z/OS
2.5

MVS Planning: Global Resource Serialization

SA23-1389-50

Note

Before using this information and the product it supports, read the information in “Notices” on page 219.

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2023-03-28

Contents

Figures ..... **ix**

Tables ..... **xiii**

About this information ..... **xv**
Who should use this information xv
How this information is organized xv
How to use this information xvi
Where to find more information xvi

How to send your comments to IBM ..... **xvii**
If you have a technical problem xvii

Summary of changes ..... **xix**
Summary of changes for z/OS MVS Planning: Global Resource Serialization for Version 2 Release 5 (V2R5) xix
Summary of changes V2R4 xix
Summary of changes V2R3 xix

Part 1. Global Resource Serialization **1**

Chapter 1. Introduction 3
Advantages of using global resource serialization 3
How global resource serialization works 3
Local and global resources 4
Using an API to serialize resources 5
ISGENQ macro 6
RESERVE macro 7
Potential problems when using RESERVE 7
Solving the RESERVE problems 8
Understanding the synchronous RESERVE feature 9
Issuing ISGQUERY and GQSCAN 10
ISGADMIN macro 10
Limiting global resource serialization requests 10
Methods of serializing global resources 12
The ring 13
The star 14
Which method to choose 15
Using IBM Health Checker for z/OS 16

Chapter 2. Selecting the data 17
Data set naming conventions 18
Authorized qnames 18
Dynamic exits and RNL processing 20
Request processing sequence 20
RNL processing 21
How the RNL is scanned 21
ISGQUERY RNL search routine 22
SYSTEM inclusion RNL 22

SYSTEMS exclusion RNL....23
RESERVE conversion RNL....23
Excluding requests from RNL processing....24
RNL processing sequence....29
RESERVE conversion....31
Migrating from GRSRNL=EXCLUDE to a set of RNLs....32
Changing the RNL....32
RNL defaults....33
CICS....35
DAE....35
DB2....36
DFSMShsm....36
SMS....36
IMS....36
ISPF or ISPF/PDF....36
JES2....37
JES3....37
System logger....38
RACF....38
Tape volumes....39
Temporary data sets....39
TSO/E....39
VIO journaling data set....40
VSAM data sets....41
RNL candidates....42
Defining the RNLs....45

Chapter 3. Using the ENQ/RESERVE/DEQ monitor tool....49
Security considerations....50
Setting up the monitor....50
Monitor execution....50
Starting the monitor....52
Option 1....53
Option 2....55
Option 3....55
Option 4....57
Monitor control....57
Messages - abends - return codes....60
Filter facility....62
Reports....66
Logs....67
Temporary data sets....68
In-stream procedures variables....68
Report programs parameters....68
Region requirements....69
ENQ/RESERVE/DEQ restrictions....79
Resources used and guidelines....80
Monitor diagnostic information....80
Monitor physical output record....80
Monitor utilization hints....81
ENQ/DEQ/RESERVE analysis aid reports....82
Records layout....85
DSECT monitor physical and logical record....85
ISGAMEDM ISGAMED1 ISGAMVOL output record fields....86
EDSORTED output record fields....86
ISGAMCTM output record fields....87

Part 2. Global Resource Serialization Star 89

Chapter 4. Star processing 91
Global ENQ/DEQ processing overview 92
ISGQUERY and GQSCAN processing overview 92
Cross-system processing option 93
Processing system failures 93
Contention management 93
Contention notification 94

Chapter 5. Planning a star complex 95
Decisions you need to make 95
Designing a star complex 95
Defining the sysplex couple data set for the star complex 96
The CFRM couple data set 97
Sizing the ISGLOCK structure 98
Getting the right structure size 98
Defining parmlib members for a star complex 99
Global resource serialization star definition parameter (GRSDEF) 99
Bringing up a star complex 101
Potential error scenarios 101
Error scenarios you can avoid 102
ISGLOCK rebuild processing 102
Coupling facility structure failure 103
Loss of connectivity to a coupling facility structure 103
Specifying global resource serialization tracing options (CTRACE) 103

Chapter 6. Operating the star complex 105
Operating the star complex 105
Building the complex 105
Normal operations in a sysplex 106
Rebuilding the ISGLOCK structure 108
Shutting down a coupling facility 109
Steps in converting to a star complex 109
Converting to a star complex 109

Part 3. Global Resource Serialization Ring 113

Chapter 7. Ring processing 115
The RSA-message 116
Processing a request for a resource 116
Request processing without ring acceleration 117
Request processing with ring acceleration 117

Chapter 8. Designing a ring complex 119
Designing a complex that matches a sysplex 120
Processing options in a sysplex 120
Designing a mixed complex (sysplex does not match complex) 125
Choosing the link configuration 126
Processing options in a mixed complex 130
Defining the complex to MVS 139

Chapter 9. Operating the ring complex 147
Operating a complex that matches a sysplex 147
Building the complex 147
Normal operations in a sysplex 148

Operating a mixed complex (complex does not match sysplex)..... 150
Building the complex..... 151
Normal operations in a mixed complex..... 153

Chapter 10. Installing and tuning the complex..... 163
Installing the complex..... 163
Installing a new complex..... 163
Migrating an existing complex into a sysplex..... 163
Tuning the complex..... 167
Average response time..... 167
Tuning factors..... 168
Tuning process..... 170
Measurements..... 171

Part 4. Global Resource Serialization Diagnosis..... 173

Chapter 11. Diagnosing global resource serialization..... 175
Discriminating between system and application problems..... 175
Check if the complex is operating normally..... 176
Tuning the global resource serialization ring..... 176
Ring disruption recovery..... 177
ISG177E and ISG178E recovery..... 177
Global resource serialization ring rebuild..... 179
Checking XCF/XES connectivity and performance..... 179
ISGLOCK structure request processing..... 179
Checking for ENQ contention problems..... 180
Checking for latch contention problems..... 185
Using SMF 87 records..... 185
Using SMF 87 subtype 1 records to identify global generic queue scan issuers..... 186
Using SMF 87 subtype 2 records to diagnose application problems related to ENQ and DEQ and monitor the usage of ENQ/DEQ/ISGENQ/RESERVE requests..... 189

Chapter 12. Measuring response time..... 195
Viewing system performance..... 195
Changing RESMIL values..... 196
Comparing response time..... 197

Appendix A. Data set ENQ contention monitor..... 199
Installing the data set ENQ contention monitor..... 199
Assembling the data set ENQ contention monitor source module..... 199
Link-editing the data set ENQ contention monitor source module..... 200
Changes to SYS1.PROCLIB..... 200
Using the data set ENQ contention monitor..... 201
Starting the monitor..... 201
Stopping the monitor..... 201
Output..... 201

Appendix B. Recovery actions for global resource serialization..... 203
Loss of connectivity to the coupling facility..... 203
Recovery when some ring complex systems are in sysplex..... 203
Recovery when sysplex and ring complex contain same systems..... 204
Recovery actions for ring complex with some systems not in a sysplex..... 204
Purge a system from the ring complex before reIPL..... 204
Actions to purge a system before reIPL..... 204
Ring disruption in progress during IPL..... 205
Recovery actions for a ring disruption during IPL..... 205
Primary link failure..... 205

Recovery actions for a primary link failure 207
Alternate link failure 207
Recovery actions for an alternate link failure 207
System failure 207
Automatic restart example 208
Manual restart 209
Restart of two-system complex 211
Reactivating a quiesced system 212
Recovery actions for reactivating quiesced systems 213

Appendix C. Accessibility 217

Notices 219
Terms and conditions for product documentation 220
IBM Online Privacy Statement 221
Policy for unsupported hardware 221
Minimum supported hardware 221
Trademarks 222

Index 223

Figures

1. The Conceptual View of the Ring Complex....13

2. Global Resource Serialization Ring Complex....14

3. A Conceptual View of the Star Complex....15

4. ENQ/DEQ exits installation flow....20

5. Alternate serialization product spans multiple GRS Complexes....26

6. Multiple alternate serialization product domains within a GRS Complex....27

7. Sample ENQ invocations to assist with converting from RNL=YES to RNL=NO....29

8. ENQ and DEQ processing summary....30

9. RESERVE Processing Summary....31

10. Contents of the default RNLs....34

11. JCL to Run the Monitor....51

12. GRS Ring Main Menu - ISPF Application....52

13. GRS Star Main Menu - ISPF Application....52

14. Option 1 - Major Name List....53

15. Selected Row - Minor Name List....54

16. Selected Row - Jobname List....55

17. Option 2 - Resource Name List Table....55

18. Option 3 - Volume List....56

19. Option 3 - Volume Entry List (S is selected)....56

20. Option 3 - Volume Active Reserve List (A is selected)....57

21. Option 4 - Filter List....57

22. AUDIT MODIFY Command Options....58

23. AUDIT MODIFY Option L Output Example....58

24. AUDIT MODIFY Option L Output Example (GRS=NONE or GRS in Star)..... 59

25. AUDIT MODIFY Option T Example.....59

26. AUDIT MODIFY Option T=Major Example..... 59

27. AUDIT MODIFY Option V Example..... 60

28. AUDIT MODIFY Option V=Volser Example.....60

29. AUDIT Messages..... 62

30. AUDIT Abends.....62

31. GFLG Filter Example.....63

32. NAME Filter Example..... 63

33. Filter Coding Example..... 66

34. In-stream Procedures Variables.....68

35. PARM Supported by Report Programs.....69

36. JCL for Trace Report for Single MVS System Input..... 70

37. JCL for Trace Report for Multiple MVS Systems Input..... 71

38. volume reserve time reportJCLvolume reserve time reportJCL for Volume Reserve Report for Single MVS System Input..... 72

39. JCL for Volume Reserve Report for Multiple MVS Systems Input..... 74

40. reportsresource usage reportJCLresource usage reportJCL for Resource Usage Report for Single MVS System Input..... 76

41. JCL for Resource Usage Report for Multiple MVS Systems Input..... 78

42. Trace Report for Single System..... 82

43. Trace Report for Multiple MVS Systems..... 83

44. Volumes Reserve Time Report..... 83

45. ENQ/RESERVE Resources Report.....84

46. Reserve/Release Trace at Time of Max..... 85

47. Overview of the global resource serialization star complex..... 91

48. Overview of GQSCAN/ISGQUERY request for global resource data.....92

49. The Star Concept....96
50. Syntax of the Global Resource Serialization Record for IXCL1DSU....96
51. Specifying Global Resource Serialization Lock Structure in the CFRM Policy....97
52. Formula for Determining SIZE Parameter for the CFRM Policy....98
53. Syntax for the GRS= System Parameter....99
54. Star-relevant parameters for GRSCNFxx....99
55. Example Global Resource Serialization Star Parmlib Definition....101
56. D GRS Star Explanation....107
57. Fully-Connected Four-System Complex....115
58. Three-System Ring — SYS1 Failed....116
59. Global Resource Request Processing without Ring Acceleration....117
60. Global Resource Request Processing with Ring Acceleration....118
61. Links in a Sysplex....120
62. Ring-relevant parameters for GRSCNFxx....121
63. Ring Acceleration Configuration....123
64. Links in a Mixed Complex....126
65. Recovery Problems with a Partially-Connected Complex....128
66. CTC Link Failure — No Alternate Available....129
67. CTC Link Failure — Alternate Available....129
68. Ring Acceleration Configuration....134
69. Sample Complex Design and Definition Diagram....140
70. Sample Design and Definition Diagram for a Mixed Complex....141
71. D GRS Explanation (Complex Matches Sysplex)....149
72. Sample mixed complex configuration....151
73. D GRS Explanation (Mixed Complex)....154

74. Using D GRS to Analyze a Problem.....155

75. Two-System Ring with Link Failure..... 160

76. Global Resource Serialization Complex before Migration..... 164

77. Global Resource Serialization Complex after Migration..... 166

78. Example of Three System Sysplex..... 181

79. Current View of Contention..... 182

80. Contention for ENQ Resources.....182

81. Final State of Contention..... 183

82. Record structure for SMF 87 subtype 1..... 186

83. Record structure for SMF 87 subtype 2..... 190

84. Sample Output of GENQRESP..... 196

85. Measurement Program Example.....196

Tables

1. API Serialization in MVS..... 5

2. ISGENQ Services..... 7

3. SYSTEM inclusion RNL guidelines.....42

4. RESERVE conversion RNL Guidelines..... 42

5. SYSTEMS exclusion RNL Guidelines..... 44

6. GRSRNLxx Worksheet..... 46

7. GRS=STAR Relationship to PLEXCFG= Options..... 102

8. GRSCNF____ Definition (Sysplex Matches Complex)..... 125

9. Sample Complex Definition Plan..... 142

10. GRSCNF____ Definition (Mixed Complex).....143

11. Transmission Delays by Signalling Path and Device..... 168

12. RESMIL Values with Ring Acceleration — ACCELSYS(2)..... 170

13. RESMIL Values without Ring Acceleration..... 170

14. SMFRCD87 DSECT..... 186

15. SMF87REQ DSECT – Maps the Requester section..... 187

16. SMF87QSCAN DSECT – Maps the QSCAN data section..... 188

17. SMFRCD87 DSECT..... 190

18. SMF87REQ DSECT – Maps the Requester section..... 191

19. SMF87ENQ DSECT – Maps the ENQ data section.....192

20. ENQ RET to ISGENQ mapping..... 193

21. System commands for reserve status.....198

About this information

This information supports z/OS (5650-ZOS).

