set size, do the following:

1. Depending on how the log stream is defined, do one of the following:
 - For a log stream defined with STG_DUPLEX(YES) and MODE(UNCOND), have all instances of the application on all systems disconnect from the log stream.
 - For a log stream defined with STG_DUPLEX(YES) and MODE(COND), either issue a structure rebuild so that the configuration no longer contains a single point of failure or else have all instances of the application on all systems disconnect from the log stream.
2. Change the STG_SIZE parameter in the LOGR policy either to a new value or the default, to take the system logger defined default.
3. Have all systems reconnect to the log stream.

For example, to change the staging data set size for logrec log streams, you would:

1. Issue the SETLOGRC DATASET command to disconnect logrec from the log stream and record logrec records in a data set.
2. Update the STG_SIZE value in the LOGR policy using the IXGINVNT service or the IXCMIAPU utility.
3. Issue the SETLOGRC LOGSTREAM command to reconnect to and reactivate the logrec log stream.

If the update of the staging data set size does not work, it might be because there are more connectors to the log stream that have not been disconnected. Use the LIST request on the IXCMIAPU utility to see if there are still connectors. Make sure that all log stream connectors have disconnected, by canceling the job if necessary, before changing the staging data set size in the LOGR policy.

Plan Staging Data Sets for DASD-Only Log Streams

For DASD-only log streams, staging data sets are a required part of the system logger configuration. System logger automatically duplexes data to the staging data set for the system at the same time it writes the data to local storage buffers. **IBM recommends** that you plan for staging data sets for each system, using SMS to manage them. If you do not plan for the staging data sets using SMS, system logger will automatically allocate staging data sets for a system connecting to a DASD-only log stream, using the defaults outlined below.

IBM recommends that all the staging data sets reside on devices that all systems in the sysplex have connectivity to through channel paths, if the application intends

to have more than one system connect to the DASD-only log stream (in sequence, not simultaneously). If the application intends to have only one system ever connect to a staging data sets, this connectivity is not necessary.

Sizing the Staging Data Sets for DASD-Only Log Streams: To size data sets for a DASD-only log stream, you can either take the system logger defined default size or define your own size on the STG_SIZE parameter in the log stream definition in the LOGR policy.

To determine the default staging data set size for a DASD-only log stream, system logger does one of the following, in the order listed:

- Uses the STG_SIZE of the log stream specified on the LIKE parameter, if specified.
- Uses the size defined in the SMS data class for the staging data sets.
- Uses dynamic allocation rules for allocating data sets, if SMS is not available.

Note that if both the STG_DATACLAS and STG_SIZE parameters are specified for a log stream, the value for STG_SIZE overrides the STG_DATACLAS value.

Changing the Staging Data Set Size for a DASD-Only Log Stream: If you need to change the staging data set size for a DASD-only log stream because of a change in conditions or in response to monitoring SMF record 88, note that the staging data set must actually be deleted and then reallocated in order to change. This is because the staging data set size and other log stream attributes are defined at allocation and remain in effect until all the connections to the DASD-only log stream have disconnected.

To change staging data set size, do the following:

1. Have all connections to the DASD-only log stream disconnect.
2. Change the STG_SIZE parameter in the LOGR policy to a new value.
3. Have all connections from the system reconnect to the log stream.

If the update of the staging data set size does not work, it might be because the system is still connected to the log stream. Use the LIST request on the IXCMIAPU utility to see if the system is still connected. Make sure that the system is disconnected, by canceling the job if necessary, before changing the staging data set size in the LOGR policy.

Monitoring Staging Data Set Usage Log Streams

Whether your staging data sets are defined by system logger or on the STG_SIZE parameter, you should carefully monitor your staging data sets. This applies to both coupling facility and DASD-only log streams, and is important because the consequences of having your staging data set fill up can be quite disruptive. When a system's staging data set fills up, system logger applications on that system will not be able to write to the log stream until log data can be offloaded to DASD, which frees up space in the staging data set. Thus, when your staging data sets are too small, system logger will perform coupling facility offloading more frequently than the **HIGHOFFLOAD** and **LOWOFFLOAD** thresholds defined for the log stream would otherwise require. This can negatively affect the performance of all the log streams in that structure.

Use SMF record 88 to monitor staging data set usage:

- Check field SMF88ETT to see whether the high threshold mark for a staging data set is being reached. If you find that the high threshold mark is never reached, your staging data set might be larger than necessary.

System Logger

- Check field SMF88ETF to see if you are getting one or more staging data set full conditions. This indicates that your staging data set might be too small and should probably be enlarged.

Set Up the SMS Environment for DASD Data Sets

System logger requires that you have SMS installed and its address space active at your installation on each system where system logger is expected to run. This is true even if you do not use SMS to manage your volumes and data sets. SMS must be active because system logger uses VSAM linear data sets. If the SMS address space is not active when you attempt to use a system logger application, system logger issues system messages that indicate allocation errors and the application will not be able to use the logger function.

Set up the SMS environment to manage DASD log data sets and DASD staging data sets. See *OS/390 DFSMSdfp Storage Administration Reference* for information on creating the following SMS classes:

1. Storage groups identifying the volumes to be used for data set allocation.
2. Storage classes to describe the performance characteristics for the data sets.
3. Data classes to specify the data set characteristics and size of the data sets.
4. Management classes to specify the migration characteristics of the data sets. The data sets might be migrated to hierarchical storage manager (HSM) or archived.
5. Automatic class selection (ACS) routines to assign the storage groups, classes, data classes and management classes, based on data set name when the DASD data sets are allocated.

You can specify the SMS characteristics for log and staging data sets in a number of ways, depending on how an installation is set up. For example, there might be installation written automatic class selection (ACS) routines to assign classes for DASD data sets, or you can specify the SMS characteristics for DASD log and staging data sets in the LOGR policy. To define the SMS characteristics in the LOGR policy, see “Specifying SMS Data Set Characteristics for DASD Data Sets” on page 219.

Define Share Options for DASD Data Sets: To define VSAM share options for a log stream’s DASD log data sets and staging data sets, set up a data class and associate it with the DASD log data sets and staging data sets in the LOGR policy. Make this association when you define or update the LOGR policy with the Administrative Data Utility (IXCMIAPU). Use the LS_DATACLAS keyword to make the association. If your sysplex has multiple systems, you must use VSAM SHAREOPTIONS(3,3) so that all the systems in the sysplex can share the data sets. See “VSAM Share Options for System Logger” on page 190 for more information.

Add the DASD Data Sets to the GRSRNL Inclusion List

Add the DASD log data sets and staging data sets to the GRSRNL inclusion list:

1. Update the GRSRNLxx parmlib member.

Add the following RNLDEF statement in the GRSRNL inclusion list to cover all log stream data sets allocated on behalf of log streams that are defined with the default high level qualifier:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(IXGLOGR)
```

IBM recommends that you add other RNLDEF statements in the GRSRNL inclusion list to cover log streams that are defined with a high level qualifier other than IXGLOGR:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(h1q)
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(h1q.l$name)
```

h1q

Specifies a high-level qualifier associated with a log stream or a staging data set. IBM supplies two log streams for customer use, the logrec log stream and the operations log (OPERLOG) log stream. You can create your own log streams and your own staging data sets. Each DASD data set that represents a log stream or staging data set begins with the specified high-level qualifier. The high-level qualifier can be specified when you define a log stream or staging data set to the LOGR policy.

l\$name

Specifies the portion of the log stream name that limits the scope of the RNLDEF statement to the system logger-created data sets.

See *OS/390 MVS Planning: Global Resource Serialization* for more information.

2. **Enter the following command from any system in the sysplex to activate the parmlib updates:**

```
SET GRSRNL=(xx,xx,...)
```

Managing Log Data: How Much? For How Long?

For installations with an active primary LOGR couple data set at an OS/390 Release 3 level or higher, system logger provides support to make it easier to archive log data and manage the amount of data kept in a log stream. This applies to both coupling facility and DASD-only log streams. You can define a retention period and automatic deletion policy for each log stream. The retention period and automatic deletion policy are specified on the RETPD and AUTODELETE parameters in an OS/390 Release 3 or higher LOGR couple data set to help manage:

- How much data you keep in a log stream.
- How long you keep data in the log stream.

The RETPD and AUTODELETE parameters are used together to set up a retention period and automatic deletion for a log stream. You can specify the RETPD and AUTODELETE parameters in an OS/390 Release 3 level or higher LOGR couple data set using either the IXCMIAPU utility or the IXGINVNT service in a program. See “Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy” on page 217.

The RETPD parameter allows you to specify a retention period for a log stream. On RETPD, you specify the number of days that you want to keep data in the log stream, even if the data has been marked for deletion using IXGDELETE. For example, if you specify RETPD(7) in the LOGR policy for a log stream, the retention period for data in that log stream is 7 days from the time the data is written to the log stream by the application. System logger processes the retention period on a log data set basis. Once the retention period for the entire log data set has expired, the data set is eligible for deletion. System logger may not physically delete data as soon as it hits the retention period. The point at which system logger physically deletes the data depends on when the data set fills and what you specify on the AUTODELETE parameter.

Note that this retention period is different from a data set retention period on a JCL DD statement; a system logger retention period applies to the age of the log data, not the data set.

The **AUTODELETE** parameter lets you specify:

- **AUTODELETE(NO)** to indicate that you do not want an automatic deletion policy for a log stream. **AUTODELETE(NO)** means that the log data can be physically deleted only after the log data has been marked for deletion via **IXGDELET** and after any retention period specified for the log stream has expired. **AUTODELETE(NO)** is the default.
- **AUTODELETE(YES)** to indicate that you want an automatic deletion policy to limit how long data is kept in the log stream. **AUTODELETE(YES)** means that log data can be physically deleted **either** when the data is marked for deletion or when a retention period specified for the log stream expires. Use care when specifying **AUTODELETE(YES)**. Automatic deletion is designed to speed physical deletion of log data, which can mean deletion of data that an application needs.

Choosing the Right Retention Period and Automatic Deletion Policy For a Log Stream

The way you use **RETPD** and **AUTODELETE** together to set up a retention period and an automatic deletion policy for a log stream depends on the data requirements of each log stream.

Example 1. Delete Data After No More Than 3 Days, No Matter What: For audit log streams, such as the **OPERLOG** log stream, the installation must often manage how much data is kept in the log stream. Using automatic deletion and a retention period, you can manage the amount of time log data resides in the log stream, cutting down also on the storage consumed by log data in this log stream. For example, let's say that log **AUDIT1** requires that data be kept for no more than 3 days and then physically deleted. For **AUDIT1**, you would specify **RETPD(3)** and **AUTODELETE(YES)** to make sure that data is kept for no more than 3 days.

Example 2. Keep Data for At Least 30 Days, Even If It Was Marked for Deletion: Installations may require that their transaction manager log streams retain data for a certain length of time to meet audit requirements. This data must be kept for the retention period, even when it has been marked for deletion via the **IXGDELET** service. Using automatic deletion and a retention period, you can archive the data for the required length of time without having to move the data out of the log stream, and without having the archived data interfere with the transaction manager. For example, log **TRANS1** needs to have data in the log stream for at least 30 days, even if it has been marked for deletion. You can specify **RETPD(30)** **AUTODELETE(NO)** to keep the data for the required amount of time. Even if data is marked for deletion via **IXGDELET** within the 30 day retention period, it is kept in the log stream and can be accessed by a system logger application using the **IXGBRWSE** service with **VIEW=ALL**.

Example 3. Keep Data in the Log Stream Until Deletion: Some log streams may not need to keep log data around after it has been marked for deletion via **IXGDELET** and do not require that data be retained for a particular length of time. For this kind of a log stream, the default settings of **RETPD(0)** and **AUTODELETE(NO)** should be just fine. System logger will consider data eligible for physical deletion any time after it has been marked for deletion via **IXGDELET**. These settings give you the same function as pre-OS/390 Release 3 processing.

Example 4. Don't Keep My Log Data At All: If you specify **RETPD(0)** and **AUTODELETE(YES)**, data is eligible for deletion as soon as you write it to the log stream. This setting is dangerous, because data can be physically deleted whether it is marked for deletion or not.

When Is Data Physically Deleted If I Specify RETPD and AUTODELETE?

If you specify the RETPD and AUTODELETE parameters to set a retention period and automatic deletion policy for a log stream, system logger deletes the data on a log data set basis. System logger physically deletes the entire log data set after all the data in the log data set has expired.

- The retention period has expired for **all** the log data in the log data set. (If you specify or default to RETPD(0), this condition has already been met.)
- The log data set is full and is no longer in use as the current log data set for the log stream.

Using a Resource Manager and Auto Deletion Together

If you have specified AUTODELETE(YES) for a log stream, and you also manage that log stream with a resource manager and a resource manager exit, note that automatic deletion processing takes precedence over the delete override processing performed by the resource manager exit. This means that log data can be deleted even when a delete request is rejected or overridden by the resource manager user exit. In general, IBM recommends AUTODELETE(NO), especially for log streams that are backed up by a resource manager.

Define Authorization to System Logger Resources

IBM recommends that installations use Security Authorization Facility (SAF) to control access to system logger resources, such as log streams or coupling facility structures associated with log streams. The following examples assume the use of OS/390 Security Server (RACF) or an equivalent security product.

Installations must define authorization to system logger resources for the following:

- System logger address space
- Userid setting up the CFRM and LOGR policies
- System logger applications (for example, logrec or OPERLOG) that will be accessing the log streams.

These steps apply to both coupling facility and DASD-only log streams.

For information about the authorization an application needs to issue system logger services, see *OS/390 MVS Programming: Assembler Services Guide*.

Define Authorization for the System Logger Address Space

IBM recommends that the system logger address space (IXGLOGR) be assigned privileged and/or trusted RACF status.

If the system logger address space for your installation is neither privileged nor trusted, make sure to give IXGLOGR the following SAF authorizations:

- Define IXGLOGR in either RACF started procedures table (SPT) or in RACF CLASS(STARTED).
- For coupling facility log streams, define alter access to RESOURCE(IXLSTR.*structure_name*) CLASS(FACILITY) for access to the log stream coupling facility structures.
- Define alter access to RESOURCE(*hlq.data_set_name*) CLASS (DATASET) for each DASD log stream and staging data set.
- Define read access to RESOURCE(*sys1.parmlib_data_set_name*) CLASS (DATASET) for access to SYS1.PARMLIB.

System Logger

Define Authorization for Setting Up Policies

The LOGR policy information describes the characteristics of each log stream and coupling facility structure that will be used when there are active connections to the log stream.

The CFRM policy contains the definition of all coupling facility structures used by system logger, if applicable.

Before setting up these policies for your installation with the IXCMIAPU Administrative Data Utility program, authorize who can use IXCMIAPU.

- Define a resource profile for the resource name MVSADMIN.LOGR to the FACILITY class. Assign ALTER access authority to users who must alter or maintain the policy; assign READ access authority to users who require reports on the policy, but who will not change the policy.
- If applicable, define a resource profile for the resource name MVSADMIN.XCF.CFRM to the FACILITY class. Assign UPDATE access to users who must define coupling facility structures in the policy.

The users of the IXCMIAPU utility require authorization to the resources accessed by the utility. Define the appropriate authorizations before users attempt to add, update, delete, or extract a report from the LOGR policy. The LOGR couple data set must be activated and available before you add log streams and structure definitions to it. The following authorizations are required to set up the LOGR policy:

- To DEFINE, UPDATE, or DELETE LOGSTREAM, specify ALTER access authority to RESOURCE(*log_stream_name*) CLASS(LOGSTRM).
- To DEFINE or DELETE STRUCTURE, specify ALTER access authority to RESOURCE(*IXLSTR.structurename*) CLASS(FACILITY).
- To specify a structure name on a LOGSTREAM definition (for example,
DEFINE LOGSTREAM(*log_stream_name*) STRUCTNAME(*structname*))

specify UPDATE access authority to RESOURCE(*IXLSTR.structurename*) CLASS(FACILITY).

Authorization for System Logger Applications

For an installation or vendor-written system logger application, the access to system logger resources, such as log streams or coupling facility structures associated with log streams, depends on which system logger services the application issues.

- For a vendor-written application, see the documentation provided by the vendor.
- For an installation-written application, see the application programmer for authorization requirements of the program.

Information on system logger services is provided in *OS/390 MVS Programming: Assembler Services Guide*.

For logrec log stream, CICS log manager, and OPERLOG system logger application you must do the following:

- For the application, define UPDATE access to SAF resource RESOURCE(*log_stream_name*) CLASS(LOGSTRM).

Format the LOGR Couple Data Set and Make it Available to the Sysplex

Do the following for both coupling facility and DASD-only log streams to format a primary and alternate LOGR couple data set and define them to the sysplex:

1. **Format the primary and alternate LOGR couple data set using the IXCL1DSU utility.**

Format a primary and alternate LOGR couple data set using the IXCL1DSU utility, but make sure they are on different volumes.

You must provide as input the maximum number of log streams and structure definitions you plan to define. For DASD-only log streams, you will not have structures to define, so do not increase the number of structures for DASD-only log streams. These values were calculated in “Plan the System Logger Configuration” on page 191.

For complete information and examples of using the IXCL1DSU utility, see “Appendix A. Format Utility for Couple Data Sets” on page 241.

2. **Update the COUPLExx member in SYS1.PARMLIB to identify the LOGR couple data set to the sysplex.**

Add your primary and alternate LOGR couple data set names to the COUPLExx parmlib member. See the following for more information:

- “Defining Policy Couple Data Sets to MVS” on page 34 for an example of defining the LOGR couple data set in the COUPLExx parmlib member.
- *OS/390 MVS Initialization and Tuning Reference* for complete information on the COUPLExx parmlib member.

3. **Make the LOGR couple data set available.**

You can use either of the following ways to make the primary and alternate couple data sets available to the sysplex:

- IPL the system with the primary and alternate LOGR couple data sets defined in the COUPLExx parmlib member.
- Issue the following SETXCF commands when you want to bring the LOGR couple data sets online without IPLing the system as follows:

```
SETXCF COUPLE,TYPE=LOGR,PCOUPLE=(primary_couple_data_set)
SETXCF COUPLE,TYPE=LOGR,ACOUPLE=(alternate_couple_data_set)
```

Either of these methods causes the empty LOGR couple data set to become available on all systems where system logger is installed. After the SETXCF command is entered, you can define structures and log streams in the LOGR policy.

Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy

Define each log stream and coupling facility structure in the LOGR policy using the IXCMIAPU utility. Note that each coupling facility structure that you define in the LOGR policy must also be defined in the current CFRM policy for the sysplex. Structure names in the LOGR policy **must match** those defined in the CFRM policy.

See “Appendix B. Administrative Data Utility” on page 263 for information on using the IXCMIAPU utility. Use the guidance included throughout this planning section to help you determine what parameters to specify in the LOGR policy. Guidance for SMS parameters and selecting a method of duplexing log data for your configuration are covered below.

Specifying Whether the Log Stream is DASD-Only

This step applies to both the DASD-only and coupling facility log streams.

System Logger

For each log stream, you must decide whether to define it as DASD-only or coupling facility (see “Determine the Right Kind of Log Stream for Each Application” on page 191). Use the DASDONLY parameter on a DEFINE LOGSTREAM request as follows:

To define a log stream as a coupling facility log stream, specify or default to DASDONLY(NO) on the log stream definition. You must also specify a structure name on the STRUCTNAME parameter in the definition for a coupling facility log stream.

To define a log stream as DASD-only, specify DASDONLY(YES) on the definition for a log stream that is not associated with a coupling facility structure. When you define a DASD-only log stream, you can also specify MAXBUFSIZE on the DEFINE LOGSTREAM request to define the maximum buffer size that can be written to the DASD-only log stream.

Specifying the Average Log Block Size

This step applies only to coupling facility log streams, DASD-only log streams may omit this step.

For each coupling facility structure associated with a coupling facility log stream or streams, use the AVGBUFSIZE parameter to specify the average log block size that system logger applications will write to log streams associated with the structure. System logger uses AVGBUFSIZE to calculate the initial entry-to-element ratio for a coupling facility structure for a log stream.

For installations using a LOGR primary couple data set formatted at the OS/390 Release 3 level or above, system logger only uses the AVGBUFSIZE specified initially to set the entry-to-element ratio for the structure. After that, system logger will automatically manage the entry-to-element ratio for a structure dynamically, based on actual structure usage.

To find the current entry-to-element ratio system logger is using for a structure associated with a log stream, use the REPORT or LIST parameters in the IXCMIAPU utility. See “Appendix B. Administrative Data Utility” on page 263.

If you would like to monitor the changes to the entry-to-element ratio system logger is maintaining, use the aggregate structure information in SMF record type 88. Using this record, you can:

- Monitor the number of entry-to-element ratio changes.
- See the time of the last entry-to-element change.
- See the allocated entry-to-element ratio before the last change.
- See the new entry-to-element target ratio after the last change.

See *OS/390 MVS System Management Facilities (SMF)* for information on SMF record type 88.

For installations using a pre-OS/390 Release 3 level LOGR couple data set, the value you select for AVGBUFSIZE determines the entry-to-element ratio for the life of the coupling facility structure. Once you have defined the AVGBUFSIZE for a coupling facility structure, you will not be able to update the value. Select a value that represents the actual average size of log blocks that will be written to this structure.

For a logrec log stream, **IBM recommends** an AVGBUFSIZE of 4068.

For an OPERLOG log stream, **IBM recommends** an AVGBUFSIZE of 512.

If you select a value for AVGBUFSIZE that is different from the actual average block size, structure space will be wasted. If you run out of structure entries for a structure, a structure full condition will result, even if you have many unused elements in the structure. This disruptive condition will cause frequent offload processing for the structure.

The only way to change the AVGBUFSIZE for a structure in an installation using a pre-OS/390 Release 3 LOGR couple data set is to delete the log streams associated with the structure, delete the structure, and then re-define the structure with the new AVGBUFSIZE value.

If you are using a pre-OS/390 Release 3 LOGR couple data set, you cannot use the IXLALTER service to change the entry-to-element ratio. If you are using an OS/390 Release 3 level or higher LOGR couple data set, **IBM recommends** that you do not try to manage the entry-to-element ratio for system logger coupling facility structures using the IXLALTER service. System logger will manage the ratio for optimal structure usage; using IXLALTER will just conflict with system logger's entry-to-element ratio.

For more information on the directory-to-entry ratio see the connection services topic in *OS/390 MVS Programming: Sysplex Services Guide*.

Specifying SMS Data Set Characteristics for DASD Data Sets

When you define or update a log stream definition in the LOGR policy, you can assign the SMS storage class, data class, and management class for both the DASD log data sets and staging data sets. For example, you can specify parameters STG_DATACLAS and LS_DATACLAS in the LOGR policy to assign a data class with the correct VSAM share options. These parameters intersect with other parameters and the system environment in ways that might produce unpredictable results if you do not plan well. So think carefully before specifying the SMS parameters in the LOGR policy. For instance:

- **SMS parameters and ACS routines:**

If you use installation-written automatic class selection (ACS) routines to assign classes for these DASD data sets, the ACS class might override SMS classes you explicitly specify in the LOGR policy using the IXCMIAPU utility. Make sure you coordinate your ACS routines and the SMS characteristics you specify in the LOGR policy to avoid surprises.

- **Using SMS parameters and the LIKE parameter together:**

If you are using the LIKE parameter for a log stream, note that any explicit definitions for the log stream you are defining override the definitions of the log stream referenced on the LIKE parameter. System logger **will not use** an SMS class from a log stream referenced on a LIKE parameter if there is any value specified for that class on an SMS parameter.

- **Optimizing Control Interval Size:**

Increasing the size of the control interval for DASD log data sets can provide performance improvements in the areas of record processing speed and storage requirements. You can modify the control interval size attribute of DFSMS Data Classes defined for DASD log data sets to gain performance improvements in I/O operations associated with the offloading of log stream data as well as during browse processing when retrieving log stream data from DASD log data sets.

A VSAM linear data set requires a control interval size of from 4096 to 32768 bytes, and can be expressed in increments of 4096 bytes. The VSAM default is

System Logger

4096 bytes. System logger is designed to obtain optimal I/O performance when DASD log data sets reside on devices with 3390 formats. IBM recommends a control interval size of 24576 bytes to realize optimal I/O performance if system logger DASD log data sets reside on devices with 3390 formats. Specify a DFSMS data class that is defined with a control interval size of 24576 on the LS_DATACLAS parameter of a log stream definition to have DASD log data sets allocated with control interval sizes of 24576.

Note: Do not change the Control Interval Size attributes for staging data sets to be anything other than 4096. System logger requires that the Control Interval Size for staging data sets be set to 4096. If staging data sets are defined with characteristics that result in a Control Interval Size other than 4096, system logger will not use staging data sets to keep a duplicate copy of log data. Operations involving staging data sets defined with a Control Interval Size other than 4096 will fail to complete successfully.

If you intend to use control interval sizes other than 4096, you may need to modify the STG_DATACLAS specifications or change your DFSMS ACS routine. This action may be required regardless of whether or not you specify STG_DUPLEX (YES) for your log streams. If you explicitly provide a STG_DATACLAS specification for the log stream, ensure the DFSMS data class uses a control interval size of 4096. If you use an ACS routine to have attributes assigned to any log stream data sets, you must ensure the staging data sets do not use a data class that has a control interval size other than 4096. For example, you may need to change the ACS routine to filter based on the data set name up to and including the last qualifier in order to recognize staging data sets.

If your IXCMIAPU request does not complete successfully, you might get return and reason codes returned in the output. These return and reason codes are documented along with IXGINVNT in *OS/390 MVS Programming: Assembler Services Reference*.

Before using the IXCMIAPU utility, you will need to define the SAF authorizations for system logger applications. See “Authorization for System Logger Applications” on page 216.

Selecting a Method of Duplexing Coupling Facility Log Data

This step applies to coupling facility log streams only. For DASD-only log streams, duplexing to staging data sets is automatic.

Using the STG_DUPLEX and DUPLEXMODE parameters in the LOGR policy, you can request how you want system logger to duplex coupling facility resident log data for a log stream. You can duplex coupling facility data in either local storage buffers for each system or in DASD staging data sets.

The method you choose depends on whether your connection contains a single point of failure and whether your log stream can tolerate a loss of data.

What is a Single Point of Failure?: A **single point of failure** is an environment where one failure can result in the simultaneous loss of both the coupling facility list structure for a log stream and the local storage buffer copy of the data on the system making the connection. If you were using local storage buffers to duplex data, this can result in loss of log data. When your configuration contains a single point of failure, you can safeguard your data by duplexing data to staging data sets.

When a configuration is not vulnerable to a single point of failure, it is **failure independent**.

System logger evaluates a connection for a single point of failure based on the location of the coupling facility and its status of volatile or non-volatile. The coupling facility can reside in one of the following configurations:

- The coupling facility executes in an LPAR, sharing a central processing complex (CPC) with a system that is connected to a log stream associated with the coupling facility.
- The coupling facility is in a stand-alone CPC and does not share the CPC with a system with a connection to the log stream.
- The coupling facility executes as an Integrated Coupling Migration Facility (ICMF) partition on a CPC. Only connections from a system on the same CPC can communicate with this coupling facility.

System logger uses the following rules for determining whether a log stream connection contains a single point of failure:

- **When a coupling facility shares a CPC** with a system that connects to log streams mapping to the same coupling facility, each active connection between the system and the coupling facility contains a single point of failure, regardless of the volatility state of the coupling facility. The connections have a single point of failure because if the CPC should fail, coupling facility resident log data will be lost when both the system, with local storage buffers, and the coupling facility fail with the CPC.
- **When a coupling facility is separate** from the system connected to the associated log stream, the volatility status of the coupling facility determines whether or not the connection is vulnerable to a single point of failure:
 - If the coupling facility is non-volatile, the connection is failure independent.
 - If the coupling facility is volatile, the connection is vulnerable to a single point of failure. A power failure could knock out both the system, with its local storage buffer copy of coupling facility data, and the coupling facility.

A connection is failure independent when it is from a system on a separate CPC from the non-volatile coupling facility it is connected to.

If a connection contains a single point of failure and `STG_DUPLEX(YES)` `DUPLEXMODE(COND)` has been specified in the log stream definition, system logger will duplex coupling facility log data to staging data sets for that connection. In the event of a coupling facility or CPC failure, system logger can retrieve lost coupling facility log data from the staging data sets for a connection with a single point of failure.

Using the *STG_DUPLEX* and *DUPLEXMODE* Parameters: The `STG_DUPLEX` and `DUPLEXMODE` parameters are as follows:

- `STG_DUPLEX(NO)`, which is the default, specifies that system logger is to maintain a copy of coupling facility data in local storage buffers on each system. Choose this option when your application is not vulnerable to data loss that might occur if your environment contains a single point of failure.
Note that in certain rebuild error cases, system logger will automatically allocate a staging data set, even when you specify `STG_DUPLEX(NO)`. See “Plan Space for Staging Data Sets” on page 208 for more information.
- `STG_DUPLEX(YES)` specifies that you want system logger to maintain a copy of coupling facility data in DASD staging data sets, one per log stream per system.

System Logger

You can only specify DUPLEXMODE with STG_DUPLEX(YES). Choose one of the following:

- DUPLEXMODE(COND), which is the default, specifies that system logger is to conditionally duplex coupling facility log data for a system **ONLY** when a system's connection to the log stream contains a single point of failure and is therefore vulnerable to loss of data.

With DUPLEXMODE(COND), note that different systems connected to a log stream might be backing up data differently. Duplexing is done on a connection basis, depending on whether a system's connection contains a single point of failure.

If you have specified STG_DUPLEX(YES) DUPLEXMODE(COND), system logger can switch duplexing methods when the system logger configuration changes. For example, if you specify conditional duplexing for a log stream that writes to a coupling facility that changes from volatile to non-volatile, system logger will stop duplexing the data to staging data sets and duplex the coupling facility data to local storage buffers, because there is no longer a single point of failure in the volatile coupling facility.

If the coupling facility changes from non-volatile to volatile, the coupling facility structure is rebuilt. When the rebuild completes, system logger re-evaluates each connection for a single point of failure and whether to use staging data sets.

Use STG_DUPLEX(YES) DUPLEXMODE=(COND) when you cannot afford any data loss. It is a flexible way to protect your application against a single point of failure on a connection by connection basis. With DUPLEXMODE=COND, duplexing to staging data sets is only used when needed for a particular connection. Connections that do not have a single point of failure can exploit the performance benefits of backing up log data to local storage buffers when duplexing is not necessary.

- DUPLEXMODE(UNCOND) specifies that system logger is to unconditionally duplex coupling facility data for log stream connectors to DASD staging data sets, regardless of whether the connection contains a single point of failure. Use this option when you always want to use staging data sets, regardless of whether a connection has a single point of failure or not.

Example - Using Staging Data Sets When the Coupling Facility Shares a CPC with a System: In the following scenario, a log stream has been defined with STG_DUPLEX=YES DUPLEXMODE=COND. There are two systems connected to the log stream. System 1, SYS1, shares the CPC with the coupling facility, so the data that SYS1 writes to the coupling facility is duplexed to a staging data set because there is a single point of failure between the system and the coupling facility. For example, a machine error might require a reset of CPC1, which would result in the loss of both the coupling facility and SYS1 with its local storage buffer copy of the coupling facility log data if it was not duplexed to staging data sets.

System 2, SYS2, is in a different CPC from the coupling facility it writes to, so the backup copy of coupling facility log data for that system connection is maintained in a local storage buffer within SYS2. If SYS2 should fail, its data is still in the coupling facility list structure.

If the coupling facility fails, SYS2 still has the coupling facility log data for this connection in the local storage buffer. The local storage buffer copy of the coupling facility data can be used to recreate the log stream in another coupling facility list structure.

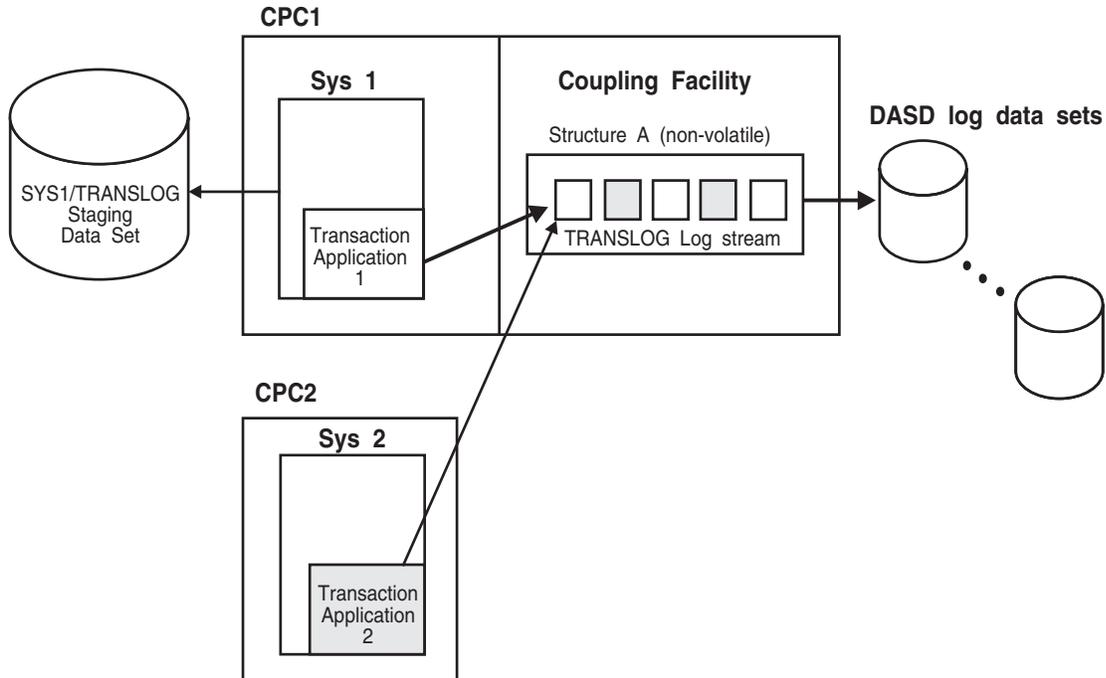


Figure 54. Example: Using Staging Data Sets When the Coupling Facility Shares a CPC with a System

If the installation had specified `STG_DUPLEX(YES) DUPLEXMODE(UNCOND)`, both systems would duplex to staging data sets, even though there is no single point of failure for SYS2.

If the installation had specified `STG_DUPLEX(NO)`, neither system would duplex to staging data sets (under normal circumstances).

Example - Using Staging Data Sets When the Coupling Facility is Volatile: In this scenario, a log stream was defined in the LOGR policy with `STG_DUPLEX(YES) DUPLEXMODE(COND)`. Two systems are connected to the log stream. The coupling facility structure associated with the log stream is in a stand-alone CPC. The coupling facility for the structure is volatile, therefore both systems will duplex the data to staging data sets because there might be a single point of failure for both systems.

System Logger

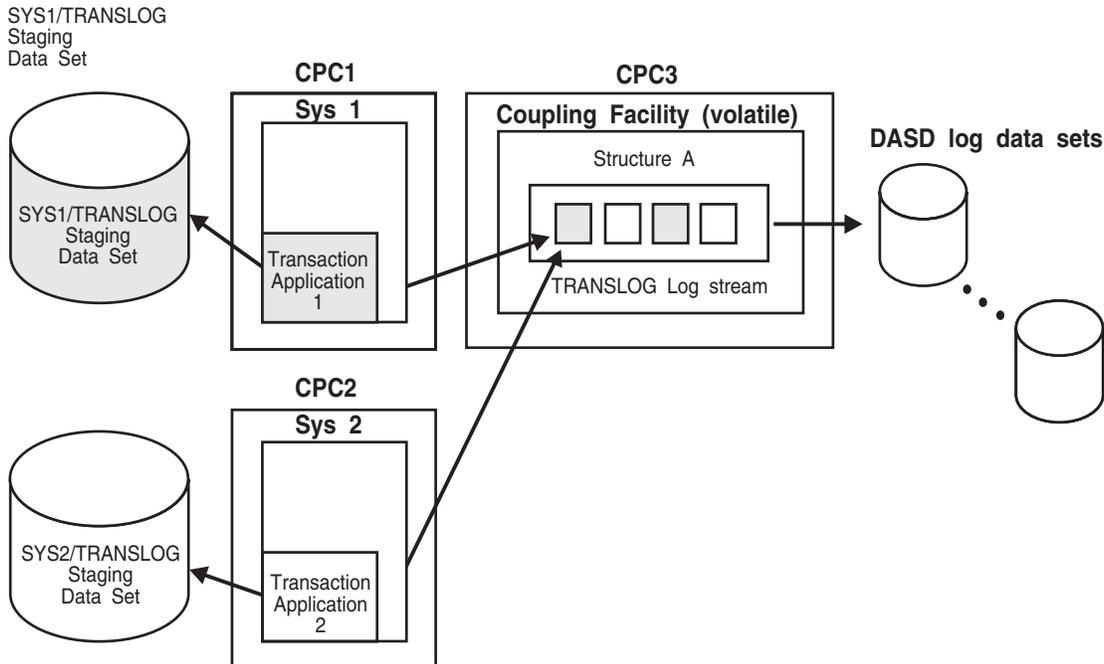


Figure 55. Example: Using Staging Data Sets When the Coupling Facility is Volatile

If the coupling facility was non-volatile, neither system would duplex their coupling facility data to staging data sets, but would maintain local buffers on each system.

If the installation had specified `STG_DUPLEX(YES) DUPLEXMODE(UNCOND)`, both systems would duplex data to staging data sets, regardless of the volatility of the coupling facility.

If the installation had specified `STG_DUPLEX(NO)`, neither system would duplex to staging data sets (under normal circumstances).

Associating a Log Stream With a Resource Manager

This step applies to both coupling facility and DASD-only log streams.

If an application is using a resource manager, you can associate a log stream with a resource manager using the `RMNAME` parameter in the `LOGR` policy. You can specify **one** resource manager name for a log stream in the log stream definition. A resource manager is an application you can write to manage resources and processing for a log stream. See the system logger chapter in *OS/390 MVS Programming: Assembler Services Guide* for more information on resource managers.

Deleting Log Streams From the LOGR Policy

To delete a log stream, you must delete it from the `LOGR` policy. To do this, you can use either:

- The `IXCMIAPU` utility for system logger, with `DELETE LOGSTREAM NAME(log_stream_name)` parameters.
- The `IXGINVNT` service, with the `REQUEST=DELETE TYPE=LOGSTREAM` parameters.

If there are active connections to the log stream you are trying to delete, or failed connections that cannot be recovered, system logger will not let you delete the log stream. The requestor of the delete will be notified if there are any connectors remaining to that log stream.

You can find out if a log stream has existing connections using the LIST parameter on the IXCMIAPU utility for system logger. For example, to get information about connectors to a log stream in the SYSPRINT output, use LIST LOGSTREAM NAME(*log_stream_name*) DETAIL(YES) on the IXCMIAPU utility for system logger. For information on the LIST parameter of IXCMIAPU, see “Appendix B. Administrative Data Utility” on page 263.

If the log stream you wish to delete still has one or more active connectors, you can do one of the following.

- Keep the connection, cancelling deletion of the log stream.
- Cause the application to disconnect from the log stream, and then delete the log stream from the LOGR policy.

Define the Coupling Facility Structures Attributes in the CFRM Policy Couple Data Set

This step applies to coupling facility log streams only. DASD-only log streams can omit this step.

Do the following:

- **Determine the attributes needed to define coupling facility structures used by system logger applications.**

See “Planning a Coupling Facility Policy” on page 41 to plan updates to the CFRM policy for system logger applications. When planning preference and exclusion list specifications, keep in mind that system logger requires a minimum CFLEVEL. See “Understand the Requirements for System Logger” on page 190 for specific requirements.

The ALLOWAUTOALT(NOIYES) parameter on a structure definition in the CFRM policy may affect structures used by system logger. Since logger connects to a coupling facility list structure using ALLOWALTER(YES), if you specify ALLOWAUTOALT(YES) on the list structure definition in the CFRM policy, Cross System Extended Services (XES) may automatically alter the size and ratio attributes of the structure as needed. If your coupling facility is size constrained and you must free up some space, you should probably consider other structures for automatic altering first. If you must allow automatic altering of the logger list structure, IBM recommends that you also specify a minimum size for the logger list structure with the MINSIZE parameter, to ensure that logger always has enough space to do its work.

The effects associated with the automatic altering of a list structure in use by logger can vary depending on the number of log streams connected to the structure. Consider the following cases:

- Case 1: A single log stream is connected to the list structure.

Depending on how your applications use the log stream, you may find that there are long periods when the log stream is relatively empty, followed by peaks of system activity when the log stream quickly approaches its high threshold. If you have specified ALLOWAUTOALT(YES), XES may decide to alter your structure size to a smaller size during a low usage period. When a period of peak usage occurs, write requests to this now smaller structure area for this log stream could result in a structure full condition, meaning there is

System Logger

- | no more space available in the structure for this log stream. You may be able
| to avoid this by specifying a sufficient MINSIZE for the list structure.
- | – Case 2: Multiple log streams are connected to the list structure.
- | In this case, since there is more than one log stream mapped and connected
| to the list structure, the overall use of the structure space may not be
| indicative of how any one log stream is performing. For example, assume one
| log stream in the structure is using most, but not all, of its share of the list
| structure space, and a second log stream in the same structure is using very
| little of its share of the list structure space. If automatic monitoring and altering
| processing is allowed it could reduce this structure's size because the overall
| use of the list structure space is relatively low. This size reduction could cause
| the log stream that was using a large portion of its available space to now
| exceed its available space. Subsequent write requests to this log stream
| would result in a structure full condition, meaning there is no more space
| available in the structure for this log stream.
- **Add information to the CFRM policy couple data set about the desired attributes of the coupling facility list structures for system logger applications.** To use the IXCMIAPU utility to update the CFRM policy for logger information, see “CFRM Parameters for Administrative Data Utility” on page 275. Note that each coupling facility structure that you define in the LOGR policy must match those defined in the CFRM policy. See “CFRM Parameters for Administrative Data Utility” on page 275 for using IXCMIAPU to update the CFRM policy.
 - **Ensure adequate dump space is specified in a CF definition in the CFRM policy that could contain list structures that will be used by the system logger.**
See “Determining the Amount of Coupling Facility Dump Space” on page 53 for setting the DUMPSPACE parameter.
 - **Activate the CFRM policy couple data set using the SETXCF command.**
See *OS/390 MVS System Commands Summary* for the SETXCF command.