This document describes how to plan for a global resource serialization complex. Global resource serialization (GRS) allows users on multiple systems to serialize access to processing and logical resources, such as data sets on shared DASD volumes.

Who should use this information

This document is intended to help you with planning the use of global resource serialization.

Using this documentation requires you to be familiar with the MVS™ operating system and the services that programs running under it can invoke.

How this information is organized

This document assumes that you are familiar with MVS and have a thorough knowledge of your installation's data processing goals, issues, and available hardware and software.

The document deals with the process of planning — the thinking and decision making that you must do before you use global resource serialization to serialize access to global resources, such as data sets on shared DASD volumes. It does not describe such details as the complete syntax of the macros, parmlib members, and system commands that you use to carry out your plan. These details are described in other documents and this document refers to them, when appropriate. Problem determination information is also included in this document.

This document is divided into five parts:

- **Part one** describes the basic elements of global resource serialization.

  Chapter 1, "Introduction," on page 3 presents an overview of the benefits of global resource serialization. It describes the methods MVS uses to serialize requests for global resources. This information can help you to decide which method of global resource serialization will meet your needs.
  Chapter 2, "Selecting the data," on page 17 describes how global resource serialization processes requests for resources. This information can help you to decide how to handle various resources at your particular installation.
  Chapter 3, "Using the ENQ/RESERVE/DEQ monitor tool," on page 49 describes the global resource serialization monitor tool. This information can help you use the monitor tool to assist you in planning the Resource Name List (RNL) for global resource serialization implementation.

- **Part two** describes operations of the star complex:

  Chapter 4, "Star processing," on page 91 is a high-level description of how a global resource serialization star complex works.
  Chapter 5, "Planning a star complex," on page 95 provides information on how to design a global resource serialization star complex.
  Chapter 6, "Operating the star complex," on page 105 explains how to build and operate a global resource serialization star complex.

- **Part three** describes operations of the ring complex:

  Chapter 7, "Ring processing," on page 115 is a high level description of how a global resource serialization ring complex works.
  Chapter 8, "Designing a ring complex," on page 119 describes the basic principles of designing a global resource serialization ring complex.

Chapter 9, "Operating the ring complex," on page 147 describes how to build and operate a global resource serialization ring complex. It presents information on the procedures that operators need to run the complex.

Chapter 10, "Installing and tuning the complex," on page 163 describes step-by-step approaches to installing the global resource serialization ring complex. It identifies installation steps as well as planning tasks and long-term considerations. It also describes the storage requirements for global resource serialization and some actions you can take to tune the performance of your complex.

- **Part four** describes problem determination:

  Chapter 11, "Diagnosing global resource serialization," on page 175 describes strategies to aid in the diagnosis and correction of global resource serialization problems in a sysplex environment.

  Chapter 12, "Measuring response time," on page 195 describes how a program measures request response time from the perspective of the problem program issuing the request.

- **Part five** contains appendixes:

  Appendix A, "Data set ENQ contention monitor," on page 199 describes a sample application that determines if contention exists for a particular data set and sends a message to the TSO/E user who is causing that contention.

  Appendix B, "Recovery actions for global resource serialization," on page 203 describes the recovery actions for global resource serialization.

  Appendix C, "Accessibility," on page 217 describes the major accessibility features in z/OS.

How to use this information

Once you have decided to build a global resource serialization complex, there are three basic planning tasks:

1. Select the data to be shared
2. Design the complex
3. Provide effective operator procedures

All of these tasks are closely related, but the design and operation of a global resource serialization complex are especially closely related. Therefore, a good way to use this document is to read it quickly all the way through to become familiar with all of the planning you need to do for a global resource serialization complex. Then, use the information and the planning aids in the document to develop your installation's plan for the use of global resource serialization.

Where to find more information

Where necessary, this document references information in other documents, using shortened versions of the document title. For complete titles and order numbers of documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page xvii.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM® Documentation function
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS MVS Planning: Global Resource Serialization, SA23-1389-50
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Summary of changes

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

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument).

Summary of changes for z/OS MVS Planning: Global Resource Serialization for Version 2 Release 5 (V2R5)

The following content is new, changed, or no longer included in V2R5.

Changed

The following content is changed.

March 2023 Refresh

- "SMS" on page 36 is updated. (APAR OA50221, which also applies to z/OS V2R4, V2R3, and V2R2).

Summary of changes for z/OS MVS Planning: Global Resource Serialization for Version 2 Release 4

The following changes are made for z/OS Version 2 Release 4 (V2R4).

Changed

July 2020 refresh

- "Checking for ENQ contention problems" on page 180 and "Checking for latch contention problems" on page 185 are updated for Runtime Diagnostic actions.

Prior to July 2020 refresh

- Figure 64 on page 126 and Figure 2 on page 14 are updated in support of OA38230. GRS can now additionally manage FICON CTCs (FCTC), which run in extended mode.
- "System logger" on page 38 is updated for system logger enhancement to support single-system scope Couple Data Set types (LOGRY and LOGRZ) for GDPS K-system environments.

Summary of changes for z/OS MVS Planning: Global Resource Serialization for Version 2 Release 3

The following changes are made for z/OS Version 2 Release 3 (V2R3).

Changed

- Clarification was added about devices that require a synchronous reserve regardless of any SYNCHRES=NO specification. For more information, see "Understanding the synchronous RESERVE feature" on page 9.

- Information was added on when to use GRS RNL=NO. For more information see "Excluding requests from RNL processing" on page 24.

Part 1. Global Resource Serialization

Chapter 1. Introduction

In a multitasking, multiprocessing environment, resource serialization is the technique used to coordinate access to resources that are used by more than one program. When multiple users share data, a way to control access to that data is needed. Users that update data, for example, need exclusive access to that data; if several users tried to update the same data at the same time, the result would be a data integrity exposure – the possibility of incorrect or damaged data. In contrast, users who only read data can safely access the same data at the same time.

z/OS provides multiple ways of providing serialized access to data on a single or multiple systems, but global resource serialization is a fundamental way for programs to get the control they need and ensure the integrity of resources in a multisystem environment.

Because global resource serialization is automatically part of your system and is present during z/OS initialization, it provides the application programming interfaces that are used by the applications on your system. This is true even in the presence of vendor products.

This introduction contains the following topics:

- "Advantages of using global resource serialization" on page 3
- "How global resource serialization works" on page 3
- "Using an API to serialize resources" on page 5
- "ISGENQ macro" on page 6
- "RESERVE macro" on page 7
- "Understanding the synchronous RESERVE feature" on page 9
- "Issuing ISGQUERY and GQSCAN" on page 10
- "ISGADMIN macro" on page 10
- "Methods of serializing global resources" on page 12
- "Using IBM Health Checker for z/OS" on page 16

Advantages of using global resource serialization

In a global resource serialization complex, programs can serialize access to data sets on shared DASD volumes at the data set level. A program on one system can access one data set on a shared volume while other programs on any system can access other data sets on the same volume.

Because global resource serialization enables jobs to serialize their use of resources at the data set level, it can reduce contention for these resources and minimize the chance of an interlock occurring between systems. This ends the need to protect resources by job scheduling. Because global resource serialization maintains information about global resources in system storage, it does away with the data integrity exposure that occurs when there is a system reset while a reserve exists. A global resource serialization complex also allows serialization of shared logical resources — resources that might not be directly associated with a data set or DASD volumes. An ENQ with a scope of SYSTEMS might be used to synchronize processing in multisystem applications.

Combining the systems that access shared resources into a global resource serialization complex can solve the problems related to using the RESERVE macro. To understand how global resource serialization overcomes the major drawbacks of using the RESERVE macro, see "RESERVE macro" on page 7.

How global resource serialization works

Depending on the level of z/OS and XCF environment the systems are running in, sysplex or non-sysplex, a global resource serialization complex consists of one or more systems:

- Connected to each other in a **ring** configuration through:

  - XCF signalling paths, through CTC or coupling facility
  - Global resource serialization managed channel-to-channel (CTC) adapter
    - Systems in the same sysplex must use XCF signalling
    - Non-sysplex or monplex systems must use GRS-managed CTCs to communicate
- Connected to each other in a **star** configuration through:
  - XCF signalling paths, through CTC or coupling facility
  - A GRS coupling facility lock structure

Guideline: IBM recommends basic or parallel sysplex technology for resource sharing. Global resource serialization uses parallel sysplex technology to achieve significant scalability and performance benefits in its star mode. Even with the basic sysplex technology, global resource serialization provides the capability to dynamically change the resource name list (RNL) and provide superior recoverability to ring disruptions. This document uses the term complex because a global resource serialization complex is not the same as a sysplex.

Before looking at the ring and star complexes in more detail, you need to know how users serialize global resources through macros and more about global resource serialization in general.

Local and global resources

When an application uses a macro (ISGENQ, ENQ, DEQ, and RESERVE) to serialize resources, global resource serialization uses the RNL, various global resource serialization installation exits, and the scope on the macro to determine whether a resource is a local resource or a global resource.

A **local resource** is a resource requested with a scope of STEP or SYSTEM. It is serialized only within the system processing the request for the resource. If a system is not part of a global resource serialization complex, all resources (with the exception of resources serialized with the RESERVE macro), regardless of scope (STEP, SYSTEM, SYSTEMS), are local resources.

A **global resource** is a resource requested with a scope of SYSTEMS or SYSPLEX. It is serialized among all systems in the complex.

The ISGENQ, ENQ, DEQ, and RESERVE macros identify a resource by its symbolic name. The symbolic name has three parts:

Major name:
qname

Minor name:
rname

Scope:
The scope is one of the following:

STEP
SYSTEM
SYSTEMS
SYSPLEX

Example: On the ISGENQ, ENQ or DEQ macro, a resource could have a symbolic name of `APPL01,MASTER,SYSTEM`. The major name (qname) is APPL01, the minor name (rname) is MASTER, and the scope is SYSTEM. Global resource serialization identifies each resource by its entire symbolic name. That means a resource that is specified as A,B,SYSTEMS is considered a different resource from A,B,SYSTEM or A,B,STEP because the scope of each resource is different.

Note: In addition, the ISGENQ macro uses the scope of SYSTEMS and the scope of SYSPLEX interchangeably, so a resource with the symbolic name of A.B.SYSTEMS would be the same as a resource with the name A.B.SYSPLEX.

GRS users define the QNAME/RNAME of resources. As such, the user defines their usage and meaning. See the ENQ/DEQ Summary table in *z/OS MVS Diagnosis: Reference* for definitions of the used ENQ resources (QNAME/RNAME). It is helpful for the users to know the usage when observing ENQ/RESERVE activity and contention.

In general, global resource serialization identifies a resource with a scope of STEP or SYSTEM as a local resource, and a resource with a scope of SYSTEMS as a global resource. Because users of some previous versions of MVS could serialize access to resources across multiple systems only through a reserve, whether the user specified SYSTEM or SYSTEMS on the ENQ macro did not affect resource serialization. Therefore, your installation might have programs that specify ENQ with a scope of SYSTEM for resources that you would want global resource serialization to identify as global resources. You might also have programs that specify ENQ with a scope of SYSTEMS for resources that you would want global resource serialization to identify as local resources.

To ensure that resources are treated as you want them to be without changes to your applications, global resource serialization provides three resource name lists (RNLs):

- The **SYSTEM inclusion RNL** lists resources requested with a scope of SYSTEM that you want global resource serialization to treat as global resources.
- The **SYSTEMS exclusion RNL** lists resources requested with a scope of SYSTEMS that you want global resource serialization to treat as local resources.
- The **RESERVE conversion RNL** lists resources requested on the RESERVE macro that you want global resource serialization to suppress the reserve.

  Note: For the correct use of the conversion RNL, see "RESERVE conversion" on page 31.

By placing the name of a resource in the appropriate RNL, you can cause global resource serialization to process it as you want. The RNL enables you to build a global resource serialization complex without first having to change your existing programs, though you might at some time want to change these programs.

Deciding how to use the RNLs to define your resource processing needs is a major part of the planning for a global resource serialization complex; the process is described in detail in Chapter 2, "Selecting the data," on page 17. To make the most effective use of the RNLs, you also need to understand how a global resource serialization complex processes requests from an application programming interface (API).

Using an API to serialize resources

You can choose from the following macros to serialize access to global resources:

- "ISGENQ macro" on page 6
- ENQ macro
- "RESERVE macro" on page 7

The ISGENQ macro provides the most flexibility because it allows you to obtain, change, release, and test resources while supporting the following modes:

- AMODE 31 and 64
- Primary or AR ASC mode

Guideline: As of z/OS V1R6, IBM recommends using the ISGENQ service for unauthorized serialization requests, although the ENQ, DEQ, and RESERVE interfaces will continue to be supported.

The following table shows how to serialize access to resources at three levels:

Table 1. API Serialization in MVS

To serialize access to a resource:	Macro	Scope
Within an address space	ISGENQ, ENQ and DEQ	STEP
Within a single system	ISGENQ, ENQ and DEQ	SYSTEM

Table 1. API Serialization in MVS (continued)

To serialize access to a resource:	Macro	Scope
Across a sysplex	ISGENQ	SYSTEMS or SYSPLEX
	ENQ, RESERVE and DEQ	SYSTEMS

Note: To serialize access to resources across sysplexes and between VM and z/OS establish a reserve through the RESERVE macro or ISGENQ macro with the RESERVEVOLUME=YES parameter. Because this serialization involves I/O to the entire volume; you lose much of the granularity of the qname and rname resource abstraction.

As of z/OS V1R6, use the ISGENQ macro to obtain, change, release resources, test an obtain request, and reserve a DASD volume.

For detailed information about using the ISGENQ macro:

- For the ISGENQ macro description, see "ISGENQ macro" on page 6.
- For the ISGENQ macro defaults, see *z/OS MVS Programming: Assembler Services Reference IAR-XCT*.

The RESERVE macro obtains access to a resource and the DEQ macro frees the resource. The RESERVE macro, by default, has a scope of SYSTEMS. The corresponding DEQ macro must specify a scope of SYSTEMS.

The ENQ macro with a scope of SYSTEMS obtains access to a resource. The DEQ macro frees the resource when you specify a scope of SYSTEMS.

Guideline: Some applications may require specific RNL changes for ENQs or related products that are installed.

Note:

1. A scope of SYSTEMS indicates that MVS is to protect the resource across multiple systems.
2. Using the ENQ macro with a scope of SYSTEMS requires a multisystem approach to resource serialization. Your installation combines the systems that must serialize access to resources in a global resource serialization complex. For more information, see "Advantages of using global resource serialization" on page 3.

ISGENQ macro

The ISGENQ macro combines the serialization abilities of the ENQ, DEQ, and RESERVE macros. ISGENQ supports AMODE 31 and 64 in primary or AR ASC mode. With the ISGENQ macro you can:

- Obtain a single resource or multiple resources with or without associated device reserves.
- Change the status of a single existing ISGENQ request or multiple existing ISGENQ requests.
- Release serialization from a single or multiple ISGENQ requests.
- Test an obtain request.
- Reserve a DASD volume.

ISGENQ enables you to specify that USERDATA can also be associated with the ENQ. ISGQUERY can retrieve the USERDATA, which allows the USERDATA to be a filter on the query.

The following table describes the ISGENQ requests.

Table 2. ISGENQ Services	
REQUEST	**SERVICE**
REQUEST=OBTAIN	• Obtain a single resource or multiple resources with or without associated devices. • Test an obtain request.
REQUEST=CHANGE	Change the status of an ISGENQ request from shared to exclusive.
REQUEST=RELEASE	Release (dequeue) from a single resource or multiple resources.

ISGENQ includes parameter checking to ensure that authorized callers use authorized qnames. See "Authorized qnames" on page 18 for a list of authorized qnames.

ISGENQ uses the ENQTOKEN parameter to match a CHANGE request (ENQ RET=CHNG) or RELEASE request (DEQ) with the original OBTAIN request (ENQ). This association helps to avoid integrity problems with regard to dynamic changes of the ISGNQXIT and ISGNQXITFAST installation exits.

For additional information about the ISGENQ macro, see *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*.

RESERVE macro

The RESERVE macro serializes access to a resource (a data set on a shared DASD volume) by obtaining control of the volume on which the resource resides to prevent jobs on other systems from using any data set on the entire volume. There is an ENQ associated with the volume RESERVE; this is required because a volume is reserved by the z/OS image and not the requester's unit of work. As such, the ENQ is used to serialize across the requesters from the same system.

If you are using the GDPS/PPRC product or the GDPS/PPRC HyperSwap® Manager product, see the following publications for GDPS® specific requirements:

- *GDPS/PPRC Installation and Customization Guide, ZG246703*
- *GDPS/PPRC HyperSwap Manager Installation and Customization Guide, ZG246746*

Potential problems when using RESERVE

Serializing access to data sets on shared DASD volumes by means of a reserve generally protects the resource, but it creates several critical problems. Below are explanations with some examples to help you better understand the problems associated with RESERVE processing:

- **Lack of granularity:** When a task on one system has issued a RESERVE macro to obtain control of a data set, no programs on other systems can access any other data set on that volume. Protecting the integrity of the resource means delay of jobs that need the volume, but do not need that specific resource, as well as delay of jobs that need only read access to that specific resource. These jobs must wait until the system that issued the reserve releases the device. Even if the owning task only needs to read the data, no programs on other systems can read the data.
- **Starvation:** From the hardware control unit perspective, only one system can RESERVE the volume at one time. z/OS's RESERVE processing holds onto the volume RESERVE when there are outstanding RESERVE requests from the local system. As such, there is no RESERVE fairness across z/OS images and a single system can monopolize a shared device when it encounters multiple reserves for the same device. No other system can use the device until the reserving system releases the device.

  Example: S1 (system 1) issues a RESERVE for a resource with qname,rname (Q1,R1) on V1 (volume 1). S2 (system 2) then issues a RESERVE for (Q2,R2) on that same volume. S1 then issues a RESERVE for resources (Q3,R3), (Q4,R4), (Q5,R5), (Q6,R6),... (Q99,R99), all on V1, which it already owns. It does not

matter that S2 made the second request. It does not matter if the majority of the RESERVE requests are DEQed. S2 cannot access that volume until there are no outstanding RESERVEs on S1 for V1.

- **Volume hierarchy deadlock:** The installation must manage the full list of all volumes potentially reserved and pay particular attention to any application that requires more than one at a time. Regardless of the underlying qname and rname resource names, mixing the order of reserves on volumes can cause an interlock (also called a deadlock), which is an unresolved contention for use of a resource.

  Example: S1 requires a resource with qname,rname of (Q1A,R1A) on V1, then (Q1B,R1B) on V2. S2 requires resource (Q2A,R2A) on V2, then (Q2B,R2B) on V1. Even though the qname,rname pairs are all completely different, the order of reserved volumes have been reversed. If these four resource requests had been converted to global ENQs, there would be no contention at all. If they are left in the current order as reserves, they can deadlock. S1 reserves V1, S2 reserves V2, and neither system will be able to complete its second reserve. Reserve processing requires that the installation know the order of all reserve requests from all applications.

- **Asynchronous I/O deadlock:** The actual first I/O to the volume can occur asynchronously to the RESERVE service completing successfully. Even if the installation maintains the volume hierarchy, it is possible for a deadlock to occur. To prevent this from happening, use the SYNCHRES=YES parameter, an option in GRSCNFxx that is dynamically adjustable through the SETGRS system command. For more information, see “Understanding the synchronous RESERVE feature” on page 9.

  Example: Neither system S1 or system S2 have the SYNCHRES option on. There are two shared volumes, V1 and V2. An application running on both systems requires resources on both V1 and V2, and issues a RESERVE for them in that order. The application gets control back from Global Resource Serialization on both images before any I/O occurs; as if both S1 and S2 have exclusive control of both resources. However, the race condition on those first I/O requests gives resource V1 to system S1 and resource V2 to system S2. Due to S1's I/O request on V2, and S2's I/O request on V1, the application becomes locked out on both images. If the installation had SYNCHRES activated on S1 and S2, this scenario would not occur.

- **Double serialization contention and deadlock:** Global Resource Serialization processes a corresponding ENQ request with every RESERVE. When Global Resource Serialization is in ring or star mode, that ENQ defaults to being scope=systems, causing unnecessary contention. In certain scenarios, the double serialization can also deadlock. Global Resource Serialization provides the installation with the choice of converting the RESERVE request into being only a global ENQ, or alternately demoting the corresponding ENQ to scope=system and keep the RESERVE. This is typically done through RNL processing.

  Example: S1 issues a RESERVE with qname,rname (Q1,R1) on V1, gaining both the global ENQ and the reserve. S2 issues a RESERVE for (Q2,R2), also on V1. The application obtains the global ENQ for (Q2,R2), but not the reserve. S1 then issues a RESERVE with qname,rname (Q2,R2) for V1. Although S1 already owns the entire volume, the global ENQ for (Q2,R2) is already held. This is not a volume hierarchy problem because there is only one volume involved. If both resources were made scope=system, or both converted to being only global ENQ requests, this scenario would not occur.

- **Data integrity exposure upon system reset:** A system reset while a reserve exists terminates the reserve. The loss of the reserve can leave the resource in a damaged state if, for example, a program had only partly updated the resource when the reserve ended. This type of potential damage to a resource is a data integrity exposure.

- **Compounding problems:** Often when an installation encounters one of the problems listed above, it compounds them into several, and any deadlocks from reserved volumes can expand to an increasing number of tasks.

Solving the RESERVE problems

One major purpose of global resource serialization is to eliminate the need to protect data sets on shared DASD volumes by issuing a RESERVE macro that causes a reserve of the entire volume. Some installations make it their policy to never issue a reserve.

You can use the following commands to determine which system is holding a RESERVE:

- DISPLAY U
- DEVSERV PATH

You can also issue the D GRS,DEV=ddd command, to see if this system holds reserve on a device.

To use these commands, see *z/OS MVS System Commands*.

When considering the conversion of reserves to global ENQs, there are some restrictions:

- **Outside the complex:** Do not convert a reserve for a resource on a volume if any system in your complex shares the volume with a system that is not part of the complex (or is part of another complex). This case can be particularly relevant for star mode where the global resource serialization complex must equal the parallel sysplex, or if the global resource serialization complex shares resources with another type of system. Some installations choose data transfer and duplication over the use of reserves here.
- **Inconsistent qnames and rnames:** Do not convert the reserve for a resource when different systems in the complex use different names for that resource. This inconsistency can occur when the resource name includes system-dependent information, such as control block addresses. Global resource serialization assumes that the different names represent different resources, and it cannot protect a single resource known by different names. Programs could be modified to use a consistent naming convention, either as reserves to be converted, or simply as global ENQs.
- **One reserve, multiple resources:** An application might be using a RESERVE against multiple resources while another application uses another form of serialization, such as multiple ENQs for resources that all exist on the same volume (this is not recommended). The first application could take advantage of owning the entire volume. This is similar to the inconsistent qname and rname restriction. Before converting this RESERVE, the installation must understand all the resources it serializes and set up consistent qname and rname conventions for all applications to share it.
- **Ring performance:** Do not convert the reserve on a ring-mode global resource serialization complex when it would adversely affect system performance. Your installation might have applications where dependencies on quick access to a resource outweigh the additional availability resulting from converting the reserve. Because of the lack of granularity and starvation issues of reserve processing, this pertains to resources that are obtained and released quickly. This restriction is not applicable to star mode. See “RNL candidates” on page 42 for basic recommendations.

The ENQ/DEQ/RESERVE monitor tool can help identify current RESERVEs on a system. For more information, see Chapter 3, “Using the ENQ/RESERVE/DEQ monitor tool,” on page 49.

Tip: To collect the same data but with less impact to the ENQ/DEQ path length, instead use SMF 87 subtype 2 records. See “Using SMF 87 records” on page 185 for more information.