Activate the LOGR Subsystem

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

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

Activate the LOGR subsystem using one of the following methods:

- In the IEFSSNxx parmlib member, add the following:
SUBSYS SUBNAME(LOGR) INITRTN(IXGSSINT)

Note that the new format for defining the LOGR subsystem is required (you cannot use the old format). Because IEFSSNxx parmlib members cannot contain the new and old format definitions in the same IEFSSNxx parmlib member, you might need to define a new parmlib member for the LOGR subsystem and concatenate it with other IEFSSNxx parmlib members.

- Use the SETSSI command to add the subsystem dynamically:
SETSSI ADD,SUBNAME=LOGR,INITRTN=IXGSSINT

The value specified for SUBNAME is defined by the installation; it does not have to be LOGR. Make sure, however, that the subsystem name specified in the SUBNAME parameter matches the corresponding name in the SUBSYS option in the JCL.

Note that you cannot issue either the SETSSI ACTIVATE or SETSSI DEACTIVATE command for the LOGR subsystem.

References

- See *OS/390 MVS Programming: Assembler Services Guide* for information on using the LOGR subsystem.
- See *OS/390 MVS Initialization and Tuning Reference* for information about the IEFSSNxx parmlib member.
- See *OS/390 MVS System Commands* for information about the SETSSI command.
- See *OS/390 MVS Installation Exits* for information about the LOGR subsystem exit.

Deleting Log Data and Log Data Sets

In general, applications should handle deletion of log data from a log stream.

If the application does not manage deletion, the installation might be able to use the retention period and automatic deletion functions in system logger to manage the amount of log data you keep in each log stream. See “Managing Log Data: How Much? For How Long?” on page 213 for information on the retention period and automatic deletion. You can also increase the amount of space available for DASD log data sets using the DSEXTENT parameter in the LOGR couple data set. See “Increasing the Space Available for DASD Log Data Sets” on page 207.

If for some reason, your installation wants to delete log data, first carefully evaluate the needs of application or applications writing data to the log stream. For some log streams, deleting log data is not a problem, while for others it could be disastrous.

This section covers the following:

- “Deleting Log Data”.
- “Deleting Log Data Sets” on page 228.
- “When Is My Log Data or Log Data Set Physically Deleted?” on page 228. System logger **does not** delete log data or log data sets immediately when you mark them for deletion.
- “Finding and Deleting Orphaned Log Data Sets” on page 228. Deletion of log data sets may result in errors that leave orphaned log data sets consuming storage.

Deleting Log Data

Use one of the following methods to delete log data:

- Use the IXGDELETE service in an application to delete data from the log stream as soon as the data is no longer needed. This will help to manage log data set storage consumption by making sure log data sets are deleted as soon as possible. See *OS/390 MVS Programming: Assembler Services Reference* for information on using the IXGDELETE service.

System Logger

- For logrec log stream data, run EREP using the LOGR subsystem with the delete option. See program IFBEREPS in SYS1.SAMPLIB for examples of archiving and deleting logrec log data sets. See *OS/390 MVS Diagnosis: Tools and Service Aids* for information on using EREP.
- For OPERLOG, use the IEAMDBLG program in SYS1.SAMPLIB with the DELETE parameter.
- For the CICS log manager, CICS automatically deletes data in a CICS system log. For CICS general logs, use the RETPD and AUTODELETE parameters in the log stream definition to manage the amount of log data automatically. You can also use CICS/VR to archive or delete CICS log data.

Note that using DFHSM to migrate log stream data sets to tape **does not** solve the data set directory full condition because the migrated data sets will still count toward the data set directory entry limit. Deleting DASD log stream data sets (using a TSO/E DELETE command, for example) also **does not** solve the data set directory full condition.

Deleting Log Data Sets

To delete a log data set, delete the log stream definition from the LOGR policy (see “Deleting Log Streams From the LOGR Policy” on page 224). System logger subsequently deletes **all** the log data sets associated with the log stream (see “When Is My Log Data or Log Data Set Physically Deleted?”).

Note that IBM **does not** recommend other methods for deleting data sets, such as the TSO/E DELETE command, because system logger cannot perform any clean-up and the deleted data sets will still count toward the data set directory limit.

When Is My Log Data or Log Data Set Physically Deleted?

After log data is marked for deletion, system logger does not physically delete the log data or log data set until an offload occurs. This means that there will often be a delay before eligible log data sets are physically deleted, since offloading will not occur until the high threshold associated with the log stream is reached. If offloads are relatively infrequent, then there may be a considerable delay before log data sets that are eligible for deletion are actually deleted.

Allocation errors can also delay log data set deletion. If system logger cannot delete a log data set because it is currently allocated, it will attempt to delete the log data set on a subsequent offload when delete processing is again performed. This situation can also delay the deletion of a log data set since system logger may attempt to delete an eligible log data set several times before it actually succeeds. See “Finding and Deleting Orphaned Log Data Sets”.

Finding and Deleting Orphaned Log Data Sets

When you delete a log data set, allocation errors may result in “orphaned” log data sets — log data sets that should have been deleted but were not. When a log data set is orphaned, system logger issues message IXG252I. System logger frees up the directory extents in this case, but the data sets consume storage. If an allocation error occurs when you delete a log stream or when system logger is deleting a log data set, you will need to manually delete the resulting unnecessary log data sets. Note that when you delete a log stream by deleting it from the LOGR policy, system logger deletes all the log data sets associated with the log stream, **NOT** all the log data sets that match the naming convention for the log stream.

Upgrading a Log Stream from DASD-Only to Coupling Facility

You can upgrade a DASD-only log stream to use coupling facility storage, making it a coupling facility log stream. You might do this, for example, if your installation goes to a multiple system configuration. To upgrade a log stream from DASD-only to coupling facility based, associate a structure with the log stream by updating the log stream definition in the LOGR policy. Use the IXGINVNT service or the IXCMIAPU utility, and on an UPDATE request specify a coupling facility structure name on the STRUCTNAME parameter. All connectors to the DASD-only log stream must be disconnected in order to make the upgrade. No failed or active connections can exist for this log stream.

Before upgrading a DASD-only log stream, read the following sections:

- “Plan for a DASD-Only Log Stream Upgrade”
- “Match the MAXBUFSIZE Values for a DASD-only Log Stream Upgrade”
- “Staging Data Sets and Upgrading a DASD-Only Log Stream” on page 230

Plan for a DASD-Only Log Stream Upgrade

Before upgrading a DASD-only log stream, make sure that you have prepared adequately for the new coupling facility log stream. You should be familiar with the coupling facility log stream requirements and planning steps covered in “Preparing to Use System Logger Applications” on page 188. For example, the structure specified on the log stream definition must be defined in the LOGR policy. And before you can connect to the upgraded log stream or issue other system logger services, you must also specify the structure in the CFRM policy (see “Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy” on page 217). In addition, the naming convention system logger uses to allocate staging data sets is different for coupling facility log streams and DASD-only log streams. See “Naming Conventions for the Log Stream and DASD Data Sets” on page 203.

Note that you cannot update a coupling facility log stream definition with a new structure name. You can only specify a structure name on an update request for a DASD-only log stream. If you issue an update request with the STRUCTNAME parameter for a coupling facility log stream, system logger returns a X'0827' reason code.

Match the MAXBUFSIZE Values for a DASD-only Log Stream Upgrade

When you upgrade a log stream from DASD-only to coupling facility based, make sure that the structure you specify on the UPDATE request has a MAXBUFSIZE value that is equal to or greater than the MAXBUFSIZE value currently defined to the DASD-only log stream. For example, if you decide to upgrade a DASD-only log stream that has a MAXBUFSIZE of 65,532 bytes (which is the default), you must associate it with a structure that has a MAXBUFSIZE of 65,532 bytes.

If the MAXBUFSIZE value defined for the structure is smaller than the MAXBUFSIZE value for the DASD-only log stream, system logger ends the UPDATE request with an X'083C' reason code. To avoid this error, do one of the following before upgrading from a DASD-only log stream to a coupling facility based one:

- When specifying log stream definitions for DASD-only log streams, plan ahead for possible upgrades by matching the MAXBUFSIZE value for the DASD-only log stream with the MAXBUFSIZE value for the structure you might assign it to

System Logger

on an upgrade request. The MAXBUFSIZE value on the DASD-only log stream definition must be the same size as or smaller than the MAXBUFSIZE value for the structure.

- On the UPDATE request to upgrade a DASD-only log stream, specify a structure with a MAXBUFSIZE value that is as large or larger than the DASD-only log stream MAXBUFSIZE value.

Note that you **cannot** issue an UPDATE request to make a MAXBUFSIZE smaller on a DASD-only log stream definition. You also cannot specify the MAXBUFSIZE parameter on an UPDATE request for a structure definition.

Staging Data Sets and Upgrading a DASD-Only Log Stream

A DASD-only log stream automatically duplexes log data to DASD staging data sets. When you upgrade a DASD-only log stream to a coupling facility based log stream, you will still get duplexing to staging data sets **unless you specifically specify otherwise**. You must specify STG_DUPELEX(NO) when you upgrade on the UPDATE LOGSTREAM request to get a coupling facility log stream that does duplexes to local storage buffers rather than to DASD staging data sets. For example, if you do not want to use staging data sets after the upgrade, specify STG_DUPLEX(NO) on the UPDATE request. See “Selecting a Method of Duplexing Coupling Facility Log Data” on page 220 for more information on duplexing options.

System Logger Recovery

This section explains how system logger handles errors, how the errors might affect applications, and what action, if any, a system programmer might need to take. For coupling facility log streams, see also “Plan Your Configuration for Log Data Recovery” on page 195.

System logger performs recovery differently for for DASD-only versus coupling facility log streams. Most of the information in this section applies to coupling facility log streams only. For DASD-only log streams, see “Recovery Performed for DASD-Only Log Streams” on page 231. Other recovery information pertinent to DASD-only log streams is noted under each topic below.

Operations Log and Logrec Application Recovery

Each system logger application will issue messages to indicate that an error has occurred and explain it:

- **OPERLOG:** If operations log encounters an unrecoverable error, such as a damaged log stream or coupling facility structure, the system attempts to switch to a different hardcopy medium, such as SYSLOG or a device, based on what is defined on the HARDCOPY statement in the CONSOLxx parmlib member. (See *OS/390 MVS Initialization and Tuning Reference* for more information on the CONSOLxx parmlib member.)

If an error occurs, the operations log issues message IEE316I indicating that OPERLOG has failed, the system where it has failed, and possibly some additional explanation of the problem. Message IEE316I might show some return and reason codes returned to operations log from system logger service; look these up in the *OS/390 MVS Programming: Assembler Services Guide*. Operations log issues message IEE710I showing the hardcopy medium that is active after OPERLOG fails.

- **Logrec log stream:** If logrec encounters a system logger error, logrec issues system message IFB100E explaining the problem. See the explanation of message IFB100E in *OS/390 MVS System Messages, Vol 4 (IEC-IFD)*.

If the error is recoverable, logrec resumes writing to the log stream. If the error is unrecoverable, an operator action might be required to switch hardcopy recording mediums or to resume writing to the log stream.

Recovery Performed for DASD-Only Log Streams

Like a coupling facility log stream, a DASD-only log stream is subject to system or system logger failure. A DASD-only log stream is not subject, however, to coupling facility or structure failures. When a failure occurs involving a DASD-only log stream, system logger releases the exclusive ENQ on the log stream name serializing the log stream for one system. No system-level or peer recovery is performed for a DASD-only log stream after a failure or as part of system initialization. System logger does not perform system-level recovery for a DASD-only log stream because data is already safeguarded on DASD staging data sets, a non-volatile medium. For a DASD-only log stream, offload of log data to DASD log data sets is not done as part of recovery processing for the same reason — log data is already on DASD staging data sets. Peer recovery is both unnecessary and not possible for a DASD-only log stream, because there are no peer systems connected to the log stream.

Recovery for a DASD-only log stream only takes place when an application reconnects to the log stream. As part of connect processing, system logger reads log data from the staging data set (associated with the last connection to the log stream) into the local storage buffers of the current connecting system. This allows the application to control recovery, by selecting which system they wish to have reconnect to the log stream and when. Note that for another system to connect to the log stream and perform recovery, the staging data sets must reside on devices accessible by both systems.

When an MVS System Fails

This section applies to coupling facility log streams only. For DASD-only log streams, see “Recovery Performed for DASD-Only Log Streams”.

When a system fails, system logger tries to safeguard all the coupling facility log data for the failed system by offloading it to DASD log data sets so that it is on a persistent media.

Recovery processing for the failing system is done by a **peer connector**, which is another system in the sysplex with a connection to a coupling facility structure that the failing system was also connected to. Note that a peer connector need only be connected to the same coupling facility structure, not the same log stream. See “Plan Your Configuration for Log Data Recovery” on page 195 for more information.

If there is no peer connection available to perform recovery for a failed system, recovery is delayed until either the failing system re-IPLs or another system connects to a log stream in the same coupling facility structure to which the failing system was connected.

If you have a failed system with no peer connectors (no other system is connected to the same coupling facility structure) and you want to minimize the time that the coupling facility data for the log stream remains unrecovered, we recommend one of the following:

- Write a dummy application to run on another system that will connect to the same log stream or coupling facility structure that the failed system was connected to.

System Logger

- Make sure that another log stream with connectors from other systems maps to the same coupling facility structure.

Either of these procedures will ensure peer connectors for a system connection to a coupling facility structure so that recovery can be performed.

Recovery for a Failing System in a Single System Sysplex

When a failing system was in a single system sysplex, there are no peer connectors existing to perform recovery for the failing system. In this case, the coupling facility resident log data is recovered when the failed system re-IPLs.

If the log data is still not recovered after IPL, do the following:

1. Delete the log stream from the LOGR data set; system logger will then write all the log stream data to DASD data sets.
2. If you receive a message that indicates that system logger cannot delete the log stream because it is 'in use', write a dummy application to run on another system that will connect to the same log stream or coupling facility structure that the failed system was connected to.

If the coupling facility that the system was connected to also fails, the system will lose its coupling facility resident log data unless staging data sets were in use. If staging data sets exist, system logger uses them to recover log data for the failing system. If there were no staging data sets in use when both the system and the coupling facility fails, any log streams affected are marked as damaged.

When a Sysplex Fails

This section applies to coupling facility log streams only. For DASD-only log streams, see "Recovery Performed for DASD-Only Log Streams" on page 231.

When all the systems in a sysplex fail, there are no peer connectors to perform the recovery processing for the failing systems, which would consist of offloading the coupling facility data for log streams to DASD log data sets. Coupling facility resident log data continues to exist. Further recovery processing depends on whether or not the coupling facility also failed.

When the Coupling Facility Survives the Sysplex Failure

When the coupling facility survives the sysplex failure, each system in the sysplex does the following:

1. Re-IPLs.
2. Identifies and reconnects to all the log streams to which it was connected before the failure occurs.
3. Performs recovery for the log streams it was connected to, which consists of offloading coupling facility log data to DASD log data sets.
4. Disconnects from the log streams that it reconnected to.
5. Performs recovery for any other log streams that were connected to the same coupling facility that the IPLing system had connections to.

Note that each IPLing system will perform recovery for log streams that it has a peer connection to, for as long as there is recovery to be done. Subsequent peer connectors IPLing will not duplicate recovery performed by other peer connectors.

When the Coupling Facility Fails with the Sysplex

When the coupling facility fails at the same time as the sysplex, each system in the sysplex does the following:

1. Re-IPLs.
2. Identifies all the log streams to which it was connected before the failure occurred.
3. Performs recovery for log streams by trying to recover log data from staging data sets:
 - If each connector to a log stream was using staging data sets at the time of the failure, system logger recreates the log stream data from the staging data sets for each connection. If the recovery from staging data sets is successful, the log stream is considered recovered.
If system logger encounters an error while trying to recover log data from staging data sets, the log stream is marked as damaged. Subsequent IPLing systems will not attempt recovery from staging data sets because the log stream has been marked as damaged.
 - If one or more connectors to a log stream did **not** use staging data sets while trying to recover log data from staging data sets, the log stream is marked as damaged.

Note that each IPLing system will perform recovery for the log streams that it has a peer connection to, for as long as there is recovery to be done. Subsequent peer connectors IPLing will not duplicate recovery performed by other peer connectors.

When the System Logger Address Space Fails or Hangs

If system logger or an application using it appears to be hung or fails, there are several things that can be done to aid in identifying the problem. The following sections list some useful diagnostic procedures related to logger.

When the System Logger Address Space Hangs

The following commands can be useful as diagnostic aids when the system logger address space hangs. For a full description of these commands, see *OS/390 MVS System Commands Summary*.

1. Issue

```
D XCF,COUPLE,TYPE=LOGR
```

to display couple data sets in use for system logger.

2. Issue

```
D XCF, STR,STRNAME=structurename
```

to display the status of a structure.

3. Issue

```
D LOGGER,STATUS
```

to display the status of the system logger address space (IXGLOGR) on this system.

4. Issue

```
D LOGGER,CONNECTION,...
```

to display the connection status of logstreams and jobs on this system.

5. Issue

```
D LOGGER,LOGSTREAM,...
```

System Logger

to display logstream activity from a sysplex view (for example, from the LOGR couple data set).

6. Issue

```
D LOGGER,STRUCTURE,...
```

to display structure activity from a sysplex view (for example, from the LOGR couple data set).

7. Issue

```
D GRS,C
```

to see if there is any contention.

8. Issue

```
D GRS,RES=(SYSZLOGR,*)
```

to see if there is a logger resource serialization contention.

9. Issue

```
D GRS,ANALYZE,BLOCK,SYSTEM=*,JOBNAME=IXGLOGR
```

to see what units of work are blocking GRS-managed resources.

10. Issue

```
D GRS,ANALYZE,WAIT,SYSTEM=*,JOBNAME=IXGLOGR
```

to see what units of work are waiting for ownership of GRS-managed resources.

11. Issue

```
D GRS,ANALYZE,DEPEND,SYSTEM=*,JOBNAME=IXGLOGR,DETAIL
```

to see the dependencies between units of work and resources that are in contention.

12. Run the IXCMIAPU utility and specify

```
DATA TYPE(LOGR) REPORT(YES)
LIST STRUCTURE NAME(structurename|*) DETAIL(YES)
LIST STRUCTURE NAME(structurename|*) DETAIL(YES)
```

See “Appendix B. Administrative Data Utility” on page 263 for a full description of the TYPE(LOGR) LIST options.

If there is a problem with a single application, attempt to stop and restart the application before stopping logger.

If there is a problem with a structure, a rebuild may help resolve the problem. Issue

```
SETXCF START,REBUILD,STRNAME=structurename
```

If there is no logger (LOGR) couple data set, issue

```
SETXCF COUPLE,TYPE=LOGR,PCOUPLE=(primarydataset,volser),
ACOUPLE=(alternatedataset,volser)
```

to define the logger couple data set(s).

If there is a problem with the existing logger couple data sets(s),

1. issue

```
SETXCF COUPLE,TYPE=LOGR,ACOUPLE=(alternatedataset,volser)
```

to change to the logger alternate data set.

2. issue

```
SETXCF COUPLE,TYPE=LOGR,PSWITCH
```

to switch the current alternate logger couple data set to be the new primary couple data set.

If you are unable to resolve the problem at this point, it may be necessary to stop and restart logger. Before doing this, obtain a dump of logger on the systems that are impacted. See *OS/390 MVS Diagnosis: Tools and Service Aids* for details on obtaining a logger dump.

To terminate logger, issue the command

```
FORCE IXGLOGR, ARM
```

Note that this will cause all active connections on the system to be terminated. Message IXG056I will be issued when the IXGLOGR address space has terminated.

It may also be necessary to restart structure connections by issuing

```
SETXCF FORCE,CON,STRNAME=strname,CONNAME=ALL
```

The system will no longer have any type of connection to the structure (*strname*).

Once logger has terminated, it can be restarted by issuing the command S IXGLOGRS. After doing this, it may be necessary to restart applications that were previously connected. Logger will issue an ENF signal to indicate when it is ready to start work.

When the System Logger Address Space Fails

This section applies to both coupling facility and DASD-only log streams, with differences noted in the text.

If the system logger address space fails, the following processing occurs:

- Log stream connections are terminated.
- The system issues a X'1C5' abend code.
- Any system logger requests from the system where the system logger component failed are rejected.

Contact the IBM Support Center for a fix for the problem, providing the SVC dump that accompanies abend code X'1C5'.

Subsystems or applications that held the lost connections may be required to be restarted. To start the system logger address space again after a failure, use the IXGLOGRS procedure in SYS1.PROCLIB:

```
START IXGLOGRS
```

This procedure starts the system logger address space as a system address space.

For coupling facility log streams, recovery processing for the system logger address space is done by peer connectors, if available, and consists of offloading the coupling facility log data for the system where the failure occurred to DASD log data sets so that it is on a persistent media. The recovery process for the system logger address spaces is the same as that for system failures. See “When an MVS System Fails” on page 231 for a description of the recovery process.

System Logger

When the Coupling Facility Structure Fails

This section applies to coupling facility log streams only.

The following coupling facility problems can occur, resulting in rebuild processing for the structure:

- Damage to or failure of the coupling facility structure.
- Loss of connectivity to a coupling facility.
- A coupling facility becomes volatile.

For complete information on rebuild processing, see *OS/390 MVS Programming: Sysplex Services Guide*.

Damage to or Failure of the Coupling Facility Structure

If a coupling facility fails or is damaged, all systems connected to the coupling facility structure detect the failure. The first system whose system logger component detects the failure initiates the structure rebuild process. The structure rebuild process results in the recovery of one or more of the affected coupling facility structure's log streams. All the systems in the sysplex that are connected to the list structure participate in the process of rebuilding the log streams in a new coupling facility list structure.

While the rebuild is in progress, system logger rejects any system logger service requests against the log stream. The status will be one of the following:

- The structure rebuild has completed successfully, the coupling facility structure and associated log streams are available, and system logger requests will be accepted.
- The structure rebuild was unsuccessful and connection to the structure is not possible because the structure is in a failed state. Log data still resides in staging data sets if they are used to duplex the log data for the log stream. If staging data sets were not used, the data persists in the local storage buffers on each system.

Loss of Connectivity to the Coupling Facility Structure

If a system loses connectivity to the coupling facility structure due to a hardware link failure, all the systems connected to the log streams associated with the coupling facility detect the failure.

Then, based on the rebuild threshold specified, if any, in the structure definition in the CFRM policy, the system that lost connectivity may initiate a rebuild for the structure.

If a rebuild is initiated, the event 48 parameter list mapped by macro IXGENF has bits IXGENFLogStreamsNotAvailable, and IXGENFStrRebuildStart, on, and field IXGENFStrName contains the name of the coupling facility structure affected. System logger rejects logger service requests issued during the rebuild process.

If XES cannot allocate a new coupling facility that all the systems affected can connect to, system logger does one of the following, depending on whether the system or systems that cannot connect to the new coupling facility structure were using staging data sets:

- If the system was using staging data sets, the rebuild process continues and the coupling facility log data for the system is recovered from the staging data sets.
- If the system was **not** using staging data sets, the rebuild process is stopped. The systems go back to using the source structure.

The systems that do not have connectivity to the old coupling facility structure issue an ENF 48 event indicating that they do not have connectivity to the log stream.

The systems that can connect to the source structure issue an ENF 48 event indicating that the log stream is available to that system and can resume use of the log stream.

The installation should either update the CFRM active policy to make the new coupling facility structure available to all the systems or else fix the hardware link problem and then have the operator initiate a rebuild for the structure so that all the original systems will have connectivity.

A Coupling Facility Becomes Volatile

If a coupling facility changes to the volatile state, the system logger on each system using the coupling facility structure is notified. A dynamic rebuild of the structure is initiated so that the log data can be moved to a non-volatile coupling facility. During rebuild processing, system logger rejects any logger service requests.

If there is not a structure available in a non-volatile coupling facility, system logger will still rebuild the data on a new volatile coupling facility. System logger may then change the way it duplexes coupling facility data since the volatile coupling facility constitutes a single point of failure:

- For log streams defined with STG_DUPLEX=YES, system logger will begin duplexing data to staging data sets, if they were not already in use.
- For log streams defined with STG_DUPLEX=NO, system logger will keep on duplexing data to local storage buffers on each system.

When the Coupling Facility Space for a Log Stream Becomes Full

This section applies to coupling facility log streams only.

Ordinarily, system logger offloads coupling facility resident log stream data to DASD log data sets before the coupling facility storage allocated to the log stream is fully utilized. Occasionally however, the coupling facility storage allocated to a log stream reaches 100% utilization, for reasons such as a sudden burst of logging activity or because your coupling facility is sized too small for the volume of log data.

You can monitor the number of times a log stream becomes full using SMF record 88 (field SMF88ESF).

If you want to enlarge your coupling facility structure, you can do it in one of two ways:

- Alter the coupling facility size dynamically by issuing the SETXCF START,ALTER command or the IXLALTER service.
- Update the CFRM policy with the new coupling facility size, activate the policy, and then have the operator initiate a rebuild of the structure.

When a Staging Data Set Becomes Full

This section applies to both coupling facility and DASD-only log streams.

The staging data sets for each system should not fill up. If they do, you probably have them sized too small for your volume of log data and should enlarge them.

System logger will immediately begin offloading log data to DASD log data sets. If your staging data set is too small, you may find that offloading occurs very frequently.

System Logger

You can monitor staging data set usage using SMF record 88, field SMF88ETF, for staging data set full conditions.

When a Log Stream is Damaged

This section applies to both coupling facility and DASD-only log streams.

System logger marks a log stream as permanently damaged when it cannot recover log data from either DASD staging data sets or the local storage buffers after a system, sysplex, or coupling facility failure. Applications are notified of the damage via system logger services. Recovery actions are necessary only if warranted for the application.

When DASD Log Data Set Space Fills

This step applies to both coupling facility and DASD-only log streams.

The number of DASD log data sets available for log streams in a sysplex depends on whether you use the default (168 per log stream) or have provided additional directory extents in the LOGR couple data set. See “Increasing the Space Available for DASD Log Data Sets” on page 207 for more information.

System logger monitors usage of the available log stream directory space, notifying you as follows if you start to run out of space:

- If you are using the DSEXTENT parameter in the LOGR couple data set system logger issues messages IXG261E and IXG262A indicating that usage of directory extents is over 85% and 95% respectively.
- If you are using the default number of log data sets allowed for a log stream (168), system logger issues message IXG257I indicating that the data set directory for the log stream is over 90% full.

If you have run out of log stream directory space, offloads may fail. When this occurs, system logger issues message IXG301I. Offload processing for the log stream cannot complete until more log stream directory space or directory extents are made available. If the last disconnect to a log stream occurs and the offload cannot complete successfully, the log stream is in a failed state. In this case, the log stream is considered 'in use' and there may be a failed-persistent connection to a structure associated with the log stream.

You can make more directory space available for a log stream in one of the ways below. Use the IXCMIAPU utility to run a report of the log stream definitions in the LOGR couple data set to help you with this step. The LIST LOGSTREAM NAME(*) DETAIL(YES) statement outputs information showing which log streams might be using large numbers of data sets and directory extents. See “LOGR Parameters for Administrative Data Utility” on page 281 for more information about IXCMIAPU.

- Format another set of LOGR couple data sets with a higher DSEXTENT value and bringing them into the sysplex as the active primary and alternate LOGR couple data sets. See “Increasing the Space Available for DASD Log Data Sets” on page 207. You must have a LOGR couple data set formatted at the OS/390 Release 3 level or above to use the DSEXTENT parameter. See “Appendix A. Format Utility for Couple Data Sets” on page 241 for the IXCL1DSU utility and the DSEXTENT parameter.
- Free directory extents currently in use in one of the following ways:
 - Use a program that issues the IXGDELET service to delete enough data from the log stream to free up space in the log stream data set directory.

Some products provide a program to delete log stream data. See “Deleting Log Data and Log Data Sets” on page 227 for information on deletion programs provided for IBM products.

- Delete log stream definitions from the LOGR couple data set.

Identify and delete the definitions for unused or unnecessary log streams. This will free the directory space associated with the log streams, which may free up directory extents for use by other log streams.

Note: Deleting DASD log data sets using a non-system logger method will not work because system logger will still count the data sets toward the data set directory entry limit. You cannot, for example:

- Use a TSO/E DELETE command to delete a log data set.
- Use DFHSM to migrate log stream data sets to tape.

When Unrecoverable DASD I/O Errors Occur

This section applies to both coupling facility and DASD-only log streams, with differences noted.

DASD I/O errors may occur against either log data sets or staging data sets. System logger tries to recover from the error, but if it cannot, the error is characterized as an **unrecoverable I/O error**. See the following:

- “When Unrecoverable DASD I/O Errors Occur During Offload”
- “When Staging Data Set Unrecoverable DASD I/O Errors Occur”

When Unrecoverable DASD I/O Errors Occur During Offload

DASD I/O errors may occur during offload processing, while log data is being written to DASD log data sets. When this happens, system logger tries to recover by closing the current log data set and allocating a new one. If this process fails, the I/O error is characterized as an unrecoverable I/O error.

In the case of unrecoverable I/O errors, system logger will accept subsequent IXGWRITE requests as follows:

- For a coupling facility log stream, system logger will accept IXGWRITE requests if the log stream is connected to a coupling facility where there is still room for log data. If the coupling facility is full or no coupling facility exists, system logger rejects IXGWRITE requests.
- For a DASD-only log stream, system logger will accept IXGWRITE requests until the staging data set for the system writing to the log stream is filled.

IXGBRWSE and IXGDELET requests may continue to work. I/O errors encountered in the process of completing these requests are reported to the application in return and reason codes.

To correct an unrecoverable I/O problem, delete the log stream definition in the LOGR policy and redefine it with different log data set attributes, such as LS_DATACLAS, in order to get the log stream data set allocated in a usable location.

When Staging Data Set Unrecoverable DASD I/O Errors Occur

DASD I/O errors may occur when log data is being duplexed to DASD staging data sets. When this occurs, system logger tries to recover by doing the following:

1. Offload current log data to DASD log data sets.
2. Delete and unallocate the staging data set.

System Logger

3. Re-allocate a new instance of the staging data set.

In the meantime, system logger continues to accept write and other requests against the log stream.

If system logger cannot re-allocate a new staging data set, the I/O error is characterized as unrecoverable. In the case of an unrecoverable staging data set I/O error, system logger does the following:

- **For a coupling facility based log stream**, system logger switches the duplexing mode to duplex log data to local storage buffers. The system issues message IXG255I indicating that the duplexing mode has changed. Normal system logger processing continues. The log stream may be more vulnerable to data loss due to system, sysplex, or coupling facility failure.
- **For a DASD-only log stream**, system logger disconnects connectors from the log stream. The system issues message IXG216I to indicate that connectors are being disconnected because a staging data set could not be allocated.

Appendix A. Format Utility for Couple Data Sets

IBM provides the couple data set format utility to allocate and format the DASD couple data sets for your sysplex. See “Chapter 3. Planning the Couple Data Sets” on page 25 for couple data set considerations before proceeding with couple data set formatting. **In particular, note the recommendations about not overspecifying the size or the parameter values in a policy, which could cause degraded performance in a sysplex.**

The control statements for the utility programs follow standard conventions for JCL statements. The utility program accepts 80-byte records that can contain one or more parameters per record. Columns 72 through 80 of the record image are not used. See *OS/390 MVS JCL Reference* for additional information about JCL statements.

Use the XCF couple data set format utility to create and format all sysplex couple data sets prior to IPLing a system that is to use the sysplex couple data sets. Other types of couple data sets do not have to be formatted prior to IPLing the system. The format utility is portable; that is, you can run the utility on MVS/XA SP 2.2.3 or a later-level system.

Using the Couple Data Set Format Utility

The name of the XCF couple data set format utility is IXCL1DSU. This program resides in SYS1.MIGLIB (which is logically appended to the LINKLIST), which makes it available through STEPLIB on a pre-OS/390 system.

IXCL1DSU in OS/390 allows you to format all types of couple data sets for your sysplex. The utility contains two levels of format control statements. The primary format control statement, DEFINEDS, identifies the couple data set being formatted. The secondary format control statement, DATA TYPE, identifies the type of data to be supported in the couple data set — sysplex data, automatic restart management data, CFRM data, SFM data, WLM data, LOGR data, or OS/390 UNIX System Services data.

- Once the ARM, CFRM, SFM, or LOGR couple data sets are formatted, you use another utility program, IXCMIAPU, to create or update administrative data for these services. See “Appendix B. Administrative Data Utility” on page 263 for a description of the IXCMIAPU utility program.
- For WLM couple data sets, you use an ISPF interface to create policy administrative data. See *OS/390 MVS Planning: Workload Management* for a description of the ISPF interface.
- For UNIX couple data sets, you do not need to perform any additional steps.

Note that sysplex couple data sets need no additional processing.

The sysplex couple data set must be formatted to contain only XCF control data. For other types of couple data sets, however, you have the option of either formatting a couple data set for each type of service separately or combining more than one type of service in a single couple data set. Thus, you could format a CFRM couple data set and an SFM couple data set, or you could combine the utility control statements to format a CFRM/SFM couple data set. The system maintains the data for each type of service separately so that data integrity is ensured.

Format Utility

If you choose to format separate couple data sets for the services you are using, you can place them on the same DASD volume or on separate volumes, depending on your capacity and performance requirements. For more information on this topic, see 28.

Understanding Couple Data Set Coexistence

The IXCL1DSU format utility allows you to specify different maximum values for some couple data set parameters. These values are based on the version of the code installed on your system. For example, the initial maximum value for the CFRM parameters STR and CONNECT was 64. The updated version of the utility allows you to specify 255 as the maximum value for STR and CONNECT. This increase causes a comparable increase in the physical size of the CFRM couple data set.

In a sysplex with a mix of the same type couple data sets formatted with different versions of code, the following coexistence concerns should be noted:

- XCF does not allow you to install an uplevel couple data set on a system in a sysplex in which one or more systems do **not** include the code for a newer version of the format utility (and for the higher maximum values specified in the utility).

For example, you cannot issue a SETXCF COUPLE command to bring into use a CFRM couple data set formatted for 255 structures in a sysplex in which the code support for the 255 maximum has not been installed on all systems.

- When defining your primary and alternate couple data sets, the system does not allow you to mix couple data sets formatted at different versions when running in a sysplex in which one or more systems do **not** include the code for a newer version of the format utility (and for the higher maximum values specified in the utility).

For example, do not specify a primary couple data set formatted for 255 structures and an alternate couple data set formatted for 64 structures (or vice versa) on a system on which the code support for the 255 maximum has not been installed.

Coding the Couple Data Set Format Utility

See the following figures for examples of how to set up your JCL to format different types of couple data sets:

- For a sysplex couple data set, see Figure 56 on page 251
- For an ARM couple data set, see Figure 58 on page 253
- For a CFRM couple data set, see Figure 59 on page 255
- For a LOGR couple data set, see Figure 60 on page 256
- For an SFM couple data set, see Figure 61 on page 257
- For a UNIX couple data set, see Figure 62 on page 258
- For a WLM couple data set, see Figure 63 on page 260.

The utility control statements are described below:

PGM=IXCL1DSU

The format utility program that is shipped with Version 5 and OS/390.

STEPLIB

Identifies the SYS1.MIGLIB library.

SYSPRINT

Describes where the output messages from the format utility are to be written. The SYSPRINT DD statement is required.

SYSIN

Describes the utility control statements used to format one or more couple data sets.

DEFINEDS

Specifies the beginning of the definition of a new couple data set. There is no limit to the number of DEFINEDS statements that you provide, and therefore to the number of couple data sets that you format. The couple data set to be created should not have previously existed and must not be found by the format utility either on the specified volume or in the catalog. This prevents the format program from being run against a couple data set that is in use.

Specific Couple Data Set Information
<p>The information for the type of couple data set being formatted is identified with parameters of the DEFINEDS statement.</p> <p>For a description of the keywords to format a couple data set type, see the respective topic, below.</p> <ul style="list-style-type: none"> • “Sysplex Parameters for Format Utility” on page 246 • “Automatic Restart Management Parameters for Format Utility” on page 246 • “CFRM Parameters for Format Utility” on page 246 • “LOGR Parameters for Format Utility” on page 248 • “SFM Parameters for Format Utility” on page 248 • “WLM Parameters for Format Utility” on page 249 • “OS/390 UNIX Parameters for Format Utility” on page 249

The physical attributes of the couple data set are controlled with the following parameters of DEFINEDS.

SYSPLEX(sysplex-name)

Specifies the name of the sysplex in which the couple data set is to be used. The sysplex-name must be 1-8 alphanumeric characters, with the first character alphabetic. The sysplex name is a required parameter.

MAXGROUP(total-group) THIS STATEMENT CAN BE USED ONLY ON A DEFINEDS STATEMENT WHEN FORMATTING A SYSPLEX COUPLE DATA SET. IT IS MAINTAINED FOR COMPATIBILITY WITH MVS/ESA SP VERSION 4 STATEMENTS.

Specifies the number of groups to be supported by the new sysplex couple data set. If this parameter is omitted, a default of 10 is used. The maximum allowable value for this parameter is release dependent. In OS/390, the maximum supported is 2045. The minimum value of MAXGROUP allowed is 10.

MAXMEMBER(members-per-group) THIS STATEMENT CAN BE USED ONLY ON A DEFINEDS STATEMENT WHEN FORMATTING A SYSPLEX COUPLE DATA SET. IT IS MAINTAINED FOR COMPATIBILITY WITH

MVS/ESA SP VERSION 4 STATEMENTS.

Specifies the maximum number of members in any one XCF GROUP. If this parameter is omitted, a default value is taken. The default number of members in a group is calculated to be the number of members which can

Format Utility

be accommodated on one track on the device type on which the sysplex couple data set resides. The maximum allowable value for this parameter is release dependent. In OS/390, the maximum supported is 1023. The minimum value of MAXMEMBER allowed is 8. Systems joining the sysplex obtain the MAXMEMBER value from the sysplex couple data set. The actual number defined by the format utility might be larger than the number specified. This is due to the fact that XCF adds one member for its own use and rounds the number of members to the next larger multiple of four.

MAXSYSTEM(maximum-systems)

Specifies the number of systems to be supported by the new couple data set. If this parameter is omitted, the format utility assigns a default value of eight. A sysplex couple data set formatted to support eight systems is compatible with and can be used by MVS/ESA SP 4.1 and beyond. The specification of a number other than eight will not allow an MVS/ESA SP Version 4 system to use the sysplex couple data set. In OS/390 the maximum useable value is 32.

Notes:

1. If MAXSYSTEM is greater than eight, and RMAX (the maximum length of WTOR reply IDs, as specified in the CONSOLxx parmlib member) is less than 99, then the system will increase RMAX to 99 when you start using this data set. This is to handle the potential increase in WTOR traffic in a sysplex of greater than eight systems. Once RMAX is increased, it cannot be decreased. The system issues message IEA403I to indicate that the maximum number of reply IDs becomes 99.
2. When a sysplex is configured with a MAXSYSTEM value greater than 8, reply IDs are no longer assigned in strict sequential order. Instead, systems obtain groups of reply IDs for assignment to WTORS, and the ids might not be assigned in sequential order. This change requires no coding changes on the installation's part, but might surprise an operator. You should consider informing operators of this change.

DSN(data-set-name)

Specifies the 1 - 44 character length name to be used for the couple data set. The format utility will allocate a new data set by this name on the volume specified by the VOLSER parameter. The valid characters are alphanumeric characters (A-Z and 0-9), national characters (\$,@,#), and a period (.). The first character must be either an alphabetic or a national character. DSN is a required parameter.

VOLSER(volser)

Specifies the 1 - 6 character length name of the volume on which the couple data set is to be allocated. The valid characters are alphanumeric characters (A-Z and 0-9) and national characters (\$,@,#).

VOLSER is a required parameter for non-SMS managed data sets. For SMS-managed data sets and VOLSER restrictions, see the STORCLAS and MGMTCLAS parameters.

UNIT(type)

Specifies the type of the unit to which the couple data set is to be allocated. You can specify an installation-defined group name, generic device type, or specific device number.

The UNIT parameter is optional. If you specify UNIT, then you must also specify VOLSER.

NOCATALOG**CATALOG**

Specifies whether the data set is to be cataloged. The default is not to catalog the data set if neither CATALOG nor NOCATALOG is specified. If the data set is SMS-managed, NOCATALOG is ignored.

STORCLAS(storage-class)

Specifies the SMS storage class for the SMS-managed volume where the data set is to reside. **storage-class** can be any valid SMS storage class name. STORCLAS is an optional parameter.

Either STORCLAS or VOLSER must be defined on the DEFINEDS statement. If STORCLAS is specified, you do not need to specify VOLSER. It is recommended that you specify the GUARANTEED SPACE=YES attribute for the storage class. You can use ISMF panels to specify names of storage classes with the guaranteed space attribute.

If your installation has assigned storage classes and management classes through automatic class selection (ACS) routines, the routines might affect where SMS places the couple data set. The format utility issues a message to indicate the volume that SMS has selected for the data set.

MGMTCLAS(management-class)

Specifies the SMS management class for the SMS-managed volume where the data set is to reside. **management-class** can be any valid SMS management class name. MGMTCLAS is an optional parameter.

If your installation has assigned storage classes and management classes through automatic class selection (ACS) routines, the routines might affect where SMS places the couple data set. The format utility issues a message to indicate the volume that SMS has selected for the data set.

DATA TYPE

Indicates the type of data to be supported by the couple data set. Use one DATA TYPE control statement to identify each function to be supported by the couple data set.

DATA TYPE(SYSPLEX)**DATA TYPE(ARM)****DATA TYPE(CFRM)****DATA TYPE(LOGR)****DATA TYPE(SFM)****DATA TYPE (WLM)****DATA TYPE(BPXMCD)**

Identifies the type of data to be supported by the couple data set.

Note that SYSPLEX data cannot be combined with other data types in the same couple data set.

ITEM NAME(data-name) NUMBER(data-number)

Specifies the number of occurrences of the data specified on the DATA TYPE statement that are to be supported on the couple data set.

Format Utility

Sysplex Parameters for Format Utility

For a sysplex couple data set — DATA TYPE(SYSPLEX), valid data names are GROUP and MEMBER.