After the installation has chosen which RESERVEs to convert, it is typically accomplished through RNL processing. For more information, see Chapter 2, “Selecting the data,” on page 17.

Many installations use job scheduling instead of a reserve to protect the integrity of data sets on shared DASD volumes. That is, jobs that need the same data set are assigned to the same job class and run at different times. Protecting resources by job scheduling works, but it complicates operations, might increase job turnaround time, and might reduce the additional workload your installation can handle.

Understanding the synchronous RESERVE feature

The synchronous RESERVE feature of global resource serialization allows an installation to specify whether a system obtains a physical hardware reserve before it returns control to the program that issued the ISGENQ or RESERVE macro. This feature is known as the SYNCHRES option. The SYNCHRES option helps prevent lockouts by eliminating the window between the time that the RESERVE macro is issued and the time that the first I/O operation to the device is successfully started.

When SYNCHRES=NO, the system places a reserve channel-command word (CCW) on the front of the next executed channel program. When SYNCHRES=YES, the system ensures that the reserve CCW is issued and successful before it returns. As of z/OS V1R8, GRS indicates to IOS to not use an internal timeout value to time the GRS SYNCHRES I/O request. Instead, the device specified timeout value is

used. So, if needed, you can change the Missing Interrupt Handler (MIH) timeout value for the device to control the time that GRS uses.

You can activate the SYNCHRES option through either the GRSCNFxx parmlib member or the SETGRS operator command. The GRSDEF statement of GRSCNFxx contains the SYNCHRES (YES or NO) parameter.

Starting with z/OS V1R6, the default value for SYNCHRES is YES. This default setting is a change from previous releases of z/OS, where the default value for SYNCHRES is NO. During normal system operation, you can modify the setting of SYNCHRES by issuing the SETGRS command. You can activate SYNCHRES by issuing SETGRS SYNCHRES=YES and deactivate it by issuing SETGRS SYNCHRES=NO. The ISGENQ macro can override the system SYNCHRES setting on any request.

The GRSCNFxx does not set the value for the entire global resource serialization complex. When you issue either the SETGRS command or the GRSCNFxx, you change only the SYNCHRES value for that system.

An ISGENQ requester can override the installation default for the request by specifying whether the reserve is to be done synchronously or not. In addition, z High Performance FICON (zHPF) devices that operate I/O in Transport Mode require that a reserve is done synchronously regardless of any SYNCHRES=NO specification.

Issuing ISGQUERY and GQSCAN

The ISGQUERY and GQSCAN macros enable you to obtain information about the status of each resource that is identified to global resource serialization, including information about the tasks that have requested the resource.

Reference documents

Complete information about using ISGQUERY and GQSCAN is found in:

- *z/OS MVS Programming: Assembler Services Guide*
- *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*
- *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*

ISGADMIN macro

You can change the maximum number of concurrent ENQ, ISGENQ, RESERVE, GQSCAN and ISGQUERY requests. This is useful for subsystems that have large numbers of concurrently outstanding ENQs, query requests, or both, such as CICS® and DB2®.

- To change an address space authorized or unauthorized request limit for a particular address space, use the ISGADMIN service in *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*
- To change system-wide default request limits for all address spaces, use the ENQMAXA and ENQMAXU parameter in:
  - SETGRS command in *z/OS MVS System Commands*.

    Restriction: Only use the SETGRS command when a subsystem or application legitimately requires a high maximum and there is no other work around. To specifically set the requirements you must change the subsystem or application to use the ISGADMIN service. Relying on the system wide defaults can reduce system protection against run away applications.
  - GRSCNFxx parmlib in *z/OS MVS Initialization and Tuning Guide*.

Limiting global resource serialization requests

Global resource serialization allows an installation to share symbolically-referenced resources between tasks. The installation must write its applications to share DASD, and is responsible for establishing the protocol to ensure data set integrity. A global resource serialization request is an ISGENQ (ENQ) or RESERVE request that causes an element to be added to any queue in the global resource serialization queue area.

If you build a multisystem sysplex, you automatically build a global resource serialization complex with it. As the systems join the sysplex, the systems join the global resource serialization complex. See *z/OS MVS Setting Up a Sysplex* for information about a sysplex and how your installation can use global resource serialization.

Use ISGQUERY or GQSCAN to obtain the status of resources and requesters of resources.

- The ISGQUERY macro is described in *z/OS MVS Programming: Assembler Services Reference IAR-XCT*.
- The GQSCAN macro is described in *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*.

Understanding concurrent request counts: Each ENQ, DEQ, RESERVE and ISGENQ request typically causes an update to either the concurrent unauthorized or authorized count, depending on the authority of the requester. An ENQ or ISGENQ REQUEST=OBTAIN request will cause the respective count to increment.

- Typically an ENQ, RESERVE or ISGENQ REQUEST=OBTAIN request will cause the respective count to increment.
- ENQ RET=CHNG, RET=TEST, ISGENQ REQUEST=CHANGE and REQUEST=OBTAIN TEST=YES do not cause the count to increment
- DEQ and ISGENQ REQUEST=RELEASE cause the count to decrement.
- GQSCAN and ISGQUERY REQINFO=QSCAN can also cause an increment of the count, but only when there is insufficient answer area space for the results and a resume token is returned.
- GQSCAN and ISGQUERY REQINFO=QSCAN requests are always added to the unauthorized count — even when the service issuer is authorized.

To determine various related ENQ statistics regarding authorized and unauthorized requesters, use ISGQUERY REQINFO=ENQSTATS by ASID to query:

- Peak address space count of concurrent ENQs
- Current address space count of concurrent ENQs
- Current address space specific concurrent ENQ maximums
- Current system wide concurrent ENQ maximums
- Largest peak address space count of concurrent ENQs across the system
- ASID of the address space with that largest peak.

There are thresholds for the number of requests that global resource serialization will accept from a single address space. These thresholds prevent one address space (job, started task, or TSO/E user) from generating enough concurrent requests to exhaust the resource queue area.

If these values do not suit your installation's needs, you can change them by using the ISGADMIN macro. The ISGADMIN service can raise subsystem-specific maximums for unauthorized and authorized requesters. For unauthorized callers, the threshold value supplied by IBM is 16,384 (X'4000'). For authorized callers, the threshold value supplied by IBM is 250,000 (X'3D090'). The threshold values apply to an individual system in your global resource serialization complex, so different systems can have different limits.

The system does not reset the values to the system defaults when a job or the cross memory owning TCB terminates. As such, it is the responsibility of the ISGADMIN REQUEST=SETENQMAX invoker to issue a corresponding ISGADMIN REQUEST=RESETENQMAX or REQUEST=SETENQMAX to the previous values when control is given up and the higher values are no longer required. The system does not stack ISGADMIN SETENQMAX requests. An ISGADMIN REQUEST=RESETENQMAX can undo a previous setting. When the higher values are no longer required, issuing an ISGQUERY REQINFO=ENQSTATS can obtain the values to be restored.

In an urgent situation, you can use the SETGRS command to set the limits overall. Applications that require higher values must take the overall system requirements into consideration and properly manage limit requirements through the ISGADMIN service.

For more information about ISGADMIN, see

- *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*

As each ISGENQ, ISGQUERY, ENQ, RESERVE, and GQSCAN request is received, global resource serialization determines if increasing the count will exceed the allowed threshold value. If the threshold will be exceeded, GRS will issue:

- A return code of X'0C01' for ISGENQ OBTAIN requests to indicate the limit of concurrent resource requests is reached.
- A return code of X'0C06' for ISGQUERY requests.
- An abend X'538' for an unconditional ENQ or RESERVE
- A return code of X'18' for a conditional ENQ or RESERVE
- A return code of X'14' for GQSCAN requests that do not fit into the caller's buffer; the requests are not queued and a TOKEN is not provided.

Before an abend, messages ISG368E and ISG369I monitor the address spaces that are approaching the maximum amount. If a particular subsystem requires more ENQs than normal, use ISGADMIN service to raise the maximum ENQ amount of that subsystem for unauthorized and authorized requesters.

ENQ or RESERVE requests from authorized callers use the MASID (matching ASID) and MTCB (matching TCB) parameters to allow a further conditional control of a resource. One task issues an ENQ or RESERVE for a resource specifying a matching ASID; if the issuing task does not receive control, it is notified whether the matching task has control (which allows the issuing task to use the resource even though it could not acquire the resource itself). This process requires serialization between the issuing and requesting tasks.

Methods of serializing global resources

A global resource serialization complex is one or more z/OS systems that use global resource serialization to serialize access to shared resources (such as data sets on shared DASD volumes). In this document, the term *sysplex* refers to a multisystem sysplex. It is possible, in a ring configuration, to have systems in a global resource serialization complex that are not in the sysplex. For the purposes of global resource serialization, these systems should be treated as if they are not part of a multisystem sysplex. The ring and star systems cannot coexist in a global resource serialization complex.

Every system in a sysplex is a member of the same global resource serialization complex. Global resource serialization is required in a sysplex because components and products that use sysplex services need to access a sysplex-wide serialization mechanism. For more information on sysplexes, see the following topics:

- Multisystem sysplex configuration without an existing global resource serialization complex
- Multisystem sysplex configuration with an existing global resource serialization complex

in *z/OS MVS Setting Up a Sysplex*.

You can set up either of the following complexes for global resource serialization:

- A **star** - In this type of complex, all of the systems in the sysplex must match the systems in the complex.
- A **ring** - In this type of complex, either:
  - The systems in the sysplex are the same as the systems in the complex **(sysplex matches complex)**. In a ring, you derive greater benefits if the sysplex and the complex match.
  - At least one system in the complex is outside of the sysplex **(mixed complex)**. A mixed complex often exists temporarily as your installation migrates one system at a time into a star complex.

The ring is IBM's original design for serializing global resources and it still has value in today's sysplex environment. But the star offers many advantages over the ring complex. “Which method to choose” on page 15 offers guidance on which method best suits your present environment and future needs.

The ring

The ring consists of one or more systems connected to each other by communication links. The links are used to pass information about requests for global resources from one system to another in the complex. Requests are made by passing a message or token, called the ring system authority message (RSA-message), between systems in a round-robin or ring fashion. There is one RSA token in a global resource serialization ring complex.

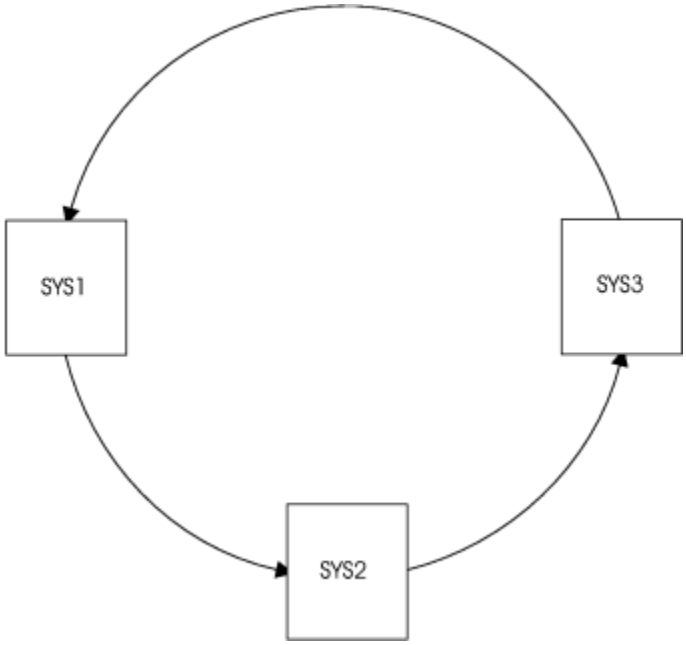


Figure 1. The Conceptual View of the Ring Complex

When a system receives the RSA, it inserts global ENQ and DEQ information, and passes it along to the next system to copy. It also makes a copy of the global ENQ/DEQ information that was inserted by other systems. When the RSA returns to the originating system, it knows that all other systems have seen its request, so the request is removed. The system can now process those requests by adding them to the global resource queues, and can now determine which jobs own resources, and which jobs must wait for resources owned by other jobs. Each system takes this information and updates its own global queues.

For various reasons, such as system or link failure, not all systems in a complex might be actively using global resource serialization. The ring is made up of all systems in a global resource serialization complex that are **actively** using global resource serialization to serialize access to global resources.

Systems participating in a ring are dependent on each other to perform global resource serialization processing. For this reason, the following areas are all adversely affected as the number of systems in a ring increase.

- Storage consumption
- Processing capacity
- Response time
- CPU consumption
- Availability and recovery time

Figure 2 on page 14 shows the basic elements of a four-system global resource serialization ring complex. These elements include the systems, the communication links, and any shared resources (such as DASD) specified by the installation.

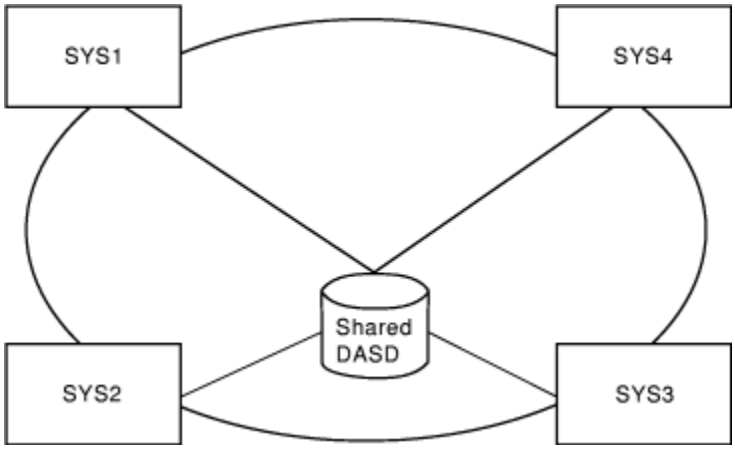


Figure 2. Global Resource Serialization Ring Complex

The systems in the complex can use global resource serialization to serialize access to global resources, such as data sets on shared DASD volumes. Global resource serialization in a ring complex uses the links (either XCF paths or the dedicated communication links) to communicate information about global resources from one system in the complex to another.

- In a ring sysplex, all the links shown are XCF signalling paths. XCF manages the communication links that global resource serialization uses between systems in a sysplex. XCF can use IBM Enterprise Systems Connection (ESCON) channels operating in CTC mode, and list structures in a coupling facility as communication links.
- Links between a system in a sysplex and one outside of the sysplex, as well as the links that are between two systems that are both outside of the sysplex, can be FICON® (FCTC), parallel CTC adapters dedicated to global resource serialization or an ESCON channel operating in basic mode (ESCON BCTC links). Global resource serialization does not use ESCON links that are defined as SCTC.

In this document, “link”, “path”, and “CTC link” can mean a parallel CTC adapter, an ESCON channel operating in basic mode, or if appropriate, a list structure in a coupling facility.

For more information on CTC definitions:

- *z/OS HCD User's Guide*
- *z/OS HCD Planning*

The star

The **star** method of serializing global resources is built around a coupling facility in which the global resource serialization lock structure, ISGLOCK, resides. In a star complex, when an ISGENQ, ENQ, DEQ, or RESERVE macro is issued for a global resource, MVS uses information in the ISGLOCK structure to coordinate resource allocation across all systems in the sysplex.

Figure 3 on page 15 shows the basic elements of a global resource serialization star complex. These elements include the systems, the coupling facility, and any shared resources (such as DASD) specified by the installation.

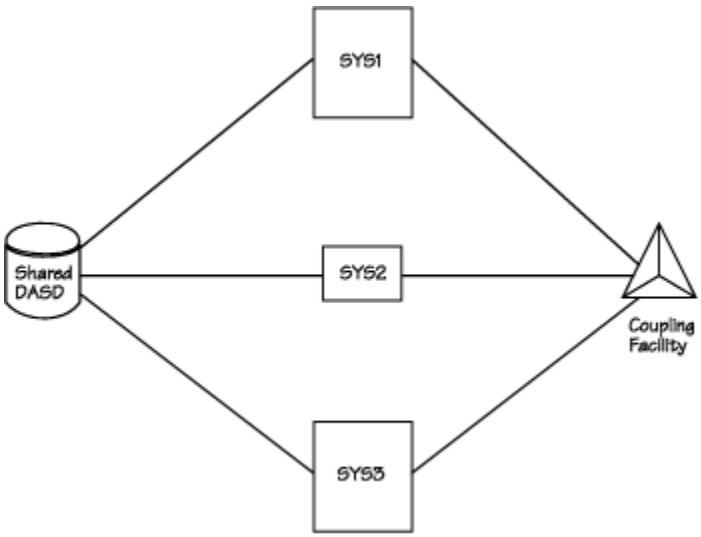


Figure 3. A Conceptual View of the Star Complex

In a star complex, global resource serialization requires the use of the coupling facility. The coupling facility makes data sharing possible by allowing data to be accessed throughout the sysplex by ensuring the data remain consistent among each system in the sysplex. The coupling facility provides the medium in global resource serialization to share information about requests for global resources between systems. Communication links are used to connect the coupling facility with the systems in the global resource serialization star sysplex. For more information on communication links, see *z/OS MVS Setting Up a Sysplex*.

Star advantages

- Processing Capacity:

  The processing capacity does not diminish as the number of systems in the complex increase. This is because all requests for global resources are handled by the coupling facility; not by passing a token between each system in the complex before a request is satisfied. The requests are processed by the coupling facility as they are received, therefore the processing capacity is improved.

- Response Time:

  Response time is improved in a star as each request for a resource that is not in contention can be completed with only two signals, rather than having to pass the RSA-message to each system in the complex.

  Also, in a star complex, the central processing unit (CPU) overhead required to process an ENQ or DEQ request is limited to the system on which the request originated, the coupling facility, and the system chosen to be the global contention manager by XES (if one is needed). Therefore, the total processing time consumed across a star complex will be less than that consumed by a ring complex.

- Availability and Recovery:

  Availability and recovery time is improved for the star because the systems that make up the complex are not dependent on each other, as they are in the ring. The complex does not have to alter processing to adjust to topological changes due to a system joining or leaving the complex. When a system joins or leaves the complex, queue manipulation for resource requests are distributed around the complex, rather than having to be repeated for every request on each system. These changes and adjustments are handled by global resource serialization and XES. Your only requirement is to determine which resource owned by a failing system is in contention and reassign it to a new system.

Which method to choose

Your choice of method, ring or star, depends primarily on hardware requirements.

- The star complex hardware requirements:

  - Coupling facility
  - Coupling facility links
- The ring hardware requirements:
  - Communications links, XCF sysplex, or both

It is also important the consider the number of systems in the global resource serialization complex, and what performance you are experiencing. IBM recommends that:

- New sysplexes start out using the star method for serializing global resources.
- Complexes that require sharing the MVS systems that are not members of a sysplex require using the ring.
- Existing complexes experiencing performance problems should migrate to the star.

Using IBM Health Checker for z/OS

The IBM Health Checker for z/OS includes several global resource serialization checks. This section gives a brief overview of the checks. For installation overrides of IBM parameters and detailed information, see Global resource serialization checks in *IBM Health Checker for z/OS User's Guide*.

The following checks examine the configuration and tuning of your global resource serialization complex against best practices.

GRS_CONVERT_RESERVES
This check looks for the recommended best practice of possibly converting all RESERVEs. Converting RESERVEs to global ENQs can help avoid deadlocks and improve reliability, availability, and serviceability. Note that some RESERVEs should not be converted. For more information on converting RESERVES, see "RESERVE conversion" on page 31.

GRS_EXIT_PERFORMANCE
This check examines exit points. The use of certain global resource serialization dynamic exits can negatively impact system performance. In some cases, removing an exit module or changing it to use a different exit point can help improve performance. For more information on dynamic exits, see *z/OS MVS Installation Exits*.

GRS_GRSQ_SETTING
This check examines the GRSQ setting and is only applicable in global resource serialization STAR mode. The recommended GRSQ setting, GRSQ=CONTENTION, produces smaller dump sizes and reduces the amount of time needed to create the dump.

GRS_MODE
This check determines if global resource serialization is running in star or ring mode. A star configuration is recommended for advantages in availability, real storage consumption, processing capacity, and response time. See "Star advantages" on page 15.

GRS_RNL_IGNORED_CONV
This check examines the RESERVE conversion list for entries that always supersede entries in the SYSTEMS exclusion list. For example, an installation expected a RESERVE to be converted into a global ENQ, but instead it is processed as a RESERVE and local ENQ because an entry is found in the SYSTEMS exclusion list first. The check generates a list of problem entries.

GRS_SYNCHRES
This check examines global resource serialization synchronous RESERVE processing. Enabling global resource serialization synchronous RESERVE processing can prevent deadlock conditions. See "Understanding the synchronous RESERVE feature" on page 9.

Chapter 2. Selecting the data

This contains the following topics:

- "Data set naming conventions" on page 18
- "Dynamic exits and RNL processing" on page 20
- "Request processing sequence" on page 20
- "RNL processing" on page 21
- "RESERVE conversion" on page 31
- "Changing the RNL" on page 32
- "RNL defaults" on page 33
- "RNL candidates" on page 42
- "Defining the RNLs" on page 45

Restrictions: Some applications are not affected by the resource name lists (RNLs) because they do not require GRS for managing ENQ beyond a single system. Therefore, if you need to, or prefer to use an alternate method of serializing global resources in a sysplex, you can specify GRSRNL=EXCLUDE on the GRSRNL system parameter. It is important to remember that EXCLUDE must be used with caution. When you specify GRSRNL=EXCLUDE, note the following facts:

- Only resources identified as SCOPE=SYSTEMS,RNL=NO on the ENQ macro are still managed by global serialization in the complex.
- You cannot specify the SET GRSRNL=xx command.
- A sysplex-wide outage is required to switch to an alternate method of serializing global resources (with only one exception, for more information, see "Migrating from GRSRNL=EXCLUDE to a set of RNLs" on page 32).

The rest of this topic assumes that you have not specified GRSRNL=EXCLUDE.

To ease migration to protecting resources as global resources, global resource serialization provides three RNLs:

- SYSTEM inclusion RNL
- SYSTEMS exclusion RNL
- RESERVE conversion RNL

The RNLs allow you to change resource scopes and convert reserves. IBM supplies default RNLs, shown in Figure 10 on page 34. You can modify the contents of the RNLs to define the resource serialization requirements of your installation. You can also use ISGNQXIT — ISGENQ / ENQ / DEQ Installation Exit and ISGNQXITFAST — Fast ISGENQ / ENQ / DEQ Installation Exit to indicate that global resource serialization is to bypass RNL processing for SYSTEM and SYSTEMS requests. This topic contains information you can use to select the resource names to place in each RNL.

Rule: A major reason for carefully planning the contents of the RNLs is that the RNLs installed in all systems in the complex must be exactly the same. During its initialization, global resource serialization checks to make sure that the RNLs on each system are indeed identical. If they are different, global resource serialization does not allow the system to join the complex.

When the sysplex matches the complex, you can use the SET GRSRNL command to change the RNLs dynamically. The procedures for changing RNLs for the ring and star complex are the same. For more information see "Changing the RNL" on page 32.

Naming resources in the appropriate RNL enables you to build a global resource serialization complex without modifying any existing programs. Therefore, your decisions about resources will probably focus on two questions:

1. What are my installation's short-term goals for the global resource serialization complex? Or, what must we do with the RNLs to get benefits from the complex as quickly as possible?
2. What are my installation's long-term goals for the global resource serialization complex? Or, what changes might we make to existing programs and procedures to make resource serialization more efficient?

Answering the first question requires you to analyze your use of data sets on shared DASD volumes to select the resources that are causing contention problems. As you do this analysis, focus on resources that are known to cause problems, such as resources that are frequently involved in interlocks. Consider whether a shared DASD volume accessed by the complex must also be accessed by systems outside the complex. Answering the second question might require you to analyze the data set naming conventions at your installation.

You must also understand how global resource serialization uses the resource names in the RNLs and what suggestions and recommendations exist for various situations.

Once you have completed your plan, see *z/OS MVS Initialization and Tuning Reference* for information on defining your RNLs in the GRSRNLxx parmlib member.

Data set naming conventions

Using global resource serialization effectively requires an established installation-wide convention for naming data sets, especially those on shared DASD volumes. If your installation does not have a standard convention for naming data sets, you should probably define and implement a standard before trying to use a global resource serialization complex to serialize access to global resources.

Restriction: Some data set naming conventions, such as one that relies heavily on system-dependent information like virtual storage addresses, do not work well with global resource serialization. If your installation has such a standard, you might want to redefine it as part of your long-term planning. A data set naming convention that works well with global resource serialization can avoid both very long RNLs and frequent changes to the RNLs.