ITEM NAME(GROUP) NUMBER()

Specifies that the sysplex couple data set supports group data. Include not only the number of XCF groups needed for multisystem applications, but also the number of XCF groups that MVS system components need. (To determine the number of MVS system groups, issue the DISPLAY XCF,G command.) It is also advisable to add a contingent number of groups for future growth. See “Considerations for All Couple Data Sets” on page 25. (Default=50, Minimum=10, Maximum=2045)

ITEM NAME(MEMBER) NUMBER()

Specifies that the sysplex couple data set supports member data. Determine which group has the largest number of members that can exist on systems in a sysplex and specify a value at least as large. (Default=number of members that can be supported on one track of the device type for which the sysplex couple data set is being formatted, Minimum=8, Maximum=1023.

ITEM NAME(GRS) NUMBER(1)

Specifies that the sysplex couple data set supports global resource serialization data.

Automatic Restart Management Parameters for Format Utility

For an automatic restart management couple data set — DATA TYPE(ARM) — valid data names are POLICY, MAXELEM, and TOTELEM.

ITEM NAME(POLICY) NUMBER()

Specifies the number of user-defined administrative policies that can be defined. (Default=3, Minimum=1, There is no maximum value.)

ITEM NAME(MAXELEM) NUMBER()

Specifies the maximum number of elements per policy. (Default=10, Minimum=1, There is no maximum value.)

ITEM NAME(TOTELEM) NUMBER()

Specifies the maximum number of elements that can be registered in the sysplex at once. (Default=500, Minimum=1, Maximum=999)

CFRM Parameters for Format Utility

For a CFRM couple data set — DATA TYPE(CFRM), valid data names are POLICY, CF, STR, CONNECT, and SMREBLD.

ITEM NAME(POLICY) NUMBER()

Specifies the number of administrative policies that can be defined within the data set. (Default=6, Minimum=1, Maximum=8)

ITEM NAME(CF) NUMBER()

Specifies the number of coupling facilities that can be defined within a policy. (Default=8, Minimum=1, Maximum=16)

ITEM NAME(STR) NUMBER()

Specifies the number of coupling facility structures that can be defined in one policy. (Default=50, Minimum=1, Maximum=512)

ITEM NAME(CONNECT) NUMBER()

Specifies the number of connections that can be defined for a coupling facility structure. (Default=32, Minimum=1, Maximum=255)

ITEM NAME(SMREBLD) NUMBER(1)

Specifies that this CFRM couple data set supports the system-managed rebuild process. Use this statement only when formatting a CFRM couple data set for systems at OS/390 Release 8 or higher. (Note that 1 is the only valid value for NUMBER for ITEM NAME(SMREBLD).)

Considerations for System-Managed Process Support

The following considerations apply when using a CFRM couple data set formatted to support system-managed processes:

- Only systems at OS/390 Release 8 and higher are capable of using a CFRM couple data set formatted for system-managed rebuild. Once a version of the CFRM couple data set that has been formatted to support the system-managed rebuild process is in use as the primary or alternate CFRM couple data set, if any systems at a level below OS/390 Release 8 join the sysplex, those systems will be unable to access the CFRM couple data set. The inability to access the CFRM couple data set means that those systems cannot exploit any CFRM function, such as connecting to a coupling facility structure, submitting coupling facility requests, and issuing DISPLAY XCF,CF and DISPLAY XCF,STR commands.
- Once a version of a CFRM couple data set that is formatted to support system-managed processes is made the primary CFRM couple data set, the sysplex cannot revert to a CFRM couple data set that was formatted at a lower version without a sysplex-wide IPL. The sysplex will not permit a data set formatted for less capability to become the alternate CFRM couple data set.
 - The SETXCF command to identify an alternate couple data set
`SETXCF COUPLE,ACOUPLE=(alternatedsname,alternatevolume)`

 will be rejected if “alternatedsname” is not also a CFRM couple data set formatted to support system-managed rebuild.
 - The first system that IPLs into a sysplex cannot specify in COUPLExx a version of a primary CFRM couple data set that has been formatted to support system-managed rebuild and a version of an alternate CFRM couple data set that has not been formatted to support that process. The sysplex will reject the alternate and will operate without an alternate until a version of a CFRM couple data set that has been formatted to support system-managed rebuild is made available as the alternate CFRM couple data set.

Considerations for Supporting More than 255 Structures

The following considerations apply when defining more than 255 structures (up to a limit of 512) in a CFRM policy:

- Only systems with OW19974 installed or which are at OS/390 Release 9 and higher are capable of using a CFRM couple data set formatted to support from 256 to 512 structures. Once a version of the CFRM couple data set that has been formatted to support more than 255 structures is made the primary CFRM couple data set, other systems in the sysplex without this support installed will be unable to access the CFRM couple data set. The inability to access the CFRM couple data set means that those systems cannot exploit any CFRM function, such as connecting to a coupling facility structure, submitting coupling facility requests, and issuing DISPLAY XCF,CF and DISPLAY XCF,STR commands. In order to allow the systems without this support to access the CFRM couple data set, it will be necessary to revert back from a primary CFRM couple data set formatted for more than 255 structures to one formatted for 255 structures or less. To do so will require a sysplex-wide IPL.
- Once a version of a CFRM couple data set that is formatted to support more than 255 structures is made the primary CFRM couple data set, the sysplex

Format Utility

cannot revert to a CFRM couple data set that was formatted at a lower version without a sysplex-wide IPL. The sysplex will not permit a data set formatted for less capability to become the alternate CFRM couple data set.

- The SETXCF command to identify an alternate couple data set
SETXCF COUPLE,ACOUPL=(alternatedsname,alternatevolume)

will be rejected if “alternatedsname” is not also a CFRM couple data set formatted to support more than 255 structures.

- The first system that IPLs into a sysplex cannot specify in COUPLExx a version of a primary CFRM couple data set that has been formatted to support more than 255 structures and a version of an alternate CFRM couple data set that has not been formatted to support more than 255 structures. The sysplex will reject the alternate and will operate without an alternate until a version of a CFRM couple data set that has been formatted to support more than 255 structures is made available as the alternate CFRM couple data set.

LOGR Parameters for Format Utility

For a LOGR policy — DATA TYPE(LOGR), valid data names are LSR and LSTRR.

ITEM NAME(LSR) NUMBER()

Specifies the maximum number of log streams that can be defined to the LOGR policy. Each LSR record consumes one track in the LOGR couple data set. (Default=1, Minimum=1, Maximum=32767)

ITEM NAME(LSTRR) NUMBER()

Specifies the maximum number of structure names that can be defined to the LOGR policy. Each LSTRR record consumes one track in the LOGR couple data set. (Default=1, Minimum=1, Maximum=32767)

ITEM NAME(DSEXTENT) NUMBER()

Specifies the number of additional log data set directory extents to define for log stream log data sets. Use this parameter to increase the potential log data set directory capacity for a sysplex. Each additional directory extent specified goes into a common pool available to any log stream in the sysplex. System logger allocates these directory extents as needed when a log stream runs out of DASD log data set directory space. Each directory extent allows a log stream to extend by 168 log data sets. Each DSEXTENT record consumes one track in the LOGR couple data set. (Default=0, Minimum=0, Maximum=99998)

See “Increasing the Space Available for DASD Log Data Sets” on page 207 for guidance on the DSEXTENT parameter.

SFM Parameters for Format Utility

For an SFM couple data set — DATA TYPE(SFM), valid data names are POLICY, SYSTEM, and RECONFIG.

ITEM NAME(POLICY) NUMBER()

Specifies the number of administrative policies that can be defined. (Default=9, Minimum=1, Maximum=50)

ITEM NAME(SYSTEM) NUMBER()

Specifies the number of systems for which actions and weights can be defined. This should be the maximum number of systems that will be in the sysplex that the policy will govern. Note that the number specified does not need to include those systems identified by NAME(*), for which policy default values are applied. (Default=8, Minimum=0, Maximum=32)

ITEM NAME(RECONFIG) NUMBER()

Specifies the number of reconfigurations involving PR/SM partitions that can be specified. (Default=0, Minimum=0, Maximum=50)

OS/390 UNIX Parameters for Format Utility

For an OMVS couple data set — DATA TYPE(BPXMCD), valid data names are MOUNTS and AMTRULES.

ITEM NAME(MOUNTS) NUMBER()

Specifies the number of MOUNTS that can be supported by OS/390 UNIX. (Default=100, Minimum=1, Maximum=50000)

ITEM NAME(AMTRULES) NUMBER()

Specifies the number of automount rules that can be supported by OS/390 UNIX. (Default=50, Minimum=50, Maximum=1000)

To use shared Hierarchical File System (HFS) in a sysplex environment, all systems must be running OS/390 Version 2 Release 9 or higher. Systems using earlier releases of OS/390 cannot share the HFS.

The NUMBER (nnnn) specified for mounts and automount rules (a generic or specific entry in an automount map file) is directly linked to function performance and the size of the OS/390 UNIX couple data set. If maximum values are specified, the size of the couple data set will increase accordingly and the performance level for reading and updating it will decline. Conversely, if the NUMBER values are too small, the function (for example, the number of mounts supported) would fail once the limit is reached. However, if such a failure occurs, a new couple data set can be formatted and switched in with larger values specified for NUMBER. The number of file systems required (factoring in an additional number to account for extra mounts), will determine your minimum and maximum NUMBER value.

After the couple data sets are defined, they must be identified to XCF for use by OS/390 UNIX through either COUPLExx or with the SETXCF command.

See “Tuning OS/390 UNIX Performance in a Sysplex” on page 131 for additional information about using a shared HFS in a sysplex.

WLM Parameters for Format Utility

For a WLM couple data set — DATA TYPE(WLM), valid data names are POLICY, WORKLOAD, SRVCLASS, SVDEFEXT, SVDCREXT, APPLENV, SVAEAEXT, SCHENV, and SVSEAEXT.

ITEM NAME(POLICY) NUMBER()

Specifies that an increment of space large enough to accommodate the specified number of policies be allocated in the WLM couple data set. (Default=5, Minimum=1, Maximum=99)

ITEM NAME(WORKLOAD) NUMBER()

Specifies that an increment of space large enough to accommodate the specified number of workloads be allocated in the WLM couple data set. (Default=32, Minimum=1, Maximum=999)

ITEM NAME(SRVCLASS) NUMBER()

Specifies that an increment of space large enough to accommodate the specified number of service classes be allocated in the WLM couple data set. (Default=128, Minimum=1, Maximum=999)

Format Utility

Note: As of OS/390 R3, WLM allows no more than 100 service classes to be defined in a service definition. The default, however, remains at the pre-OS/390 R3 value of 128. This will set aside as much space as you will ever need for service classes, as well as a little extra for other WLM objects.

If you are migrating from a pre-OS/390 R3 system and you previously reserved space for more than 128 service classes, do the following:

- If you intend to use SETXCF PSWITCH to make the new WLM couple data set active, continue to reserve space for the same number of service classes as you did in the earlier release. SETXCF PSWITCH will not allow a smaller WLM couple data set to be added as an alternate couple data set.
- If you intend to IPL the sysplex with the new WLM couple data set and install the service definition from a saved ISPF data set, then reserve space for as many service classes as you ever expect to define.

ITEM NAME(SVDEFEXT) NUMBER()

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition (IWMSVDEF) be allocated in the WLM couple data set. (Default=0, Minimum=0, Maximum=8092)

ITEM NAME(SVDCREXT) NUMBER()

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition Classification Rules (IWMSVDCR) be allocated in the WLM couple data set. (Default=0, Minimum=0, Maximum=8092)

ITEM NAME(APPLENV) NUMBER()

Specifies that an increment of space large enough to accommodate the specified number of application environments be allocated in the WLM couple data set. (Default=100, Minimum=1, Maximum=999)

ITEM NAME(SVAEAEXT) NUMBER()

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition Application Environment Area (IWMSVAEA) be allocated in the WLM couple data set. (Default=0, Minimum=0, Maximum=8092)

ITEM NAME(SCHENV) NUMBER()

Specifies that an increment of space large enough to accommodate the specified number of scheduling environments be allocated in the WLM couple data set. (Default=100, Minimum=1, Maximum=999)

ITEM NAME(SVSEAEXT) NUMBER()

Specifies that an exact amount of space (in K bytes) for extension areas to the WLM Service Definition Scheduling Environment Area (IWMSVSEA) be allocated in the WLM couple data set. (Default=0, Minimum=0, Maximum=8092)

Return Codes

The return codes in decimal format from the format utility are:

Return Codes Description

0	Successful completion. All requested data sets are formatted.
4	One or more couple data sets could not be formatted.
8	An I/O error occurred on the SYSPRINT or SYSIN file. Some couple data sets might have been formatted.
12	The SYSPRINT DD or SYSIN DD was not specified or could not be used. No data sets are formatted.

- 16 An abend occurred during format utility processing. Some couple data sets might have been formatted.
- 20 Unknown error while parsing command input. Some couple data sets might have been formatted.

Examples of Using the IXCL1DSU Format Utility

The examples shown here for OS/390 are available in SYS1.SAMPLIB.

Formatting a Sysplex Couple Data Set

Sample JCL to run the format utility to format a sysplex couple data set is shipped with OS/390 in SYS1.SAMPLIB member IXCSYSPF. Figure 56 shows the sample JCL, with comments.

```
//IXCSYSPF JOB
/*
/* SAMPLE JCL TO FORMAT THE PRIMARY AND/OR ALTERNATE COUPLE DATA SETS
/* - SYSPLEX COUPLE DATA SETS
/*
/* COUPLE DATA SET ALLOCATION RULES:
/*
/* 1. SYSPLEX NAME IS REQUIRED AND IS 1-8 CHARACTERS
/* 2. STATUS IS OPTIONAL AND DEFAULTS TO SYSTEM, RECOVERY, GROUP
/* AND MEMBER
/* 3. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
/* MESSAGES
/* 4. SYSIN DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
/* CONTROL STATEMENTS
/*
/*
/******
//STEP1 EXEC PGM=IXCL1DSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        DEFINEDS SYSPLEX(PLEX1)
                DSN(SYS1.XCF.CDS01) VOLSER(3380X1)
                MAXSYSTEM(8)
                CATALOG
                DATA TYPE(SYSPLEX)
                        ITEM NAME(GROUP) NUMBER(50)
                        ITEM NAME(MEMBER) NUMBER(120)
                        ITEM NAME(GRS) NUMBER(1)
        DEFINEDS SYSPLEX(PLEX1)
                DSN(SYS1.XCF.CDS02) VOLSER(3380X2)
                MAXSYSTEM(8)
                CATALOG
                DATA TYPE(SYSPLEX)
                        ITEM NAME(GROUP) NUMBER(50)
                        ITEM NAME(MEMBER) NUMBER(120)
                        ITEM NAME(GRS) NUMBER(1)
/*
```

Note: If you want the couple data set to reside on an SMS-managed volume, specify the storage and management classes on the DEFINEDS statement. The parameter for storage class is STORCLAS. The parameter for management class is MGMTCLAS. See the explanation of the parameters for DEFINEDS.

Figure 56. IXCSYSPF: Sample OS/390 JCL for Formatting Sysplex Couple Data Set

Format Utility

Sample JCL Provided Prior to OS/390

The following sample JCL to run the format utility was shipped with MVS/ESA SP Version 4. The sample JCL, with comments, is shown in Figure 57.

```
//SAMPXCF JOB
//*
/* JOB TO FORMAT PRIMARY AND/OR ALTERNATE COUPLE DATA SETS
/* USED BY THE XCF COMPONENT OF THE MVS BCP.
/*
/* COUPLE DATA SET ALLOCATION RULES:
/*
/* 1. SYSPLEX NAME IS REQUIRED AND IS 1-8 CHARACTERS
/* 2. MAXMEMBER IS OPTIONAL AND DEFAULTS TO UTILIZE FULL TRACKS
/*    MAXMEMBER MUST BE AT LEAST 8 AND LESS THAN OR EQUAL TO
/*      1023
/* 3. MAXGROUP IS OPTIONAL AND DEFAULTS TO 10.
/*    MAXGROUP MUST BE AT LEAST 10 AND LESS THAN OR EQUAL TO
/*      2045
/* 4. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
/*    MESSAGES
/* 5. SYSIN DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
/*    CONTROL STATEMENTS
/*
//STEP1 EXEC PGM=IXCL1DSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
    DEFINEDS DSN(SYS1.XCF.CDS01) VOLSER(3380X1) SYSPLEX(PLEX1)
             MAXMEMBER(50) MAXGROUP(20) CATALOG
    DEFINEDS DSN(SYS1.XCF.CDS02) VOLSER(3380X2) SYSPLEX(PLEX1)
             MAXMEMBER(50) MAXGROUP(20) CATALOG
/*
```

Note: If you want the couple data set to reside on an SMS-managed volume, specify the storage and management classes on the DEFINEDS statement. The parameter for storage class is STORCLAS. The parameter for management class is MGMTCLAS. See the explanation of the parameters for DEFINEDS.

Figure 57. SAMPXCF: Sample SP Version 4 JCL for Formatting Couple Data Sets

The format utility control statements and JCL in Figure 57 are described previously. The MAXGROUP and MAXMEMBER keywords are maintained for compatibility and are mutually exclusive with the DATA keyword. If the MAXGROUP and/or MAXMEMBER keywords are specified, then a sysplex couple data set is created, and the DATA and ITEM keywords are ignored.

Formatting an Automatic Restart Management Couple Data Set

Sample JCL to run the format utility for format couple data sets for automatic restart management is shipped in SYS1.SAMPLIB member IXCARMF. The sample shows formatting both a primary and an alternate couple data set for automatic restart management.

```
//IXCARMF JOB
//*
/* SAMPLE JCL TO FORMAT A PRIMARY AND AN ALTERNATE COUPLE DATA SET
/* FOR AUTOMATIC RESTART MANAGEMENT. DATASETS SYS1.MIGLIB.ARMCPLO1
/* AND SYS1.MIGLIB.ARMCPLO2, WILL BE ALLOCATED. IF THE SIZE OF
/* THE DATASETS IS DIFFERENT, THE LARGER SHOULD BE USED AS THE
/* ALTERNATE.
/*
/* NOTES:
/* 1. REGION=2M IS RECOMMENDED WHEN INVOKING IXCL1DSU TO ENSURE
/* ENOUGH SPACE.
/* 2. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
/* MESSAGES
/* 3. SYSIN DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
/* CONTROL STATEMENTS
/* 4. SYSPLEX NAME IS REQUIRED AND IS 1-8 CHARACTERS
/* 5. DSN AND VOLSER ARE REQUIRED FOR DATA SET SPECIFICATION
/* 6. DATA TYPE(ARM) STATEMENT IS REQUIRED TO INDICATE WHAT TYPE
/* OF COUPLE DATASET IS TO BE ALLOCATED
/*
/******
/*
```

Figure 58. IXCARMF - JCL for Utility to Format ARM Couple Data Set (Part 1 of 4)

```
/*
//STEP1 EXEC PGM=IXCL1DSU,REGION=2M
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSIN DD *
/* Allocating a primary Couple Dataset*/
DEFINEDS SYSPLEX(PLEX1)
DSN(SYS1.MIGLIB.ARMCPLO1) VOLSER(3380X1)
NOCATALOG
```

Figure 58. IXCARMF - JCL for Utility to Format ARM Couple Data Set (Part 2 of 4)

```
DATA TYPE(ARM)
/* Note that, if any of the
/* following ITEM NAME(xxx) and
/* its NUMBER(n) are not specifi-
/* ed, default value for that
/* item will take effect. */
ITEM NAME(POLICY) NUMBER(5) /* Maximum number of user-defined
/* ARM policies that will be on
/* the couple data set at once
/* Defaults to 3. */
ITEM NAME(MAXELEM) NUMBER(25) /* Maximum number of elements
/* per policy.
/* Defaults to 10. */
```

Figure 58. IXCARMF - JCL for Utility to Format ARM Couple Data Set (Part 3 of 4)

Format Utility

```
ITEM NAME(TOTELEM) NUMBER(20) /* Maximum number of elements that
                                are anticipated to be register-
                                ed with ARM across the sysplex
                                at once.
                                Defaults to 500.          */
                                /* Allocating an alternate Couple Dataset*/
DEFINEDS SYSPLEX(PLEX1)
        DSN(SYS1.MIGLIB.ARMCPLO2) VOLSER(3380X2)
        NOCATALOG
DATA TYPE(ARM)
        ITEM NAME(POLICY) NUMBER(7)
        ITEM NAME(MAXELEM) NUMBER(35)
        ITEM NAME(TOTELEM) NUMBER(30)
/*
```

Figure 58. IXCARMF - JCL for Utility to Format ARM Couple Data Set (Part 4 of 4)

Formatting a CFRM Couple Data Set

Sample JCL to run the format utility for formatting couple data sets for the coupling facility resource management service is shipped in SYS1.SAMPLIB member IXCCFRMF. The sample shows formatting both a primary and an alternate couple data set for coupling facility resource management.

```

//IXCCFRMF JOB
//*
//* SAMPLE JCL TO FORMAT THE PRIMARY AND/OR ALTERNATE COUPLE DATA SETS
//* - COUPLE DATA SETS FOR COUPLING FACILITY RESOURCE MANAGEMENT (CFRM)
//*
//* COUPLE DATA SET ALLOCATION RULES:
//*
//* 1. SYSPLEX NAME IS REQUIRED AND IS 1-8 CHARACTERS
//* 2. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
//*    MESSAGES
//* 3. SYSIN DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
//*    CONTROL STATEMENTS
//* 4. FORMATTING OF LARGE CFRM COUPLE DATA SETS (E.G., WITH MORE THAN
//*    255 STRUCTURES DEFINED) MAY REQUIRE MODIFICATION OF THE REGION
//*    PARAMETER AND/OR THE IEFUSI EXIT SO AS TO SATISFY THE VIRTUAL
//*    STORAGEREQUIREMENTS OF THE FORMAT UTILITY.
//*
//*
//*****
//STEP10 EXEC PGM=IXCLDSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        DEFINEDS SYSPLEX(PLEX1)
            DSN(SYS1.XCF.FDS01) VOLSER(3380X1)
            NOCATALOG
            DATA TYPE(CFRM)
                ITEM NAME(POLICY) NUMBER(6)
                ITEM NAME(CF) NUMBER(8)
                ITEM NAME(STR) NUMBER(50)
                ITEM NAME(CONNECT) NUMBER(32)
        DEFINEDS SYSPLEX(PLEX1)
            DSN(SYS1.XCF.FDS02) VOLSER(3380X2)
            NOCATALOG
            DATA TYPE(CFRM)
                ITEM NAME(POLICY) NUMBER(6)
                ITEM NAME(CF) NUMBER(8)
                ITEM NAME(STR) NUMBER(50)
                ITEM NAME(CONNECT) NUMBER(32)
/*

```

Note: If you want the couple data set to reside on an SMS-managed volume, specify the storage and management classes on the DEFINEDS statement. The parameter for storage class is STORCLAS. The parameter for management class is MGMTCLAS. See the description of the parameters for DEFINEDS.

Figure 59. IXCCFRMF - JCL for Utility to Format CFRM Couple Data Set

Formatting a LOGR Couple Data Set

Use the sample JCL to run the format utility for formatting couple data sets for the system logger service. The sample shows formatting for both a primary and alternate couple data set for the LOGR policy.

Format Utility

```
//INVCDS    JOB
//*
//* SAMPLE JCL TO FORMAT THE PRIMARY AND ALTERNATE
//* COUPLE DATA SETS FOR SYSTEM LOGGER (LOGR)
//*
//* COUPLE DATA SET ALLOCATION RULES:
//*
//* 1. SYSPLEX NAME IS REQUIRED AND IS 1-8 CHARACTERS
//* 2. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
//*    MESSAGES
//* 3. SYSIN DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
//*    CONTROL STATEMENTS
//*
//*
//*****
//STEP1     EXEC  PGM=IXCL1DSU
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
        DEFINEDS SYSPLEX(XLSDEV)
                DSN(SLC.FDSS1) VOLSER(3380X1)
        DATA TYPE(LOGR)
                ITEM NAME(LSR) NUMBER(10)
                ITEM NAME(LSTRR) NUMBER(10)
                ITEM NAME(DSEXTENT) NUMBER(20)
        DEFINEDS SYSPLEX(XLSDEV)
                DSN(SLC.FDSS2) VOLSER(3380X2)
        DATA TYPE(LOGR)
                ITEM NAME(LSR) NUMBER(10)
                ITEM NAME(LSTRR) NUMBER(10)
                ITEM NAME(DSEXTENT) NUMBER(20)
/*
```

Note: If you want the LOGR policy to reside on an SMS-managed volume, specify the storage and management classes on the DEFINEDS statement. The parameter for storage class is STORCLAS. The parameter for management class is MGMTCLAS. See the description of the DEFINEDS parameters.

Figure 60. JCL for Utility to Format LOGR Couple Data Set

Formatting an SFM Couple Data Set

Sample JCL to run the format utility for formatting couple data sets for the sysplex failure management service is shipped in SYS1.SAMPLIB member IXCSFMF and is shown in Figure 61 on page 257. The sample shows formatting for both primary and alternate for the couple data set for Sysplex Failure Management.

```

//IXCSFMF JOB
//*
//* SAMPLE JCL TO FORMAT THE PRIMARY AND/OR ALTERNATE COUPLE DATA SETS
//* - COUPLE DATA SETS FOR SYSPLEX FAILURE MANAGEMENT (SFM)
//*
//* COUPLE DATA SET ALLOCATION RULES:
//*
//* 1. SYSPLEX NAME IS REQUIRED AND IS 1-8 CHARACTERS
//* 2. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
//*    MESSAGES
//* 3. SYSIN DD IS A REQUIRED DD STATEMENT FOR FORMAT UTILITY
//*    CONTROL STATEMENTS
//*
//*
//*****
//STEP10 EXEC PGM=IXCL1DSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
        DEFINEDS SYSPLEX(PLEX1)
                DSN(SYS1.XCF.SFMCDS01) VOLSER(3380X1)
                NOCATALOG
        DATA TYPE(SFM)
                ITEM NAME(POLICY)          NUMBER(9)
                ITEM NAME(SYSTEM)         NUMBER(8)
                ITEM NAME(RECONFIG)       NUMBER(4)

        DEFINEDS SYSPLEX(PLEX1)
                DSN(SYS1.XCF.SFMCDS02) VOLSER(3380X2)
                NOCATALOG
        DATA TYPE(SFM)
                DATA NAME(POLICY)        NUMBER(9)
                DATA NAME(SYSTEM)       NUMBER(8)
                DATA NAME(RECONFIG)     NUMBER(4)

/*

```

Note: If you want the couple data set to reside on an SMS-managed volume, specify the storage and management classes on the DEFINEDS statement. The parameter for storage class is STORCLAS. The parameter for management class is MGMTCLAS. See the description of the parameters for DEFINEDS.

Figure 61. IXCSFMF - JCL for Utility to Format SFM Couple Data Set

Formatting an OS/390 UNIX System Services Couple Data Set

Sample JCL to run the format utility to format an OS/390 UNIX System Services couple data set is shipped with OS/390 in SYS1.SAMPLIB member BPXISCD5. It allows for specification of the number of mount records that will be supported by the couple data set. Figure 62 on page 258 shows the sample JCL with comments. The sample shows formatting for both primary and alternate for the couple data set for OS/390 UNIX System Services.

Format Utility

```
//BPXISCD S JOB
//*
//* SAMPLE JCL TO FORMAT THE PRIMARY AND ALTERNATE COUPLE DATA SETS
//* FOR OS/390 UNIX SYSTEM SERVICES
//*
//*****
//*
//* Before running this job:
//* 1. Update the job card as required for your installation
//* 2. The STEPLIB DD should indicate the library containing
//*    the XCF CDS Utility (IXCL1DSU) and OS/390 UNIX exit support
//*    (BPXMCDS P and BPXMCDS D)
//* 3. Change the control statement input to the XCF CDS Utility
//*    as indicated below
//*****
//*
//* JCL statements:
//* 1. SYSPRINT DD is a required DD statement for XCF CDS Utility
//*    messages
//* 2. SYSIN DD is a required DD statement for XCF CDS Utility
//*    control statements
//*
//* Control statements:
//* 1. SYSPLEX name is required and is 1-8 characters
//*    The name should be for the sysplex in which the OS/390 UNIX
//*    CDS is to be used.
//* 2. Update the DSN and VOLSER
//* 3. Indicate CATALOG or NOCATALOG
//* 4. Update NUMBER(nnnn) for MOUNTS
//* 5. Update NUMBER(nnnn) for AMTRULES.
//*****
```

Figure 62. BPXISCD S: Sample OS/390 JCL for Formatting an OS/390 UNIX Couple Data Set (Part 1 of 2)

```

//*****
//STEP10 EXEC PGM=IXCL1DSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
/* Begin definition for OS/390 UNIX couple data set (1) */
   DEFINEDS SYSPLEX(PLEX1) /* Name of the sysplex in
                             which the OS/390 UNIX couple
                             data set is to be used. */
   DSN(SYS1.OMVS.CDS01) VOLSER(3390X1) /* The name and
                                         volume for the OS/390 UNIX
                                         couple data set. The
                                         utility will allocate a
                                         new data set by the name
                                         specified on the volume
                                         specified. */
   NOCATALOG /* Default is to CATALOG. */
   DATA TYPE(BPXMCD) /* The type of data in the
                        data set being created is
                        for OS/390 UNIX. BPXMCD is
                        the TYPE for OS/390 UNIX. */
   ITEM NAME(MOUNTS) NUMBER(500) /* Specifies the number of
                                   MOUNTS that can be supported
                                   by OS/390 UNIX.
                                   Default = 100
                                   Minimum = 1
                                   Maximum = 50000 */
   ITEM NAME(AMTRULES) NUMBER(50) /* Specifies the number
                                   of automount rules that can
                                   be supported by OS/390 UNIX.
                                   Default = 50
                                   Minimum = 50
                                   Maximum = 1000 */

/* Begin definition for OS/390 UNIX couple data set (2) */
   DEFINEDS SYSPLEX(PLEX1)
   DSN(SYS1.OMVS.CDS02) VOLSER(3390X2)
   NOCATALOG
   DATA TYPE(BPXMCD)
   ITEM NAME(MOUNTS) NUMBER(500)
   ITEM NAME(AMTRULES) NUMBER(50)
/*

```

Figure 62. BPXISCD: Sample OS/390 JCL for Formatting an OS/390 UNIX Couple Data Set (Part 2 of 2)

Formatting a WLM Couple Data Set

Sample JCL to run the format utility for formatting a primary couple data set for workload management is shipped in SYS1.SAMPLIB member IWMFTCDS. Figure 63 on page 260 shows a sample JCL job for formatting a couple data set.

Format Utility

```
//STEP1 EXEC PGM=IXCL1DSU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  DEFINEDS SYSPLEX(PLEX1)
           DSN(SYS1.WLMCDS01) VOLSER(TEMPAK)
           MAXSYSTEM(32)
           CATALOG
  DATA TYPE(WLM)
        ITEM NAME(POLICY) NUMBER(10)
        ITEM NAME(WORKLOAD) NUMBER(35)
        ITEM NAME(SRVCLASS) NUMBER(30)
        ITEM NAME(SVDEFEXT) NUMBER(5)
        ITEM NAME(SVDCREXT) NUMBER(5)
        ITEM NAME(APPLENV) NUMBER(100)
        ITEM NAME(SVAEAEXT) NUMBER(5)
        ITEM NAME(SCHENV) NUMBER(100)
        ITEM NAME(SVSEAEXT) NUMBER(5)
/*
```

Figure 63. Sample JCL to Format WLM Couple Data Set

Sample Output

Following are input statements and output messages from IXCL1DSU formatting a single couple data set for CFRM, SFM, WLM, ARM and LOGR.

```

DEFINEDS SYSPLEX(XLSDEV)
  DSN(SYS1.XCF.FDS01) VOLSER(3380X1)
  MAXSYSTEM(32)
  CATALOG
  DATA TYPE(CFRM)
    ITEM NAME(POLICY)  NUMBER(6)
    ITEM NAME(CF)      NUMBER(5)
    ITEM NAME(STR)     NUMBER(20)
    ITEM NAME(CONNECT) NUMBER(32)
  DATA TYPE(SFM)
    ITEM NAME(POLICY)  NUMBER(9)
    ITEM NAME(SYSTEM)  NUMBER(8)
    ITEM NAME(RECONFIG) NUMBER(4)
  DATA TYPE(WLM)
    ITEM NAME(POLICY)  NUMBER(10)
    ITEM NAME(WORKLOAD) NUMBER(35)
    ITEM NAME(SRVCLASS) NUMBER(30)
    ITEM NAME(SVDEFEXT) NUMBER(5)
    ITEM NAME(SVDCREXT) NUMBER(5)
    ITEM NAME(APPLENV)  NUMBER(100)
    ITEM NAME(SVAEAEXT) NUMBER(5)
    ITEM NAME(SCHENV)   NUMBER(100)
    ITEM NAME(SVSEAEXT) NUMBER(5)
  DATA TYPE(ARM)
    ITEM NAME(POLICY)  NUMBER(5)
    ITEM NAME(MAXELEM) NUMBER(25)
    ITEM NAME(TOTELEM) NUMBER(20)
  DATA TYPE(LOGR)
    ITEM NAME(LSR)     NUMBER(20)
    ITEM NAME(LSTRR)   NUMBER(50)

```

Figure 64. Sample Output of Couple Data Set Format Utility (Part 1 of 2)

```

IGD100I 0182 ALLOCATED TO DDNAME COUPLE  DATACLAS (      )
IXC292I DATA SET FORMATTING COMPLETE: DATA SET REQUIRES 208 TRACKS ON VOLSER
3380X1
IXC292I 1 RECORDS FORMATTED WITH 6 POLICY ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 5 CF ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 20 STR ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 32 CONNECT ITEMS EACH

IXC292I 1 RECORDS FORMATTED WITH 9 POLICY ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 8 SYSTEM ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 4 RECONFIG ITEMS EACH

IXC292I 10 POLICY RECORDS FORMATTED
IXC292I 35 WORKLOAD RECORDS FORMATTED
IXC292I 30 SRVCLASS RECORDS FORMATTED
IXC292I 5 SVDEFEXT RECORDS FORMATTED
IXC292I 5 SVDCREXT RECORDS FORMATTED
IXC292I 100 APPLENV RECORDS FORMATTED
IXC292I 5 SVAEAEXT RECORDS FORMATTED
IXC292I 100 SCHENV RECORDS FORMATTED
IXC292I 5 SVSEAEXT RECORDS FORMATTED

IXC292I 1 RECORDS FORMATTED WITH 20 TOTELEM ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 5 POLICY ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 25 MAXELEM ITEMS EACH

IXC292I 1 RECORDS FORMATTED WITH 20 LSR ITEMS EACH
IXC292I 1 RECORDS FORMATTED WITH 50 LSTRR ITEMS EACH

```

Figure 64. Sample Output of Couple Data Set Format Utility (Part 2 of 2)

Appendix B. Administrative Data Utility

Use the administrative data utility to add, update, or delete policy data on the formatted ARM, CFRM, LOGR, and SFM couple data sets. The number of administrative policies you can add is determined by the number specified on the POLICY control statement in the couple data set format utility.

WLM USERS

Note that the administrative data utility does not support WLM policy information. Those policies are stored in a formatted WLM couple data set with an ISPF application. See *OS/390 MVS Planning: Workload Management* for a description of the ISPF application and how to use it.

The control statements for this utility program follow standard conventions for JCL statements. The utility accepts 80-byte records with one or more parameters per record. See *OS/390 MVS JCL Reference* for additional information about JCL statements.

The administrative data utility is portable; that is, you can run the utility on MVS/ESA Version 3 systems and above.

Using the Administrative Data Utility

The name of the administrative data utility is IXCMIAPU. The program resides in SYS1.MIGLIB (which is logically appended to the LINKLIST), which makes it available through STEPLIB on a non-OS/390 MVS system.

Authorizing Use of the Utility

You must control the use of the IXCMIAPU utility through the OS/390 Security Server, which includes RACF, or your installation's security package. The security administrator must define a resource profile for the resource name associated with the following policies if the installation plans to use them:

- For automatic restart management, the resource name is 'MVSADMIN.XCF.ARM'.
- For CFRM, the resource name is 'MVSADMIN.XCF.CFRM'.
- For LOGR, the resource name is 'MVSADMIN.LOGR'.
- For SFM, the resource name is 'MVSADMIN.XCF.SFM'.

If you are using the OS/390 Security Server (RACF), define these FACILITY class profiles.

```
RDEFINE FACILITY MVSADMIN.XCF.ARM UACC(NONE)
RDEFINE FACILITY MVSADMIN.XCF.CFRM UACC(NONE)
RDEFINE FACILITY MVSADMIN.LOGR UACC(NONE)
RDEFINE FACILITY MVSADMIN.XCF.SFM UACC(NONE)
```

Assign UPDATE access authority to users who must alter or maintain the policy; assign READ access authority to users who require reports on the policy, but who will not change the policy. (These FACILITY class profiles will be used for authority checking on an ACTIVE couple data set. When a couple data set is INACTIVE, the rules for normal data set protection apply.)

Administrative Data Utility

After giving access authority to the appropriate users, activate the FACILITY class.
SETROPTS CLASSACT(FACILITY)

Coding the Administrative Data Utility

For examples of how to set up your JCL and code the control statements for administrative policy data, see the following figures:

- For automatic restart management, see Figure 65 on page 303.
- For CFRM, see Figure 68 on page 306.
- For LOGR, see Figure 69 on page 307.
- For SFM, see Figure 70 on page 308.

The utility control statements are described below:

PGM=IXCMIAPU

The administrative data utility program that is shipped with SP 5.1. (Program IXCM2APU also exists as an alias for IXCMIAPU.)

SYSPRINT

Describes where the output messages from the administrative data utility are to be printed. The SYSPRINT DD statement is required.

SYSABEND

Describes where SYSABEND data from the administrative data utility is to be printed. The SYSABEND DD statement is required.

SYSIN

Describes the utility control statements that are input to the utility.

DATA

Indicates that a couple data set is to be updated. The couple data set is identified with the following parameters of DATA.

TYPE(ARM)

TYPE(CFRM)

TYPE(LOGR)

TYPE(SFM)

Specifies the type of administrative data that the data set is to contain.

TYPE is a required parameter.

LOGR Policy-Specific Information

The information for the LOGR **type** of policy being updated is identified with specific parameters of the DATA statement.

See “LOGR Parameters for Administrative Data Utility” on page 281 for a description of the keywords to identify log streams and structures in a LOGR policy.

DSN(data-set-name)

Specifies the 1 - 44 character length name of the couple data set that was formatted with the IXCL1DSU format utility to contain the specified type of administrative data. The valid characters are alphanumeric characters (A-Z and 0-9), national characters (\$,@,#), and a period (.). The first character

Administrative Data Utility

must be either an alphabetic or a national character. DSN is an optional parameter when you are updating administrative data. When you do not specify DSN when updating administrative data, the utility applies the changes to the data set (or data sets, if both primary and alternate exist) that contains the active policy — CFRM, SFM, or ARM, and any administrative policies. The DSN parameter is not supported for TYPE(LOGR), since you can only update the active LOGR administrative policy. If you specify DSN for TYPE(LOGR), the IXCMIAPU utility ends with a non-zero return code and error message IXG432E.

Note that the changes are applied to the administrative policy. If the changes are to an administrative policy that is also the current active policy, the changes are applied to the administrative copy of the active policy and not directly to the active policy itself. To activate the changes in the active policy in the sysplex, issue a SETXCF command to make the updated administrative policy the active policy.

If you want to obtain a report of the policy information contained in the active couple data set, ensure that you omit the DSN keyword. Otherwise, the system issues allocation error messages.

When to Specify DSN

(An active couple data contains the current active policy.)

- DO specify DSN if you are creating your administrative policies initially or want to have a secondary set of policies in a couple data set that is not active in your sysplex.
- DO NOT specify DSN if you want to add a new policy or update an existing policy in the couple data set that is active in your sysplex.

See “Updating a CFRM Policy” on page 54 for an example of how to update a policy.

VOLSER(serial number)

Specifies the 1 - 6 character length name of the volume on which the couple data set resides if the data set is not cataloged. The valid characters are alphanumeric characters (A-Z and 0-9) and national characters (\$,@,#). VOLSER is an optional parameter and can be specified only if DSN parameter also is specified.

The VOLSER parameter is not supported for TYPE(LOGR), since you can only update the active LOGR administrative policy. If you specify VOLSER for TYPE(LOGR), the IXCMIAPU utility ends with a non-zero return code and error message IXG432E.

REPORT(YES)

REPORT(NO)

Specifies whether a report of the contents of the policy data for all administrative policies defined in the data set is to be output to the SYSPRINT file. The default is YES, to output a report.

Administrative Data Utility

Note: If you specify REPORT(YES) (or accept the default for the report keyword) with a DATA TYPE of LOGR, the requestor must have SAF read access to the MVSADMIN.LOGR resource to successfully obtain a report.

DEFINE POLICY NAME(*polname*)

Specifies the beginning of the definition of a new administrative policy. The limit for the number of policies that can be defined in a couple data set is established when the couple data set is formatted.

polname specifies the name of the policy. The name can be from 1 - 8 characters in length. Valid characters are A-Z and 0-9. The policy name must start with an alphabetic character.

A policy name can be defined only once in a single run of the administrative data utility.

ARM, CFRM, and SFM Policy-Specific Information

The information for the ARM, CFRM, and SFM **type** of policy being updated is identified with specific parameters of the DEFINE POLICY statement.

- See “Automatic Restart Management Parameters for Administrative Data Utility” on page 267 for a description of the keywords to define elements and restart groups in an automatic restart management policy.
- See “CFRM Parameters for Administrative Data Utility” on page 275 for a description of the keywords to identify coupling facilities and structures in a CFRM policy.
- See “SFM Parameters for Administrative Data Utility” on page 297 for a description of the keywords to identify systems in the sysplex and PR/SM reconfiguration scenarios in an SFM policy.

The LOGR policy does not use the DEFINE POLICY statement. See “LOGR Parameters for Administrative Data Utility” on page 281 for LOGR policy-specific keywords on the DATA statement.

REPLACE(NO)

REPLACE(YES)

Specifies whether or not the existing policy identified by *polname* is to be replaced.