In general, a data set naming convention that works well is one where the high-level qualifier or qualifiers have a meaning that proceeds from the general to the specific. For example, assume that an application named ACCOUNTS has three data sets: MASTER, TRANS, and ERRORS. If the data set names are ACCOUNTS.MASTER, ACCOUNTS.TRANS, and ACCOUNTS.ERRORS, you can use the high-level qualifier (ACCOUNTS) to cause all of the ACCOUNTS data sets to be either local resources or global resources.

Data set names that proceed from the general to the specific tend to work well with global resource serialization. Such names make it easy to identify and, if necessary, subdivide large groups of resources with a minimum number of entries in the RNLs. If your installation uses TSO/E, the format of the TSO/E userid is also important because it affects the names of user data sets.

Authorized qnames

An authorized caller of the ENQ services needs to protect itself from an unauthorized caller blocking or releasing the resource prematurely. Although an unauthorized caller cannot specify a task other than its own, an authorized caller could still be vulnerable if an unauthorized caller can acquire ENQs under the same task

GRS users define the QNAME/RNAME of resources. As such, the user defines the usage and meaning. See the ENQ/DEQ Summary table in *z/OS MVS Diagnosis: Reference* for definitions of the used ENQ resources (QNAME/RNAME). It is helpful to know the usage when observing ENQ/RESERVE activity and contention.

GRS provides the following list of qnames that only authorized callers may specify:

- ADRDFRAG
- ADRDSN
- ARCENQG
- BWODSN

- SYSCTLG
- SYSDSN
- SYSIEA01
- SYSIEECT
- SYSIEFSD
- SYSIGGV1
- SYSIGGV2
- SYSPSWRD
- SYSVSAM
- SYSVTOC
- SYSZ* (Where '*' is a wildcard. For example, SYSZABC is an authorized QNAME.)

Beginning with z/OS V1R13, global resource serialization provides an additional list of qnames that are conditionally authorized:

- ARCDSN
- ARCBTAPE
- ARCGPA
- ARCBACV
- ARCMIGV

You can set the AUTHQLVL parameter in the GRSCNFxx parmlib member to indicate whether the system is to recognize the second list of authorized qnames in addition to the original list. For information, see *z/OS MVS Initialization and Tuning Reference*.

An option on the DISPLAY GRS system command allows you to display the value, and an option on the SETGRS system command allows you to fall back to the original IBM default list of authorized qnames if you have enabled your system to recognize both lists. For information, see *z/OS MVS System Commands*.

Also, the Health Checker for z/OS health check GRS_AUTHQLVL_SETTING exists to help you determine the need of authorized qname protection for your installation. See *z/OS Upgrade Workflow*.

Authorized programs that are currently using unauthorized qnames in their ENQ requests should consider changing the qnames to authorized usage. This change is not trivial especially if the resources might be global. Also, other products that might interact with the qname resource can also have an impact on any changes that you make. The following protocol for transitioning an ENQ resource to use an authorized qname can help with planning such changes:

1. Add an ENQ resource with an authorized qname and the same rname and scope as the unauthorized ENQ. The following conditions apply:
   - The ENQ resource needs to be used wherever the unauthorized resource is used.
   - The ENQ resource needs to be in a list request to ensure that global resource serialization processes the set atomically.

   Performing a rolling IPL should be sufficient for global requests.
2. Communicate the intention to transition from one qname to another. Consider the following factors:
   - Synchronize the possible RNL definitions in the global resource serialization complex.
   - Be sure to update any separate product that interacts with the ENQ resource. Consider a waiting period of two years. If no other product interacts with this resource, this waiting period might not be necessary.
   - A
3. Before you make the change to remove the unauthorized ENQ resource and leave the new authorized ENQ resource in place, you must require that all systems are using both ENQ resources. The EQDQ

Monitor can filter on qname to help diagnose usage. Performing a rolling IPL should be sufficient for global requests.

To avoid any possible integrity problems, ISGENQ checks that an authorized caller uses an authorized qname. For COND=YES, a return and reason code of 040D is returned. For COND=NO, ABEND338 is issued for an OBTAIN or CHANGE request or ABEND330 is issued for a RELEASE request.

For more information, see the ISGENQ description of ISGENQRsn_UnprotectedQName and ISGENQRsn_UnprotectedExitQName in *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*.

Dynamic exits and RNL processing

Prior to RNL processing, your system can implement the ISGNQXIT or the ISGNQXITFAST installation exits. These exit routines will affect the way ISGENQ, ENQ, DEQ, and RESERVE requests are handled. The following attributes can be altered (note that for certain types of requests, the exits can change the return/reason code):

- QName
- RName
- Scope
- Device UCB address
- Convert Reserve
- Bypass RNLs.

For more details on possible resource alteration that are provided by ISGNQXIT or ISGNQXITFAST, see ISGYNQXP mapping macro in *z/OS MVS Data Areas* in the z/OS Internet Library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

The exit routines cannot alter the parameter list for the original request. To avoid potential integrity problems, ensure that the original DEQ parameters match the original ENQ parameters.

See *z/OS MVS Installation Exits* for complete information about:

- ISGNQXIT — ISGENQ / ENQ / DEQ Installation Exit
- ISGNQXITFAST — Fast ISGENQ / ENQ / DEQ Installation Exit

Request processing sequence

The following chart describes the installation flow of ENQ/DEQ exits.

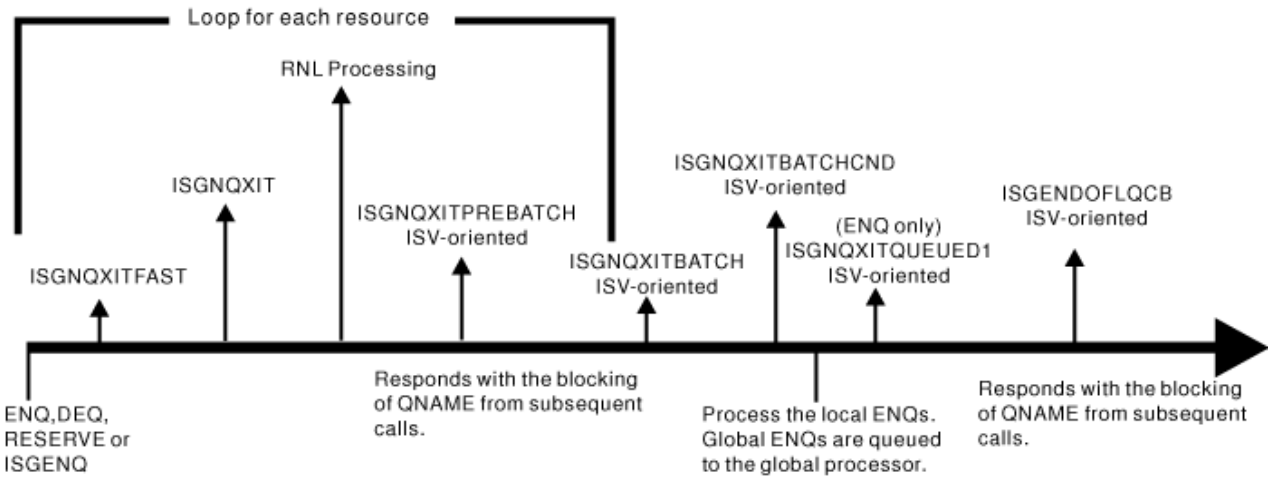


Figure 4. ENQ/DEQ exits installation flow

System programmers only use RNLs and either the ISGNQXITFAST or ISGNQXIT installation exit. All of the other exits are intended for the independent software vendors (ISVs).

As of V1R9 and later, the installation exits that are used by the ISVs and the ISGNQXITFAST exit (not the ISGNQXIT exit) can get control in a cross memory environment.

See "Global Resource Serialization Exits" topic in *z/OS MVS Installation Exits* for more information about the GRS exits.

RNL processing

To evaluate how well your installation's data set naming conventions will work with global resource serialization, and to make decisions about specific resources, you need to understand how global resource serialization processes entries in the RNLs to determine if a specific resource is a local resource or a global resource.

Note: An active ISGNQXIT or ISGNQXITFAST installation exit routine can request that RNL processing be bypassed.

How the RNL is scanned

Whenever global resource serialization encounters a request for a resource with a scope of SYSTEM or SYSTEMS, global resource serialization searches the appropriate RNL to determine the scope of the requested resource. Global resource serialization will not scan the RNL if one of the following is true:

- The request specifies RNL=NO.
- During the system IPL, GRSRNL=EXCLUDE was specified.
- During the system IPL, GRS=NONE in IEASYSxx was specified.
- The ISGNQXIT or ISGNQXITFAST installation exit routines are installed and active on the system.

To scan an RNL, global resource serialization compares the input search argument — the resource name — to the resource name entries in the RNL.

Entries in an RNL can be specific, generic, or can include wildcard characters.

Specific resource name entry
Matches a search argument only when they are exactly the same.

Generic resource name entry
Matches a portion of a rname (minor name) within the resource name. A match occurs whenever the specified portion of the generic rname entry matches the beginning of the same portion of an input search argument. The qname (major name) must match exactly.

Pattern resource name entry
Contains wildcard characters that extend the matching specification. The wildcard characters can be used within both parts of the resource name.

*****
Allows matching for a substring of any characters for any length, including zero.

?
Allows matching for any single character.

For the correct RNL syntax rules, see the Syntax rules for GRSRNLxx in *z/OS MVS Initialization and Tuning Reference*.

Each RNL entry indicates whether the name is specific (SPECIFIC), generic (GENERIC), or contains wildcard characters (PATTERN).

Global resource serialization finds a match when:

- A specific resource name entry in the RNL matches the specific resource name in the search argument.

For example, the specific entry APPL01,MASTER in the RNL matches APPL01,MASTER as a search argument, but does not match a search argument of APPL01,MASTER2.

The length of the specific rname is important; for example, a specific rname of 'ABC' does not match a resource named 'ABC '.

- A generic qname entry in the RNL matches the qname of the search argument.

  For example, a generic qname entry of APPL01 in the RNL matches any search argument with a qname of APPL01, such as APPL01,MASTER or APPL01,TRANS.

- A generic qname,rname entry in the RNL matches the corresponding portion of the resource name in the search argument.

  For example, a generic qname,rname entry of APPL01,MASTER in the RNL matches any search argument that begins with APPL01,MASTER, such as APPL01,MASTER or APPL01,MASTER2. However, APPL01A,MASTER would not match.

Effective use of generic entries and entries containing wildcard characters simplifies the management of the RNLs by reducing the number of entries and the complexity of the entries.

Note: All of the specific entries in a RNL are searched before scanning the generic and pattern RNL entries for a match. If no specific entry matches, the first generic or pattern entry that matches is used.

Global resource serialization scans the RNL to determine if there is a match. The actions it takes, however, are different for each RNL.

ISGQUERY RNL search routine

The ISGQUERY REQINFO=RNLSEARCH service provides an interface to search the RNLs for a match to a specific resource name (QNAME/RNAME pair).

See *z/OS MVS Programming: Assembler Services Reference IAR-XCT* for complete information about using the ISGQUERY macro.

SYSTEM inclusion RNL

When global resource serialization encounters an ENQ or DEQ request for a resource with a scope of SYSTEM, it scans the SYSTEM inclusion RNL. If there is no match, global resource serialization processes the resource as a local resource.

If there is a match, global resource serialization changes the scope of the request from SYSTEM to SYSTEMS and processes the resource as a global resource. For example, if the resource a task requests is A,B,SYSTEM, and there is a match in the SYSTEM inclusion RNL, global resource serialization changes the resource name to A,B,SYSTEMS and processes A,B,SYSTEMS as a global resource. That is, global resource serialization then scans the SYSTEMS exclusion RNL for the resource name. See Figure 8 on page 30.

Thus, you can specify a generic name in the SYSTEM inclusion RNL, changing the scope of all resources with that generic name to SYSTEMS. You can then place specific resource names with that generic name in the SYSTEMS exclusion RNL, making those specific resources local resources. (The default RNLs use this technique to make certain system data sets local resources. See Figure 10 on page 34.

The following are examples of a SYSTEM inclusion RNL:

```
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(SYS1.PROD)
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(SYS1.PRD1)
RNLDEF RNL(INCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(SYS1.PRD2)
```

The following is an example of a SYSTEM inclusion list using the RNL wildcard:

```
RNLDEF RNL(INCL) TYPE(Pattern) QNAME(SYSDSN) RNAME(SYS1.*)
```

SYSTEMS exclusion RNL

When global resource serialization encounters a request for a resource with a scope of SYSTEMS, it scans the SYSTEMS exclusion RNL. Thus, it scans the SYSTEMS exclusion RNL when:

- A task issues a RESERVE macro (which has a default scope of SYSTEMS) or specifies a scope of SYSTEMS on the ENQ or DEQ macro.
- The resource name matched an entry in the SYSTEM inclusion RNL, which caused global resource serialization to change the scope to SYSTEMS.
- The ISGNQXIT and ISGNQXITFAST exit routine changed the scope to SYSTEMS.

If there is a match, global resource serialization changes the scope of the request from SYSTEMS to SYSTEM and processes the resource as a local resource. For example, if the resource name a program requests is A,B,SYSTEMS, and there is a match in the SYSTEMS exclusion RNL, global resource serialization changes the resource name to A,B,SYSTEM and processes A,B,SYSTEM as a local resource.

If there is a match and the task issued the RESERVE macro to request the resource, global resource serialization processes the resource as a local resource and does not scan the RESERVE conversion RNL.

If there is no match, global resource serialization processes the request as a global resource. If the task issued the RESERVE macro to request the resource, global resource serialization then scans the RESERVE conversion RNL to determine if the system is to issue the reserve.

The following are examples of a SYSTEMS exclusion RNL:

```
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSDSN) RNAME(SYS1.PRD1.LOGREC)
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSDSN) RNAME(SYS1.PRD2.LOGREC)
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSDSN) RNAME(SYS1.PRD1.MANX)
RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSDSN) RNAME(SYS1.PRD2.MANX)
RNLDEF RNL(EXCL) TYPE(GENERIC) QNAME(SYSDSN) RNAME(SYS1.TEST)
```

The following are examples of a SYSTEMS exclusion list using the RNL wildcard:

```
RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSDSN) RNAME(SYS1.*.LOGREC)
RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSDSN) RNAME(SYS1.*.MANX??)
```

RESERVE conversion RNL

The purpose of the RESERVE conversion RNL is to convert a hardware RESERVE to just a global (SYSTEMS) ENQ. Even though double serialization (ENQ and hardware reserve) can result in deadlock, it is the default behavior for RESERVE requests. Thus, RESERVE requests should result in only a global ENQ or a hardware reserve and a local (SYSTEM) ENQ.

A hardware reserve request can originated by:

- A task issuing the RESERVE macro or the ISGENQ macro with RESERVEVOLUME=YES
- A task issuing the ENQ macro or the ISGENQ macro with RESERVEVOLUME=NO and a GRS installation exit change the ENQ request to a reserve request by adding a UCB@ to it.

The RESERVE conversion RNL is scanned for all reserve requests under the following conditions:

- A GRS installation exit did not request that RNLs should be bypassed. Note that the RESERVE API does not support RNL=NO.
- After processing installation exits and the exclusion RNL, the ENQ associated with a reserve has a scope of SYSTEMS.

If there is a match, global resource serialization suppresses the reserve for the global resource. It issues an ENQ with a scope of SYSTEMS to serialize access to the resource.

If there is no match, global resource serialization allows the system to issue the reserve. Thus, the RESERVE conversion RNL does not affect the scope of the resource but does determine whether or not the system is to issue the reserve.

Choosing the reserves to be converted is a critical planning task; for a full explanation of the considerations involved, see "Potential problems when using RESERVE" on page 7.

The following is an example of a RESERVE conversion RNL using a wildcard:

```
RNLDEF RNL(CON) TYPE(PATTERN) QNAME(*)
```

Excluding requests from RNL processing

Guideline: Use the default RNL=YES

Programs should use the default RNL=YES unless the resource *must under all possible user configurations*, always be serialized at the specified scope. Do not use RNL=YES for a SYSTEMS scope when the sharing system resource domain cannot be a sub-set or extension of the systems in the GRS Complex. An ENQ or DEQ request can be excluded from RNL **and installation exit processing** by specifying RNL=NO on the request.

Many applications specify RNL=YES to allow customers to change the resource domain based on their product's various configuration capabilities. See Figure 5 on page 26 and Figure 6 on page 27 below for some different possible configurations. For example, it is common for customers using an alternate serialization product to share VSAM datasets across GRS complexes or Sysplexes. However, some dataset types require Sysplex communications for their management. As such, their ENQs are issued with RNL=NO to prevent them from being managed at a domain greater than or less than the GRS Complex or Sysplex level.

When RNL=NO is specified, the ENQ request will be ignored by alternative serialization products, and the ENQ will not be used in a resource domain outside or only within a subset of the GRS complex. It also ensures that early SYSTEMS scope ENQs that are issued prior to system Master Scheduler Initialization (MSI), when an alternate serialization product has not initialized, will always intersect with ENQs that are issued on other systems when the alternate serialization product is active or not active. Use RNL=NO only after understanding and considering the impacts of its specification to the control of the resource by an installation-specified RNL and alternate serialization product functionality.

Rules

Consistent usage of the RNL= and resource scope specifications is important:

- If RNL=NO is specified for a resource on the ENQ request, it must also be specified on the DEQ request. ISGENQ release does not support the RNL keyword as it always ensures that the release matches what was specified on the obtain.
- Do not mix ENQ RNL=NO and RNL=YES requests that use the same QNAME and RNAME unless you are dynamically migrating from RNL=YES to RNL=NO. See below.
- Do not mix ENQ scope SYSTEM and SYSTEMS requests that use the same QNAME and RNAME.
- RNL=NO must be specified for resources that must be serialized exclusively by two or more members of a multisystem Sysplex and no other systems outside the Sysplex. If RNL=YES was being used previously and it was determined that RNL=NO must now be used, then making the change must be done with care or serialization can be lost. See "When to Use RNL=NO" on page 24 when required to do so.

Failure to follow these rules can result in abends, incorrectly obtaining serialization, failure to obtain serialization, or failure to release serialization.

When to Use RNL=NO

For the proper serialization of a resource, the correct resource identity, consisting of the QNAME, RNAME, and SCOPE, must be used within the correct resource domain. The resource domain consists of the systems and units of work within those systems that take part in the sharing of the serialized resource. For example, a SCOPE=STEP ENQ can only be used to serialize units of work within a single job step as there are no means to alter or extend that scope from what is specified on the API. However, via GRS RNLs and GRS installation exits (which alternate serialization products exploit), an initial scope of SYSTEM specified with RNL=YES can result in the resource being shared beyond the system and an initial scope of

SYSTEMS can result in the resource only be shared on the local system. GRS RNLs allow for the resource to either be included or excluded from the GRS global resource domain consisting of all the systems in the GRS complex while installation exits allow for many changes. Such changes may include skipping GRS RNLs via an alternate serialization product, extending the resource domain to systems in multiple GRS complexes that can cross Sysplexes, or limiting the systems to a subset of the GRS complex potentially resulting in a subset of systems from the Sysplex from which the GRS API was issued.

Defining and documenting the resource sharing rules, including how the scope can be altered, ultimately lies with the owner of the resource. GRS provides the means for the installation and alternate serialization products to alter the scope and extend or limit the domain of systems and Sysplexes that serialize a shared resource. In some cases, a resource sharing rule may not allow the altering of the resource identity and the owner must enforce the rules by specifying RNL=NO on associated GRS APIs. For example, when the installation has no means of configuring a resource to prevent the need for all multisystem Sysplex members to be included in the resource domain, the resource owner must use SCOPE=SYSTEMS and RNL=NO.

Note: Global ENQs are issued via ENQ/DEQ/ISGENQ macros with SCOPE=SYSTEMS or SCOPE=SYSPLEX. SCOPE=SYSPLEX is misleading and is the same as specifying SYSTEMS.

Use cases for specifying RNL=NO

1. The resource cannot be shared across systems.
   - Any program that has a system only resource that can never be shared by multiple systems must specify SCOPE=SYSTEM and RNL=NO to prevent:
     - the scope from being changed to SYSTEMS
     - an alternate serialization product from extending the scope to multiple systems.
2. The resource cannot be shared across GRS Complexes or Sysplexes.
   - Any program that has a resource that cannot be shared across GRS Complexes or Sysplexes must specify RNL=NO to prevent alternate serialization products from doing so.
3. The resource cannot be limited to be shared *only* by a subset of the GRS Complex or Sysplex systems.
   - Any program that requires that any system in the Sysplex or GRS Complex must be able to serialize the resource, must specify RNL=NO in order to prevent an alternate serialization product from limiting the scope to a subset of the Sysplex or Complex.

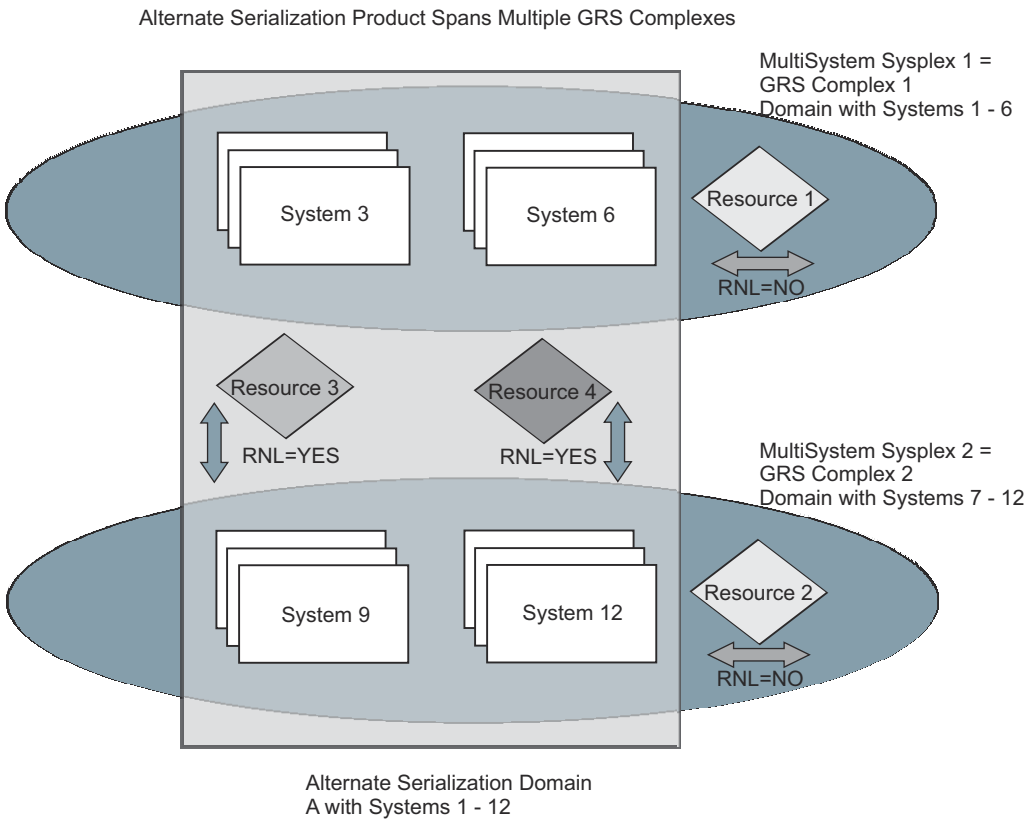


Figure 5. Alternate serialization product spans multiple GRS Complexes

Figure 5 on page 26 shows a configuration where an alternate serialization product is used to serialize most SCOPE=SYSTEMS resources under a single resource domain that includes members of two GRS Complexes. Within each GRS Complex, GRS is managing SCOPE=SYSTEMS resources that are not managed by an alternate serialization product. The only means of ensuring that an alternate serialization product does not manage a resource is to specify RNL=NO on the GRS APIs.

Resources 1 and 2 are unique within their Sysplexes and must always be shared only by all members of their respective Sysplexes. However, the resource must also be shared by all systems in the Sysplex as with resource 1 in the Figure 6 on page 27 configuration where there is a sub-Sysplex resource domain managed by an alternate serialization product. As such, RNL=NO is specified on the GRS APIs that serialize the resources. If the installation could create more than one unique instance of the resource with unique QNAMEs and/or RNAMEs, as needed within each sub-Sysplex, then RNL=YES would have been used.

Resources 3 and 4 can be shared across the two Sysplexes since their use has no dependencies on any other resources that are specific to any given GRS Complex or Sysplex. If they were, then the RNL=YES could still be specified but the installation would have to be able to create unique instances. This is also true in the configuration of Figure 6 on page 27 such for resource 3 and 4. However, for Figure 6 on page 27, if they had the same QNAME and RNAME, they still would not intersect as they are in different domains.