Note: If the policy name currently exists in the data set, REPLACE(YES) causes the policy definition to be replaced with the policy being defined. REPLACE(NO) will not allow the policy definition in the data set to be replaced.

DELETE POLICY NAME(*polname*)

Specifies that the named administrative policy is to be deleted from the set of policies previously defined by the administrative data utility on the couple data set.

Automatic Restart Management Parameters for Administrative Data Utility

The automatic restart management policy identifies elements and restart groups with restart parameter values that differ from the policy default values.

Your installation does not need to specify a policy if the default values are acceptable, or if the application overrides the defaults when coding the IXCARM macro. The default values (if any) are described under each parameter, along with the IXCARM macro specifications that can be used to override these defaults.

Note: The automatic restart management policy values are determined when an element is restarted, not when the batch job or started task registers with automatic restart management.

RESTART_ORDER

Specifies the order in which elements in the same restart group are to become ready after they are restarted.

The RESTART_ORDER parameter applies to all restart groups.

RESTART_ORDER is optional. The default values for RESTART_ORDER are:

- LEVEL(0) - ELEMENT_TYPES: SYSIRLM, SYSLVL0
- LEVEL(1) - ELEMENT_TYPES: SYSDDB2, SYSIMS, SYSVTAM, SYSTCPIP, SYSLVL1
- LEVEL(2) - ELEMENT_TYPES: SYSCICS, SYSMQMGR, SYSMQCH, SYSLVL2
- LEVEL(3) - ELEMENT_TYPES: SYSCB, SYSLVL3

Any elements that do not match a default or policy-specified ELEMENT_TYPE or a policy-specified ELEMENT_NAME.

LEVEL(level)

Specifies the level associated with elements that must be restarted in a particular order. The elements are restarted from the lowest level to the highest level. The LEVEL keyword can be specified multiple times.

The set of elements associated with a particular level is identified by the ELEMENT_NAME or ELEMENT_TYPE parameters. ELEMENT_NAME or ELEMENT_TYPE must be specified with each LEVEL specification. (You can specify both ELEMENT_NAME and ELEMENT_TYPE.)

Level must be a decimal number from 0 through 65535.

The default is LEVEL(2). The IXCARM macro parameter that overrides the default specification is ELEMTYPE (which allows an element to give a name or type that is associated with another LEVEL).

ELEMENT_NAME(name-list)

Specifies the name of each element to be restarted at the *level* specified by LEVEL. The element name can contain the wildcard characters '?' or '*'. See "Using Wildcard Characters" on page 274 for more information about wildcard characters.

Name-list can be one or more element names separated by commas.

IBM reserves the use of element names starting with the characters A through I and "SYS" for IBM use.

Administrative Data Utility

ELEMENT_TYPE(type-list)

Specifies the element types that are to be restarted at the *level* specified by LEVEL. The element type is a name given to elements with similar characteristics. For example, PAYROLL might be the element type of multiple payroll-related programs (elements).

Type-list can be one or more element types separated by commas. You can specify the element type on the IXCARM macro by coding the parameter ELEMENTYPE.

IBM reserves the use of element types starting with the characters A through I and "SYS" for IBM use.

RESTART_GROUP(*Iname)

Identifies related elements that are to be restarted as a group if the system on which they are running fails.

Not all elements in a given restart group need to be running on the same system, or running at all.

The restart group name can:

- Be up to 16 characters long with no imbedded blanks
- Be a combination of alphabetic characters, numbers, and \$, #, and @
- Not begin with a number
- Be named DEFAULT to change the characteristics (target system, CSA requirements, and so on) of the default restart group
- Not be repeated
- Not contain wildcard characters (except for RESTART_GROUP(*))

The RESTART_GROUP parameter can be specified more than once in a policy.

RESTART_GROUP(*) is a special category and can be specified only once. A global substitution of the values specified here are applied to all restart groups that are not specified in the policy. For example, when the following is specified:

```
RESTART GROUP(*)  
  TARGET_SYSTEM(SYS1,SYS3)
```

only systems SYS1 and SYS3 can be the targets of a cross-system restart.

Another example: the following restart groups are specified in a policy:

```
RESTART_GROUP(*)  
  TARGET_SYSTEM(SYS1,SYS3)  
RESTART_GROUP(JIM)  
  TARGET_SYSTEM(SYS2)  
  ELEMENT(CICS1)  
  ELEMENT(DB28)
```

If a system fails, the elements listed in the restart group named JIM would be restarted only on SYS2. All other elements would be restarted on SYS1 or SYS3.

RESTART_GROUP is a required parameter. All elements that are not put into a specific restart group by a policy are in the restart group named DEFAULT.

TARGET_SYSTEM(*!system-list)

Specifies the systems on which elements can be restarted in a cross-system restart. Specify this parameter when you do not want all systems to be the targets of a cross-system restart.

Administrative Data Utility

The *system-list* must be one or more system names separated by commas. The order of the systems in *system-list* is irrelevant.

TARGET_SYSTEM(*) indicates that any system that supports automatic restart management might be the target of a cross-system restart. When the sysplex is running with WLM in goal mode, automatic restart management will choose the system that has the greatest available CPU capacity and that has the required available CSA (if the FREE_CSA parameter was specified for this restart group).

TARGET_SYSTEM is optional. TARGET_SYSTEM(*) is the default.

FREE_CSA(below,above)

Specifies the minimum amount of CSA and ECSA that must be available on a system for it to be eligible for a cross-system restart of the elements in the associated restart group.

Note: This parameter value will not disqualify a system as eligible for a cross-system restart if WLM is running in compatibility mode on that system.

below

Indicates the number of kilobytes (KB) of CSA that must be available on a target system for this restart group to be restarted. A value of 0 indicates that the amount of free CSA is not a requirement for eligible systems for this restart group.

The maximum decimal value that can be specified for this keyword is 16384.

The default is 0.

above

Indicates the number of kilobytes (KB) of ECSA that must be available on a target system for this restart group to be restarted. A value of 0 indicates that the amount of free ECSA is not a requirement for eligible systems for this restart group.

The maximum decimal value that can be specified for this keyword is 2080767.

The default is 0.

FREE_CSA is optional. The default is FREE_CSA(0,0).

RESTART_PACING(delay)

Specifies the amount of time in seconds that automatic restart management should delay between the restart of each element in the restart group. This value provides the ability to stagger the restarts of the elements to minimize contention on a system.

The value for *delay* must be a decimal number from 0 to 21600 seconds (6 hours).

This value applies only for cross-system restarts.

RESTART_PACING is optional. The default is RESTART_PACING(0).

Administrative Data Utility

ELEMENT(*Iname)

Specifies a batch job or started task that can register as an element of automatic restart management. The wildcard characters '?' or '*' can be used. See "Using Wildcard Characters" on page 274 for more information about wildcard characters.

In addition, automatic restart management provides two system symbols that relate to the policy syntax for automatic restarts:

IBM reserves the use of element names starting with the characters A through I and "SYS" for IBM use.

&SYSELEM.

Specifies that the element name from the previous ELEMENT statement should be substituted for this system symbol. For example:

```
ELEMENT(KAREN??)
  RESTART_METHOD(SYSTEM,STC,'S &SYSELEM.')
```

If an element named KAREN1A needs to be restarted, MVS substitutes 'S KAREN1A' for the command text.

Element names used for &SYSELEM. can not exceed 8 characters in length.

&SYSSUF.

Substitutes the suffix of an element name. The characters following the first wildcard character are considered the suffix.

For example, if element ELWOOD was being restarted and the best match from an installation policy was ELEMENT(EL*), then the suffix for this element would be WOOD. Similarly, when element PKASKOVICH matches the policy entry ELEMENT(PKAS??VI*), then the suffix for this element would be KOVICH.

The substitution text for &SYSSUF. cannot exceed 7 characters.

See "Using System Symbol Substitution" on page 274 for general information about using system symbols.

You can specify ELEMENT(*) only when RESTART_GROUP(*) or RESTART_GROUP(DEFAULT) is specified. When ELEMENT(*) is specified with RESTART_GROUP(*), the element-related parameters apply to all elements in all restart groups (if these parameters do not have explicitly specified values for those restart groups). When ELEMENT(*) is specified with RESTART_GROUP(DEFAULT), the associated parameters apply to all elements in this restart group.

ELEMENT is a required parameter.

RESTART_ATTEMPTS(max-number,time-interval)

Specifies the maximum number of times that automatic restart management should attempt to restart the specified element within a given interval. This limit prevents automatic restart management from continually restarting an element that is recursively terminating.

When this threshold has been reached, no further restarts will be attempted and the system will deregister the element.

max-number

Must be a decimal value from 0 to 3. When RESTART_ATTEMPTS(0) is specified, no restarts will be attempted for this element.

The default is 3.

time-interval

Must be a decimal number from 1 to 86400 seconds (24 hours).

The default is 300.

RESTART_ATTEMPTS is optional, but if specified, must refer to a specific ELEMENT.

The default is RESTART_ATTEMPTS(3,300).

RESTART_TIMEOUT(time-interval)

Specifies the maximum amount of time, in seconds, that an element is expected to take to issue the IXCARM macro with REQUEST=REGISTER after automatic restart management has restarted it.

If the re-registration does not occur within this interval, automatic restart management issues message IXC803I to inform the operator of the situation and deregisters the element. Any elements that were waiting for this element to become ready will be released from their wait.

time-interval

Must be a decimal number from 1 to 86400 seconds (24 hours).

Note: Many factors can affect how long it will take for a job to restart, so this value should not be set to less than 60 seconds. If *time-interval* is too small, automatic restart management can deregister an element that would have re-registered successfully. When the subsequent re-registration is received, automatic restart management will treat it as an initial registration.

RESTART_TIMEOUT is optional. The default is RESTART_TIMEOUT(300). The IXCARM macro parameter that overrides the default specification is RESTARTTIMEOUT.

READY_TIMEOUT(time-interval)

Specifies the maximum amount of time, in seconds, to wait for an element being restarted to issue the IXCARM macro with the REQUEST=READY parameter.

If this interval is exceeded, automatic restart management will release any elements that were waiting for this element to become ready. No message is issued for this condition and the element is not deregistered. (If a DISPLAY XCF,ARMSTATUS is done for this element, it will be in the available-to state.)

The value for *time-interval* must be a decimal number from 1 to 86400 seconds (24 hours).

READY_TIMEOUT is optional. The default is READY_TIMEOUT(300).

Administrative Data Utility

TERMTYPE(ALLTERM)

TERMTYPE(ELEMTERM)

Specifies under which conditions MVS should restart this element.

ALLTERM

Indicates that the element should be restarted for all unexpected failures as appropriate. The value specified on an IXCARM REQUEST=REGISTER request for the ELEMbind keyword determines what types of failures are appropriate.

ELEMTERM

Indicates that the element should be restarted only if the element itself terminates. It should not be restarted if the system terminates.

TERMTYPE is optional. The default is TERMTYPE(ALLTERM). The IXCARM macro parameter that overrides the default specification is TERMTYPE.

When the element is an abstract resource (has a bind to the system on which it is registered), the following applies:

- If the policy specifies TERMTYPE(ELEMTERM), the policy specification for TERMTYPE is ignored. The element will be restarted when the system fails. To suppress all restarts for an element, specify RESTART_ATTEMPTS(0).
- If the policy specifies TERMTYPE(ALLTERM), the policy specification for TERMTYPE is honored. (The element will be restarted when the system fails.)

RESTART_METHOD(event,restart-type)

Specifies restart text that automatic restart management is to use to restart the element.

Specify one of the following for *event*:

ELEMTERM

Indicates that the persistent restart text is to be overridden by the JCL data set or the command text specified in *restart-type* only when the **element** itself terminates.

SYSTEM

Indicates that the persistent restart text is to be overridden by the JCL data set or the command text specified in *restart-type* only when the **system** the element was running on terminates.

BOTH

Indicates that the persistent restart text is to be overridden by the JCL data set or the command text specified in *restart-type* when either the system the element was running on terminates, or when the element itself terminates.

Specify one of the following for *restart-type*:

PERSIST

Indicates that MVS is to use the persistent restart text. The persistent restart text is either the JCL or the started task command that previously started the element.

JOB,'jcl-source'

Indicates that MVS is to restart the element as a batch job. *'jcl-source'* is the name of the data set that contains the JCL for

restarting the element. This data set name must be enclosed within single quotes. The data set must have the same data set characteristics (for instance, LRECL) as standard procedure libraries.

The data set name can:

- Contain symbolic-substitution keywords (such as &SYSCONE)
- Be up to 72 characters in length (after symbolic-substitution keywords are resolved, the data set name cannot exceed 44 characters in length, and the member name cannot exceed 8 characters in length and must be enclosed in parentheses).

Note: Data set names are not validated when a policy is defined except as noted above. They are validated only when a restart is necessary for the element.

STC,'command-text'

Indicates that automatic restart management is to restart this element by issuing the command provided in '*command-text*'.

The command text:

- Must be enclosed in single quotes
- Can include symbolic substitution keywords
- Length can not exceed 126 characters
- Can contain a single quote(') but must be represented by specifying two single quotes('') in succession.
- Can contain lowercase characters but will be changed to uppercase when the utility program runs. Your command text must not be dependent on lowercase characters.

Note: Command text strings are not validated when a policy is defined except as specified above. They are validated only when a restart is necessary for this element.

Notes:

1. If the *jcl-source* or *command-text* will not fit on one line, it can be continued on another line by using single quotes to indicate segments. The quotes are not counted toward the input length. For example:

```
RESTART_METHOD(BOTH,STC,'S ELEMENT1,PARM1=ABC,'  
                  'PARM2=XYZ')
```

is equivalent to specifying:

```
RESTART_METHOD(BOTH,STC,'S ELEMENT1,PARM1=ABC,PARM2=XYZ')
```

2. The RESTART_METHOD can be specified once for ELEMTERM and once for SYSTERM per element. (Specifying RESTART_METHOD more than once for an element when BOTH is specified results in an error.)
3. The RESTART_METHOD can be specified by the element restart exit, IXC_ELEM_RESTART, if one is coded, and by the STARTTXT parameter of the IXCARM macro. If more than one restart method is specified for an element, the order of priority, from highest to lowest, is:
 - a. Element restart exit
 - b. Installation-written policy (if RESTART_METHOD is specified)
 - c. STARTTXT parameter of IXCARM macro

Administrative Data Utility

- d. Policy defaults
- 4. Blanks specified within quotes are significant.

RESTART_METHOD is optional. The default is RESTART_METHOD(BOTH, PERSIST). For started tasks and for elements that represent abstract resources, the IXCARM macro parameter that overrides the default specification is STARTTXT.

To specify the restart text for an element that has a bind to the system on which it is registered (an abstract resource) and restarts only on system termination, use one of the following:

```
RESTART_METHOD(SYSTEM,STC,'command-text')
```

where *command-text* is a command that causes the element to restart.

```
RESTART_METHOD(SYSTEM,JOB,'jcl-source')
```

where *jcl-source* is the name of the data set that contains the JCL for restarting the element.

Using Wildcard Characters

The ELEMENT_NAME and ELEMENT parameters can contain wildcard characters. The wildcard characters are:

- ? A question mark matches one and only one character, and can be specified more than once anywhere in the name.
- * An asterisk matches a series of characters, and can be specified only at the end of a name.

These characters allow a single set of element-related parameters in a policy to apply to multiple elements with similar names and needs.

If two or more element names with wildcard characters in a policy match the name of the element being restarted, the most specific takes priority. For example, if MVS is restarting ELEM1, and the policy contains elements with the names: ELEM?, ELEM*, and ELEM1*, the information for ELEM1* will be used.

Examples of Matching: If the element name is ABC?, this would match:

ABCD

but not:

ABCDE or ABC

If the element name is AB?D*, this would match

ABCDEFGHIK

and:

ABCD and ABCDE and ABYDEFGHI

Using System Symbol Substitution

Automatic restart management supports both system symbols that the system provides and static system symbols that your installation defines for:

- *jcl-source* on the RESTART_METHOD parameter
- *command-text* on the RESTART_METHOD parameter

When symbols are used with RESTART_METHOD, note that the symbols are resolved using the symbol table from the system where the job **initially registered** as an element of automatic restart management.

See the section on sharing parmlib members in *OS/390 MVS Initialization and Tuning Reference* for lists of system symbols that automatic restart management supports.

CFRM Parameters for Administrative Data Utility

The CFRM policy information describes the coupling facilities that will be used in the sysplex, and specifies the requirements for structure size and allocation within a coupling facility.

CF

Specifies the definition of a coupling facility within the scope of the named policy.

The location of the information needed to define the coupling facility in a CFRM policy depends on the processor on which the coupling facility is installed.

- For ES/9000 processors, the information is available from the processor controller. Values for the NAME and PARTITION parameters appear on the LPDEF frame. The TYPE, MFG, PLANT, SEQUENCE, SIDE, and CPCID values appear on the ECSTAT frame.
- For the S/390 CMOS processors, the information is available from the Hardware Management Console. The NAME and PARTITION can be obtained by viewing the IOCDS to be used at POR. The TYPE, MFG, PLANT, SEQUENCE, and CPCID appear on the CPC Details panel. The SIDE parameter does not apply to the CMOS processor.

The limit for the number of coupling facilities that can be defined in a policy is established when the CFRM couple data set is formatted.

The coupling facility must appear in the preference list of at least one coupling facility structure defined within the policy or the administrative policy utility flags the statement as an error.

NAME(cfname)

Specifies the 1 - 8 character length name of the coupling facility. The valid characters are uppercase alphabetic characters (A-Z), numeric characters (0-9), national characters (\$,@,#), or an underscore (_). The **cfname** must start with an alphabetic character (A-Z).

IBM recommends that you define your coupling facility name to match the name of the LPAR in which the coupling facility is to run.

NAME is a required parameter.

The following set of parameters describe the unique coupling facility that is defined in the policy.

TYPE(tttttt)

Specifies the 6-character machine type. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary.

TYPE is a required parameter.

Administrative Data Utility

MFG(mmm)

Specifies the 3-character manufacturer identification. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9).

MFG is a required parameter.

PLANT(pp)

Specifies the 2-character plant of manufacture code. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9).

PLANT is a required parameter.

SEQUENCE(nnnnnnnnnnnn)

Specifies the 12-character sequence number. The valid characters are uppercase alphabetic characters (A-Z) and numeric characters (0-9), padded with leading zeros if necessary.

SEQUENCE is a required parameter.

PARTITION(h)

Specifies the 1-hexadecimal digit qualifier (0-F) to uniquely identify and associate a coupling facility to a specific PR/SM partition.

PARTITION is a required parameter.

SIDE(0)

SIDE(1)

Specifies the 1-numeric character to identify the coupling facility to a specific physical side of a CPC running in Physically Partitioned mode. You should not split or merge physical sides on which a coupling facility resides.

Use caution when splitting or merging physical sides of a processor that contains a coupling facility. The action might change the SIDE information that identifies the coupling facility.

SIDE is an optional parameter. Omit this parameter when the coupling facility resides on a CPC that is not physically partitioned.

- SIDE(0) — Side 0 of a physically partitioned machine. Do not specify for a Single Image CPC.
- SIDE(1) — Side 1 of a physically partitioned machine. Do not specify for a Single Image CPC.

CPCID(nn)

Specifies the 2-hexadecimal digit qualifier (00-FF) to identify a coupling facility to a specific CPC in a CMOS processor. CPCID is an optional parameter.

DUMPSPACE(size)

Specifies the amount of space to be reserved in the coupling facility for dumping structures allocated in the coupling facility. The number is specified in units of 1K (1024 bytes.) The number can be 1 - 6 decimal digits long.

DUMPSPACE is an optional parameter. If omitted, no storage in the coupling facility is reserved for dumping.

See “Determining the Amount of Coupling Facility Dump Space” on page 53 for information about specifying the amount of dump space.

STRUCTURE

Specifies the definition of a structure within the scope of the named policy.

The limit for the number of structures that can be defined in a policy is established when the CFRM couple data set is formatted.

NAME(strname)

Specifies the 1-16 character name of the structure. The valid characters are numeric characters, uppercase alphabetic characters, national characters (\$,@,#), or an underscore (_). The *strname* must start with an alphabetic character (A-Z). IBM names begin with SYS, or the letters A through I.

Note that structures used for XCF signaling must begin with the letters IXC.

NAME is a required parameter.

SIZE(size)

Specifies the maximum amount of space to be allocated for the structure in the coupling facility. The number is specified in units of 1K (1024 bytes). The number must be 1 - 8 decimal digits long.

The maximum structure size is the largest to which the structure can be altered. If a structure size larger than the maximum is required, you must modify the CFRM policy to specify the larger size. When you start the modified policy, the system can reallocate the new structure, which likely will be disruptive to your operation.

Specifying too large a maximum structure size can waste coupling facility resources. Especially when the SIZE is significantly larger than INITSIZE, it may even cause the structure to become unallocatable, depending on the coupling facility and its CFLEVEL.

SIZE is a required parameter.

INITSIZE(initsize)

Specifies the initial amount of space to be allocated for the structure in the coupling facility. The number is specified in units of 1K (1024 bytes). The number must be 1 - 8 decimal digits long. The INITSIZE value must be less than or equal to the SIZE value. Otherwise, the system issues error message IXC745I.

INITSIZE is an optional parameter. If not specified, the system uses the SIZE parameter.

MINSIZE(minsize)

Specifies the smallest size in units of 1K (1024 bytes) to which the structure can ever be altered, either as a result of automatic alter processing or any alter process. MINSIZE also serves as a minimum bound for the structure size on all structure allocation requests that occur on systems at OS/390 Release 10 and above (including connect and rebuild connect) except that new structure allocation during system-managed rebuild may cause the structure to be allocated with a size smaller than the value specified or defaulted to for MINSIZE. This can occur when there is not enough space in any of the coupling facilities in the preference list to satisfy the MINSIZE requirement for the rebuild new structure, but there is enough space in at least one of the coupling facilities in the preference list to allocate the rebuild new structure and copy all the inuse objects from the rebuild old

Administrative Data Utility

structure. In this case, instead of stopping the system-managed rebuild because the MINSIZE requirement could not be satisfied, XCF will allow the rebuild to continue.

The number can be 1 - 8 decimal digits long. The MINSIZE value must be less than or equal to the INITSIZE value, or less than or equal to the SIZE value if INITSIZE is not specified.

MINSIZE is an optional parameter. If not specified, the system will use 50% of the structure INITSIZE value, or 50% of the SIZE value if INITSIZE is not specified.

ALLOWAUTOALT(NO)

ALLOWAUTOALT(YES)

Specifies the installation's request to allow system-initiated alter processing for this structure.

If a value other than zero is specified or defaulted to for FULLTHRESHOLD and ALLOWAUTOALT(NO) is specified or defaulted to, structure full monitoring will monitor the structure, but automatic alter processing will not perform any system-initiated alters against the structure.

ALLOWAUTOALT is an optional parameter. NO is the default, when ALLOWAUTOALT is not specified.

FULLTHRESHOLD(value)

Specifies a percentage value used by automatic alter processing and structure full monitoring. This value represents a percent full threshold for the structure.

- For a cache structure, the value represents a percent full threshold in terms of changed coupling facility structure objects.
- For a list or lock structure, the value represents a percent full threshold in terms of inuse coupling facility structure objects.

The value is specified as a percentage and can be 1 - 3 decimal digits long (0-100). Specifying 0 will result in no structure full monitoring or automatic alter processing for the structure. If a value other than 0 is specified or defaulted to for FULLTHRESHOLD, then the value will be used as a percent full threshold for structure full monitoring.

If the exploiter allows alter processing, and ALLOWAUTOALT(YES) is specified, the FULLTHRESHOLD value can be used by automatic alter processing. When the FULLTHRESHOLD value is reached, XCF will alert the installation through message IXC585E and automatically start an alter against the structure to relieve the storage shortage for the object whose storage is in short supply. Before starting the alter request, XCF will issue message IXC588I to the system log to externalize the alter request.

If a value other than 0 is specified or defaulted to for FULLTHRESHOLD, and alter processing is not allowed by the exploiter, then the value will be used as a percent full threshold for structure full monitoring.

PREFLIST(cfname1,cfname2,...,cfname8)

Specifies an ordered list of coupling facility names from which the system is to choose when allocating a structure in a coupling facility. If ENFORCEORDER(NO) is specified, the system attempts to allocate the

structure in the first coupling facility in the preference list that meets the following allocation criteria, listed in order of relative importance from most important to least important:

- Has connectivity to the system trying to allocate the structure
- Has a coupling facility operational level (CFLEVEL) equal to or greater than the requested CFLEVEL
- Is a failure-independent coupling facility in relation to the coupling facility containing the old structure in user-managed duplexing rebuild processing. The system will give preference to failure-independent coupling facilities when allocating the new structure during user-managed duplexing.
- Has space available that is greater than or equal to the requested structure size
- Meets the volatility requirement requested by the connector
- Meets the failure-independence requirement requested by the connector
- Does not contain a structure in this structure's exclusion list.

Note that the system assumes certain criteria when selecting a coupling facility for new structure allocation in user-managed structure duplexing. The system always assumes LOCATION=OTHER when selecting the coupling facility. As listed above, the coupling facility chosen by the system will, if possible, be failure-independent with respect to the coupling facility containing the old structure. Lastly, if the level of connectivity to the new structure is less than that to the old structure, the action taken by the system is LESSCONNECTION=TERMINATE.

If there is no coupling facility in the preference list that meets all these criteria, then the system determines the coupling facility that most closely meets the criteria. See "How MVS Uses the Lists" on page 46 for information about how the system chooses the coupling facility in the preference list that most closely meets the requirements of the connect request.

If the structure you are defining can only reside in a coupling facility with a certain attribute (such as CFLEVEL or nonvolatility), be sure to include only those coupling facilities in the preference list for the structure. Note that two or more coupling facilities are required when you specify DUPLEX(ALLOWED) or DUPLEX(ENABLED).

If ENFORCEORDER(YES) is specified, the system will not reorder the coupling facilities in the preference list, but will attempt to allocate the structure in the exact order specified in the preference list. Therefore, when constructing your preference list, be sure to place the coupling facilities with the preferred attributes before any coupling facilities without the desired attributes when ENFORCEORDER(YES) is specified.

ENFORCEORDER(YES) is mutually exclusive with EXCLLIST.

PREFLIST is a required parameter.

EXCLLIST(strname1,strname2,...,strname8)

Specifies the list of 1 to 8 coupling facility structure names with which this structure should not share the same coupling facility. The system attempts to honor the exclusion request, but will not fail a request to allocate a structure when all other requirements for structure allocation have been satisfied and the exclusion list cannot be honored.

Administrative Data Utility

EXCLLIST is mutually exclusive with ENFORCEORDER(YES).

EXCLLIST is an optional parameter.

REBUILDPERCENT(value)

Specifies, as a percent of lost connectivity to a structure, when MVS is to initiate a structure rebuild. Use of this parameter requires that all active connections to the structure support structure rebuild. The specified percent value is compared with a value calculated by MVS when a loss of coupling facility connectivity occurs. MVS uses the system weights assigned in the active SFM policy to calculate:

- (a) the total weight of all systems with active connections to the structure
- (b) the total weight of all systems with active connections which have lost connectivity to the coupling facility containing the structure.

If the value of $(b*100) / a$ is greater than or equal to the percent value specified, MVS initiates a rebuild of the structure.

MVS only initiates rebuild if all of the following are true:

1. All active connections to the structure support rebuild.
2. The structure is not being used for XCF signaling.
3. Rebuild is not already in progress.
4. SFM policy is active.

Note: CONNFAIL(YES) is no longer required by SFM to use the system weights specified in the SFM active policy.

5. CFRM policy specified REBUILDPERCENT > 0.

If REBUILDPERCENT is not specified, MVS will use 1% as the default value if SFM policy is active for systems at OS/390 Release 10 and higher and for those systems at Release 3 and above with OW41959 installed.

If there is not an active SFM policy in the sysplex and all systems in the sysplex are at OS/390 Release 3 or higher with OW30814 installed, MVS will initiate a rebuild for structures that have experienced any loss of connectivity to systems in the sysplex, regardless of the REBUILDPERCENT specification, if structure rebuild is supported.

REBUILDPERCENT is an optional parameter. If not specified, when there is an active SFM policy in the sysplex, the system uses the value 1. In this case, if any system with active connections to the structure loses connectivity to the coupling facility in which the structure resides, MVS initiates a rebuild (if structure rebuild is supported).

DUPLEX(DISABLED)

DUPLEX(ALLOWED)

DUPLEX(ENABLED)

Specifies the installation's request for duplexing of the structure.

DISABLED

User-managed duplexing cannot be started for the specified structure.

ALLOWED

The application can initiate its own duplexing or an operator can start duplexing with the SETXCF START,DUPLEX command if the

Administrative Data Utility

application supports user-managed duplexing. However, MVS will not make any attempts to maintain the duplexed status of the structure. The application or operator must initiate another duplexing operation if the current duplexing is stopped.

ENABLED

MVS will initiate and attempt to maintain a user-managed duplexing operation for the structure. When the duplexing is stopped, MVS will attempt to initiate duplexing again. If the operator stops the duplexing operation, MVS will ensure that the coupling facility most recently used to contain the previous instance of the duplexed structure is not used again to contain the structure. This provides a way to easily move an instance of a duplexed structure so that maintenance can be performed on the vacated coupling facility.

DUPLEX is an optional parameter. DUPLEX is ignored if the structure is not a cache structure, or the users of the structure do not support user-managed duplexing.

DUPLEX(DISABLED) is the default when DUPLEX is not specified. DUPLEX(ALLOWED) and DUPLEX(ENABLED) require at least two coupling facilities to be specified in the preference list.

ENFORCEORDER(NO)

ENFORCEORDER(YES)

Specifies whether or not the ordering of coupling facilities in the preference list will be strictly enforced for structure allocation purposes. Using this option can help ensure failure-independence between a coupling facility and the systems that use it when these resources reside on the same CEC.

If you specify ENFORCEORDER(NO), which is the default, the system will use its allocation algorithms to reorder the list of coupling facilities in the preference list in order to choose the one that most closely fits the application's requirements.

If you specify ENFORCEORDER(YES), the system will not reorder the list of coupling facilities in the preference list. A coupling facility may be chosen based on its order in the list rather than on how well it satisfies the structure allocation requirements.

ENFORCEORDER(YES) is mutually exclusive with EXCLLIST.

LOGR Parameters for Administrative Data Utility

The LOGR policy information describes the characteristics of a log stream and a coupling facility structure that will be used when there are active connections to the log stream. You can specify the REPORT keyword for any of the requests.

The LOGR couple data set must be activated and available before you add log stream and structure definitions to it. See "Format the LOGR Couple Data Set and Make it Available to the Sysplex" on page 216.

For guidance on planning your LOGR couple data set parameters, see "Chapter 9. Planning for System Logger Applications" on page 173.

Administrative Data Utility

The return and reason codes for LOGR parameter errors are displayed in system message IXG002E. They are documented in the information for the IXGINVNT service in *OS/390 MVS Programming: Assembler Services Reference*.

The requests you can specify for LOGR include:

- DEFINE LOGSTREAM
- DEFINE STRUCTURE
- UPDATE LOGSTREAM
- DELETE LOGSTREAM
- DELETE STRUCTURE
- LIST LOGSTREAM
- LIST STRUCTURE

The following restrictions apply when defining, updating, deleting, or listing a log stream or structure for system logger:

- If the Security Authorization Facility (SAF) is available, the system performs SAF authorization checks on all IXGINVNT requests and IXCMIAPU TYPE(LOGR) requests.

To define or delete a structure entry in the LOGR policy, the caller must have alter access to RESOURCE(MVSADMIN.LOGR) in SAF class CLASS(FACILITY).

To define or update a log stream entry in the LOGR policy, the caller must have update access to RESOURCE(log_stream_name) in SAF class CLASS(LOGSTRM).

If the log stream entry specifies the STRUCTNAME parameter, then the caller must also have update access authority to the coupling facility structure through RESOURCE(IXLSTR.structure_name) in SAF class CLASS(FACILITY).

To define a log stream and model it after another log stream using the LIKE parameter, the caller must also have update access to the model log stream, RESOURCE(model_log_stream_name) in class CLASS(LOGSTRM).

- If SAF is not available or if there is no CLASS(LOGSTRM) or CLASS(FACILITY) class defined for the log stream or structure, no security checking is performed.
- To list a log stream or structure entry from the LOGR policy, the caller must have read access to RESOURCE(MVSADMIN.LOGR) in SAF class CLASS(FACILITY).

REPORT(YES|NO)

Indicates whether the utility should list out the LOGR policy. The default is REPORT(YES). IXCMIAPU generates the report after all control statements have been processed, regardless of whether errors were encountered during the processing of the control statements.

Note: To run a report, the caller must have read access to RESOURCE(MVSADMIN.LOGR) in SAF class CLASS(FACILITY).

DEFINE LOGSTREAM Keywords

The DEFINE LOGSTREAM specification requests that an entry for a log stream is to be defined in the LOGR policy. The keywords are explained as follows:

NAME(streamname)

Specifies the name of the log stream that you want to define in the LOGR policy. The name can be made up of one or more segments separated by periods, up to the maximum length of 26 characters. The following rules apply:

- Each segment can contain up to eight numeric, alphabetic, or national (\$, #, or @) characters.

- The first character of each segment must be an alphabetic or national character.
- Each segment must be separated by periods, which you must count as characters.

NAME is required when defining a log stream.

RMNAME(rmname)

Specifies the name of the resource manager program associated with the log stream. RNAME must be 8 alphanumeric or national (\$,#,or @) characters, padded on the right with blanks if necessary.

You must define RMNAME in the LOGR policy before the resource manager can connect to the log stream.

See the system logger chapter in *OS/390 MVS Programming: Assembler Services Guide* for information on writing a resource manager program to back up a log stream.

DESCRIPTION(description)

Specifies the user defined data describing the log stream. DESCRIPTION must be 16 alphanumeric or national (\$,#,@) characters, underscore (_) or period (.), padded on the right with blanks if necessary.

DASDONLY(NO)

DASDONLY(YES)

Specifies whether the log stream being defined is a coupling facility or a DASD-only log stream.

If you specify DASDONLY(NO), which is the default, the log stream is defined as a coupling facility log stream.

If you specify DASDONLY(YES) the log stream is defined as a DASD-only log stream and does not use the coupling facility for log data.

With DASDONLY(NO), you can also specify STG_DUPLEX and DUPLEXMODE parameters to select a method of duplexing for a coupling facility log stream.

STRUCTNAME(structname)

Specifies the up to 16 character name of the coupling facility structure associated with the coupling facility log stream being defined. The structure specified is a list structure defined in the CFRM policy where all of this log stream's log blocks will be written before being written to DASD.

For a coupling facility log stream, you must define STRUCTNAME in the log stream definition in the LOGR policy via this parameter or the STRUCTNAME defined for the log stream referenced by the LIKE parameter before you can connect to the log stream.

For a DASD-only log stream, omit the STRUCTNAME parameter, since there is no coupling facility associated with the log stream.

STRUCTNAME must be 16 alphanumeric or national (\$,#,@) characters, or underscore (_), padded on the right with blanks if necessary. The first character must be alphabetic.

MAXBUFSIZE(maxbufsize)

Administrative Data Utility

MAXBUFSIZE(65532)

Specifies the size, in bytes, of the largest log block that can be written to the DASD-only log stream being defined in this request.

The value for MAXBUFSIZE must be between 1 and 65,532 bytes. The default is 65,532 bytes.

This parameter is only valid with DASDONLY(YES).

STG_DUPLEX(NO)

STG_DUPLEX(YES)

Specifies whether the log stream data for a coupling facility log stream should be duplexed in DASD staging data sets.

If you specify STG_DUPLEX(NO), which is the default, log data for a coupling facility log stream will be duplexed in local buffers, which might be vulnerable to system failure if your configuration contains a single point of failure.

If you specify STG_DUPLEX(YES), log data for a coupling facility log stream will be duplexed in staging data sets when the conditions defined by the DUPLEXMODE parameter are met. This method safeguards data on DASD staging data sets.

You can use the DUPLEXMODE parameter with STG_DUPLEX to specify whether you want conditional or unconditional duplexing. STG_DUPLEX is only valid when you specify or default to DASDONLY(NO) to define a coupling facility log stream.

DUPLEXMODE(COND)

DUPLEXMODE(UNCOND)

Specifies the conditions under which the coupling facility log data for a coupling facility log stream should be duplexed in DASD staging data sets.

If you specify DUPLEXMODE(COND), which is the default, the coupling facility log data will be duplexed in staging data sets only if a system's connection to the coupling facility log stream contains a single point of failure and is therefore vulnerable to permanent log data loss:

- A connection to a log stream contains a single point of failure if the coupling facility is volatile and/or resides on the same CPC as the MVS system connecting to it. The coupling facility log data for the system connection containing the single point of failure will be duplexed.
- A connection to a log stream is failure-independent when the coupling facility for the log stream is non-volatile and resides on a different central processor complex (CPC) than the MVS system connecting to it. The coupling facility log data for that system connection will not be duplexed.

If you specify DUPLEXMODE(UNCOND), the log data for the coupling facility log stream will be duplexed in staging data sets, unconditionally, even if the connection is failure independent.

DUPLEXMODE is valid only when STG_DUPLEX(YES) is specified.

DUPLEXMODE and STG_DUPLEX are only valid when you specify or default to DASDONLY(NO) to define a coupling facility log stream.

STG_DATACLAS(NO_STG_DATACLAS)

STG_DATACLAS(stg_dataclas)

Specifies the up to 8-byte name of the SMS data class that will be used for allocation of the DASD staging data set for this log stream. The first character must be an alphabetic or national character.

If you specify STG_DATACLAS(NO_STG_DATACLAS), which is the default, the data class is defined by standard SMS processing. See *OS/390 DFSMS: Using Data Sets* for more information about SMS. An SMS value specified on the STG_DATACLAS parameter, including NO_STG_DATACLAS, **always** overrides one specified on a model log stream used on the LIKE parameter.

STG_DATACLAS is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified.

STG_MGMTCLAS(NO_STG_MGMTCLAS)**STG_MGMTCLAS(stg_mgmtclas)**

Specifies the up to 8-byte name of the SMS management class for the allocation of the DASD staging data set for this log stream. The first character must be an alphabetic or national character.

If you specify STG_MGMTCLAS(NO_STG_MGMTCLAS), which is the default, the management class is defined by standard SMS processing. See *OS/390 DFSMS: Using Data Sets* for more information about SMS. An SMS value specified on the STG_MGMTCLAS parameter, including NO_STG_MGMTCLAS, **always** overrides one specified on a model log stream used on the LIKE parameter.

STG_MGMTCLAS is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified.

STG_STORCLAS(NO_STG_STORCLAS)**STG_STORCLAS(stg_storclas)**

Specifies the up to 8-byte name of the SMS storage class for allocation of the DASD staging data set for this log stream. The first character must be an alphabetic or national character.

If you specify STG_STORCLAS(NO_STG_STORCLAS), which is the default, the storage class is defined by standard SMS processing. See *OS/390 DFSMS: Using Data Sets* for more information about SMS. An SMS value specified on the STG_STORCLAS parameter, including NO_STG_STORCLAS, **always** overrides one specified on a model log stream used on the LIKE parameter.

STG_STORCLAS is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified.

STG_SIZE(stg_size)

Specifies the size, in 4K blocks, of the DASD staging data set for the log stream being defined.

If you omit STG_SIZE for a **coupling facility log stream**, system logger does one of the following, in the order listed, to allocate space for staging data sets:

- Uses the STG_SIZE of the log stream specified on the LIKE parameter, if specified.
- Uses the maximum coupling facility structure size for the structure to which the log stream is defined. This value is obtained from the value defined on the SIZE parameter for the structure in the CFRM policy.

Administrative Data Utility

If you omit STG_SIZE for a **DASD-only log stream**, system logger does one of the following, in the order listed, to allocate space for staging data sets:

- Uses the STG_SIZE of the log stream specified on the LIKE parameter, if specified.
- Uses the size defined in the SMS data class for the staging data sets.
- Uses dynamic allocation rules for allocating data sets, if SMS is not available.

Note that if both the STG_DATACLAS and STG_SIZE are specified, the value for STG_SIZE overrides the space allocation attributes for the data class specified on the STG_DATACLAS value.

STG_SIZE is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified.

LS_DATACLAS(NO_LS_DATACLAS)

LS_DATACLAS(ls_dataclas)

Specifies the up to 8-byte name of the SMS data class that will be used for log stream data set allocation. The first character must be an alphabetic or national character.

If you specify LS_DATACLAS(NO_LS_DATACLAS), which is the default, the data class is defined by standard SMS processing. See *OS/390 DFSMS: Using Data Sets* for more information about SMS. An SMS value specified on the LS_DATACLAS parameter, including NO_LS_DATACLAS, **always** overrides one specified on a model log stream used on the LIKE parameter.

LS_MGMTCLAS(NO_LS_MGMTCLAS)

LS_MGMTCLAS(ls_mgmtclas)

Specifies the up to 8-byte name of the SMS management class for the log stream data set allocation. The first character must be an alphabetic or national character.

If you specify LS_MGMTCLAS(NO_LS_MGMTCLAS), which is the default, the management class is defined by standard SMS processing. See *OS/390 DFSMS: Using Data Sets* for more information about SMS. An SMS value specified on the LS_MGMTCLAS parameter, including NO_LS_MGMTCLAS, **always** overrides one specified on a model log stream used on the LIKE parameter.

LS_STORCLAS(NO_LS_STORCLAS)

LS_STORCLAS(ls_storclas)

Specifies the up to 8-byte name of the SMS storage class for the log stream data set allocation. The first character must be an alphabetic or national character.

If you specify LS_STORCLAS(NO_LS_STORCLAS), which is the default, the storage class is defined by standard SMS processing. See *OS/390 DFSMS: Using Data Sets* for more information about SMS. An SMS value specified on the LS_STORCLAS parameter, including NO_LS_STORCLAS, **always** overrides one specified on a model log stream used on the LIKE parameter.

LS_SIZE(ls_size)

Specifies the size, in 4K blocks, of the log stream DASD data sets for the log stream being defined.

If you omit LS_SIZE, system logger does one of the following, in the order listed, to allocate space for staging data sets:

- Uses the LS_SIZE of the log stream specified on the LIKE parameter, if specified.
- Uses the size defined in the SMS data class for the log stream data sets.
- Uses dynamic allocation rules for allocating data sets, if SMS is not available.

Note that if both the LS_DATACLAS and LS_SIZE are specified, the value for LS_SIZE overrides the space allocation attributes for the dataclass specified on the LS_DATACLAS value.

AUTODELETE(NO)

AUTODELETE(YES)

Specifies when system logger can physically delete log data.

If you specify AUTODELETE(NO), which is the default, system logger physically deletes an entire log data set only when **both** of the following are true:

- Data is marked for deletion by a system logger application using the IXGDELET service or an archiving procedure (like CICS/VR for CICS log streams or IEFEREPS for logrec log stream).
- The retention period for all the data in the log data set has expired.

If you specify AUTODELETE(YES), system logger automatically physically deletes log data whenever data is either marked for deletion (using the IXGDELET service or an archiving procedure) or the retention period for all the log data in a data set has expired.

Be careful when using AUTODELETE(YES) if the system logger application manages log data deletion using the IXGDELET service. With AUTODELETE(YES), system logger may delete data that the application expects to be accessible. If you specify AUTODELETE=YES with RETPD=0, data is eligible for deletion as soon as it is written to the log stream.

The LOGR couple data set must be formatted at the OS/390 Release 3 level or above to use this keyword.

RETPD(0)

RETPD(retpd)

Specifies the retention period, in days, for log data in the log stream. The retention period begins when data is written to the log stream. Once the retention period for an entire log data set has expired, the data set is eligible for physical deletion. The point at which system logger physically deletes the data depends on what you have specified on the AUTODELETE parameter.

System logger processes RETPD when a log data set fills and system logger switches to a new one for a log stream. System logger will not process a retention period or delete data on behalf of log streams that are not connected and being written to by an application.

The value specified for RETPD must be between 0 and 65,536.

The active primary LOGR couple data set must be formatted at the OS/390 Release 3 level or above to use this keyword.

HLQ(NO_HLQ)

Administrative Data Utility

HLQ(hlq)

Specifies the up to 8-byte high level qualifier for both the log stream data set name and the staging data set name. HLQ must be 8 alphanumeric or national (\$,#,or @) characters, padded on the right with blanks if necessary. The first character must be an alphabetic or national character.

If you do not specify a high level qualifier, or if you specify HLQ(NO_HLQ), the log stream will have a high level qualifier of IXGLOGR. If you specified the LIKE parameter, the log stream will have the high level qualifier of the log stream specified on the LIKE parameter. The value specified for HLQ overrides the high level qualifier for the log stream specified on the LIKE parameter.

HIGHOFFLOAD(80)

HIGHOFFLOAD(highoffload)

Specifies the percent value you want to use as the high offload threshold for the coupling facility space allocated for this log stream. When the coupling facility is filled to the high offload threshold point or beyond, system logger begins offloading data from the coupling facility to the DASD log stream data sets.

The default HIGHOFFLOAD value is 80%. You can specify the default in the following ways:

- HIGHOFFLOAD(80)
- HIGHOFFLOAD(0)
- Omit the HIGHOFFLOAD parameter.

The value specified for HIGHOFFLOAD must be greater than the LOWOFFLOAD value.

LOWOFFLOAD(0)

LOWOFFLOAD(lowoffload)

Specifies the percent value you want to use as the low offload threshold for the coupling facility for this log stream. The low offload threshold is the point in the coupling facility, in percent value of space consumed, where system logger stops offloading coupling facility log data to log stream DASD data sets. The value of LOWOFFLOAD is the percent of log data system logger leaves in the coupling facility.

If you specify LOWOFFLOAD(0), which is the default, or omit the LOWOFFLOAD parameter, system logger uses the 0% usage mark as the low offload threshold.

The value specified for LOWOFFLOAD must be less than the HIGHOFFLOAD value.

LIKE(NO_LIKE)

LIKE(like_streamname)

Specifies the name of a log stream defined in the LOGR policy. The characteristics of this log stream (such as storage class, management class, high level qualifier and data class) will be copied for the log stream you are defining only if those characteristics are not explicitly coded on the referencing log stream. The parameters explicitly coded on this request, however, override the characteristics of the log stream specified on the LIKE parameter.

MODEL(NO)

MODEL(YES)

Specifies whether the log stream being defined is a model, exclusively for use with the LIKE parameter to set up general characteristics for other log stream definitions.

If you specify MODEL(NO), which is the default, the log stream being defined is not a model log stream. Systems can connect to and use this log stream. The log stream can also be specified on the LIKE parameter, but is not exclusively for use as a model.

If you specify MODEL(YES), the log stream being defined is only a model log stream. It can be specified only as a model for other log stream definitions on the LIKE parameter.

Programs **cannot** connect to a log stream name that is defined as a model (MODEL(YES)) using an IXGCONN request.

No log stream data sets are allocated on behalf of a model log stream.

The attributes of a model log stream are syntax checked at the time of the request, but not verified until another log stream references the model log stream on the LIKE parameter.

DIAG(NO)

DIAG(YES)

Specifies whether or not dumping or additional diagnostics should be provided by Logger for certain conditions.

If you specify DIAG(NO), which is the default, this indicates no special Logger diagnostic activity is requested for this logstream regardless of the DIAG specifications on the IXGCONN, IXGDELET and IXGBRWSE requests.

If you specify DIAG(YES), this indicates special Logger diagnostic activity is allowed for this logstream and can be obtained when the appropriate specifications are provided on the IXGCONN, IXGDELET or IXGBRWSE requests.

DEFINE STRUCTURE Keywords

The DEFINE STRUCTURE specification requests that an entry for a coupling facility structure be defined in the LOGR policy for a coupling facility log stream. The keywords are explained as follows:

STRUCTNAME(structname)

Specifies the up to 16-byte name of the coupling facility structure you are defining. STRUCTNAME must be 16 alphanumeric or national (\$, #, @) characters, or underscore (_), padded on the right with blanks if necessary. The first character must be alphabetic.

LOGSNUM(logsnum)

Specifies the number of log streams that can be allocated to the coupling facility list structure being defined. *logsnum* must be a value between 0 and 512.

LOGSNUM is required when defining a structure.

MAXBUFSIZE(maxbufsize)

Specifies the size, in bytes, of the largest log block that can be written to log streams allocated to the coupling facility specified in this request.

Administrative Data Utility

The value for MAXBUFSIZE must be between 1 and 65532 bytes. The default is 65532 bytes.

AVGBUFSIZE(avgbufsize)

Specifies the average size in bytes, of log blocks written to all the log streams using this coupling facility structure.

System logger uses the average buffer size to control the entry-to-element ratio for this coupling facility structure.

When the active primary LOGR couple data set is at an OS/390 Release 3 level or higher, system logger uses the AVGBUFSIZE specified simply to make an initial determination of the entry-to-element ratio for the structure. After that, system logger monitors structure usage and dynamically manages the entry-to-element ratio accordingly. System logger uses the last entry-to-element ratio in effect for a structure for subsequent structure reallocation requests.

When the active primary LOGR couple data set is at a pre-OS/390 Release 3 level, system logger uses the AVGBUFSIZE specified to calculate an entry-to-element ratio that lasts for the life of this coupling facility structure. You cannot update the average buffer size for a structure without first deleting the structure definition (and all the log stream definitions associated with the structure) and then redefining the structure with a new average buffer size.

avgbufsize must be between 1 and the value for MAXBUFSIZE. The default value is 1/2 of the MAXBUFSIZE value.

UPDATE LOGSTREAM Keywords

The UPDATE LOGSTREAM specification requests that an entry for a coupling facility or DASD-only log stream be updated in the LOGR policy. Except for the RETPD, AUTODELETE and DIAG parameters, you cannot update a log stream while there are active connections (active or failed) to it. The keywords are explained as follows:

NAME(streamname)

Specifies the name of the log stream that you want to update in the LOGR policy.

NAME is required when updating a log stream.

STRUCTNAME(structname)

With UPDATE LOGSTREAM specifies the name of the coupling facility structure where all of this log stream's log blocks will be written before being offloaded to DASD. This keyword can only be specified to upgrade a DASD-only log stream to a coupling facility log stream. The structure specified is a list structure defined in the CFRM policy.

STRUCTNAME must be 16 alphanumeric or national(\$,#,or @) characters, or underscore (_), padded on the right with blanks if necessary. The first character must be alphabetic.

Note that the MAXBUFSIZE value in the structure definition for this structure must be equal to or greater than the MAXBUFSIZE specified in the DASD-only log stream definition you are upgrading to a coupling facility log stream. For example, let's say the MAXBUFSIZE value in the DASD-only log stream definition you are upgrading is 50,000 bytes. When you specify a structure name to upgrade the DASD-only log stream to a coupling facility log stream, make sure you specify one with a MAXBUFSIZE value of 50,000 bytes or more.

RMNAME(rmname)

Specifies the name of the resource manager program associated with the log stream. RNAME must be 8 alphanumeric or national (\$,#, or @) characters, padded on the right with blanks if necessary.

You must define RMNAME in the LOGR policy before the resource manager can connect to the log stream.

See the system logger chapter in *OS/390 MVS Programming: Assembler Services Guide* for information on writing a resource manager program to back up a log stream.

DESCRIPTION(description)

Specifies the user defined data describing the log stream. DESCRIPTION must be 16 alphanumeric or national (\$,#,@) characters, underscore (_) or period (.), padded on the right with blanks if necessary.

MAXBUFSIZE(maxbufsize)

MAXBUFSIZE(65532)

Specifies the size, in bytes, of the largest log block that can be written to the DASD-only log stream being updated in this request. Use this parameter to update the MAXBUFSIZE for a DASD-only log stream.

The value for MAXBUFSIZE must be between 1 and 65,532 bytes and cannot be less than the current MAXBUFSIZE specified for the DASD-only log stream. (You can increase the MAXBUFSIZE, but you cannot decrease it.)

There is no default for the MAXBUFSIZE parameter on an UPDATE request. If you omit this parameter, there will be no change to the MAXBUFSIZE for this log stream definition.

STG_DUPLEX(NO)

STG_DUPLEX(YES)

Specifies whether the log stream data for a coupling facility log stream should be duplexed in DASD staging data sets.

If you specify STG_DUPLEX(NO), log data for a coupling facility log stream will not be duplexed.

If you specify STG_DUPLEX(YES), the log data for a coupling facility log stream will be duplexed when the conditions defined by the DUPLEXMODE parameter are fulfilled. STG_DUPLEX is only valid for a coupling facility log stream.

There is no default for the STG_DUPLEX parameter on an UPDATE request. If you omit this parameter, there will be no change to the duplexing status for this log stream definition.

You can use the DUPLEXMODE parameter with STG_DUPLEX to specify whether you want conditional or unconditional duplexing.

DUPLEXMODE(COND)

DUPLEXMODE(UNCOND)

Specifies the conditions under which log data for a coupling facility log stream should be duplexed in DASD staging data sets.

Administrative Data Utility

If you specify DUPLEXMODE(COND), the coupling facility log data will be duplexed only if an application's connection to the log stream contains a single point of failure and is therefore vulnerable to permanent log data loss:

- A connection to a log stream contains a single point of failure if the coupling facility is volatile and/or resides on the same central processing complex as the MVS system connection to it. The coupling facility log data will be duplexed.
- A connection to a log stream is failure-independent when the coupling facility for the log stream is non-volatile and resides on a different central processing complex than the MVS system connecting to it. The coupling facility log data will not be duplexed.

If you specify DUPLEXMODE(UNCOND), the log data for the coupling facility log stream will be duplexed in staging data sets, unconditionally, even if the coupling facility is failure-independent.

There is no default for the DUPLEXMODE parameter on an UPDATE request. If you omit this parameter, there will be no change to the duplexing mode for this log stream definition.

See *OS/390 MVS Programming: Assembler Services Guide* for complete information on using staging data sets to duplex coupling facility log data.

DUPLEXMODE is valid only when STG_DUPLEX(YES) is specified. DUPLEXMODE and STG_DUPLEX are only valid for a coupling facility log stream.

HIGHOFFLOAD(highoffload)

Specifies the percent value you want to use as the high offload threshold for the coupling facility space allocated for this log stream. When the coupling facility is filled to the high offload threshold point or beyond, system logger begins offloading data from the coupling facility to the DASD log stream data sets.

IBM recommends that you are careful in considering to define your HIGHOFFLOAD value to greater than 80%. Defining a higher high offload threshold can leave you vulnerable to filling your coupling facility space for the log stream, which means that system logger will reject all write requests until the coupling facility log data can be offloaded to DASD log data sets.

The value specified for HIGHOFFLOAD must be higher than the LOWOFFLOAD value.

LOWOFFLOAD(lowoffload)

Specifies the percent value you want to use as the low offload threshold for the coupling facility for this log stream. The low offload threshold is the point in the coupling facility, in percent value of space consumed, where system logger stops offloading coupling facility log data to log stream DASD data sets. The value of LOWOFFLOAD is the percent of log data system logger leaves in the coupling facility.

There is no default for the LOWOFFLOAD parameter on an UPDATE request. If you omit this parameter, there will be no change to the low offload value for this log stream definition.

The value specified for LOWOFFLOAD must be less than the HIGHOFFLOAD value.

STG_DATACLAS(stg_dataclas)

Specifies the name of the SMS data class that will be used for allocation of the DASD staging data set for this log stream. The first character must be an alphabetic or national character.

If you omit the STG_DATACLAS parameter, system logger makes no changes to the data class defined for this log stream.

If you specify STG_DATACLAS(NO_STG_DATACLAS) system logger deletes the STG_DATACLAS defined for this log stream.

STG_DATACLAS is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified for the log stream.

STG_MGMTCLAS(stg_mgmtclas)

Specifies the name of the SMS management class for the allocation of the DASD staging data set for this log stream. The first character must be an alphabetic or national character.

If you omit the STG_MGMTCLAS parameter, system logger makes no changes to the management class defined for this log stream.

If you specify STG_MGMTCLAS(NO_STG_MGMTCLAS) system logger deletes the STG_MGMTCLAS defined for this log stream.

STG_MGMTCLAS is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified for the log stream.

STG_STORCLAS(stg_storclas)

Specifies the name of the SMS storage class for allocation of the DASD staging data set for this log stream. The first character must be an alphabetic or national character.

If you omit the STG_STORCLAS parameter, system logger makes no changes to the storage class defined for this log stream.

If you specify STG_STORCLAS(NO_STG_STORCLAS) system logger deletes the STG_STORCLAS defined for this log stream.

STG_STORCLAS is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified for the log stream.

STG_SIZE(stg_size)

Specifies the size, in 4K blocks, of the DASD staging data set for the log stream being defined.

If you omit this parameter, there will be no change to the DASD staging data size in this log stream definition.

Note that if both the STG_DATACLAS and STG_SIZE are specified, the value for STG_SIZE overrides the space allocation attributes for the data class specified on the STG_DATACLAS value.

STG_SIZE is valid only when either STG_DUPLEX(YES) or DASDONLY(YES) is specified for the log stream.

Administrative Data Utility

LS_DATACLAS(Is_dataclas)

Specifies the name of the SMS data class that will be used for log stream data set allocation. The first character must be an alphabetic or national character.

If you omit the LS_DATACLAS parameter, system logger makes no changes to the data class defined for this log stream.

If you specify LS_DATACLAS(NO_LS_DATACLAS) system logger deletes the LS_DATACLAS defined for this log stream.

LS_MGMTCLAS(Is_mgmtclas)

Specifies the name of the SMS management class for the log stream data set allocation. The first character must be an alphabetic or national character.

If you omit the LS_MGMTCLAS parameter, system logger makes no changes to the management class defined for this log stream.

If you specify LS_MGMTCLAS(NO_LS_MGMTCLAS) system logger deletes the LS_MGMTCLAS defined for this log stream.

LS_STORCLAS(Is_storclas)

Specifies the name of the SMS storage class for the log stream data set allocation. The first character must be an alphabetic or national character.

If you omit the LS_STORCLAS parameter, system logger makes no changes to the storage class defined for this log stream.

If you specify LS_STORCLAS(NO_LS_STORCLAS) system logger deletes the LS_STORCLAS defined for this log stream.

LS_SIZE(Is_size)

Specifies the size, in 4K blocks, of the log stream DASD data sets for the log stream being defined.

If you omit this parameter, there will be no change to the DASD log data set size for the log stream being updated.

Note that if both the LS_DATACLAS and LS_SIZE are specified, the value for LS_SIZE overrides the space allocation attributes for the dataclass specified on the LS_DATACLAS value.

AUTODELETE(NO)

AUTODELETE(YES)

Specifies when system logger can physically delete log data. This parameter can be specified even when the log stream has connections. The log stream definition is updated next time there is a switch to a new DASD log data set for this log stream.

If you specify AUTODELETE(NO), which is the default, system logger physically deletes an entire log data set only when **both** of the following are true:

- Data is marked for deletion by a system logger application using the IXGDELET service or an archiving procedure (like CICS/VR for CICS log streams or IEFEREPS for logrec log stream).
- The retention period for all the data in the log data set expires.

If you specify AUTODELETE(YES), system logger automatically physically deletes log data whenever data is either marked for deletion (using the IXGDELET service or an archiving procedure) or the retention period for all the log data in a data set has expired.

Be careful when using AUTODELETE(YES) if the system logger application manages log data deletion using the IXGDELET service. With AUTODELETE(YES), system logger may delete data that the application expects to be accessible. If you specify AUTODELETE=YES with RETPD=0, data is eligible for deletion as soon as it is written to the log stream.

The LOGR couple data set must be formatted at the OS/390 Release 3 level or above to use this keyword.

RETPD(0)

RETPD(retpd)

Specifies the retention period, in days, for log data in the log stream. The retention period begins when data is written to the log stream. Once the retention period for an entire log data set has expired, the data set is eligible for physical deletion. The point at which system logger physically deletes the data depends on what you have specified on the AUTODELETE parameter. This parameter can be specified even when the log stream has connections. The log stream definition is updated next time there is a switch to a new DASD log data set for this log stream.

System logger processes RETPD when a log data set fills and system logger switches to a new one for a log stream. System logger will not process a retention period or delete data on behalf of log streams that are not connected and being written to by an application.

The value specified for RETPD must be between 0 and 65,536.

The active primary LOGR couple data set must be formatted at the OS/390 Release 3 level or above to use this keyword.

DIAG(NO_DIAG)

DIAG(NO)

DIAG(YES)

Specifies whether or not dumping or additional diagnostics should be provided by Logger for certain conditions..

If you specify DIAG(NO_DIAG), which is the default, this indicates that the DIAG attribute of the log stream should not be updated.

If you specify DIAG(NO), this indicates no special Logger diagnostic activity is requested for this logstream regardless of the DIAG specifications on the IXGCONN, IXGDELET and IXGBRWSE requests.

If you specify DIAG(YES), this indicates special Logger diagnostic activity is allowed for this logstream and can be obtained when the appropriate specifications are provided on the IXGCONN, IXGDELET or IXGBRWSE requests.

Administrative Data Utility

This keyword can be updated regardless of whether the log stream is actively connected or not. The change will immediately update the log stream definition but will not take effect on a particular system until the next first connection to the log stream from that system.

Omitting the DIAG parameter will not change the log stream's special diagnostics setting.

DELETE LOGSTREAM Keyword

The DELETE LOGSTREAM specification requests that an entry for a log stream be deleted from the LOGR policy. Note that you cannot delete a log stream while there are active connections to it (see "Deleting Log Streams From the LOGR Policy" on page 224). The keyword is explained as follows:

NAME(streamname)

Specifies the name of the log stream you want to delete from the LOGR policy.

NAME is required when deleting a log stream.

DELETE STRUCTURE Keyword

The DELETE STRUCTURE specification requests that an entry for a coupling facility structure be deleted from the LOGR policy. Note that you cannot delete a structure while there are active connections to any of the log streams mapped to that structure. The keyword is explained as follows:

NAME(structname)

Specifies the name of the coupling facility structure you are deleting from the LOGR policy.

NAME is required when deleting a structure.

LIST LOGSTREAM Keywords

The LIST LOGSTREAM specification requests that an entry for a log stream be listed in SYSPRINT output. The keywords are explained as follows:

NAME(streamname)

Specifies the name of the log stream you want to list. You will get the log stream definition from the LOGR policy in the output.

NAME is required.

A log stream name can be specified with wildcards. See "Using Wildcards With the LIST Request" on page 297.

DETAIL(NO)

DETAIL(YES)

Specifies whether you want detailed information about the log stream from the LOGR policy.

If you specify DETAIL(NO), which is the default, the output will only include the log stream definition.

If you specify DETAIL(YES) the output will include the log stream data as well as the definition.

LIST STRUCTURE Keywords

The LIST STRUCTURE specification requests that an entry for a structure be listed in the SYSPRINT output. The keywords are explained as follows:

NAME(structname)

Specifies the name of the structure you want to list. You will get the structure definition from the LOGR policy in the output.

NAME is required.

A structure name can be specified with wildcards. See “Using Wildcards With the LIST Request”.

DETAIL(NO)

DETAIL(YES)

Specifies whether you want detailed information about the log stream from the LOGR policy.

If you specify DETAIL(NO), which is the default, the output will only include the structure definition.

If you specify DETAIL(YES), the output will include the structure data as well as the definition.

Using Wildcards With the LIST Request

You can use wildcards on LIST requests to select multiple log streams or structures in your SYSPRINT output. The wildcards you can use include:

- * Substitutes for zero or more characters, up to the maximum length of the string. An * can start the string, end it, appear in the middle or in multiple places in the string. A single * for the name will select all the structure or log stream names.
- ? Substitutes for one character. One or more ? can start the string, end it, appear in the middle or in multiple places in the string.

Examples: Examples of using wildcards follow:

- *A* selects all names containing an A, including the name A.
- *A*B selects all names that contain an A followed by and ending with a B, with or without intervening characters. The names can have characters before the A.
- ?A? selects all 3-character names with an A as the second character.
- ?A?B selects all 4-character names with A as the second character and B as the fourth character.
- ?A* selects all names of 2 or more characters with A as the second character.

SFM Parameters for Administrative Data Utility

The SFM policy information refers to each system in the sysplex. The policy assigns each system a weight, to be used if signaling connectivity errors occur, and an action to be taken when a status update missing condition occurs. Only one action — DEACTTIME, RESETTIME, ISOLATETIME, or PROMPT can be specified for a given system. The system also uses the assigned weights in conjunction with the REBUILDPERCENT value specified in the CFRM policy when determining whether to initiate a rebuild for a structure in a coupling facility to which a system has lost connectivity.

The policy information also specifies whether SFM is to be used to automatically recover XCF signaling connectivity failures and what reconfiguration actions are to be taken when the PR/SM Automatic Reconfiguration Facility is being used.

CONNFAIL(YES)

Administrative Data Utility

CONNFAIL(NO)

Specifies whether or not the system is to perform recovery actions in the event of loss of signaling connectivity failures between one or more systems in the sysplex. Specifying YES indicates that the sysplex is to perform sysplex partitioning actions using the WEIGHT values specified for each system in the sysplex as a basis to determine the best set of systems to remain in the sysplex, given the connectivity failures that have occurred. Specifying NO indicates that the system is to prompt the operator to decide which system or systems to partition from the sysplex.

CONNFAIL is an optional parameter. The default is YES.

SYSTEM

Specifies the definition of a system within the scope of the named SFM policy.

NAME(sysname)

NAME(*)

Specifies the 1-8 character system name that is in accordance with MVS naming conventions. The limit for the number of systems that can be defined in a policy is established when the SFM couple data set is formatted. Use NAME(*) to specify policy default values for any system in the sysplex that doesn't have one or more specifically defined SYSTEM parameters. You can specify * only once per policy for the NAME parameter of the SYSTEM keyword. See 301 for an example of using the * to specify installation default values in a policy.

NAME is a required parameter.

WEIGHT(weight)

Specifies a value in the range from 1 to 9999 that represents the relative importance value of this particular system in the sysplex. This value is used in the determination of potential sysplex partitioning actions which would be required in the event of a signaling connectivity failure between one or more systems in the sysplex. This value is also used in conjunction with the REBUILDPERCENT value specified in the CFRM policy when determining whether MVS-initiated rebuild of coupling facility structures should take place when loss of connectivity to a coupling facility occurs.

WEIGHT is an optional parameter. The system default value is 1. You can change the default value for your policy by using the NAME(*) WEIGHT (your policy default value) option.

DEACTTIME(nostatus-interval)

Specifies the time interval in seconds after which the logical partition on which the failing system (identified by NAME) resides is to be deactivated. For example, specifying DEACTTIME(10) causes an LPAR DEACTIVATE of the failing LPAR 10 seconds after the system status update missing condition is detected. This action is performed only if there is an active system in the sysplex running in LPAR mode on the same PR/SM CPC as the named system.

You can specify a time interval for DEACTTIME from 0 to 86400 seconds. Specifying a value of 0 implies that you want the system to deactivate the logical partition as soon as the status update missing condition is detected. You can set a default value for your installation by using the NAME(*) DEACTTIME(your default value) option.

Notes:

1. If the failing system resumes its status update before the interval has expired, the deactivate function is not performed.
2. If the OPNOTIFY interval expires before the time specified for DEACTTIME, the IXC402D message will be issued before the deactivate is attempted.

RESETTIME(nostatus-interval)

Specifies the time interval in seconds after which the named system's PR/SM logical partition is to be system reset. For example, specifying RESETTIME(10) causes the failing system to be reset 10 seconds after a system status update missing condition is detected. This action is performed only if there is an active system in the sysplex running in LPAR mode on the same PR/SM CPC as the named system.

You can specify a time interval for RESETTIME from 0 to 86400 seconds. Specifying RESETTIME(0) causes the failing system to be reset as soon as XCF detects the system status update missing condition. You can set a default value for your installation by using the NAME(*) RESETTIME(your default value) option.

Notes:

1. If the failing system resumes its status update before the interval has expired, the system reset function is not performed.
2. If the OPNOTIFY interval expires before the time specified for RESETTIME, the IXC402D message is issued before the reset is attempted.

ISOLATETIME(nostatus-interval)

Specifies the time interval in seconds after which the named system is to be isolated using the fencing services through the coupling facility. For example, specifying ISOLATETIME(10) causes the failing system to be fenced 10 seconds after the System Status Update Missing condition is detected. This action is performed only if there is an active system in the sysplex that shares connectivity to one or more coupling facilities with the named system.

You can specify a time interval for ISOLATETIME of from 0 to 86400 seconds. Specifying a value of 0 implies that you want the system to isolate the named system as soon as the status update missing condition is detected. You can set a default value for your installation by using the NAME(*) ISOLATETIME (your default value) option.

Notes:

1. If the failing system resumes its status update before the interval has expired, the ISOLATE function is not performed.
2. If the isolation fails, message IXC102A prompts the operator to reset the system manually.

Coordinating Intervals

You should understand the relationship among several time intervals to which you assign a value.

- The INTERVAL specified in COUPLExx
- The spin loop timeout interval specified in EXSPATxx
- The RESETTIME, DEACTTIME, or ISOLATETIME interval described here

For information on the relationship between INTERVAL and SPINTIME values, see “Planning the Failure Detection Interval and Operator Notification Interval” on page 140.

PROMPT

Specifies that an operator prompt will be issued if the named system enters a status update missing condition. The time interval used is the OPNOTIFY interval specified in COUPLExx.

PROMPT is the default when any of the DEACTTIME/RESETTIME/ISOLATETIME parameters is not specified.

RECONFIG

Specifies the LPAR reconfiguration action to be taken after a system has been removed from the sysplex through sysplex partitioning. The limit for the number of reconfiguration actions that can be defined for a system is established when the SFM couple data set is formatted.

FAILSYS(sysname)

Specifies the name of the failing system that has been partitioned out of the sysplex. Use of this parameter is intended to name the primary system responsible for processing a workload in the sysplex. Should this system fail, a PR/SM reconfiguration would allow another system in the sysplex to acquire the necessary resources to take over the workload processing responsibilities.

FAILSYS is a required parameter.

ACTSYS(sysname)

Specifies the name of the “acting” system which is to perform the reconfiguration action in the sysplex. Use of this parameter is to name the system in the sysplex that would acquire the resources of the target system identified by TARGETSYS.

The logical partition identified by ACTSYS must be authorized to perform reconfiguration actions. See *PR/SM Planning Guide* for additional information about cross-partition control authority.

ACTSYS is a required parameter.

Note: To reactivate TARGETSYS, storage resources that have been taken over by ACTSYS must be configured offline to ACTSYS before the TARGETSYS logical partition is reactivated.

TARGETSYS(sysname|ALL)

Specifies whether the logical partition where the *sysname* resides is to be deactivated or that ALL other logical partitions in the same addressing range as the “acting” system are to be deactivated. Note that deactivation

of a specific system's logical partition can be performed only when both the target system and the acting system are running on the same PR/SM CPC and in the same sysplex.

Use of this parameter is intended to name the system (or all systems) that are to be deactivated so that the system identified by ACTSYS can take over its resources.

TARGETSYS is a required parameter.

STORE(NO)

STORE(YES)

Specifies whether or not the acting system is to acquire central storage from potentially deactivated logical partitions.

If YES is specified and the LPAR deactivation is successful, the system issues the command to reconfigure the central storage. If YES is specified and the LPAR deactivation is not successful because the target LPAR is not active, the system still attempts to issue the storage reconfiguration command.

STORE is an optional parameter. The default is NO.

ESTORE(NO)

ESTORE(YES)

Specifies whether or not the acting system is to acquire expanded storage from potentially deactivated logical partitions.

If YES is specified, the system issues the command to reconfigure the expanded storage.

ESTORE is an optional parameter. The default is NO.

Assigning Defaults Using an *

The following example shows the use of the * to assign installation default values in an SFM policy.

```
DEFINE POLICY NAME(POLICY1) ...
  SYSTEM NAME(SYS01)
  DEACTTIME(100)
  SYSTEM NAME(*)
  WEIGHT(10)
```

The example shows that for POLICY1, a system in the sysplex named SYS01 requires a DEACTTIME of 100 seconds and accepts the default values for all other parameters.

The second system defined, identified by *, sets the default value for the WEIGHT parameter to 10 (instead of the system default of 1) for systems in the sysplex that do not explicitly specify a WEIGHT value. In this example, SYS01 would use the WEIGHT value of 10.

Return Codes

The return codes in decimal format from the administrative data utility are:

Administrative Data Utility

Return Codes	Description
0	Successful completion. All requested processing completed successfully.
4	An I/O error occurred on the SYSPRINT or SYSIN file, or the SYSIN file was empty. Some processing might have completed.
8	The SYSPRINT DD or the SYSIN DD was not specified or could not be used.
12	A request was made to process the active data set and the system is not an SP 5.1.0 level or higher system. An error occurred parsing the input or the utility was unable to obtain the data space used for I/O processing.
16	The level of system is not at the correct level for running the utility. The system must be at SP 3.1.0 or higher.
20	An abend occurred during utility processing.
24	An environmental error occurred. The utility must not be linked to or called from another program.

Examples of Using the IXCMIAPU Utility

The examples shown here for OS/390 are shipped with the product in SYS1.SAMPLIB.

Administrative Data Utility for Automatic Restart Manager Policy Data

Sample JCL to run the utility to create or update the automatic restart management administrative data is shown in Figure 65 on page 303. The SYS1.SAMPLIB member name is IXCARMP0.

Administrative Data Utility

```
//IXCARMP0 JOB
/**
/** SAMPLE JCL TO UPDATE THE ADMINISTRATIVE POLICY DATA IN THE
/** COUPLE DATA SET FOR ARM (AUTOMATIC RESTART MANAGER)
/**
/** NOTES:
/**
/** 1. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR THE UTILITY
/** OUTPUT.
/** 2. SYSIN DD IS A REQUIRED DD STATEMENT FOR THE UTILITY
/** CONTROL STATEMENTS.
/** 3. DATA TYPE(ARM) STATEMENT IS REQUIRED TO SPECIFY WHAT TYPE
/** OF COUPLE DATA SET IS TO BE UPDATED.
/** 4. REPORT KEYWORD IS OPTIONAL. WHEN REPORT(YES) IS SPECIFIED,
/** AN ARM ADMINISTRATIVE POLICY REPORT WILL BE GENERATED IN
/** THE OUTPUT. THE DEFAULT VALUE FOR REPORT IS YES.
/** 5. REPLACE KEYWORD IS OPTIONAL. WHEN REPLACE(YES) IS SPECIFIED
/** FOR A POLICY, THE POLICY WILL BE REPLACED IF IT ALREADY
/** EXISTED IN THE COUPLE DATA SET.
/** IF REPLACE(NO) IS SPECIFIED FOR AN EXISTING POLICY, THE UPDATE
/** JOB WILL BE FAILED AND NO CHANGES WILL BE MADE TO THE COUPLE
/** DATA SET.
/** 6. TO DELETE AN EXISTING POLICY IN A COUPLE DATA SET,
/** INCLUDE THE FOLLOWING LINE IN THE SYSIN DD CARD:
/** DELETE POLICY NAME(POLNAME)
/** WHERE POLNAME IS THE NAME OF THE POLICY TO BE DELETED.
/**
/**
/*******
//STEP1 EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSIN DD *

DATA TYPE(ARM)
DSN(SYS1.MIGLIB.ARMCPLO1) VOLSER(3380X1) REPORT(YES)

DEFINE POLICY NAME(POLICY1) REPLACE(YES)

/* The following example shows the usage of wild card '?'
specification in the level assignments. Elements
CREDITDEPTx0 to CREDITDEPTx9, where x can be A-Z or 0-9,
will be associated with LEVEL(5). However, CREDITDEPTA1,
CREDITDEPTB9, and CREDITDEPT99 are assigned to LEVEL(8). */

RESTART_ORDER

*/
```

Figure 65. IXCARMP0 - Administrative Policy Data in ARM Couple Data Set (Part 1)

Administrative Data Utility

```
LEVEL(5)
  ELEMENT_NAME(CREDITDEPT?0,CREDITDEPT?1,CREDITDEPT?2,
               CREDITDEPT?3,CREDITDEPT?4,CREDITDEPT?5,
               CREDITDEPT?6,CREDITDEPT?7,CREDITDEPT?8,
               CREDITDEPT?9)
  ELEMENT_TYPE(AUTHREQ,
               CUSTFILE,
               SAVINGS)
LEVEL(6)
  ELEMENT_NAME(CORPNEWSSERVER)
LEVEL(7)
  ELEMENT_TYPE(DEPOSITA,DEPOSITB,
               DEPOSITC)
LEVEL(8)
  ELEMENT_NAME(CREDITDEPTA1,CREDITDEPTB9,CREDITDEPT99)

/* The following RESTART_GROUP(*) segment sets the values of
   RESTART_PACING and READY_TIMEOUT for any restart groups
   that do not explicitly specify these fields
```

Figure 66. IXCARMP0 - Administrative Policy Data in ARM Couple Data Set (Part 2)

```

RESTART_GROUP(*)
  RESTART_PACING(250)
  ELEMENT(*)
    READY_TIMEOUT(250)

/* Elements not defined in any restart groups in a policy are
   considered to be in the restart group named DEFAULT.
   The following RESTART_GROUP(DEFAULT) segment shows how
   to override some of the system assigned default values
*/

RESTART_GROUP(DEFAULT)
  FREE_CSA(250,700)
  RESTART_PACING(1000)
  ELEMENT(*)
    RESTART_ATTEMPTS(1,150)
    RESTART_TIMEOUT(150)
    READY_TIMEOUT(150)
RESTART_GROUP(CREDIT$AUTH)
  TARGET_SYSTEM(CONTROLA,CONTROLB,CONTROLC,CONTROLD,
                EASTOPS4,EASTOPS5,EASTOPS8,
                SAVEOPS1,SAVEOPS2)
  FREE_CSA(100,500)
  RESTART_PACING(50)
  ELEMENT(CREDITDEPT07)
    RESTART_ATTEMPTS(2,500)
    RESTART_TIMEOUT(30)
    READY_TIMEOUT(500)
    TERMTYPE(ALLTERM)
    RESTART_METHOD(ELEMTERM,STC,
                  'S CR07,PARAM=('DEPT','07','U3055'),'
                  'OPT=NO')
    RESTART_METHOD(SYSTEM,JOB,
                  'CREDIT.DEPT.PROGRAM.RESTART.BY.SYSTEM'
                  '(&SYSNAME.)') /* Symbolic-substitution parameters
                                   can be specified in the
                                   RESTART_METHOD. In this case,
                                   &SYSNAME will be substituted
                                   with the system name on which
                                   the element initially registered
                                   */
  ELEMENT(CREDITDEPTMSTF11)
    RESTART_ATTEMPTS(3,100)
    RESTART_TIMEOUT(20)
    READY_TIMEOUT(200)
    TERMTYPE(ALLTERM)
    RESTART_METHOD(ELEMTERM,PERSIST)

DEFINE POLICY NAME(POLICY2) REPLACE(YES)
  RESTART_GROUP(NEWS@SERVER)
  TARGET_SYSTEM(*)
  ELEMENT(CORPNEWS@SERVER)
    RESTART_METHOD(BOTH,STC,'S CICSNEWS')
/* To delete POLICY1, POLICY2, please see note 6 above */
/*

```

Figure 67. IXCARMP0 - Administrative Policy Data in ARM Couple Data Set (Part 3)

Administrative Data Utility for CFRM Policy Data

Sample JCL to run the utility to create or update CFRM administrative data is shown in Figure 68 on page 306. The SYS1.SAMPLIB member name is IXCCFRMP.

Administrative Data Utility

```

//IXCCFRMP JOB
//*
//* EXAMPLE JCL TO CREATE OR UPDATE THE ADMINISTRATIVE POLICY DATA IN THE
//* COUPLE DATA SET FOR COUPLING FACILITY RESOURCE MANAGER (CFRM)
//*
//* NOTES:
//*
//* 1. SYSPRINT DD IS A REQUIRED DD STATEMENT FOR THE UTILITY
//*    OUTPUT
//* 2. SYSIN DD IS A REQUIRED DD STATEMENT FOR THE UTILITY
//*    CONTROL STATEMENTS
//* 3. POLICY DEFINITION OR GENERATION OF POLICY REPORTS FOR LARGE
//*    CFRM COUPLE DATA SETS (E.G., DATA SETS FORMATTED FOR MORE THAN
//*    255 STRUCTURES) MAY REQUIRE MODIFICATION OF THE REGION PARAMETER
//*    AND/OR THE IEFUSI EXIT SO AS TO SATISFY THE VIRTUAL STORAGE
//*    REQUIREMENTS OF THE ADMINISTRATIVE DATA UTILITY.
//*
//*
//*****
//STEP20 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSIN DD *

DATA TYPE(CFRM) REPORT(YES)

DEFINE POLICY NAME(POLICY1) REPLACE(YES)

CF NAME(FACIL01)
TYPE(123456)
MFG(IBM)
PLANT(02)
SEQUENCE(123456789012)
PARTITION(1)
CPCID(00)
SIDE(0)
DUMPSPACE(2000)
CF NAME(FACIL02)
TYPE(123456)
MFG(IBM)
PLANT(02)
SEQUENCE(123456789012)
PARTITION(2)
CPCID(00)
SIDE(1)
DUMPSPACE(2000)
STRUCTURE NAME(LIST_01)
SIZE(10000)
INITSIZE(1000)
MINSIZE(500)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(FACIL01,FACIL02)
EXCLLIST(CACHE_01)
STRUCTURE NAME(CACHE_01)
SIZE(1000)
REBUILDPERCENT(25)
PREFLIST(FACIL02,FACIL01)
EXCLLIST(LIST_01)

/* DELETE POLICY NAME(POLICY2)*/
/*

```

Figure 68. IXCCFRMP - Administrative Policy Data in CFRM Couple Data Set

Administrative Data Utility for the LOGR Policy

Sample JCL to run the utility to define, update, delete and list entries in the LOGR policy. You can also view member IFBLSJCL, in SYS1.SAMPLIB, as an example of how to use IXCMIAPU with system logger.

Note: All staging data set keywords must be specified after STG_DUPLEX(YES) and before all non-staging data set keywords.

```
//DEFINE JOB
//STEP1 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *

DATA TYPE(LOGR) REPORT(YES)

DEFINE STRUCTURE NAME(STRUCTURE1)
LOGSNUM(32)

DEFINE LOGSTREAM NAME(STREAM1)
STRUCTNAME(STRUCTURE1)
LS_MGMTCLAS(STANDARD)
HLQ(LOGGER)

DEFINE LOGSTREAM NAME(STREAM2)
LIKE(STREAM1)
LS_STORCLAS(STANDARD)

DEFINE LOGSTREAM NAME(STREAM3)
DASDONLY(YES)
MAXBUFSIZE(40000)

UPDATE LOGSTREAM NAME(STREAM1)
LS_DATACLAS(VSAMLS)

UPDATE LOGSTREAM NAME(STREAM2)
RETPD(60)

UPDATE LOGSTREAM NAME(STREAM2)
AUTODELETE(YES) RETPD(30)
UPDATE LOGSTREAM NAME(STREAM3)
STRUCTNAME(STRUCTURE1)

DEFINE LOGSTREAM NAME(STREAM4)
STRUCTNAME(STRUCTURE1)
STG_DUPLEX(YES)
DUPLEXMODE(UNCOND)
STG_DATACLAS(VSAMLS)
STG_MGMTCLAS(STANDARD)
LS_MGMTCLAS(STANDARD)
HLQ(LOGGER)

DELETE LOGSTREAM NAME(STREAM2)
LIST LOGSTREAM NAME(STREAM*) DETAIL(YES)
LIST STRUCTURE NAME(STRUCTURE1) DETAIL(YES)
/*
```

Figure 69. Administrative Inventory Data in LOGR Couple Data Set

Administrative Data Utility for SFM Policy Data

Sample JCL to run the utility to create or update an SFM administrative policy is shown in Figure 70 on page 308. The SYS1.SAMPLIB member name is IXCSFMP.

```

//STEP20 EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSIN DD *

DATA TYPE(SFM) REPORT(YES)

DEFINE POLICY NAME(POLICY1) CONNFALL(NO) REPLACE(YES)

SYSTEM NAME(*)
ISOLATETIME(0)

DEFINE POLICY NAME(POLICY2) CONNFALL(YES) REPLACE(YES)

SYSTEM NAME(*)
ISOLATETIME(0)
WEIGHT(5)

SYSTEM NAME(SYS1)
PROMPT
WEIGHT(25)

DEFINE POLICY NAME(POLICY3) REPLACE(YES)

SYSTEM NAME(SYS1)
DEACTTIME(10)
WEIGHT(3)

SYSTEM NAME(SYS2)
DEACTTIME(10)
WEIGHT(10)

RECONFIG
FAILSYS(SYS1) ACTSYS(SYS2) TARGETSYS(ALL)
ESTORE(YES) STORE(YES)

/* DELETE POLICY NAME(POLICY1) */

/*

```

Figure 70. IXCSFMP - Administrative Policy Data in SFM Couple Data Set

Appendix C. Calculating Sizes for Signaling Services

This appendix contains detailed information for calculating the size of a coupling facility list structure for signaling and for calculating the size of the message buffers.

Calculating the Size of a List Structure for Signaling

To calculate the size of a list structure for signaling, use the formula in *PR/SM Planning Guide*.

A list structure for signaling requires the following attributes:

- A list-structure-type that uses adjunct and data but neither keys nor names.
- An entry-to-element ratio of 1
- A list-element-characteristic value of 4
- A maximum data-list-entry size value of 16
- A lock-table-entry characteristic value of 0
- A lock-table-entry count value of 0
- The number of list headers required depends on the number of systems using the structure for pathin and/or pathout. Use the following formula to calculate the number of list headers:

$$\#lists = 8 + \#paths$$

where:

$$\begin{aligned} \#paths = & (\#systems \text{ using the CF list structure as PATHOUT} \\ & * \#systems \text{ using the CF list structure as PATHIN}) - \\ & \#systems \text{ using the CF list structure as both} \\ & \text{PATHOUT and PATHIN} \end{aligned}$$

- List entries are tracked by entries (not by elements). XCF sets list limits on the individual lists in the structure.
- The structure disposition is DELETE.
- The connection disposition is DELETE.
- Structure rebuilds are allowed.
- Non-volatility is not required.
- List monitoring is used.

A list notification vector is allocated on each system to perform this monitoring. You need one vector for each signaling structure on each system in the sysplex that is using the structure. The number of entries in the vector is equal to the number of systems from which the system will receive signals through that signaling structure.

Additional Considerations

The following topics provide more background on calculating the size of a signaling structure.

Required Space

- OS/390 requires eight list headers to manage a coupling facility list structure for signaling.

Calculating Signaling Sizes

- So that at least one signaling path can be established, a list structure must be allocatable with at least one list header more than the number needed by MVS (#XCF_lists).
- To allow the maximum size signal supported by the MVS signaling service (61K bytes) to be inserted into a single list entry, a list structure used for signaling must be allocated so that a list entry can contain up to 64K bytes of data (up to sixteen 4096 byte data elements per data entry).

Therefore, in OS/390, a coupling facility list structure defined for signaling must have a minimum of twenty-five list entries. If this absolute minimum of 25 list entries is not available, the list structure remains defined to MVS but is considered inoperative and cannot be used for signaling.

Recommended Space

The size specified for the coupling facility list structure in the CFRM policy should be large enough to allow for the recommended number of list headers and list entries, as described below. Specifying more than this minimum amount of space allows MVS to queue more signals on a list simultaneously and improves signaling performance.

- It is recommended that the coupling facility list structure be large enough to have at least one signal queued simultaneously on every list header used as a signaling path.

The number of data elements needed for these list entries depends on the size of the signals that are sent. For example, for a 61K byte signal (the largest signal supported by the MVS signaling service), 16 data elements would be needed. Since the entry to element ratio is 1:1, 16 list entries would also be allocated. Thus, to allow for one 61K byte signal to be queued simultaneously over every signaling path, the list structure must be large enough to accommodate 16 list entries per path.

- It is also desirable for the list structure to be large enough to accommodate the peak number of signals simultaneously queued over each list header for a signaling path.

Otherwise, MVS is forced to queue the signal on the sending system. This delays signal delivery and increases the amount of real storage needed for message buffers.

- MVS also needs list entries to manage the coupling facility list structure. The number and size of these list entries varies with:
 - The number of list headers in the list structure,
 - The maximum number of systems that can connect to the list structure,
 - The release of MVS that those systems are running.

In OS/390, it is recommended that the list structure be large enough to provide one list entry per connected system, plus eight additional list entries for control data. Each list entry requires only one data element.

If fewer than the recommended number of list entries are available, MVS might continue to use the coupling facility list structure for signaling, although possibly with some degradation.

Possible Consequences of Too Little Space

If either the coupling facility policy or the coupling facility list structure itself provides less than the recommended minimum amount of storage, signaling may be affected in undesirable ways. For example:

- If there is not enough space for one signal per signaling path, the potential for sympathy sickness exists because the failure of a system to process a signal

Calculating Signaling Sizes

could prevent other systems from transferring a signal. Thus, one non-operational system could interfere with the signal traffic between other unrelated systems, possibly causing a loss of signaling connectivity.

- A shortage of space (list entries) for a particular signaling path could cause signals to be queued on the sending system, thus delaying signal delivery and increasing the amount of real storage needed on the sending system for signal buffers.
- A shortage of space for control data could delay the establishment of signaling paths through the coupling facility list structure. Such delays could prevent an IPLing system from establishing signaling connectivity in a timely manner, or cause Sysplex Failure Management to partition systems out of the sysplex because signaling connectivity could not be reestablished quickly enough.

Vector Space Required on the MVS Systems

The MVS signaling service needs to establish monitoring of every list in the coupling facility list structure (not every system must monitor every list though). A list notification vector is allocated on each system to perform this monitoring.

Basically you need one vector for each signaling structure on each system in the sysplex that is using that structure. The number of entries in the vector is equal to the number of systems from which the system will be receiving signals via that signaling structure.

If the vector cannot be allocated with enough capacity to support the minimum requirements for monitoring, the coupling facility list structure remains defined but cannot be used for signaling. The vector might satisfy the minimum requirements for monitoring, but might not be large enough to allow the system to monitor all the desired lists used for signaling paths. Signaling paths that cannot be established for this reason are considered inoperative.

When MVS detects circumstances in which successful allocation of the list notification vector might be possible, MVS automatically attempts to start using the coupling facility list structure for signaling again. If a vector was allocated, but was not large enough to support all the desired signaling paths, MVS attempts to increase the length of the vector. A rebuild of the coupling facility list structure is not a suitable recovery action since the resource problem is associated with the system itself.

The particular hardware configuration of the system, including the connectors to coupling facility list structures and their use of local vectors, will determine whether there is sufficient resource to allow allocation of the list notification vector. MVS cannot detect all the changes that might make it possible for the list transition vector to be allocated. Therefore, manual intervention may be required, either to make modifications to the configuration, or to cause the MVS signalling service to attempt to start using the coupling facility list structure for signaling again. When configuring their systems, customers should ensure that there is enough resource available to allow the needed list transition vectors to be established for each coupling facility list structure defined to the MVS signaling service.

Message Information for Some MVS Groups

This topic provides information on the lengths, frequency, and performance requirements for messages sent by some MVS groups. This information can be useful when defining transport classes.

- Consoles

Calculating Signaling Sizes

For the consoles XCF groups, SYSMCS and SYSMCS2, the length of messages vary. The maximum length of a message is 4K.

- DAE

The dump analysis elimination (SYSDAE) group issues one message per dump request, whether or not the dump is suppressed. Each message is approximately 400 bytes long.

- Global resource serialization

For the global resource serialization XCF group, SYSGRS, the most important message to consider is the RSA-message, which holds information about requests for global resources. The RSA-message can contain up to 32K bytes of data. If global resource serialization is used to serialize resources in your sysplex, then the RSA-message will be sent frequently and will have high performance requirements. (For example, the performance of data base programs and the speed with which access to data sets is granted will depend on the speed with which the RSA-message is processed.) If an alternate serialization method is used, then the performance requirements for global resource serialization signaling are not as critical.

- JES2

The JES2 XCF group issues two kinds of messages:

- During a sysplex reconfiguration, messages of less than 1K in length will flow regularly between the JES2 XCF group members in the sysplex.
- As a JES2 system in a sysplex joins a network, it receives JES2 XCF messages with information about the other systems in the network. These messages can be up to 32K bytes in length and contain approximately 30 - 40 bytes of data about each system connected to the network. (Messages longer than 32K, are sent in two or more separate messages.)

- JES3

The JES3 group uses XCF messages to transport JES3 staging areas between the JES3 XCF group members in the sysplex. Most staging areas are less than 1K in length; however, the JES3 XCF group combines multiple staging areas and transports them in a single message that can be up to 64K bytes long.

- WLM

The WLM group, SYSWLM, sends a message approximately once every ten seconds. The length of the message depends on the number of service classes defined in the WLM policy definition. It is advantageous, but not critical, for the message to be received at once.

Estimate the size needed as follows: 320 bytes * the number of service class periods (in all service classes) that have a response time or velocity goal.

The message is sent by MVS images running in GOAL mode to other MVS images running in GOAL mode. If you have not yet migrated two systems to GOAL mode, no messages will be sent.

Calculating Message Buffer Space — An Example

To limit the number of message buffers in your system and maintain adequate signaling performance, you need to know how much buffer space is required to support a particular message length. (The message buffer space required for a particular message length can vary from release to release.) For this release of MVS, compute the value as follows: subtract 1K from the message length, round the result up to the nearest multiple of 4K, then add 2K. For example, if the message length is 40,000 bytes:

Calculating Signaling Sizes

40,000 - 1,024 = 38,976
 (Rounding 38,976 up to the nearest multiple of 4K equals 40K)
 40K + 2K = 42K

So, a MAXMSG value of 42K provides enough message buffer space to send a message of 40,000 bytes.

“Example 3 — Handling a Range of Message Traffic Considerations” on page 90 also has information about calculating message buffer space.

Total Buffer Space

To compute the maximum message buffer space that MVS can allocate per system, use the following formula:

total = paths + classes + LOCALMSG value

Where: paths = the sum of all message buffer spaces specified for all paths.
 classes = the sum of all message buffer spaces calculated for all transport classes.
 LOCALMSG value = the sum of local system message buffer spaces for all transport classes

To calculate the buffer space for a transport class, multiply 64K or the specified message buffer space (whichever is greater) by the number of systems in the sysplex.

For local message buffer spaces, add the LOCALMSG value for each transport class specified for the system.

For example, in a sysplex of four systems, suppose COUPLxx is defined as follows for one of the systems:

```
COUPLE  SYSPLEX(PLEXOF4)  MAXMSG(16)

CLASSDEF CLASS(DEFAULT)  MAXMSG(96)
CLASSDEF CLASS(TCGRS)    MAXMSG(32)

PATHIN  DEVICE(110,111)
PATHIN  DEVICE(112,113)  MAXMSG(48)

PATHOUT DEVICE(120,121)  CLASS(TCGRS)
PATHOUT DEVICE(8C40)    MAXMSG(50)
PATHOUT DEVICE(8C50)

LOCALMSG CLASS(DEFAULT)  MAXMSG(20)
```

For this system, the buffer space values for the paths and transport classes are:

Signal Path	Buffer Space	
110	16K	(default from COUPLE MAXMSG)
111	16K	(default from COUPLE MAXMSG)
112	48K	(specified in PATHIN statement)
113	48K	(specified in PATHIN statement)
120	32K	(default from class TCGRS)
121	32K	(default from class TCGRS)
8C40	50K	(specified in PATHOUT statement)
8C50	96K	(default from class DEFAULT)

Transport class	Buffer space
-----------------	--------------

Calculating Signaling Sizes

-----	-----
DEFAULT	404K (96K x 4 systems, + 20K local increment specified for LOCALMSG)
TCGRS	256K (Minimum 64K x 4 systems)

The total maximum buffer space for the system is:

$$\begin{aligned} \text{Total} &= 16K + 16K + 48K + 48K + 32K + 32K + 50K + 96K + 404K + 256K \\ &= 998K \end{aligned}$$

Appendix D. Sysplex Configurations

IBM recommends that you provide redundant channels, signaling paths, Sysplex Timers, and shared couple data sets in order to avoid having a single point of failure in the sysplex.

For example, if a 3088 fails and there is not a second 3088 to maintain signaling connectivity between MVS systems in the sysplex, all but one system must be removed from the sysplex and put in a wait state. Or, if you do not have an alternate couple data set and the primary couple data set fails, all systems are put in a non-restartable wait state.

When a sysplex system is running under VM or in a logical partition that is sharing processor resources, you must ensure that its processor resources are sufficient to ensure good overall system performance. A system that is running without sufficient resources can degrade the performance of sysplex-wide communications functions, such as global resource serialization, signaling, and console communications.

- “Under VM - Physical Configuration of a Sysplex With MVS Guests”
- “On PR/SM - Physical Configuration of a Sysplex on LPARs” on page 317
- “Configuration of a Sysplex on LPAR and Non-LPAR Systems” on page 318

Under VM - Physical Configuration of a Sysplex With MVS Guests

Figure 71 on page 316 shows the physical connections needed for VM to host three MVS systems in a sysplex on one processor.

A sysplex under VM cannot span processors. However, it is possible to simulate a sysplex comprised of systems on different CECs. See “Controlling the Verification of CPU Identification” on page 316.

The guest MVS systems can be set up as a sysplex with each having a COUPLExx parmlib member and so on.

- VM handles the devices assigned to the MVS guests. In the figure, 3088s are shown, but the signaling paths could be on an ESCON channel operating in CTC mode.
- A channel on the processor is connected to the primary couple data set and a second channel is connected to the alternate couple data set. Both are accessed by the VM system and assigned to the MVS guests.
- A coupling facility cannot be used in this configuration.
- A Sysplex Timer is not used in this sysplex because MVS must use the clock of the processor for synchronized timestamps. SIMETRID is used instead.

Sysplex Configurations

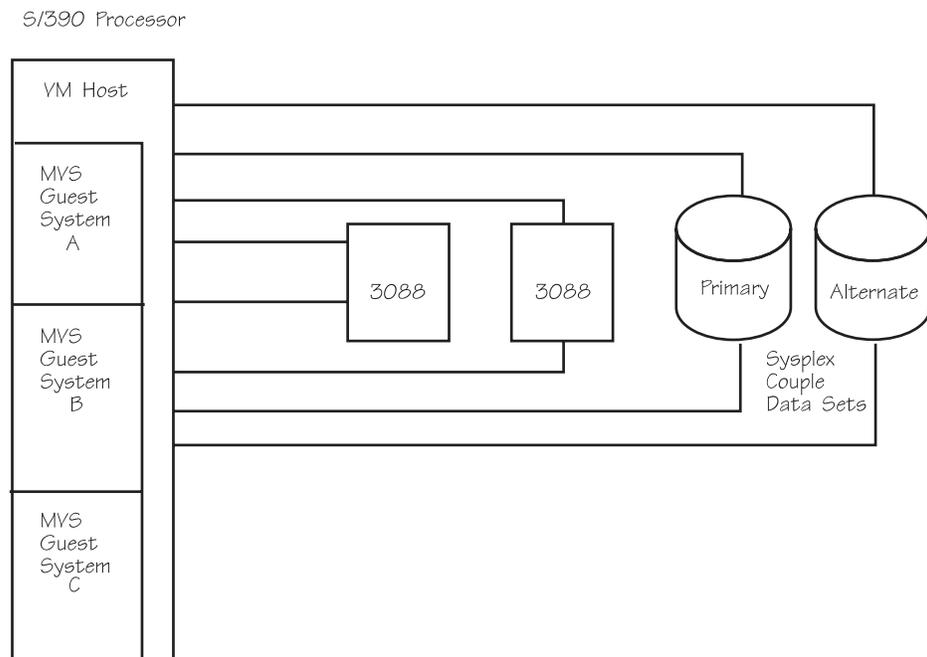


Figure 71. Under VM - Physical Configuration of a Sysplex With MVS Guests

Note: XCF requires VM/ESA to provide virtual IBM 3088 Multisystem Channel Communication Unit support. On VM/XA, the IBM 3088 Multisystem Channel Communication Unit is required.

Controlling the Verification of CPU Identification

Prior to OW27741, XCF required each MVS guest system in the sysplex to have a CPUID (CPU identifier) with the same serial number and model number. You had to ensure that the serial number and model number were the same for each system in the sysplex either by using a VM command to set the CPUID or by using the system-generated CPUID.

With OW27741, which added a keyword to the COUPLExx parmlib member, it is possible to define VM guest systems participating in a sysplex to have disparate CPU identification information. This configuration is useful when using VM to simulate a sysplex comprised of systems on different CECs. The new COUPLExx keyword, VMCPUIDTOLERATION(NOIYES) allows you to specify whether the VM guest system is to perform CPU identification information verification with respect to other active systems in the sysplex. The verification takes place only at XCF initialization time.

See *MVS Initialization and Tuning Reference* for information about how to specify the VMCPUIDTOLERATION keyword of the COUPLExx parmlib member.

Installing OW27741

The following scenario describes a migration path from a sysplex of VM guest systems with identical CPUIDs to a sysplex comprised of systems with disparate CPU identification information. Note that no sysplex-wide IPL is required to install OW27741.

1. Start with a sysplex of VM guest systems having identical CPU identification information and with OW27741 not installed.

2. Take down each VM guest system, one at a time, and re-IPL it into the sysplex with OW27741 installed.
3. Modify the COUPLExx parmlib member for all VM guest systems to specify VMCPUIDTOLERATION(YES).
4. Take down each VM guest system, one at a time, and re-IPL it into the sysplex after having redefined its virtual CPU identification to have the desired, unique value.

At this point, the sysplex will be comprised of systems with disparate CPU identification information, simulating a sysplex of systems resident on different CECs. XCF will ensure that all systems share a common SIMETRID specification, but because VMCPUIDTOLERATION(YES) is specified, will tolerate the disparate CPU identification information that the systems have defined.

Another migration path for installing OW27741 would be to combine steps 2, 3, and 4 above into one step and re-IPLing each system only once. Each IPLed system would have OW27741 installed, VMCPUIDTOLERATION(YES) specified, and unique virtual CPU identification defined for it. This method simplifies the overall migration scenario, but might cause problems if one of the downlevel systems needs to be re-IPLed.

Handling Downlevel Systems

Once any system in the sysplex has had OW27741 installed and been IPLed into the sysplex with VMCPUIDTOLERATION(YES) and its own unique CPU identification information specified, a downlevel system might be prevented from being IPLed into the sysplex. (A downlevel system is one that does not have OW27741 installed, or has OW27741 installed but does not specify VMCPUIDTOLERATION(YES).) The downlevel system will compare its virtual CPU identification information to that of other active systems in the sysplex when it attempts to IPL into the sysplex, and if a system with disparate CPU identification is found, the IPL of the downlevel system will fail. Once the sysplex has been migrated to a point where unique CPU identification information is in use on different systems in the sysplex, it might not be possible to IPL a downlevel system back into the sysplex, and therefore it might be impossible to fall back to a sysplex comprised of downlevel systems without a system-wide re-IPL.

On PR/SM - Physical Configuration of a Sysplex on LPARs

Figure 72 on page 318 shows the physical connections needed to connect two MVS systems on two logical partitions (LPARs) in one PR/SM processor and provide redundant channels, signaling paths, and couple data sets.

In this configuration, a failure on any one channel, path, ESCD, or couple data set does not bring down the sysplex. In the event of a failure, MVS can use a different channel, path, ESCD, or alternate couple data set to maintain sysplex operation.

- The figure shows channel to channel connections through ESCDs.
- A channel on each LPAR is connected to the primary couple data set and a second channel is connected to the alternate couple data set; both data sets are shared by the two LPARs.
- A Sysplex Timer is not needed in this sysplex because MVS can use the clock of the processor to synchronize timestamps. However, in this sysplex, a Sysplex Timer could be used.
- A coupling facility could be used in this configuration.

Sysplex Configurations

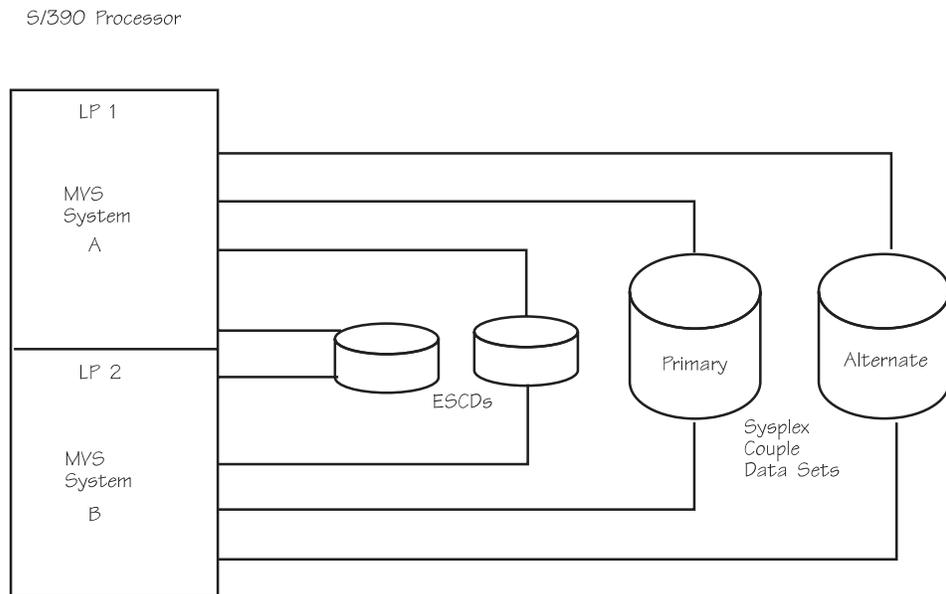


Figure 72. On PR/SM - Physical Configuration of a Sysplex on LPARs

Note: If you have PR/SM installed with the ESCON Multiple Image Facility (EMIF), you can share ESCON channels among logical partitions. For information about EMIF and shared channel paths, see *PR/SM Planning Guide* and *OS/390 HCD Planning*.

Configuration of a Sysplex on LPAR and Non-LPAR Systems

Figure 73 on page 319 shows the connections required to connect two processors, one of which runs two MVS images on PR/SM. It also illustrates redundant sysplex couple data sets, channels, signaling paths, and Sysplex Timers.

In this configuration, a failure on any one channel, path, ESCD, or sysplex couple data set does not bring down the sysplex. In case of a failure, MVS can use a different sysplex couple data set, channel, path, ESCD, or Sysplex Timer to maintain sysplex operation.

- The figure shows channel to channel connections through ESCDs.
- A channel on each LPAR in CPC1 is connected to the primary couple data set and a second channel is connected to the alternate couple data set. A channel on CPC2 is connected to the primary couple data set and a second channel is connected to the alternate couple data set. Both data sets are shared by all three systems.
- A Sysplex Timer is needed to synchronize timestamps between the two processors.
- A coupling facility can be used in this configuration.

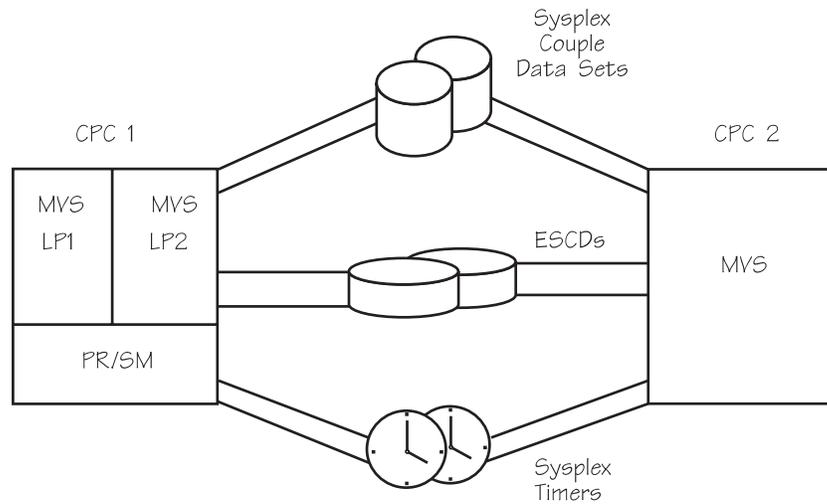


Figure 73. Configuration of a Sysplex on LPAR and Non-LPAR Systems

Note: If you have PR/SM installed with the ESCON Multiple Image Facility (EMIF), you can share ESCON channels among logical partitions. For information about EMIF and shared channel paths, see *PR/SM Planning Guide* and *OS/390 HCD Planning*.

Sysplex Configurations

Appendix E. Coupling Facility Guidelines

There may be occasions when it is necessary to alter the Parallel Sysplex configuration — to add or modify processors, to upgrade coupling facilities, or to remove and replace hardware or software components of the configuration. This section provides guidelines specifically for handling the configuration of a coupling facility in your Parallel Sysplex environment. It emphasizes the relationship between the structures that may already be allocated in a coupling facility and the coupling facility and structure definitions in a CFRM policy. It is important to understand these relationships thoroughly before beginning any configuration changes for a coupling facility.

The following topics are included:

- “Understanding Some Basic Concepts”.
- “Guidelines for Planned Outages” on page 325.
- “Guidelines for Unplanned Outages” on page 340.

Understanding Some Basic Concepts

This section reviews the concept of persistence and the role of coupling facility resource management (CFRM) in a Parallel Sysplex environment. Both topics should be understood thoroughly before reconfiguring a coupling facility in your installation.

Reviewing the Use of Structure Data

The use of data stored in a coupling facility structure is application-dependent. Applications can store data in a structure that, in the event of an application failure, can be recovered by a new instance of the application. Alternately, applications can store data in a structure along with pertinent connector data that will be required for recovery purposes in the event of a failure. The concept of a bond between the data in the structure and the connector to the structure is referred to as a **recovery bind**. Once a connector has performed the recovery processing required by the application, XCF removes the recovery bind. For example, when an instance of IRLM fails, the failed IRLM has a recovery bind to data in the structure (in this case, information about retained locks). The data is required for IRLM recovery processing.

Reviewing the Concept of Persistence

Structure persistence implies that the structure remains allocated in the coupling facility even after there are no longer any active or failed-persistent connectors to the structure. Generally speaking, applications use a persistent structure for data that needs to remain in a coupling facility structure even if there are no connectors to that structure. A persistent structure is not deleted when all active and failed-persistent connectors to the structure cease to exist. Non-persistent structures are generally used by applications with transient data, or data that must persist only when there is a recovery bind associated with a particular set of failed connectors to the structure. A non-persistent structure is deleted when all active and failed-persistent connectors to the structure cease to exist.

Connector persistence implies the existence of a recovery bind between a connector to the structure and some data that is stored in the structure. For example, when an instance of IRLM fails, the failed IRLM becomes a failed-persistent connector. This indicates that the instance of IRLM has a recovery

Coupling Facility Guidelines

bind to data in the structure, which should persist even after the connector has terminated. Non-persistent connectors, on the other hand, have no such recovery binds.

Note that a failed-persistent connector is bound to a particular allocated instance of the structure; MVS will not deallocate that instance of the structure, regardless of structure persistence or non-persistence, as long as any failed-persistent connector to the structure remains in existence.

Specifying Structure and Connector Persistence

Connector and structure persistence are for the most part independent choices that the application using a structure makes based on design philosophy and considerations. For example:

- An application that stored permanent data in the coupling facility, but that did not have a concept of recovery binds between data in the coupling facility structure and any particular connector, would probably use a persistent structure with non-persistent connectors.
- An application that stored transient data in the coupling facility, but that needed some connector-related data in the structure to persist for recovery purposes in the event of a connector failure, would probably use a non-persistent structure with persistent connectors.

Sample Persistence Scenarios

The following examples describe the effect of persistence in a sysplex.

Persistent Scenarios from a Connector's View: The behavior of persistent connectors and structures is fairly straightforward in scenarios where it is the connectors to the structures who come and go over time. The following general rules apply to a connector's actions:

Normal disconnect: When a persistent connector to a structure disconnects normally, it does not become a failed-persistent connector. (It becomes "undefined"). If no other active or failed-persistent connectors to the structure remain at the time of the disconnect, the structure is deallocated (if the structure is non-persistent) or remains allocated (if the structure is persistent).

Abnormal disconnect: When a persistent connector to a structure disconnects abnormally or terminates abnormally, indicating that it needs to do recovery at some point in the future, the connection is placed into the failed-persistent state. The structure will never be deallocated while this failed-persistent connector exists.

Reconnect: When a failed-persistent connector reconnects to the structure, the connector will perform the recovery processing that was associated with its recovery bind to the structure (the details of this recovery processing are, of course, specific to the application). It will then resume normal use of the structure as an active connector.

Persistent Scenarios in Coupling Facility Failure Situations: When the coupling facility fails, the scenarios involving persistent connectors are a little more complicated and perhaps less intuitive. Generally, the following occurs:

- When the coupling facility fails, a loss of coupling facility connectivity is recognized by MVS and reported to the coupling facility structure connectors.
- If rebuild is supported and succeeds, the connectors remain active and switch over to using the new rebuilt structure in a different coupling facility.

Coupling Facility Guidelines

- If rebuild is not supported or fails, the connectors disconnect abnormally, indicating a recovery bind to the structure for which connectivity was lost. The persistent connectors enter the failed-persistent state.
- When a failed-persistent connector attempts to reconnect to the structure, because there is still a recovery bind to the old instance of the structure (which, as far as MVS knows, is still allocated in the coupling facility to which connectivity was lost), MVS will attempt to reconnect to the structure from which connectivity was lost. There are several cases here.
 - If the coupling facility is still unavailable, the attempt to connect to the structure will fail for that reason. If the coupling facility remains unavailable indefinitely, then attempts to connect to the structure will continue to fail indefinitely.

In a case like this, if it is necessary to restart the application(s) before the coupling facility containing the structure to which a recovery bind exists can be restored, the recovery bind must be broken. To do this, use the SETXCF FORCE,CON command to delete the failed-persistent connectors and, if applicable, use the SETXCF FORCE,STR command to delete the persistent structure, so that a new instance of the structure may be allocated and brought into use. Generally, when a SETXCF FORCE is done, the application has some alternative means to carry out the recovery processing represented by the bind to the data in the structure.
 - If the coupling facility has been restored and the structure is still allocated in that coupling facility (that is, there was only a loss of connectivity to the coupling facility, as opposed to a loss of the coupling facility and its data), then the reconnect will be successful and the connector will do recovery processing using the data in the structure.
 - If the coupling facility has been restored and the structure is not allocated in that coupling facility (that is, the coupling facility failed and the structure was lost), MVS realizes that the old instance of the structure no longer is allocated, allocates a new instance of the structure, and reports to the connector that the connection request succeeded but resulted in the allocation of a new structure instance. The application will need to accomplish its recovery through alternative means, because the data in the structure is not present in this case. This might mean recovery from logs or other manual recovery procedures that must be performed.

For more information on structure and connector persistence, and on forcing failed-persistent structures and connectors, see *OS/390 MVS Programming: Sysplex Services Guide*.

Reviewing the Role of CFRM

“Chapter 4, Managing Coupling Facility Resources” provides detailed information about coupling facility resource management and how it maintains status for coupling facilities and structures in a Parallel Sysplex configuration. Important points to remember when changing a coupling facility configuration are:

- CFRM uses the CFRM couple data set to maintain status data about the use of coupling facility resources in the sysplex.
- A change to the CFRM policy may need to remain pending if coupling facility or structure resources defined in the active CFRM policy are in use. Operator intervention may be required.

Structures allocated in any coupling facility that is undergoing changes need to be dealt with before any coupling facility reconfiguration. The CFRM policy describes not only the characteristics of the coupling facilities in the sysplex but also the

Coupling Facility Guidelines

definitions of the structures that might or might not be allocated in those coupling facilities. In addition, the active CFRM policy contains real-time data about the structures and their connectors currently allocated in or not yet totally deallocated from the coupling facility. Before attempting any coupling facility modifications, you must understand this relationship and prepare for the necessary CFRM policy adjustments.

Modifying a CFRM Policy before Configuration Changes

The CFRM policy may need to be modified prior to changing a coupling facility configuration. See “Chapter 4, Managing Coupling Facility Resources” for general information about CFRM policy modifications.

Adding a New Coupling Facility: Create a new CFRM policy (or update an existing CFRM policy) to add the coupling facility in the coupling facility definition statement, and to add the coupling facility in all appropriate structure preference lists. Activate the new CFRM policy using the SETXCF START,POLICY command.

Removing a Coupling Facility: Create a new CFRM policy (or update an existing CFRM policy) to delete any reference to the coupling facility to be removed, including structure preference lists. Activate the new CFRM policy using the SETXCF START,POLICY command.

When you start the new CFRM policy, structures that are currently allocated in the coupling facility being removed will enter a policy-change-pending state. Structures that enter a policy-change-pending state remain in that state until the structure is deallocated and reallocated through a new policy. Structures that reside in a coupling facility that is not being removed might remain in a policy-change-pending state until the original policy is restored.

Adding a Structure: Create a new CFRM policy (or update an existing CFRM policy) to establish the new structure definition. Activate the new CFRM policy using the SETXCF START,POLICY command.

Moving a Structure to Another Coupling Facility: To move a structure from one coupling facility to another coupling facility:

- If necessary, create a new CFRM policy (or update an existing CFRM policy) to add additional coupling facilities to the preference list for the structure that is to be moved. The structure may only be allocated in a different coupling facility if that coupling facility is in the structure’s preference list. It might also be necessary or desirable to remove the coupling facility in which the structure is currently allocated from the structure’s preference list, particularly for the case where the users of the structure do not support structure rebuild. Activate the new CFRM policy using the SETXCF START,POLICY command. (Note that it is generally recommended that a structure always have at least two coupling facilities in its preference list.)
- When the structure is being used by an application that supports structure rebuild, use SETXCF START,REBUILD with LOCATION=OTHER to cause the structure to be rebuilt into a different coupling facility from the one in which it is currently allocated, in accordance with the current structure preference list. There must be at least one other usable coupling facility in the structure’s preference list in order for this command to be processed successfully.
- When the structure is being used by an application that does not support structure rebuild, follow the application’s recommended procedures for discontinuing use of the old structure. Ensure that the old coupling facility structure has been deallocated from the old coupling facility, and that the CFRM policy change to remove the old coupling facility from the structure’s preference

Coupling Facility Guidelines

list has taken effect, using the DISPLAY XCF,STR command. Then re-initialize the applications. This will cause a new instance of the structure to be allocated in accordance with the currently active CFRM policy definition.

See also “Using the POPULATECF Function to Rebuild Coupling Facility Structures” on page 63 for a description of how to restore structures to the most preferred coupling facility after it has been added or returned to the configuration.

Deleting a Structure from a Coupling Facility: To delete a structure from a coupling facility:

- Create a new CFRM policy (or modify an existing CFRM policy) to delete the structure definition. Activate the new CFRM policy using the SETXCF START,POLICY command. Assuming that the structure is allocated at the time, the policy change to delete the structure definition will become pending.
- Follow the application’s recommended procedures for discontinuing use of the structure that is to be removed.
- If necessary, use the SETXCF FORCE command to delete persistent connections and/or the persistent structure. See “Reviewing the Concept of Persistence” on page 321 for information about persistence.
- When the old structure is deallocated, the pending policy change to remove the structure from the policy will take effect. Ensure that this has occurred using the DISPLAY XCF,STR command.

Guidelines for Planned Outages

This section provides guidelines for configuring one or more coupling facilities in your Parallel Sysplex environment and includes the following topics:

- Adding a New Coupling Facility
- Planning for a Coupling Facility Shutdown
- Removing a Structure from a Coupling Facility
- Removing a Coupling Facility
- Replacing or Moving a Coupling Facility

Adding a Coupling Facility to an Active Sysplex (MVS Image Side)

Follow these steps for the MVS images that you are connecting to a new coupling facility:

1. Add hardware ISC links and attach them to the new coupling facility.
2. Create a new IODF containing the new coupling facility definition. For the MVS image side, this is the definition of the coupling facility sender CHPIDs. For the coupling facility image side, this is the definition of the coupling facility receiver CHPIDs.

Both the MVS image side and the coupling facility image side must be defined in the same IODF.

3. Use HCD to connect the coupling facility sender CHPIDs to the coupling facility receiver CHPIDs. This completes the coupling facility definition in the IODF.
4. Use the HCD Build IOCDS function to create a new IOCDS that defines a configuration containing the new coupling facility definition. This is for the MVS image side.
5. If the coupling facility is configured to the Hardware Management Console (HMC) for remote IOCDS management, use the HCD Build IOCDS function to create and download an IOCDS containing the coupling facility receiver definitions.

Coupling Facility Guidelines

6. If the coupling facility is not configured to the HMC for remote IOCDs management, on the MVS image side use the HCD Build IOCP Input Data Set function to create IOCP input for use on the coupling facility image CPC. Note, for the S/390 9674 Coupling Facility, this input must be on diskette.
7. Dynamically activate the new configuration on the MVS image side.
8. Create a new CFRM policy (or modify an existing CFRM policy) to add a definition for the new coupling facility. Update the preference lists for one or more existing structure definitions to reference the new coupling facility, or add new structure definitions that reference the new coupling facility. See “Chapter 4, Managing Coupling Facility Resources” for information on how to change your CFRM policy.
Activate the new CFRM policy using the SETXCF START,POLICY command.
9. Complete steps listed in “Adding a Coupling Facility to an Active Sysplex (Coupling Facility Image Side)”. Then return here.
10. Configure ISC links online to new coupling facility.
11. Structures can now be allocated in the new coupling facility based on the CFRM policy definitions that are now in effect. Note that structures that are currently allocated in other coupling facilities will not automatically be reallocated in the new coupling facility, regardless of any changes that may have been made in the structures’ preference lists. However, you might choose to rebuild structures to cause a NEW instance of those structures to be allocated in the new coupling facility, using the SETXCF START,REBUILD command. The new instances of the rebuilt structures will be allocated in accordance with their current CFRM policy preference list definitions.

Adding a Coupling Facility to an Active Sysplex (Coupling Facility Image Side)

The following tasks are required for the coupling facility image side:

1. Install hardware and make the necessary hardware connections to the MVS image CPCs.
2. If the IOCDs has not been downloaded as a result of using the remote IOCDs management function, use IOCP to create an LPAR IOCDs that contains the coupling facility receiver definitions for the new coupling facility. Use the HCD Build IOCP Input Data Set function, on the MVS image CPC, to create the IOCP input from the coupling facility definition in the IODF.
3. Power-on-reset the CPC and use the IOCDs containing the new coupling facility definition. Coupling facility receiver CHPIDs cannot be added dynamically.
4. Define and activate the new coupling facility partition.

Planning for a Coupling Facility Shutdown

Before removing a coupling facility from a Parallel Sysplex configuration, whether for replacement or maintenance, consider the following:

- Remove only one coupling facility at a time from service. Perform maintenance for the coupling facility during off-peak periods.
- To prevent any structures from being created in the coupling facility that you are removing, define a new CFRM policy that does not include any reference to that coupling facility. See step 1 of “Removing a Coupling Facility When Multiple Coupling Facilities Exist” on page 327.
- Resolve FAILED-PERSISTENT and NO CONNECTOR structure conditions before shutting down the coupling facility.

Coupling Facility Guidelines

If a failed-persistent connector exists or if a structure with no connector exists, you should understand whether this is a normal state for the connector or structure and you should follow the procedures specified by the owning application before shutting down the coupling facility. See “Actions to Remove a Structure from a Coupling Facility” on page 331.

- Most configurations should have multiple coupling facilities installed for improved availability. To rebuild structures from the coupling facility to be removed to an alternate coupling facility, ensure that all systems currently using the structures have connectivity to the alternate coupling facility. Also, ensure that the structures to be rebuilt have the name of the alternate coupling facility in their preference list of the CFRM policy. Structures can be rebuilt to one or more alternate coupling facilities.
- If the coupling facility that you plan to remove contains an XCF signaling structure, ensure that you have a full set of redundant signaling paths in place. For example, if signaling is implemented only through a single signaling structure and you do not have any other structures or CTC connections for signaling in the sysplex, you lose system-to-system connectivity when the XCF structure is rebuilt. The loss of system-to-system connectivity lengthens the time it takes to rebuild the signaling structure and might result in XCF timeouts, especially if multiple structures are being rebuilt at the same time.

If you do not have any redundant signaling paths defined in the sysplex, you must rebuild XCF signaling structures one at a time and not concurrent with any other structure rebuild. If redundant signaling paths for all systems are not available during the maintenance procedure, consider increasing the following values in SYS1.PARMLIB to prevent an XCF timeout:

- INTERVAL in COUPLExx for the XCF failure detection interval
- TOLINT in GRSCNFxx for the global resource serialization toleration interval.
- To allow structures to be rebuilt on an alternate coupling facility, ensure that enough capacity (for example, storage, CP cycles, link capacity, and structure IDs) exists on the alternate coupling facility.

If there is not enough capacity on the alternate coupling facility to accommodate the XCF, JES2, and RACF structures to be rebuilt, consider the following:

- For XCF signaling structures, temporarily stop XCF signaling through the structure on the coupling facility that you plan to remove. Instead, use the redundant signaling paths, such as CTCs or a signaling structure in an alternate coupling facility. (See “Deleting the XCF Signaling Structure” on page 336.)
- For JES2 structures, place the checkpoint on DASD and remove the structure from the coupling facility that you plan to shut down. (See “List Structure for the JES2 Checkpoint Data Set” on page 332.)
- For RACF, use RACF in non-data sharing mode and delete the RACF structure on the coupling facility to be removed. (See “Deleting the RACF Database Cache Structures” on page 335.)

Removing a Coupling Facility When Multiple Coupling Facilities Exist

To remove a coupling facility for maintenance or replacement when multiple coupling facilities exist in the configuration, use the following procedure.

1. Create a new CFRM policy that does the following:
 - Deletes any reference to the coupling facility to be removed.
 - Includes the name of the coupling facility where all structures are to be rebuilt in the preference list.

Coupling Facility Guidelines

Use a name for the new CFRM policy that is different from the name of the original CFRM policy so that when coupling facility maintenance is complete, the original policy can be restored.

2. To determine the CHPIDs in use by the coupling facility to be removed, issue the following command on each system connected to the coupling facility: `D CF,CFNAME=cfname`

Information about coupling facility connectivity to systems and coupling facility characteristics is returned in message IXL150I. You must know in advance system connectivity and other information about the coupling facility to be removed. Record the following information that uniquely identifies the coupling facility to be removed:

- NODE DESCRIPTOR (type.mfg.plant.sequence)
- PARTITION
- CPCID
- CHPID (listed under SENDER PATH heading)

You will need this information for step 5.

3. Start the new CFRM policy. `SETXCF START,POLICY,TYPE=CFRM,POLNAME=newpolicyname`
4. To obtain the names of all allocated structures in the coupling facility to be removed, issue the following command: `D XCF,CF,CFNAME=cfname` where **cfname** is the name of the coupling facility to be removed.

If structures exist in the coupling facility, you receive a message that indicates which structures are allocated. If structures are allocated, go to “Actions to Remove a Structure from a Coupling Facility” on page 331. When you have moved all the structures from the coupling facility, go to step 5.

If the response to the command indicates that the new CFRM policy is active or that there are no structures in the coupling facility, go to step 5.

5. Configure all CHPIDs offline to the coupling facility that you are removing. If you are deactivating an ICMF partition, DO NOT perform this step. See “Considerations for Removing an ICMF Partition” on page 330.
 - To take the CHPIDs identified in step 2 offline, issue the following command on each system connected to the coupling facility that you are removing:
`CONFIG CHP(xx,yy),OFFLINE,UNCOND`
 - To ensure that all CHPIDs were taken offline, issue the `D CF` command on all systems connected to the coupling facility to be removed. Identify the coupling facility by the NODE DESCRIPTOR, PARTITION, and CPCID descriptions from the information you recorded in step 2. Verify that all CHPIDs were taken offline.

If the CHPIDs do not come offline using the UNCOND keyword, there are probably one or more structures still allocated in the coupling facility. Recheck the previous steps to ensure that all structures and connectors have been removed from the coupling facility that you are shutting down.

If a structure dump exists on the coupling facility and an SVC dump is not in progress, you can issue the following command to force the deletion of the structure dump: `SETXCF FORCE,STRDUMP,STRNAME=strname,STRDUMPID=strdumpid`

If you need to force CHPIDs that remain online to the coupling facility, you can use the FORCE option on `CONFIG CHP(xx,yy),OFFLINE` to force the CHPIDs offline.

When you enter the CONFIG command with FORCE on a system to take the last path to the coupling facility offline, any active connector to a structure loses connectivity to the structure.

6. Power off the coupling facility. When the coupling facility is deactivated, any remaining structure data is lost.

When the maintenance procedure is complete, bring the coupling facility back into service by restoring the original CFRM policy and moving the structures back into the coupling facility. See “Restoring the Coupling Facility to the Sysplex” on page 338.

Removing the Only Coupling Facility in the Sysplex

It is not recommended to remove the only remaining coupling facility from the sysplex because you might lose important structure data when you power off the coupling facility. Consider activating at least one alternate coupling facility in the sysplex and follow the steps in “Removing a Coupling Facility When Multiple Coupling Facilities Exist” on page 327.

If you are removing the only coupling facility from the sysplex, you cannot continue to run subsystems like IMS and DB2 that participate in coupling facility data sharing. Before you deactivate the coupling facility, you must delete all structures in the coupling facility:

1. To determine the CHPIDs in use by the coupling facility to be removed, issue the following command on each system connected to the coupling facility: `D CF,CFNAME=cfname`

Information about coupling facility connectivity to systems and coupling facility characteristics is returned in message IXL150I. You must know in advance system connectivity and other information about the coupling facility to be removed. Record the following information that uniquely identifies the coupling facility to be removed:

- NODE DESCRIPTOR (type.mfg.plant.sequence)
- PARTITION
- CPCID
- CHPID (listed under SENDER PATH heading)

You will need this information for step 5.

2. To stop the current CFRM policy, issue the following command: `SETXCF STOP,POLICY,TYPE=CFRM`
3. To determine which structures are allocated in the coupling facility, use the following command: `D XCF,CF,CFNAME=cfname`

For procedures to delete IBM structures, see “Deleting Structures that Do Not Support Rebuild” on page 332 and “Alternatives to Rebuild” on page 334. For non-IBM software that uses the coupling facility, see the documentation for the function or application that owns the structure.

4. For each remaining structure, issue the following command to check the status of the structure: `D XCF,STR,STRNAME=stname`

If the number of connections is 0, a no-connector condition exists for the structure. If the number of connections is not 0 and all connectors are failed-persistent, a failed-persistent state exists for the structure. To handle these conditions, see “Deleting Connectors and Structures with SETXCF FORCE” on page 338.

For any structure that has active connectors, follow the procedure for deleting the structure as indicated by the owning function or application.

Coupling Facility Guidelines

Note: Some structures remain active throughout the shutdown process (for example, IEFAUTOS and VTAM structures).

5. Configure all CHPIDs offline to the coupling facility that you are removing. If you are deactivating an ICMF partition, DO NOT perform this step. See “Considerations for Removing an ICMF Partition”.
 - To take the CHPIDs identified in step 1 offline, issue the following command on each system connected to the coupling facility that you are removing:
`CONFIG CHP(xx,yy),OFFLINE,UNCOND`
 - To ensure that all CHPIDs were taken offline, issue the D CF command on all systems connected to the coupling facility to be removed. Identify the coupling facility by the NODE DESCRIPTOR, PARTITION, and CPCID descriptions from the information you recorded in step 1. Verify that all CHPIDs were taken offline.

If the CHPIDs do not come offline using the UNCOND keyword, you might need to use the FORCE option on the `CONFIG CHP(xx,yy),OFFLINE` command.

When you enter the CONFIG command with FORCE on a system to take the last path to the coupling facility offline, any active connector to a structure loses connectivity to the structure.

Note: For IEFAUTOS, VTAM, and other structures that remain active during the shutdown process, you must use the FORCE option of CONFIG to remove the remaining CHPID offline.

6. Power off the coupling facility. When the coupling facility is deactivated, any remaining structure data is lost.

To restore the coupling facility, see “Restoring the Coupling Facility to the Sysplex” on page 338.

Considerations for Removing an ICMF Partition

Consider the following:

- You plan to deactivate an ICMF partition and an alternate coupling facility exists in the configuration.
- You plan to deactivate the only ICMF partition on the processor, or shut down the processor that contains one or more ICMF partitions and no other coupling facility exists in the configuration.

Deactivating an ICMF Partition

If you are deactivating an ICMF partition on a processor and you plan to move structures to another ICMF partition on the processor, do the following:

1. Follow steps 1, 3, and 4 in “Removing a Coupling Facility When Multiple Coupling Facilities Exist” on page 327.
2. When you have moved or deleted structures from the ICMF partition to be deactivated, DO NOT configure the CHPIDs offline.
3. Deactivate the ICMF partition.

Deactivating the only ICMF Partition or Shutting down the Processor

If you are using only one ICMF partition on the processor and you want to deactivate the partition, or you plan to shut down the processor on which one or more ICMF partitions exist, use the following procedure. If you have more than one ICMF partition, follow these steps to deactivate each partition:

1. Follow steps 2 through 4 in “Removing the Only Coupling Facility in the Sysplex” on page 329.
2. Do NOT configure the CHPIDs offline.
3. If you plan to power down the processor, shut down all of the MVS images that exist on the processor. Then power off the processor for maintenance.
If you do not plan to power down the processor, deactivate the ICMF partition.

For information about ICMF, see *PR/SM Planning Guide*.

Replacing or Moving Coupling Facilities in an Active Sysplex

Coupling facility reconfiguration includes:

- Replacing one coupling facility with another coupling facility
- Moving a coupling facility from one CPC to another CPC
- Moving a coupling facility from one PR/SM partition to another PR/SM partition on the same CPC.

You can reconfigure a coupling facility in your sysplex in one of two ways:

- Follow the procedures for “Removing a Coupling Facility When Multiple Coupling Facilities Exist” on page 327, including deleting its definition from the channel subsystem for the affected MVS image CPCs. Then follow the procedures for “Adding a Coupling Facility to an Active Sysplex (MVS Image Side)” on page 325 and “Adding a Coupling Facility to an Active Sysplex (Coupling Facility Image Side)” on page 326 to add the new coupling facility to your sysplex. This will ensure configuration-sensitive products are synchronized with the reconfiguration.
- Alternatively, based on the needs of the installation, the reconfiguration can be accomplished without affecting the channel subsystem definition. Existing channel subsystem constructs can be “reused” as long as CHPID information for the coupling facility has not changed as defined for the MVS images. Similarly, if the coupling facility is configured to the Hardware Management Console for remote IOCDS management, the SNA address used for the new coupling facility must be the same as the address used for the old coupling facility.

A configuration-sensitive products, such as HCD, might display and/or report downlevel configuration data. With HCD, for example, reports comparing the IODF to the actual configuration will show disparity.

In using this alternative, follow the procedures as defined in deleting and adding coupling facilities from an active sysplex ignoring those steps dealing with IODF and IOCDS activities.

For additional information see:

OS/390 MVS System Commands
PR/SM Planning Guide
OS/390 HCD User's Guide
IOCP User's Guide

Actions to Remove a Structure from a Coupling Facility

You have several choices when removing a structure from a coupling facility, depending on the state of the structure and its use by an application. You must resolve any failed-persistent connectors or any no-connector conditions with a structure. To determine failed-persistent connectors for any structure, issue the following command `D XCF,STR,STRNAME=ALL,STATUS=(FPCONN)`

Coupling Facility Guidelines

To determine no-connector conditions for any structure, issue the following command `D XCF,STR,STRNAME=ALL,STATUS=(NOCONN)`

Generally, the function or application needs to be reinitialized to resolve failed-persistent or no-connector conditions. For information about resolving failed-persistent or no-connector conditions, see the documentation for the application or function that owns the structure.

The following procedure is a brief overview for removing structures from a coupling facility with pointers to detailed information:

1. Delete any structure that cannot be rebuilt. For a list of the IBM structures that cannot be rebuilt and procedures for deleting each of them, see “Deleting Structures that Do Not Support Rebuild”.
2. Rebuild structures that support the rebuild process. Rebuilding a structure allows the application to remain active during the reconfiguration. Rebuilding a structure is the preferred way to move a structure from a coupling facility. If you are unsure whether a structure supports rebuild, you can attempt a rebuild of the structure. If rebuild is not supported, the system will indicate that the rebuild operation cannot occur, and you must use an alternative procedure for moving the structure. For a procedure on rebuilding structures, see “Rebuilding Structures” on page 334.
3. If for some reason you cannot rebuild a structure (for example, you are shutting down the only coupling facility in the configuration), follow the procedures in “Alternatives to Rebuild” on page 334.
4. Force the deletion of the structure. Forcing a structure removes the structure from the coupling facility. See “Deleting Connectors and Structures with SETXCF FORCE” on page 338.

Deleting Structures that Do Not Support Rebuild

Some structures do not support the rebuild process. The following procedures describe how to delete IBM software structures that do not support rebuild. Non-IBM software structures should provide similar information in their documentation.

List Structure for the JES2 Checkpoint Data Set: With OS/390 Release 8, JES2 supports system-managed rebuild for its JES2 checkpoint data set structure. If you are not at that release level or higher, use the following procedure; otherwise, see “Rebuilding Structures” on page 334.

1. Use the JES2 reconfiguration dialog to move the checkpoint data set to another coupling facility or to DASD.
To use the dialog, ensure that you have defined a NEWCKPTx in the JES2 CKPTDEF initialization statement in one of the following ways:
 - JES2 initialization parameters
 - Dynamically through the \$T CKPTDEF command
2. Issue the following JES2 command to switch to the alternate checkpoint data set: `$T CKPTDEF,RECONFIG=YES`
Follow the reconfiguration dialog prompts.
3. When you have finished, issue the following JES2 display command to verify that the structure in the coupling facility to be removed is not still in use by JES2: `$D CKPTDEF`
4. To force deletion of the structure from the coupling facility, issue the following MVS command: `SETXCF FORCE,STR,STRNAME=strname`
5. If necessary, change the JES2 initialization parameters to point to the new checkpoint data set for the next JES2 warm start.

Coupling Facility Guidelines

Restoring the Structure: To move the JES2 checkpoint back into the coupling facility, use the JES2 reconfiguration dialog and issue the following command: \$T CKPTDEF,RECONFIG=YES

DB2 Cache Structure for Group Buffer Pools: If the cache structure for the DB2 group buffer pools are in the coupling facility to be removed, delete the structure as follows:

1. Stop all DB2 members in the data sharing group.
2. Make the necessary maintenance changes to the coupling facility.

If you plan to move the structures to an alternate coupling facility, do the following:

1. Stop all DB2 members in the data sharing group.
2. With the new CFRM policy in effect, start DB2 so that the structures for the group buffer pools are allocated in the alternate coupling facility indicated in the policy.
3. Make the necessary maintenance changes to the coupling facility that you have removed.

Restoring the Structure:

1. To move the DB2 structures back to the coupling facility you have removed, restore the coupling facility and the original CFRM policy.
2. If DB2 has been stopped, restart DB2, or if DB2 is already active, stop and restart DB2 so that the structures are allocated in the coupling facility that you have restored as indicated by the original CFRM policy.

If it is not possible to stop all the DB2 members, see *DB2 Data Sharing: Planning and Administration* for information.

CICS Temporary Storage Queue Pool Structures:

1. To delete a CICS structure for a temporary storage (TS) queue pool, stop all TS queue pool servers for the structure. If there are no connectors to the structure, use the MVS STOP command to stop the TS server.

If there are connectors, issue the MVS MODIFY command to disconnect active CICS connections: F *server*,CANCEL where *server* is the name of the TS queue pool server.

2. When all servers are disconnected, if required, unload the structure contents to a sequential data set by running the queue pool server program DFHXQMN with the following option:

```
FUNCTION=UNLOAD
```

3. Issue the following MVS command to delete the structure: SETXCF FORCE,STR,STRNAME=DFHXQLS_*poolname*

Restoring the Structure: To reload the TS pool queue structure from the sequential data set, run the queue pool server program DFHXQMN with the following option:

```
FUNCTION=RELOAD
```

If you do not reload the queue pool, CICS allocates a new empty queue pool structure when the queue pool server is started.

For a description of the program DFHXQMN and other information about the queue pool, see *CICS System Definition Guide*.

Cache Structure for Enhanced Catalog Sharing:

Coupling Facility Guidelines

1. To disconnect all users from the STSIGGCAS_ECS structure, issue the command `F CATALOG,ECSHR(DISCONNECT)`. The structure will be deleted when there are no connections to it.
2. To move the structure to an alternate coupling facility, activate a new CFRM policy, identifying the alternate coupling facility. Once the policy is active, CAS will attempt to connect to the structure automatically.

Restoring the Structure: To move the SYSIGGCAS_ECS structure back to the original coupling facility, restore the coupling facility and the original CFRM policy. Follow the procedure described above to move the structure.

For a description of the Enhanced Catalog Sharing structure and other information about its function, see *OS/390 DFSMS: Managing Catalogs*.

Rebuilding Structures

Issue the following command to initiate the rebuild: `SETXCF START,REBUILD,STRNAME=strname,LOCATION=OTHER`

Rebuilding structures in another coupling facility might take several minutes to complete. A message in response to the `SETXCF START,REBUILD,...` command, indicates if a structure does not support rebuild.

Additional Processing for Rebuilding Structures: Structures that can be rebuilt might require additional processing to move them to an alternate coupling facility. Rebuild scenarios that require additional processing are:

- The structure has no connectors; a structure cannot be rebuilt without an active connector.
- The structure has failed-persistent connectors; the existence of failed-persistent connectors to a structure can prevent its rebuilding or deletion.

To determine if any structure that you want to rebuild does not have an active connector, issue the following command: `D XCF,STR,STRNAME=ALL,STATUS=(NOCONN)`

For structures with no connectors, initialize the application associated with the structure to obtain a connector to the structure. Then, attempt a structure rebuild by issuing: `SETXCF START,REBUILD,STRNAME=strname,LOCATION=OTHER`

To determine if any structure that you want to rebuild has at least one FAILED-PERSISTENT connector, issue the following command: `D XCF,STR,STRNAME=ALL,STATUS=(FPCONN)`

Recovery actions should be handled by the owning application. Initialize the application associated with the structure to recover the connection and then attempt a structure rebuild.

For information on handling rebuild problems, see “Structure Rebuild Recovery” on page 338.

Alternatives to Rebuild

In some cases where a structure can be rebuilt, you might decide not to rebuild the structure but delete it from the coupling facility. For example, although IBM does not recommend relying on a single coupling facility in the sysplex, you might have only one coupling facility and cannot move structures to an alternate. For IBM structures that you do not want to rebuild, consider the following topics.

Coupling Facility Guidelines

Deleting the RACF Database Cache Structures: Follow this procedure if any of the primary or back-up RACF database cache structures exist in the coupling facility to be removed:

1. Issue the following RACF command to remove RACF from data sharing mode:
RVARY NODATASHARE

If the installation has defined RACF as a subsystem in IEFSSNxx of SYS1.PARMLIB, you can issue the command from an MVS operator console, but you need to use the subsystem recognition character that the installation has defined through the command prefix facility. Otherwise, you must issue the command from an authorized TSO/E user ID. For IEFSSNxx, see *OS/390 MVS Initialization and Tuning Reference*.

2. When RACF is no longer in data sharing mode, the RACF database cache structures are deleted.

When you have performed coupling facility maintenance and installed the coupling facility online, issue the following command to restore RACF data sharing mode and reallocate the structures: RVARY DATASHARE

Deleting IMS OSAM and VSAM Cache Structures: Shut down each subsystem that uses the coupling facility. When the last connector is stopped, the system deletes the structure.

Deleting the IRLM Lock Structure:

1. To determine if any IMS or DB2 subsystem in the data sharing group retains a lock, issue the following IMS command: F ir1mproc,STATUS,ALLD where **ir1mproc** is the IRLM associated with IMS or DB2.
2. If an IMS or DB2 subsystem retains a lock, restart that IMS or DB2 subsystem to clean up the locks.
3. Initiate a normal shutdown of all IMS or DB2 subsystems identified with IRLM to cause IRLM to disconnect from the data sharing group.
4. For IMS DB applications, stop IRLM.
5. Issue the following command to force the deletion of the FAILED-PERSISTENT IRLM connector: SETXCF FORCE,CON,CONNAME=ALL,STRNAME=strname
6. Issue the following command to force the deletion of the IRLM lock structure in the coupling facility. SETXCF FORCE,STR,STRNAME=strname
See “Deleting Connectors and Structures with SETXCF FORCE” on page 338.

Deleting the DB2 List (SCA) Structure: When you cannot rebuild the DB2 list (SCA) structure, do the following:

1. Use the IRLM status command to determine if any DB2 member that is in the data-sharing group has retained locks. If a member has retained locks, reinitialize the associated DB2 member to recover the retained locks.
2. Use the following command to provide a normal shutdown of all DB2 members:
STOP DB2 MODE(QUIESCE).

The DB2 cache structures for the group buffer pools are also deleted automatically.

3. To delete the DB2 list (SCA) structure, issue the SETXCF FORCE command. (See “Deleting Connectors and Structures with SETXCF FORCE” on page 338.)

For detailed information, see *DB2 Administration Guide* and *DB2 Data Sharing: Planning and Administration*.

Coupling Facility Guidelines

Deleting the XCF Signaling Structure:

1. Ensure that you have redundant XCF signaling paths and capacity to ensure that system communications in the sysplex are not disrupted when you delete the signaling structure.
2. To stop XCF signaling through the structure on the coupling facility to be removed, issue the following commands on each system that is connected to the coupling facility:

```
SETXCF STOP,PATHOUT,STRNAME=strname  
SETXCF STOP,PATHIN,STRNAME=strname
```

When you have stopped all paths (PATHOUT and PATHIN), XCF deletes the XCF signaling structure.

3. When you have completed coupling facility maintenance, restored connectivity to the coupling facility, and reactivated the original CFRM policy, issue the following commands on each system connected to the coupling facility to reestablish XCF communication: `SETXCF START,PATHIN,STRNAME=strname`
`SETXCF START,PATHOUT,STRNAME=strname`

Deleting the System Logger Structure: When you cannot rebuild the system logger structure, do the following:

1. If any subsystem is using the system logger, provide a normal shutdown of the subsystem.
2. Use the VARY Command with the HARDCPY option to deactivate OPERLOG and, if necessary, change the hardcopy medium.
3. Use the SETLOGRC command to stop LOGSTREAM recording of LOGREC entries.
4. Exit from all SDSF browse logger functions.

Once all users of system logger have disconnected, the system deletes the structure in the coupling facility.

Deleting the VSAM RLS Lock and Cache Structures: When you cannot rebuild the VSAM lock and cache structures, do the following:

1. Before shutting down VSAM RLS processing, follow your installation procedures for shutting down applications that use open VSAM data sets in RLS mode. For CICS, use one of the following ways to quiesce the data sets:
 - Use CEMT to shut down all CICS subsystems
 - Issue EXEC CICS, SET DSNAME(name) QUIESCED

You can shut down some CICS subsystems by issuing CEMT P SHUT.

2. Issue the following command to shut down VSAM RLS processing on all systems: `V SMS,SMSVSAM,SHUTDOWN`
3. Wait for all VSAM RLS address spaces (SMSVSAM) to end.
4. When all the address spaces have ended, all the cache structures are deleted automatically. Lock structure IGWLOCK00 still has persistent connections that you must delete.

To delete persistent connections to IGWLOCK00, see “Deleting Connectors and Structures with SETXCF FORCE” on page 338.

5. When you have completed coupling facility maintenance, restored connectivity to the coupling facility, and reactivated the original CFRM policy, issue the following command on each system:

```
V SMS,SMSVSAM,ACTIVE
```

Coupling Facility Guidelines

6. VSAM RLS processing is again available. To make available for record-level sharing any data set with unresolved locks (lost-locks condition), perform all necessary forward and backward recovery for the data sets.

Note: When a lost-locks condition exists for CICS, there is no need to perform forward recovery. CICS automatically performs backout recovery when it is restarted.

Deleting IMS Shared-Queue or EMH Structures: When you cannot rebuild the IMS structures, do the following:

1. Initiate normal shutdown of all IMS subsystems sharing the queues on the coupling facility.
2. Issue the following command to force the deletion of FAILED-PERSISTENT connections: `SETXCF FORCE,CON,CONNAME=ALL,STRNAME=strname` where *strname* is the name of the structure.
3. Issue the following command to force the deletion of the IMS structure (either shared queue or EMH): `SETXCF FORCE,STR,STRNAME=strname` where *strname* is the name of the structure.

Deleting the Global Resource Serialization Lock Structure: When the global resource serialization lock structure cannot be rebuilt in an alternate coupling facility, follow this procedure:

1. Shutdown all production work in the sysplex.
2. Shutdown all applications that use the coupling facility by following the procedures for each application.
3. Shutdown the coupling facility.

Global resource serialization places all systems into a wait state (0A3/D0) and issues the following message:

```
ISG309W GRS PROCESSING TERMINATED -  
UNRECOVERABLE FAILURE DURING LOCK STRUCTURE REBUILD PROCESSING.
```

4. Re-IPL systems when a coupling facility is available for use by global resource serialization.

Deleting the VTAM Structure for Generic Resources: If any network resource supports LU 6.2 level syncpoint protocol, you should rebuild the VTAM structure into an alternate coupling facility. If are not able to rebuild into an alternate, you can delete the VTAM structure, but you must ensure that all VTAM nodes remain active. The VTAM nodes contain a local copy of the generic resource information for the sysplex. When the coupling facility is restored, the VTAM structure can be rebuilt from the copy. If any VTAM node fails after you delete the VTAM structure, you can lose structure data. Because of the loss of critical VTAM structure data, results can be unpredictable when you restore the coupling facility.

The VTAM structure remains active until you shut down the coupling facility, at which point it is deleted. For detailed information, see *OS/390 IBM Communications Server: SNA Network Implementation Guide*.

Deleting the IEFAUTOS Structure: For IEFAUTOS used with automatic tape switching, there are no special procedures to follow. The IEFAUTOS structure remains active until you shut down the coupling facility. When you remove the coupling facility, the structure is deleted.

For IEFAUTOS, you might be able to operate without the structure while you perform coupling facility maintenance. See *OS/390 MVS System Commands*.

Coupling Facility Guidelines

Deleting Connectors and Structures with SETXCF FORCE

The FORCE option of the SETXCF command allows you to delete persistent structures or failed-persistent connectors in a coupling facility. However, use this option with caution. The FORCE operation might cause a loss of structure data, so be sure you understand how the application is using the structure. Also, you can only force structures with no-connectors or structures with failed-persistent connectors. You cannot force active connectors or structures with active connectors.

1. If the structure has no connectors, issue the following command to delete the structure: `SETXCF FORCE,STR,STRNAME=strname`
2. If the structure has **only** failed-persistent connectors, issue the following command to delete the connector: `SETXCF FORCE,CON,STRNAME=strname,CONNAME=ALL`

Note that you must force the failed-persistent connectors **before** you force the structure. To delete the structure if the structure is persistent, issue `SETXCF FORCE,STR,STRNAME=strname`

For information about the FORCE option on MVS commands, see *OS/390 MVS System Commands* .

Restoring the Coupling Facility to the Sysplex

The following procedure describes how to restore the coupling facility and move structures that support rebuild to the coupling facility that you have restored.

1. Restore the original CFRM policy.
2. For structures that have been rebuilt to an alternate coupling facility, issue the following command to rebuild the structures to the original coupling facility. `SETXCF START,REBUILD,STRNAME=strname`
Depending on the CFRM policy structure preference list, you might need the `LOC=OTHER` parameter.
3. Restart any other subsystems that have been stopped during the coupling facility shutdown procedure.

Note: When the coupling facility and CFRM policy are restored, some functions might reconnect to the coupling facility automatically depending on the method used to remove the structures.

For IBM structures that do not support rebuild, see the procedure for the structure in “Deleting Structures that Do Not Support Rebuild” on page 332.

Structure Rebuild Recovery

The following topic describes a recovery scenario for a structure rebuild that does not complete (a structure rebuild hang).

A structure rebuild might not complete for many reasons. The application or connector is the most likely source of the rebuild hang; however, the hang might be the result of a problem with either of the MVS components XES or XCF. For information about sysplex recovery, see *OS/390 MVS Diagnosis: Reference*.

Steps for Handling a Structure Rebuild Hang

The following are general steps to follow when a rebuild hang occurs:

1. Look for and respond to any messages that indicate problems with rebuild.
2. To determine the state of the rebuild process, issue the following command: `D XCF,STR,STRNAME=strname`

Coupling Facility Guidelines

The rebuild hang might be the result of a problem with one or more connectors associated with the structure. For connectors that have not completed the rebuild process, check the status of the system, subsystem, or address space where the connector resides and try to correct the problem.

3. Collect documentation as follows:
 - Take an SDUMP of the connectors that are hanging, the XCF address space, and all dataspaces owned by each address space to be dumped. (For additional SDATA parameters, information about the couple data set dump utility, IPCS and trace parameters. See *OS/390 MVS Diagnosis: Reference*.)
 - Dump the CFRM and sysplex couple data sets.
4. Try to recover from the failure.
 - To stop the rebuild process, issue the following command: SETXCF STOP,REBUILD,STRNAME=*strname* where **strname** is the name of the structure.
 - To determine if the SETXCF STOP,REBUILD command is successful, issue the following command: D XCF,STR,STRNAME=*strname* where **strname** is the name of the structure.

If the SETXCF STOP,REBUILD command does not complete, collect documentation as indicated in step 3.

If the SETXCF STOP,REBUILD command is successful, attempt to rebuild the structure again.

If the structure has failed-persistent connectors, you might need to clean up the connectors before you can rebuild the structure.

If a connector problem exists and cannot be resolved, consider shutting down the subsystem associated with the connector, cancelling the connector address space, or varying the system that contains the connector out of the sysplex.

5. When you have resolved the problem with the connector, attempt to rebuild the structure again.
6. Using the information collected in step 3, diagnose and correct the cause of the hang.

Example of a Rebuild Hang

Consider a sysplex with three systems (SYSTEM1, SYSTEM2, SYSTEM3). The SETXCF START,REBUILD command is issued for the XCF structure IXC1, but the rebuild has not completed. The D XCF,STR,STRNAME=IXC1 command is entered and the response to the command indicates that a connector to IXC1 on SYSTEM2 has not completed the rebuild process.

Figure 74 on page 340 shows the output to the commands. Information about the connector that has not responded to the rebuild request is indicated by an asterisk next to the connector name (SIGPATH_02000002 in this example). Based on the information, the system programmer is able to collect the appropriate documentation and determine the recovery action.

Coupling Facility Guidelines

```
===> SETXCF START,REBUILD,STRNAME=IXC1

IXC367I THE SETXCF START REBUILD REQUEST FOR STRUCTURE 924
IXC1 WAS ACCEPTED.
IXC467I REBUILDING PATH STRUCTURE IXC1 925
      RSN: PARTICIPANT
      DIAG073: 08840001 0000000A 00000000 04000000 00000000

      /
      /
      /
      /
      /

===> D XCF,STR,STRNAME=IXC1

IXC360I 09.53.28 DISPLAY XCF 942
STRNAME: IXC1
===> STATUS: REASON SPECIFIED WITH REBUILD START:
      OPERATOR INITIATED
      REBUILD PHASE: QUIESCE
      POLICY SIZE : 10000 K
      POLICY INITSIZE: N/A
      REBUILD PERCENT: N/A
      PREFERENCE LIST: TESTCF
      EXCLUSION LIST IS EMPTY

ACTIVE STRUCTURE
-----
ALLOCATION TIME: 08/30/95 09:50:49
CFNAME : TESTCF
COUPLING FACILITY: CF01...
      PARTITION: 0 CPCID: 00
ACTUAL SIZE : 10240 K
STORAGE INCREMENT SIZE: 256 K
VERSION : AB991335 42205400
DISPOSITION : DELETE
ACCESS TIME : 0
MAX CONNECTIONS: 32
# CONNECTIONS : 3
NEXT USYNC EVENT : 00000001
NEXT USYNC STATE : 00000000 00000000 00000000 00000000
                  00000000 00000000 00000000 00000000

* ASTERISK DENOTES CONNECTOR WITH OUTSTANDING REBUILD RESPONSE
CONNECTION NAME ID VERSION SYSNAME JOBNAME ASID STATE
-----
SIGPATH_01000001 01 00010001 SYSTEM1 XCFAS 0006 ACTIVE
===> *SIGPATH_02000002 02 00020001 SYSTEM2 XCFAS 0006 ACTIVE
SIGPATH_03000003 03 00030001 SYSTEM3 XCFAS 0006 ACTIVE
```

Figure 74. Message Output from a Rebuild Hang

Guidelines for Unplanned Outages

This section provides guidelines for recovering from coupling facility failures. Included are scenarios describing situations when another coupling facility is available and when another coupling facility is not available.

Guidelines for Recovering from Coupling Facility Failures

The following information assumes that a complete failure of a coupling facility occurred (for example, a power failure, system reset, power-on reset, LPAR reset or deactivate). For information about how to handle structure failure and connectivity failure scenarios, see *OS/390 Parallel Sysplex Test Report*. Note that coupling facility failures are observed by the MVS systems as a loss of coupling facility connectivity. There is no indication that distinguishes a true loss of connectivity (in which the coupling facility stays up but the link(s) to the coupling facility fail) from other types of coupling facility failure, such as a power failure or system reset.

When a coupling facility without power or battery backup fails, all of the structures and the data contained in the structures are lost. However, because these failures all appear to MVS as loss of connectivity to the coupling facility (in which case the structures and data are not lost), MVS makes the conservative assumption that the “loss of connectivity” is in fact just a loss of connectivity to the coupling facility and thus, that the structures have not been lost.

In these coupling facility failure scenarios, it is expected that the applications that own structures on the failed coupling facility will recover either by rebuilding those structures in an alternate coupling facility or through some other means. There are scenarios where application recovery of structures might not be successful (for example, your sysplex might not have an alternate coupling facility in which to rebuild structures, or a concurrent application failure prevents a successful structure rebuild).

The CASE 1 scenario describes procedures to follow when an alternate coupling facility is available to replace the failed coupling facility. The CASE 2 scenario describes the situation in which an alternate coupling facility is not available in the sysplex.

CASE 1 - An Alternate Coupling Facility Is Available

When an alternate coupling facility is available, you might be able to rebuild the structures from the failed coupling facility to an alternate coupling facility. The possible results of the rebuild are:

- All structures successfully rebuild in the alternate coupling facility. See “All Structures Rebuild”.
- Some structures are not rebuilt in the alternate coupling facility. See “Some Structures Do Not Rebuild”.
- Structures in the failed coupling facility become persistent or structure connectors become failed-persistent. See “Structure or Connector Persistence Occurs” on page 342.
- Structure rebuild fails to complete. See “Structure Rebuild Fails” on page 343.
- Structures cannot be rebuilt because there are storage constraints in the alternate coupling facility. See “Coupling Facility Storage Constraints Exist” on page 343.

All Structures Rebuild: All structures successfully rebuild in the alternate coupling facility.

Once the failed coupling facility is restored, use the SETXCF START,REBUILD command to move appropriate structures back into the restored coupling facility.

Some Structures Do Not Rebuild: Some applications that use coupling facility structures do not support rebuild. Structures such as the DB2 cache structures (group buffer pools) might require manual intervention to reallocate the structures in

Coupling Facility Guidelines

the alternate coupling facility. Follow application-specific recovery procedures for applications that do not support rebuild. Likewise, when the failed coupling facility is restored, the manual procedure will be necessary to move these structures back into the restored coupling facility. For additional information see *OS/390 Parallel Sysplex Test Report*.

Structure or Connector Persistence Occurs: Structures on the failed coupling facility become persistent or structure connectors become failed-persistent. This might happen for a number of different reasons, such as:

- Structure rebuild failed.
- An alternate coupling facility did not exist.
- Structure rebuild is not supported by the owning application.
- An application failure occurred.

Restarting an Application: If the application that owns a persistent structure or connector is restarted while the coupling facility remains unavailable, its attempt to connect to the structure will not succeed. MVS maintains a recovery bind, represented by the persistent structure or failed-persistent connection, to the instance of the structure that was in use at the time of the failure. Until and unless connectivity to the coupling facility containing that structure is restored, MVS does not know whether the coupling facility has truly failed (with the loss of structure data that implies), or MVS has simply lost connectivity to the coupling facility (with the implication that access to the data will be restored when connectivity to the coupling facility is restored). MVS will maintain this recovery bind to the old instance of the structure until one of the following events occurs:

- Connectivity to the coupling facility is restored, and the structure is found to still exist (a loss and regain of connectivity scenario), or
- Connectivity to the coupling facility is restored, and the structure is found to no longer exist (a true coupling facility failure scenario), or
- You use the SETXCF FORCE command to delete the recovery bind by deleting the failed-persistent connector or the persistent structure via the SETXCF FORCE command.

Breaking the Recovery Bind: Thus, if you want to restart the application and have it allocate another instance of the structure in an alternate coupling facility, you first must break the recovery bind to the old instance of the structure by deleting the persistent connectors or the structure before structure allocation will succeed.

See “Reviewing the Concept of Persistence” on page 321 before proceeding with structure/connector deletion.

Handling Failed Coupling Facility Structures: To determine which structures remain allocated on the failed coupling facility, issue the following command: `D XCF,CF,CFNAME=failed_cfname`

To delete the allocated structures do the following for each structure in the failed coupling facility:

Display structure information: `D XCF,STR,STRNM=strname`

The structure might have failed-persistent connectors, or the structure itself might be persistent, or both.

- Deleting Failed-Persistent Connectors

If the structure has failed-persistent connectors, issue SETXCF FORCE to delete all failed-persistent connectors: `SETXCF FORCE,CON,CONNM=ALL,STRNM=strname`

Because the MVS system does not have connectivity to the failed coupling facility, you will receive message

```
IXC354I THE SETXCF FORCE REQUEST FOR CONNECTION
connname IN STRUCTURE strname WAS ACCEPTED:
REQUEST WILL BE PROCESSED ASYNCHRONOUSLY
```

Display the structure again: D XCF,STR,STRNM=*strname*

The number of connectors to the structure should now be zero.

- **Deleting Persistent Structures**

If the structure is persistent (DISPOSITION: KEEP in IXC360I) and the number of connectors to the structure in message IXC360I is greater than zero, then delete the structure as described below.

If ACTIVE connectors exist (see connection state in IXC360I), invoke recovery procedures for the connector, or CANCEL the connector's address space to make the connector disconnect from the structure. Then, if the connector enters a failed-persistent state, force the connector using the SETXCF FORCE,CON.

Then, to delete the structure, issue: SETXCF FORCE,STR,STRNM=*strname*

Because the MVS system does not have connectivity to the failed coupling facility, you will receive message

```
IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE
strname WAS ACCEPTED:
REQUEST WILL BE PROCESSED ASYNCHRONOUSLY
```

Once all failed-persistent connectors are forced or the structure is forced, the structure should enter an "in transition" state. If you display the structure again, message IXC360I should contain the following statements:

```
STRUCTURE IN TRANSITION
-----
REASON IN TRANSITION: CONNECT OR DISCONNECT IN PROGRESS
```

When the structure enters an "in transition" state, it means that the structure is "targeted for deletion" by CFRM. At this point, a new instance of the structure can be allocated in another coupling facility. Depending on the application that owns the structure, you might need to restart the application for the structure to become allocated in the alternate coupling facility.

Note: When the structure is allocated in an alternate coupling facility, the DISPLAY XCF,STR,STRNAME=*strname* command will produce information about both the new instance of the structure in the alternate coupling facility and the "structure in transition" which is pending deletion from the failed coupling facility. When the failed coupling facility is eventually restored, XCF will resolve the "structure in transition" condition and this information will no longer be displayed.

Structure Rebuild Fails: If structure rebuild fails to complete, see "Rebuilding the Structure" on page 346 for information on how to handle a rebuild hang.

Coupling Facility Storage Constraints Exist: If the alternate coupling facility does not have enough capacity to handle the workload from the failed coupling facility, consider deleting less important structures from the alternate coupling facility to allow the coupling facility resources to drive your mission critical structures. You can delete structures such as the XCF, JES2, and RACF structures by:

- Stopping XCF signaling through the alternate coupling facility, provided there are sufficient alternate paths (CTCs) available for XCF signaling

Coupling Facility Guidelines

- Moving JES2 checkpoint to DASD
- Running RACF in non data-sharing mode.

CASE 2 - An Alternate Coupling Facility Is Not Available

In this case, if there are structures critical to the operation of your sysplex, consider activating a coupling facility in a backup logical partition. Follow the procedures for “Adding a Coupling Facility to an Active Sysplex (MVS Image Side)” on page 325 or “Adding a Coupling Facility to an Active Sysplex (Coupling Facility Image Side)” on page 326. Then proceed as in CASE 1.

Handling Failing or Disconnecting Connector States

This information provides guidelines for recovering from structure connectors that are hung in a FAILING or DISCONNECTING state. The term “connector” refers to a job or subsystem that has established a linkage (or connection) to a coupling facility structure. The connector may be:

- A job that is active in the system and therefore may have an ACTIVE connection to a structure, or
- A job that no longer has an active connection to a structure. (For example, a job that does not have an active connection to a structure but for which residual information about the connection persists after the connector disconnects from the structure, is a FAILED-PERSISTENT connector.) For additional information about connection states, see *OS/390 MVS Programming: Sysplex Services Guide*.

Unlike a connector state of ACTIVE or FAILED-PERSISTENT, a FAILING or DISCONNECTING state is a transitory state. When a connector disconnects from a structure, the disconnect process may place the connector into a DISCONNECTING or FAILING state. This is normal. However, a connector should not remain in the FAILING or DISCONNECTING state indefinitely.

Connectors hung in a FAILING or DISCONNECTING state will likely be discovered when the associated connector job tries to reconnect to the structure and the reconnect (IXLCONN macro) request is rejected.

Determining the Connectors’ States

To determine the state of all connectors for a given structure, issue the following command: D XCF,STR,STRNAME=strname

Message IXC360I will provide information about the structure and its connectors. Here is an example of displaying information about an IRLM lock structure: D XCF,STR,STRNAME=IRLMLOCK1

```
IXC360I 12.23.28 DISPLAY XCF xxx
STRNAME: IRLMLOCK1
STATUS:  ALLOCATED
      |
      |
      | / (only relevant portions or message IXC360I
      | / are shown here)
      |
      |
CONNECTION NAME  ID  VERSION  SYSNAME  JOBNAME  ASID  STATE
-----
IRLMGRP1$IRLA001 03  0003002F  SYS1     IRLMA    001D  ACTIVE
IRLMGRP1$IRLB002 02  0002008C  SYS2     IRLMB    0058  ACTIVE
IRLMGRP1$IRLC003 01  00010099  SYS3     IRLMC    003F  FAILING
IRLMGRP1$IRLC004 04  0004001A  SYS3     IRLMC    0180  DISCONNECTING
```

In the above example, connectors 03 and 02 are ACTIVE (normal), and connectors 01 and 04 are in transition.

Role of an Active Connector

It should be noted that the connector that is hung in the FAILING or DISCONNECTING state is not the cause of the problem. The problem lies with one of the ACTIVE connectors. When a connector disconnects from a structure, all the other ACTIVE connectors to the structure must perform recovery for the FAILING or DISCONNECTING connector. If one or more of the ACTIVE connectors fails to perform and confirm recovery completion with MVS, the FAILING or DISCONNECTING state cannot be exited. Hence, the problem is likely due to one of the ACTIVE connectors not responding to the FAIL or DISCONNECT recovery event.

Steps for Diagnosis and Recovery

The following procedure allows you to identify the ACTIVE connector that is causing the problem, determine what recovery actions should be taken, collect documentation, and recover from the problem. You may wish to modify this procedure to suit your specific needs.

Identify the Connector or Systems that Require Recovery Actions: By the time you find out that a connector is in a FAILING or DISCONNECTING state, it is likely that it had been in that state for awhile. However, if the connector recently went into a FAILING or DISCONNECTING state (for example, you just recycled a subsystem), then you should ensure that enough time has elapsed to allow the FAIL and DISCONNECT events to be processed by the active connectors. Normally, this should not take more than a few minutes.

- Determine if the FAILING or DISCONNECTING connector is owed a response.

Issue the following command: D

```
XCF,STR,STRNAME=strname,CONNAME=failing_conname or D
```

```
XCF,STR,STRNAME=strname,CONNAME=disconnecting_conname
```

Message IXC360I will provide detailed information for the connector specified in the command. If the connector is owed a response (such as a response to a FAIL or DISCONNECT event), the detailed information might include the following:

```
CONNECTION ID(s) OWING A RESPONSE FOR THIS CONNECTOR:
```

```
id id id... id
```

The IDs owing a response to the FAILING or DISCONNECTING connector identify the connector (job) that is not responding. Those jobs owing a response are candidates for recovery actions. If CONNECTION ID(s) OWING A RESPONSE was included in the IXC360I output, check the health of the job or system to see if there is anything obvious that is wrong with the job or system on which the job resides. If so, go to “Determine Recovery Actions” on page 346. Otherwise, continue to find information about the structure itself.

- Display all relevant information about the structure that might identify connectors or systems requiring recovery actions.

```
D XCF,STR,STRNAME=strname
```

For each ACTIVE connector (as displayed in IXC360I), check the health of the connector (job), and check the health of the system on which the ACTIVE connector resides.

- Issue commands that are processed by the ACTIVE connector. For example, if the connector is an IRLM job, issue: F IRLMx,STATUS F IRLMx,STATUS,ALLD
If the job owing the ACTIVE connector does not respond as normal, then it is a candidate for recovery actions.

Coupling Facility Guidelines

- Issue commands on the system on which the ACTIVE connector resides. For example: D T
If the system does not respond as normal, then it is a candidate for recovery actions.
- Issue D XCF,S,ALL to ensure that all systems are ACTIVE in the sysplex.
- Issue D R,L,CN=(ALL) to see if there are any outstanding events that might be holding up the processing of the FAILING or DISCONNECTING connector.
- Examine syslogs for signs of trouble such as ABENDs or DUMPs around the suspected time of the connector hang.
If a job owning an ACTIVE connector has experienced ABENDs or DUMPs, then it is a candidate for recovery actions. If the system has experienced problems such as ABENDs, DUMPs, SPINLOOPs, or other anomalies, then it is a candidate for recovery actions.

Determine Recovery Actions

The following section describes recovery actions depending on what was found in the preceding steps:

- If something obvious was found, such as a critical WTOR that needed a response, respond as appropriate and verify that the FAILING or DISCONNECTING connector condition was resolved.
- If a job was identified as owing a response to a FAILING or DISCONNECTING connector, or if a job or system was identified as a candidate for recovery actions, go to “Collect Documentation” on page 347.
- If no job or system was identified as requiring a recovery action (everything appears to be normal), and if the structure supports rebuilding, continue with “Rebuilding the Structure”. Otherwise, continue with “Collect Documentation” on page 347.

Rebuilding the Structure

Rebuild the structure that has the hung connectors by issuing the following command. (See *OS/390 MVS System Commands* for additional information.) If the structure does not support rebuild, go to “Collect Documentation” on page 347.

```
SETXCF START,REBUILD,STRNAME=strname
```

After the REBUILD command is issued, display the structure information by issuing the following command: D XCF,STR,STRNAME=*strname*

Examine the output of the command (IXC360I) and locate information about the REBUILD phase and the state of the structure connectors. For example,

```
IXC360I 12.23.28 DISPLAY XCF xxx
STRNAME: IRLMLOCK1
STATUS: REASON SPECIFIED WITH REBUILD START:
OPERATOR INITIATED
REBUILD PHASE: QUIESCE
|
|
|
|
| (only relevant portions of message IXC360I are
| shown here)
|
|
|
```

* ASTERISK DENOTES CONNECTOR WITH OUTSTANDING REBUILD RESPONSE

CONNECTION NAME	ID	VERSION	SYSNAME	JOBNAME	ASID	STATE
-----------------	----	---------	---------	---------	------	-------

Coupling Facility Guidelines

```
* IRLMGRP1$IRLA001 03 0003002F SYS1 IRLMA 001D ACTIVE
IRLMGRP1$IRLB002 02 0002008C SYS2 IRLMB 0058 ACTIVE
IRLMGRP1$IRLC003 01 00010099 SYS3 IRLMC 003F FAILING
IRLMGRP1#IRLC004 04 0004001A SYS3 IRLMC 0180 DISCONNECTING
```

Determine if the REBUILD is “hung” by displaying the structure information again. Examine the REBUILD phase of the structure and the state of the connectors. If the structure REBUILD is not progressing (for example, the structure is stuck in the REBUILD QUIESCE phase), look for connectors that have an outstanding REBUILD response (denoted by a * to the left of the connection name).

Note: Allow some time between issuing the display commands to ensure that the ACTIVE connectors have had a chance to process the REBUILD request. Structure REBUILD may take a few minutes. Display the structure a few times to verify that the REBUILD phase and the outstanding connector responses (*) are not changing. The REBUILD phase is displayed in message IXC360I.

If the REBUILD is not progressing and one or more of the ACTIVE connectors has an outstanding REBUILD response, the job associated with the connectors that are not responding to REBUILD (IRLMA in the above example), requires recovery actions. Continue with “Collect Documentation”.

Collect Documentation

Dump the XCFAS address space and its dataspace on all systems containing connectors (ACTIVE or otherwise) to the structure with the hung connectors.

Dump the address spaces and dataspace of all ACTIVE connectors to the structure with the hung connectors. In the example above, you would dump the XCFAS address space and dataspace on SYS1, SYS2, and SYS3, and IRLMA and IRLMB address spaces and their dataspace.

If a REBUILD was attempted and resulted in a REBUILD hand, and if the purpose of the REBUILD was simply to identify an ACTIVE connector that owed a response to a connector that was in a FAILING or DISCONNECTING state, stop the REBUILD by issuing the following: SETXCF STOP,REBUILD,STRNAME=*strname*

Continue with “Recover the Hung Connectors”.

Recover the Hung Connectors

In this step you will be shutting down one or more jobs, systems, or both to recover the hung connector(s). If it becomes necessary to shut down more than one job or system, consider shutting down one job or system at a time and then check to see if that action recovered the hung connector. If not, proceed to the next job or system shut down.

- If a job was identified as a candidate for recovery actions, shut down that job.
- If a system was identified as a candidate for recovery actions, shut down that system.
- If no jobs or systems were identified as candidates for recovery actions, shut down all jobs containing ACTIVE connectors to the structure with hung connectors.

After the shut down is complete, display the structure by issuing the following command: D XCF,STR,STRNAME=*strname*

Coupling Facility Guidelines

Examine the output of the command (IXC360I) and locate information about the structure connectors. (In the following example, IRLMA on SYS1 was shut down.)

CONNECTION NAME	ID	VERSION	SYSNAME	JOBNAME	ASID	STATE
IRLMGRP1\$IRLA001	03	0003002F	SYS1	IRLMA	001D	FAILED- PERSISTENT
IRLMGRP1\$IRLB002	02	0002008C	SYS2	IRLMB	0058	ACTIVE
IRLMGRP1\$IRLC003	01	00010099	SYS3	IRLMC	003F	FAILED- PERSISTENT

- If the FAILING and DISCONNECTING states have been resolved as indicated by the FAILING connector entering a FAILED-PERSISTENT or undefined state, and the DISCONNECTING connector becoming undefined, reinitialize the jobs that were shut down. Those jobs should restart and reconnect to the structure successfully.
- If the FAILING and DISCONNECTING states have not been resolved:
For the structure with the hung connectors, shut down all the remaining ACTIVE connectors. With all the active connectors shut down, the connectors should all be in an undefined state or a FAILED-PERSISTENT state depending on the exploiter of the structure. Issue `D XCF,STR,STRNAME=strname` and verify that all connectors have entered a FAILED-PERSISTENT or undefined state. Once verified, restart all jobs.
- If all the connector jobs were shut down and the connectors still have not entered a FAILED-PERSISTENT or undefined state, contact XCF Programming Support for additional assistance.

Key Sources of Information

You should be familiar with the following information:

- Parallel Sysplex website (<http://www.s390.ibm.com/pso/>).
- *OS/390 Parallel Sysplex Test Report*.
- *OS/390 Parallel Sysplex Recovery*, which contains information about recovery in a sysplex.

Appendix F. Defining an IEFAUTOS Structure for Automatic Tape Switching

For general information about automatic tape switching, see *OS/390 HCD Planning*.

The systems in a sysplex store the status of online automatically switchable tape devices in a structure called IEFAUTOS, which you define in the CFRM policy. “Chapter 4. Managing Coupling Facility Resources” on page 37 provides considerations for setting up a CFRM policy.

You set up your CFRM policy using the administrative data utility (IXCMIAPU), which resides in SYS1.MIGLIB. See “Appendix B. Administrative Data Utility” on page 263.

To define the IEFAUTOS structure in the CFRM policy, you specify the name of the structure (IEFAUTOS), its size, and a preference list (a prioritized list of the coupling facilities on which you want the structure allocated). Optionally, you can also specify an exclusion list — the names of any structures you do not want allocated in the coupling facility in which the IEFAUTOS structure is actually allocated. Do not specify the INITSIZE parameter when defining the IEFAUTOS structure because the structure does not support structure alter.

Recommendations for Enhancing the Availability of Automatic Tape Switching

To prevent possible failures in automatic tape switching caused by a failure of the coupling facility on which the IEFAUTOS structure is stored or of the IEFAUTOS structure itself, IBM recommends that you:

- Provide a backup coupling facility

If a coupling facility or the connection to the coupling facility should fail, the system tries to rebuild the IEFAUTOS structure in the backup coupling facility. Specify this coupling facility in the preference list.

- Limit access to IEFAUTOS

To prevent corruption of the IEFAUTOS structure, use a security product, such as RACF, to limit access to the structure. Only the system, not applications, should be able to access the structure. Create profiles that prevent applications from connecting to IEFAUTOS (a UACC of NONE). Define the resource profile IXLSTR.IEFAUTOS in the FACILITY class and ensure that the FACILITY class is active. For more information, see *OS/390 SecureWay Security Server RACF Security Administrator's Guide*.

- Provide sufficient room in the coupling facility to rebuild IEFAUTOS if it should fail.

If the IEFAUTOS structure fails and there is enough room in the coupling facility, the system will attempt to rebuild the IEFAUTOS structure. Note that the IEFAUTOS structure supports the rebuild process but does not support the structure alter process.

Determining the Size of an IEFAUTOS Structure

The size of the IEFAUTOS structure depends on the number of MVS systems that will be connected to the structure, the level of the coupling facility, the level of MVS you are running, and the number of automatically switchable devices that are online to those systems. When determining the size of the IEFAUTOS structure, include sufficient storage to accommodate any automatically switchable devices that your installation might add dynamically. Specify the size of the IEFAUTOS structure with the SIZE parameter in the CFRM policy.

Inadequate storage for the IEFAUTOS structure can cause one of the following conditions:

- Automatically switchable devices cannot be allocated, deallocated, or varied online.
- New systems cannot connect to the structure.
- Accurate status information about the structure is not available.

(To address the problems caused by a lack of storage, you can vary some of the automatically switchable devices offline or increase the size of the structure in the CFRM policy, activate the updated policy, and issue a SETXCF START,REBUILD command to rebuild the structure with a larger size.)

The following table shows the minimum number of 1K blocks to be allocated for IEFAUTOS in the coupling facility if you are implementing signaling with MVS/ESA SP 5.2 and Coupling Facility Level 0 or 1.

Table 13. Determining the Size of IEFAUTOS

Number of Systems	Number of Devices									
	10	20	30	40	50	60	70	80	90	100
2	154	165	177	188	200	212	223	235	246	258
3	157	173	188	204	219	235	250	266	281	297
4	161	181	200	219	239	258	277	297	316	335
5	165	188	212	235	258	281	304	328	351	374
6	169	196	223	250	277	304	331	358	386	413
7	173	204	235	266	297	328	358	389	420	451
8	177	212	246	281	316	351	386	420	455	490
9	181	219	258	297	335	374	413	451	490	529
10	184	227	270	312	355	397	440	482	525	567
11	188	235	281	328	374	420	467	513	560	606
12	192	242	293	343	393	444	494	544	594	645
13	196	250	304	358	413	467	521	575	629	683
14	200	258	316	374	432	490	548	606	664	722
15	204	266	328	389	451	513	575	637	699	761
16	208	273	339	405	471	536	602	668	734	799
17	212	281	351	420	490	560	629	699	768	838
18	215	289	362	436	509	583	656	730	803	877
19	219	297	374	451	529	606	683	761	838	915
20	223	304	386	467	548	629	710	792	873	954
21	227	312	397	482	567	652	737	823	908	993
22	231	320	409	498	587	676	765	853	942	1031
23	235	328	420	513	606	699	792	884	977	1070
24	239	335	432	529	625	722	819	915	1012	1109
25	242	343	444	544	645	745	846	946	1047	1147
26	246	351	455	560	664	768	873	977	1082	1186
27	250	358	467	575	683	792	900	1008	1116	1225
28	254	366	478	590	703	815	927	1039	1151	1263
29	258	374	490	606	722	838	954	1070	1186	1302
30	262	382	502	621	741	861	981	1101	1221	1341

Table 13. Determining the Size of IEFAUTOS (continued)

Number of Systems	Number of Devices									
	10	20	30	40	50	60	70	80	90	100
31	266	389	513	637	761	884	1008	1132	1256	1379
32	270	397	525	652	780	908	1035	1163	1290	1418

A formula for more precisely calculating the size for an IEFAUTOS structure appears in *ES/9000 and ES/3090 PR/SM Planning Guide*. Figure 75 provides values required by the formula.

Number of automatically switchable devices

The total number of tape devices that will be defined as automatically switchable among all the systems. Add contingency for devices that may be dynamically defined as automatically switchable.

Number of systems

The number of systems connected to IEFAUTOS.

List element characteristic

Value of 0.

List count

Value of 266.

Number of lock entries

The number is 16.

Maximum number of data elements

The maximum number of data elements that are to be associated with a single list entry in IEFAUTOS is 1.

Initial entry-to-element ratio

The ratio is 1.

Maximum list-set-entry count

The product of two numbers that you multiply:

- The number of automatically-switchable devices
- The number of systems connected to IEFAUTOS + 1.

Lock table entry characteristic

The value is 0.

Figure 75. Variables Required to Determine a Precise Size of IEFAUTOS

Updating SYS1.PARMLIB for Automatic Tape Switching

To support automatic tape switching, you also need to update the GRSRNLxx member of SYS1.PARMLIB by adding the following system inclusion RNL:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSZVOLS)
```

This specification prevents a system from holding a tape device while it waits for a mount of a volume that is in use by another system in the sysplex.

Appendix G. 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
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

Notices

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- CICS
- DB2
- DFSMS
- DFSMSdfp
- DFSMSdss
- DFSMS/MVS
- eNetwork
- ES/3090
- ES/9000
- ESCON
- IBM
- IBMLink
- IMS
- IMS/ESA
- MVS/ESA
- MVS/XA
- OS/390
- PR/SM
- Processor Resource/Systems Manager
- RACF
- RETAIN
- RMF
- S/390
- System/390
- Sysplex Timer
- VM/ESA
- VTAM

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

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

Glossary

Sources of Terms and Definitions

This glossary defines technical terms and abbreviations used in sysplex documentation. If you do not find the term you are looking for, view *IBM Glossary of Computing Terms*, located at:

<http://www.networking.ibm.com/nsg/nsgmain.htm>

A

Automatic restart management. Automatic restart management is an MVS recovery function that improves the availability of batch jobs and started tasks. When a job fails, or the system on which it is running unexpectedly fails, MVS can restart the job without operator intervention.

C

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

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

CFRM. Coupling facility resource management.

channel-to-channel (CTC). Refers to the communication (transfer of data) between programs on opposite sides of a channel-to-channel adapter (CTCA).

couple data set. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the MVS systems in a sysplex. See also *sysplex couple data set*.

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

coupling facility channel. A high bandwidth fiber optic channel that provides the high-speed connectivity required for data sharing between a coupling facility and the central processor complexes directly attached to it.

coupling services. In a sysplex, the functions of XCF that transfer data and status between members of a group residing on one or more MVS systems in the sysplex.

CP. Central processor.

CPC. Central processor complex.

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

cross-system extended services (XES). XES is a set of services that allow authorized applications or subsystems running in a sysplex to share data using a coupling facility.

cross-system restart. If a system fails, automatic restart management restarts elements on another eligible system in the sysplex.

CTC. Channel-to-channel.

D

DASD. Direct access storage device.

data sharing. The ability of concurrent subsystems (such as DB2 or IMS DB) or application programs to directly access and change the same data while maintaining data integrity.

DB2. DATABASE2.

DFSMS/MVS. Data Facility Storage Management Subsystem.

E

element. A batch job or started task that uses the IXCARM macro to register for automatic restart management services. In the event of an unexpected failure of the element itself or of the system it is running on, MVS automatically restarts the element.

Enterprise Systems Connection (ESCON). A set of products and services that provides a dynamically connected environment using optical cables as a transmission medium.

ESCD. ESCON Director.

ESCON. Enterprise Systems Connection.

ETR. External Time Reference. See also *Sysplex Timer*.

G

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

global resource serialization complex. One or more MVS systems that use global resource serialization to serialize access to shared resources (such as data sets on shared DASD volumes).

I

IMS DB. Information Management System Database Manager.

J

JES2. Job Entry Subsystem 2.

JES3. Job Entry Subsystem 3.

L

LIC. Licensed Internal Code.

logical partition (LP). A subset of the processor hardware that is defined to support an operating system. See also *logically partitioned (LPAR) mode*.

logically partitioned (LPAR) mode. A central processor complex (CPC) power-on reset mode that enables use of the PR/SM feature and allows an operator to allocate CPC hardware resources (including central processors, central storage, expanded storage, and channel paths) among logical partitions.

LP. Logical partition.

LPAR. Logically partitioned (mode).

M

member. A specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group.

mixed complex. A global resource serialization complex in which one or more of the systems in the global resource serialization complex are not part of a multisystem sysplex.

monoplex. A sysplex consisting of one system that uses a sysplex couple data set.

multiprocessing. The simultaneous execution of two or more computer programs or sequences of instructions. See also *parallel processing*.

multiprocessor (MP). A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application. An application program that has various functions distributed across MVS images in a multisystem environment.

multisystem environment. An environment in which two or more MVS images reside in one or more processors, and programs on one image can communicate with programs on the other images.

multisystem sysplex. A sysplex in which two or more MVS images are allowed to be initialized as part of the sysplex. See also *single-system sysplex*.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

MVS system. An MVS image together with its associated hardware, which collectively are often referred to simply as a system, or MVS system.

MVS/ESA. Multiple Virtual Storage/ESA.

N

n-way. The number (*n*) of CPs in a CPC. For example, a 6-way CPC contains six CPs.

P

parallel processing. The simultaneous processing of units of work by many servers. The units of work can be either transactions or subdivisions of large units of work (batch).

PR/SM. Processor Resource/Systems Manager.

Processor Resource/Systems Manager (PR/SM). The feature that allows the processor to use several MVS images simultaneously and provides logical partitioning capability. See also *LPAR*.

persistent command text. The command text that MVS saves when a started task is started, and uses to restart the job through automatic restart management.

persistent JCL. The JCL that MVS saves when a job is started, and uses to restart the job through automatic restart management.

R

RACF. Resource Access Control Facility.

register. To issue the IXCARM macro to become an element of automatic restart management, an MVS recovery function that restarts jobs in the event of an unexpected failure.

restart group. A group of related jobs, registered as elements of automatic restart management, that need to be restarted together on the same system, should a system fail unexpectedly.

RMF. Resource Measurement Facility.

S

SFM. Sysplex failure management

single-system sysplex. A sysplex in which only one MVS system is allowed to be initialized as part of the sysplex. In a single-system sysplex, XCF provides XCF services on the system but does not provide signaling services between MVS systems. See also *multisystem sysplex*, *XCF-local mode*.

SMS. Storage Management Subsystem.

structure. A construct used by MVS to map and manage storage on a coupling facility.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. See also *MVS system*.

sysplex couple data set. A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All MVS systems in a sysplex must have connectivity to the sysplex couple data set. See also *couple data set*.

Sysplex Timer. An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the MVS generic name for the IBM Sysplex Timer (9037).

System/390 microprocessor cluster. A configuration that consists of central processor complexes (CPCs) and may have one or more coupling facilities.

W

WLM. MVS workload manager.

X

XCF. Cross-system coupling facility.

XCF PR/SM policy. In a multisystem sysplex on PR/SM, the actions that XCF takes when one MVS system in the sysplex fails. This policy provides high availability for multisystem applications in the sysplex.

XCF-local mode. The state of a system in which XCF provides limited services on one system and does not provide signaling services between MVS systems. See also *single-system sysplex*.

XES. Cross-system extended services

Index

A

ACOUPLE keyword 32
administrative data utility
 coding 264
 description 263
 return code 301
 using 263
 utility control statements 264
ALLOWAUTOALT keyword 278
APPC/MVS
 finding information 187
authorization requirements
 system logger 215
AUTODELETE parameter
 system logger 213, 214, 215
automatic deletion policy
 system logger 208, 213, 214, 215
 with a resource manager 215
automatic restart management 139
 checkpoint/restart 154
 parameters 246
 planning 139
 requirements for 154
 setting up a policy 155
automatic structure alter 51
average log block size
 system logger 218
 OS/390 Release 3 or above 218
 pre-OS/390 Release 3 218
AVGBUFSIZE
 system logger 218
 OS/390 Release 3 or above 218
 pre-OS/390 Release 3 218

C

capacity planning
 signaling path 128
CFRM
 system logger
 coupling facility attributes in CFRM 225
 define the coupling facility structure attributes in
 CFRM 225
CFRM (coupling facility resource management)
 couple data set, formatting 35
 policies 55
CICS log manager
 finding information 184, 185
CLASS keyword 78, 84
class length, tuning 118
CLASSDEF statement 78, 79, 118
CLASSLEN keyword 79, 118
cleanup interval 141
CLEANUP keyword 141
CLOCKxx parmlib member
 planning 15, 23
component trace 21

configurations
 multisystem 161
 with existing complex 163
 without existing complex 161
 on PR/SM, sysplex on LPARs 317
 single system 160
 under VM 315
connector persistence
 description 321
CONSOLxx parmlib member 31, 244
 planning 21
control interval size 220
couple data set 30
 ARM, formatting 252
 CFRM, formatting 254
 CFRM parameters 246
 connectivity requirements 36
 considerations 25
 automatic restart management 25, 34
 CFRM 25, 34
 LOGR 25, 34
 security 29
 SFM 25, 34
 sysplex 25
 system joining sysplex 32
 WLM 25, 34
 format 241
 joining a sysplex 32
 LOGR 178
 LOGR, formatting 255
 LOGR parameters 248
 OS/390 UNIX, formatting 257
 OS/390 UNIX parameters 249
 performance considerations 28
 planning 25
 policies 5
 automatic restart management 5
 CFRM 5
 LOGR 5
 SFM 5, 25
 WLM 5, 25
 SFM, formatting 256
 SFM parameters 248
 SMS-managed volume 29
 switching 26
 system logger 178
 WLM, formatting 259
 WLM parameters 249
COUPLE statement 10, 32, 76, 140, 141, 142
COUPLE system parameter
 options 19
COUPLExx parmlib member
 CLEANUP keyword 141
 component trace options 21
 COUPLE statement 140
 examples 85
 defaults, using 86

COUPLExx parmlib member (*continued*)

- message traffic considerations, handling range 90
- signaling paths, defining through coupling facility 100
- signaling through coupling facility and CTC connections 102
- transport class, creating for group 98
- transport class, creating for large messages 88
- failure-related information 139
- INTERVAL keyword 140
- modifying values 21
- OPNOTIFY keyword 140
- status update missing condition 142
- syntax 19
- with PLEXCFG=MONOPLEX 10
- with PLEXCFG=MULTISYSTEM 10
- with PLEXCFG=XCFLOCAL 10

coupling facility

- adding 324, 325
- control code support 38
- description 37, 41
- dump space 53
- failure-independence 39
- guidelines 321
- identifying 41
- nonvolatility 39
- operator commands 62
- planning, physical 38
- POPULATECF function 63
- power save state 39
- rebuilding structures 334
- reconfiguring 321, 331
- removing 324
- removing the only one from a sysplex 329
- replacing 321
- resetting 61
- storage allocation 59
- storage increment 61
- structure threshold for rebuild 47
- system logger
 - damage 236
 - failure 236, 237
 - full condition 237
 - loss of connectivity to 236
 - system logger data on 175
 - types of storage 59

coupling facility structure

- identifying 42
- space needed, per log stream
 - calculating 197
 - formula 197
- system logger
 - space needed, per log stream 197

D

DASD log data set

- space 238

DASD log data sets

- increasing space for 207

DASD log data sets (*continued*)

- SMS management of 179

DASD-only log stream

- DASDONLY 217
- system logger 217
- system logger applications
 - staging data sets 210
- system logger data on 175

DASDONLY

- LOGR policy 217
- system logger 217

data deletion

- system logger 215

DATA keyword 252

DATA statement 264

DEFAULT statement 31

DEFINE POLICY statement 266

DEVICE keyword 72

DFW (DASD fast write) feature

- using 28

dump space

- in coupling facility 53

duplexing

- conditional 222
- system logger
 - example 222, 223
 - methods 220
- unconditional 222

E

ENFORCEORDER keyword 281

entry-to-element ratio 179

ETRMODE statement 16

ETRZONE statement 16

example

- log stream configuration 193
- merging log data across the sysplex 193
- system logger
 - duplexing 222, 223
 - system logger applications
 - multiple log streams in the same coupling facility structure 194

exclusion list

- in CFRM policy 45

EXSPATxx parmlib member 140

F

failed-persistent connectors

- deleting 338

failure detection interval 140

format utility

- examples of using 251
- XCF couple data set 241
 - coding 242
 - description 241
 - return code 250
 - sample JCL 252
 - sample output 261
 - using 241

FULLTHRESHOLD keyword 278

G

- global resource serialization
 - installation 163
 - planning 12
- GROUP keyword 78
- GRS system parameter
 - planning 12
 - SYSGRS (group name) 31
- GRSCNF system parameter
 - planning 14
- GRSRNL system parameter
 - planning 15

I

- IEASYSxx system parameter
 - planning 9, 23
- IEFAUTOS structure
 - determining the size 350
- IMS EMH structure
 - deleting 337
- IMS log manager
 - finding information 188
- IMS shared-queue structure
 - deleting 337
- INTERVAL keyword 140, 149
- IXCL1DSU utility program 241
- IXCMIAPU utility program 263
 - examples of using 302
 - ARM policy data 302
 - CFRM policy data 305
 - LOGR data 307
 - SFM policy data 307
- IXGINVNT service
 - duplexing
 - DUPLEXMODE parameter 221
 - STG_DUPLEX parameter 221
 - DUPLEXMODE parameter 221
 - STG_DUPLEX parameter 221

J

- JCL (job control language)
 - OS/390 couple data set format utility 251
 - Pre-OS/390 couple data set format utility 252
- JCL control statements 263
- JES3
 - complex in a sysplex 32

L

- LOCAL sysplex name 19
- local time, adjusting 16
- LOCALMSG statement 79
- log data
 - deleting 227
 - on DASD log data sets 175
- log data deletion
 - offloading 182
- log data sets 175

- log data sets 175 (*continued*)
 - allocation 175
 - data class 212
 - define DASD space for 206
 - deleting 227
 - GRSRNL inclusion list 212
 - naming convention for 203
 - share options 212
 - SMS environment for 212
- log stream
 - calculating
 - space needed, per log stream 197
 - configuration, examples 193
 - merging log data across the sysplex
 - example 193
 - multiple log streams in the same coupling facility structure
 - example 194
 - resource manager managed 224
 - upgrading
 - from DASD-only to coupling facility 229
- log stream subsystem
 - start 226
- log streams
 - deleting 224
- LOGR
 - couple data set 178
 - policy 178
- LOGR couple data set 178
 - define the coupling facility structure attributes in CFRM 225
 - DSEXTENT parameter 207
 - format and define the LOGR couple data sets 216
- LOGR policy 178
 - DASDONLY 217
 - define coupling facility structures in 217
 - define log streams in 217
 - defining a log stream 282
 - defining a structure 289
 - deleting a log stream 296
 - deleting a structure 296
 - listing a log stream 296
 - wildcards 297
 - listing a structure 296
 - wildcards 297
 - updating a log stream 290
- LOGR subsystem
 - read log data in data set format 226
 - start 226
- logrec log stream
 - finding information 184, 186
- LPAR (logical partition) configuration 318

M

- managing log data 213
- MAXGROUP keyword 252
- MAXMEMBER keyword 252
- MAXMSG keyword 81, 84
- MAXSYSTEM keyword 171
- message buffer
 - space 80

- message buffer (*continued*)
 - tuning 118
- message traffic, segregating 120
- MGTCLAS keyword 30
- MINSIZE keyword 277
- MODIFY keyword 21
- multisystem sysplex
 - description 163, 168, 169, 170
 - greater than eight systems 170
 - installation 161, 163, 168, 169, 170
 - on one processor 168
 - on PR/SM 168
 - with coupling facility 169
- MVS components and subsystems
 - group names and members 31
 - APPC 31
 - console services 31
 - DAE 31
 - DFSMS 31
 - JES2 MAS 31
 - VLF 32
 - WLM 32
 - XES 32

N

- Notices 353

O

- offloading
 - system logger data 180, 181
- operator notification interval 140
- OPERLOG log stream
 - finding information 184, 186
- OPNOTIFY keyword 140

P

- parmlib members
 - CLOCKxx 168
 - CONSOLxx 31, 244
 - COUPLExx 5, 9, 32, 118, 139, 159
 - EXSPATxx 140
 - GRSCNFxx 162
 - GRSRNL 15
 - GRSRNLxx 162
 - IEASYMxx 9
 - IEASYSxx 4, 9
 - XCFPOLxx 139, 148
- PATHIN statement 10, 76, 80, 83
- PATHOUT statement 10, 76, 79, 80
- PCOUPLE keyword 32
- performance
 - couple data set considerations 28
- persistence
 - description 321
- planning
 - availability 1, 139
 - cleanup interval 141
 - couple data sets 25

- planning (*continued*)
 - failure detection interval 140
 - IEASYSxx parmlib member 9, 23
 - inbound signaling path 72
 - message buffer space 80
 - operator notification interval 140
 - outbound signaling path 72
 - PR/SM reconfigurations 146
 - recovery 139
 - signaling services 69
 - status update missing condition 142
 - system logger applications 188
 - automatically allocated staging data sets 209
 - configuration 191
 - coupling facility space for log streams 197
 - DASD space 206
 - DASD space for log data sets 206
 - DASD space for staging data sets 208
 - data class 212
 - GRSRNL inclusion list 212
 - map log streams to structures 192
 - naming convention for system logger
 - resources 203
 - naming conventions for coupling facility
 - structures 206
 - naming conventions for log stream and staging
 - data sets 203
 - number of log streams 191
 - share options 212
 - SMS environment for DASD data sets 212
 - transport class 77
 - XCFPOLxx, controlling availability and recovery 148
- PLEXCFG system parameter
 - planning 9
- policy 155
 - automatic restart management 139
 - CFRM 41
 - defining and activating 35
 - LOGR 178
 - SFM 139, 141
 - system logger 178
 - updating 35
- POLICY control statement 263
- POPULATECF function 63
- PR/SM (Processor Resource/Systems Manager)
 - changing the XCF policy 148
 - in sysplex configuration 317
 - sysplex migration 168
- preference list
 - in CFRM policy 45
- preparing for
 - system logger applications 188
 - automatically allocated staging data sets 209
 - configuration 191
 - coupling facility space for log streams 197
 - DASD space 206
 - DASD space for log data sets 206
 - DASD space for staging data sets 208
 - data class 212
 - GRSRNL inclusion list 212
 - map log streams to structures 192

- preparing for (*continued*)
 - naming convention for system logger
 - resources 203
 - naming conventions for coupling facility
 - structures 206
 - naming conventions for log stream and staging
 - data sets 203
 - number of log streams 191
 - share options 212
 - SMS environment for DASD data sets 212
- PRSMPOLICY keyword 148

R

- read
 - log data in data set format
 - LOGR subsystem 226
- REBUILDPERCENT keyword 47, 280
- recovery
 - system logger 230
 - coupling facility failure 236
 - coupling facility full 237
 - DASD log data set space fills 238
 - example, system failure 196, 197
 - log stream damage 238
 - peer connector 195, 231
 - staging data set full 237
 - sysplex failure 232
 - system failure 195, 231
 - system failure, single system sysplex 232
 - system logger address space failure 233
 - unrecoverable I/O errors 239
- rejoining a sysplex 172
- removing a system from a sysplex 171
- requirements
 - system logger 190
 - system logger coupling facility 190
- resource manager
 - associating a log stream with 224
 - system logger
 - with automatic deletion 215
- retention period
 - system logger 208, 213, 214, 215
- RETPD parameter
 - system logger 213, 214, 215
- RETRY keyword 76, 84
- RMAX keyword 31
- RMF (Resource Measurement Facility)
 - XCF Activity Report 129

S

- SETXCF command 21
- SFM (sysplex failure management) 25
- signaling
 - description 6
 - examples 85
 - creating class for large messages 88
 - defaults, using 86
 - message traffic considerations, handling
 - range 90

- signaling (*continued*)
 - examples 85 (*continued*)
 - signaling paths, defining through coupling
 - facility 100
 - signaling through coupling facility and CTC
 - connections 102
 - transport class, creating for group 98
 - options 69
 - recovery capabilities 72
 - services 69
 - CFRM policy 72
 - COUPLExx parameters 76
 - list structure, determining size 73
 - message buffer space 80
 - performance 117
 - setting up 72
 - signaling paths 72
 - single point of failure, avoiding 73
 - sizes, calculating 309
 - transport classes 77
 - tuning 118
 - XCF groups 77
 - systems management 71
- signaling path
 - capacity planning 124, 128
 - CLASS keyword 79
 - defining through coupling facility 100
 - inbound 72
 - message counts 126
 - none in class 126
 - outbound 72
 - performance 125
 - planning 72
 - planning for transport class 79
 - restarts 126
 - retry limit 75
 - STRNAME keyword 80
 - tuning 124
- SIMETRID statement 16
- single point of failure
 - system logger 220
- SMS (Storage Management Subsystem) 29
- SMS parameters
 - system logger
 - ACS routines, interaction 219
 - LIKE parameter, interaction 219
- staging data sets 180
 - automatically allocated 209
 - connectivity to 209, 210
 - DASD-only log stream 210
 - data class 212
 - define DASD space for 208
 - full condition 237
 - GRSRNL inclusion list 212
 - naming convention for 203
 - share options 212
 - size 183
 - SMS environment for 212
 - system logger 180
- START keyword 83

- status update missing
 - planning for 142
- STOP keyword 83
- STORCLAS keyword 30
- STRNAME keyword 80
- structure
 - deleting persistent structures 338
 - rebuilding in a coupling facility 334
 - removing the coupling facility 331
- Structure Full Monitoring 49
- structure persistence
 - description 321
- structure rebuild
 - alternatives to 334
 - initiated by MVS 47
 - procedure for non-supported IBM structures 332
 - process 334
- sysplex
 - automatic restart management 154
 - configurations 315
 - couple data set requirement 1
 - COUPLExx defaults, using 86
 - description 1
 - format utility 241
 - hardware requirements 1
 - illustration 2
 - installing 7
 - local time, adjusting 16
 - MVS systems, installing 159
 - parameter values (SYS1.PARMLIB) 4
 - parmlib member, required values 9
 - PDSEs, sharing 31
 - policies 5
 - defining in couple data set 6
 - description 5
 - purpose 5
 - SFM 139
 - rejoining 172
 - services, planning 69
 - SFM policy 141
 - defining 147
 - PR/SM reconfigurations 141
 - signaling connectivity failures 141, 144
 - system availability and recovery, controlling 141
 - system failure responses 141
 - signaling 6
 - description 6
 - software requirements 1
 - Sysplex Timer 2
 - tuning 7, 117
 - XCF 1
 - Year 2000 Testing 8
- Sysplex Timer
 - description 2
 - logical time offset 8
- SYSPRINT DD statement 242
- system logger
 - APPC/MVS
 - finding information 187
 - authorization requirements 215
 - AUTODELETE parameter 213, 214, 215
 - system logger (*continued*)
 - automatic deletion policy 208, 213, 214, 215
 - automatic deletion with a resource manager 215
 - average log block size 218
 - OS/390 Release 3 or above 218
 - pre-OS/390 Release 3 218
 - AVGBUFSIZE 218
 - CICS log manager
 - finding information 185
 - component 177
 - configuration 176
 - example 193, 194
 - illustration 176, 177
 - couple data set 178
 - coupling facility requirements 190
 - coupling facility size 182, 197
 - DASD log data sets
 - increasing space for 207
 - DASD-only log stream 175, 217
 - DASDONLY 217
 - data deletion 215
 - definitions in the LOGR policy 217
 - deleting log data 227
 - deleting log data sets 227
 - deleting log streams 224
 - duplexing
 - DASD-only 180
 - example 222, 223
 - methods 220
 - duplexing log data 179
 - DASD-only 180
 - from a coupling facility log stream 179
 - entry-to-element ratio 179
 - example
 - duplexing 222, 223
 - examples 193
 - IMS log manager
 - finding information 188
 - log data deletion 182
 - log data on the coupling facility 175
 - log data sets
 - allocation 175
 - SMS management of 179
 - log stream
 - resource manager managed 224
 - upgrading 229
 - LOGR couple data set 178
 - DSEXTENT parameter 207
 - LOGR policy 178
 - LOGR subsystem 226
 - logrec log stream
 - finding information 186
 - low offload threshold 181
 - low threshold 181
 - managing log data 213
 - offloading 181
 - coupling facility size 182, 197
 - log data 180
 - log data deletion 182
 - staging data set size 183
 - thresholds 181

- system logger (*continued*)
 - OPERLOG log stream
 - finding information 186
 - peer connector
 - in recovery 195, 231
 - planning 173, 188
 - automatically allocated staging data sets 209
 - configuration 191
 - coupling facility space for log streams 197
 - DASD space for log data sets 206
 - DASD space for staging data sets 208
 - data class 212
 - GRSRNL inclusion list 212
 - map log streams to structures 192
 - naming convention for system logger
 - resources 203
 - naming conventions for coupling facility
 - structures 206
 - naming conventions for log stream and staging
 - data sets 203
 - number of log streams 191
 - share options 212
 - SMS environment for DASD data sets 212
 - policy 178
 - preparing for 188
 - read
 - log data in data set format 226
 - recovery 230
 - coupling facility failure 236
 - coupling facility full 237
 - DASD log data set space fills 238
 - example, system failure 196, 197
 - log stream damage 238
 - staging data set full 237
 - sysplex failure 232
 - system failure 195, 231
 - system failure, single system sysplex 232
 - system logger address space failure 233
 - unrecoverable I/O errors 239
 - requirements 190
 - retention period 208, 213, 214
 - RETPD parameter 213, 214, 215
 - single point of failure 220
 - SMS parameters and ACS routines 219
 - SMS parameters and LIKE parameter 219
 - SMS requirements 190
 - staging data sets 180
 - connectivity to 209, 210
 - size 183
 - system logger applications 173
 - thresholds and offloading 181
- system logger address space
 - authorization requirements 215
- system logger applications 173
 - authorization requirements 215
 - DASD-only log stream
 - staging data sets 210
 - DUPLEXMODE parameter 221
 - finding information 184
 - planning 188
 - preparing for 188

- system logger applications 173 (*continued*)
 - preparing to use 188
 - automatically allocated staging data sets 209
 - configuration 191
 - coupling facility space for log streams 197
 - DASD space for log data sets 206
 - DASD space for staging data sets 208
 - data class 212
 - map log streams to structures 192
 - naming convention for system logger
 - resources 203
 - naming conventions for coupling facility
 - structures 206
 - naming conventions for log stream and staging
 - data sets 203
 - number of log streams 191
 - share options 212
 - SMS environment for DASD data sets 212
 - STG_DUPLEX parameter 221
 - system parameters for IEASYSxx
 - CLOCK 15
 - COUPLE 19
 - GRS 12
 - GRSCNF 14
 - GRSRNL 15
 - PLEXCFG 9

T

- temporary storage (TS) queue pool structure
 - deleting 333
- time stamps, synchronizing 16
- TIMEZONE statement 15
- TOD clock
 - resetting, consequences 15
 - setting backward 18
 - setting forward 18
- transport class
 - assigning path 79
 - CLASSLEN keyword 79
 - creating class for group 98
 - creating class for large messages 88
 - default 78
 - DEFAULT name 78
 - defining 78
 - GROUP keyword 78
 - message length 79
 - planning 77
 - tuning 127
 - undesignated groups 78
 - XCF groups, assigning 78
- tuning sysplex
 - capacity planning 128
 - class length 118
 - introduction 117
 - message buffer size 118
 - message buffer space 121
 - adjusting 121
 - description 121
 - inbound, analyzing lack 124
 - outbound, analyzing lack 123

tuning sysplex (*continued*)
 RMF reports, using to analyze 122
 signaling path 124
 capacity planning 124, 128
 message counts 126
 none in class 126
 performance 125
 restarts 126
 signaling service 118
 transport classes 127

XCFPOLxx parmlib member 139, 148 (*continued*)
 planning 23

U

upgrading
 log stream
 from DASD-only to coupling facility 229
utility program
 to create policy data 263
 to format couple data set 241

V

VM System Product
 sysplex migration 168
VOLSER keyword 30

W

WLM (workload management) 25

X

XCF (cross-system coupling facility)
 component 1
 configurations 315
 format utility 241
 groups 77
 assigning to transport class 78
 description 77
 GROUP keyword 78
 message characteristics 77
 names 77
 undesignated 78
 local mode 10
 PDSEs, sharing 31
 PR/SM policy 148
 changing 148
 one processor, example 149, 151
 PRSMPOLICY keyword 148
 two processors, example 150
 RETAIN option 171
 SETXCF command 83
 add path 83
 changing the PR/SM policy 148
 delete path 83
 START keyword 83
 STOP keyword 83
 tuning a sysplex 117
XCF-local mode 159
XCFPOLxx parmlib member 139, 148

Readers' Comments — We'd Like to Hear from You

OS/390
MVS Setting Up a Sysplex

Publication No. GC28-1779-09

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



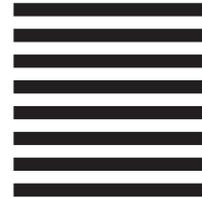
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5647-A01



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC28-1779-09