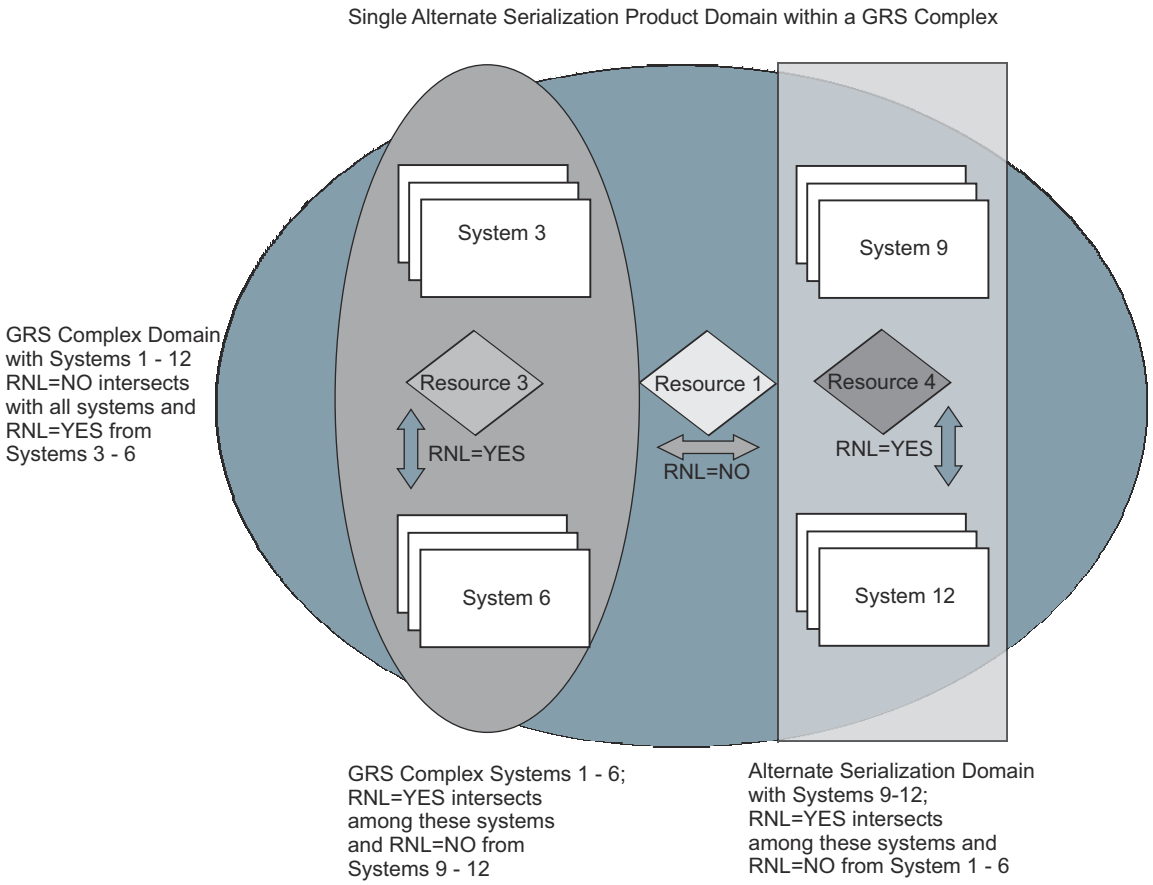


Figure 6. Multiple alternate serialization product domains within a GRS Complex

Figure 6 on page 27 depicts a configuration that has 1 GRS Complex (the circle) consisting of a multisystem Sysplex that has a sub-GRS Complex resource domain (the rectangle) managed by an alternate serialization product. The RNL keyword would control which resource domain is used as follows:

- RNL=NO ENQs issued from any system will intersect with RNL=NO ENQs issued on any system and RNL=YES ENQs issued on systems 1-6 as they are always managed by GRS.
- RNL=YES ENQs (assuming no GRS RNL change to SYSTEM scope) issued on systems 1-6 will intersect with systems 1-6 and RNL=NO ENQs issued on any system.
- RNL=YES ENQs (assuming no alternate serialization product RNL equivalent change to system) issued on systems 9-12 will intersect only with ENQs issued on systems 9-12 as they are only managed by the alternate serialization product.

RNL=YES allows the installation to ensure that resource 4, which is unique and only used by systems 9-12 and might have the same name as resource 1 or 2, does not have related ENQs intersecting with systems 1-6. Similarly, it would also allow resource 3 to have related ENQs that only intersect with systems 1-6. The product using resource 4 documents how it is related to other product resources so they can all be managed within either or both of the resource domains. For example, a unique resource is needed for each domain.

The required RNL=NO for resource 1, implies that there can only be one resource 1 in the GRS Complex and thus its related ENQs must intersect across all systems regardless of being in an alternate serialization product domain. The product using resource 1 documents that there must be one of these resources within the GRS Complex or Sysplex and that the installation cannot configure it any other way.

Products that are dependent on serializing a resource across all members of a Sysplex because there can be only one instance of that resource, must specify RNL=NO on its ENQ/DEQ serialization related service requests for that resource. Failure to do so can cause serialization errors as some Sysplex members may be in different resource domains which do not intersect.

To provide installation guidance on how to best share resources for an application, it is imperative that applications clearly document all resources and how they can be shared. This includes QNAME, RNAME, SCOPE, and RNL= API specifications, when it is appropriate to alter their scope, and any configuration changes or procedures that might be required. RNL=NO must be specified when the resource, such as resource 1 and 2 in Figure 5 on page 26 and resource 1 in Figure 6 on page 27, must only be used by all members of a given multisystem Sysplex. With RNL=YES, alternate serialization product scope domains, within a multisystem Sysplex, would result in the required serialization not intersecting, and therefore result in data integrity exposures. Consider the following examples:

1. An application has a resource that must be shared by any member that is running on a system with the same multisystem Sysplex and cannot be shared across members of different Sysplexes. For example, an application that has a resource that is used to manage Sysplex unique resources in a multisystem Sysplex. In this case, RNL=NO must be specified by each instance to ensure that the source is managed by GRS with a SCOPE=SYSTEMS that is used across the Sysplex members. Since GRS complexes cannot span multisystem Sysplexes, the resource's scope cannot span multiple multisystem Sysplexes. Specifying RNL=YES allows environments with multiple alternate serialization product instances to manage the SYSTEMS's scope within their unique and potentially smaller than the Sysplex domains, rather than across the Sysplex. Therefore, the resource would be incorrectly serialized causing data integrity issues.
2. An application wants to provide the means of sharing a resource across multiple GRS complexes when there is no requirement that each member of the GRS complex must be able to access the resource. In this example, an application has a configuration, or rules resource, that is handy to have shared across multiple domains of the application when the rules must always be consistent for all or some instances of the application. In cases where a unique resource across specific instances is desired, a specific name can be defined by the user for each domain instance of the resource. This name is reflected in the ENQ QNAME/RNAME resource so references to different resources don't collide. The application has no requirement that each member of the GRS complex must be able to access the resource. In fact, it keeps data in the resource so when accessing it, it can verify that its serialization domain matches the one being used by the application instance. If not, it complains and terminates what it was doing. In this case, the application always specifies RNL=YES so that an alternate serialization product can be used to share the resource across multiple Sysplexes. Unique resource names can be used when different resources must be used and the application provides a means and instructions for the instance to migrate from one resource to another.

How to migrate from RNL=YES to RNL=NO when required to do so

When an application has a requirement that a resource be appropriately serialized by each instance of a multiple Sysplex, already has code using RNL=YES, and now needs to specify RNL=NO, the application will have to provide a means for the installation to roll out the change to RNL=NO. The following are suggested ways for rolling out the change across multiple instances in a serialized manor:

1. Provide a serialized means of migrating all instances of ENQ, DEQ, etc. requests to the new RNL=NO specification so no RNL=YES requester is holding or is requesting the resource while another instance might be doing so with RNL=NO. This can be done by:
   a. A complete shutdown of the application rolling out the new RNL=NO update to all instances and restarting them.
   b. Using an existing alternate synchronization that has already been exploited for making synchronized software updates. Unfortunately, most software is not dynamically updateable in such a manor but some may be.
2. Roll out the change in two stages. The first stage needs to coexist with n-1 instances by using two ENQs, one with RNL=YES and one with RNL=NO. Once all of the systems are running with the two ENQs, start to migrate back to one ENQ with RNL=NO. The first stage's two ENQs will intersect with ENQs in both the GRS complex resource domain and the alternate serialization product domain. See Figure 7 on page 29 for a coded example of using this technique. Rules for the ENQs are as follows:
   a. The first ENQ/DEQ is new and specifies RNL=NO to serialize the resource under the GRS complex resource domain

b. The second ENQ/DEQ exists and specifies RNL=YES to serialize the resource under the existing GRS complex or alternate serialization resource domain. However, it is modified to specify RET=HAVE to indicate that it is acceptable to already hold the ENQ by the current unit of work. This covers cases where both ENQs fall into the GRS complex resource domain that occurs when an alternate serialization product is not installed or is not managing the resource under its domain.

Note:

1. Unacceptable return codes for the RET=HAVE ENQ are not checked in the example but must be. Failure cases must result in the previously successful ENQ being DEQed.
2. Recovery scenarios are not included in the example. For example, a retry scenario that is executing the sequence after an asynchronous abend after obtaining the first ENQ but not attempting the second, would have to be coded.
3. The ENQ and DEQ APIs are used. ISGENQ can also be used if required or preferred.
4. Using an ECB would be similar in that the ENQs are issued in the same order while waiting for each one to complete.
5. For list requests containing more than one ENQ resource, the return codes of the ENQs will have to be checked for similar values as with the non-list example. All the ENQs that were obtained must be released when an unacceptable return code is found. For example, it is acceptable for the second RNL=YES ENQ for the resource that must be RNL=NO to fail indicating it is already held in environments where there is no alternate serialization product.

```
* Sample ENQ invocations to assist with converting from RNL=YES to RNL=NO
* You could use your ENQ or ISGENQ invocations and modify accordingly

*Issue the first ENQ for unconditional, exclusive control of the resource
*with a scope of Systems, and do not allow conversion.
      ENQ  (MYQNAME,MYRNAME,E,16,SYSTEMS),RNL=NO
* Issue the second ENQ for conditional, exclusive control of the resource
* with a scope of Systems. The request may be converted to System.
      ENQ   (MYQNAME,MYRNAME,E,16,SYSTEMS),RNL=YES,RET=HAVE
* Acceptable return codes from this request are, anything else back out:
*   0 – Resource held (was converted to System)
*   8 – Resource not held (but held with Systems, was not converted

* Now have serialized the resource, perform function….

* Dequeue the first resource without conversion
         DEQ   (MYQNAME,MYRNAME,16,SYSTEMS),RNL=NO
* Dequeue the second resource conditionally (RET=HAVE), possibly with conversion
* RET=HAVE is required when an alternate serialization product is not present
* resulting the second ENQ having failed as it was a duplicate
         DEQ   (MYQNAME,MYRNAME,16,SYSTEMS),RNL=YES,RET=HAVE
* DECLARES
MYQNAME  DC    C'MYQNAME '
MYRNAME  DC    C'THIS IS MY RNAME'
```

Figure 7. Sample ENQ invocations to assist with converting from RNL=YES to RNL=NO

RNL processing sequence

Global resource serialization searches the RNLs, unless an ENQ or DEQ is coded with RNL=NO or an ISGNQXIT or ISGNQXITFAST installation exit changes the request indicating to bypass RNL processing. When RNL=NO, global resource serialization bypasses the RNL search. The default, however, is RNL=YES. Figure 8 on page 30 summarizes the processing for a resource requested by an ENQ or DEQ macro. The figure shows how global resource serialization uses the scope of the request, changes (if any) made by an ISGNQXIT or ISGNQXITFAST exit, and the RNLs to determine if the resource is a local or global resource. It also shows when global resource serialization internally changes the scope of the resource.

For an ENQ or DEQ request with a scope of SYSTEM, global resource serialization first scans the SYSTEM inclusion RNL. If it finds a match, it changes the scope to SYSTEMS and then scans the SYSTEMS exclusion RNL.

For an ENQ or DEQ request with a scope of SYSTEMS, global resource serialization scans only the SYSTEMS exclusion RNL.

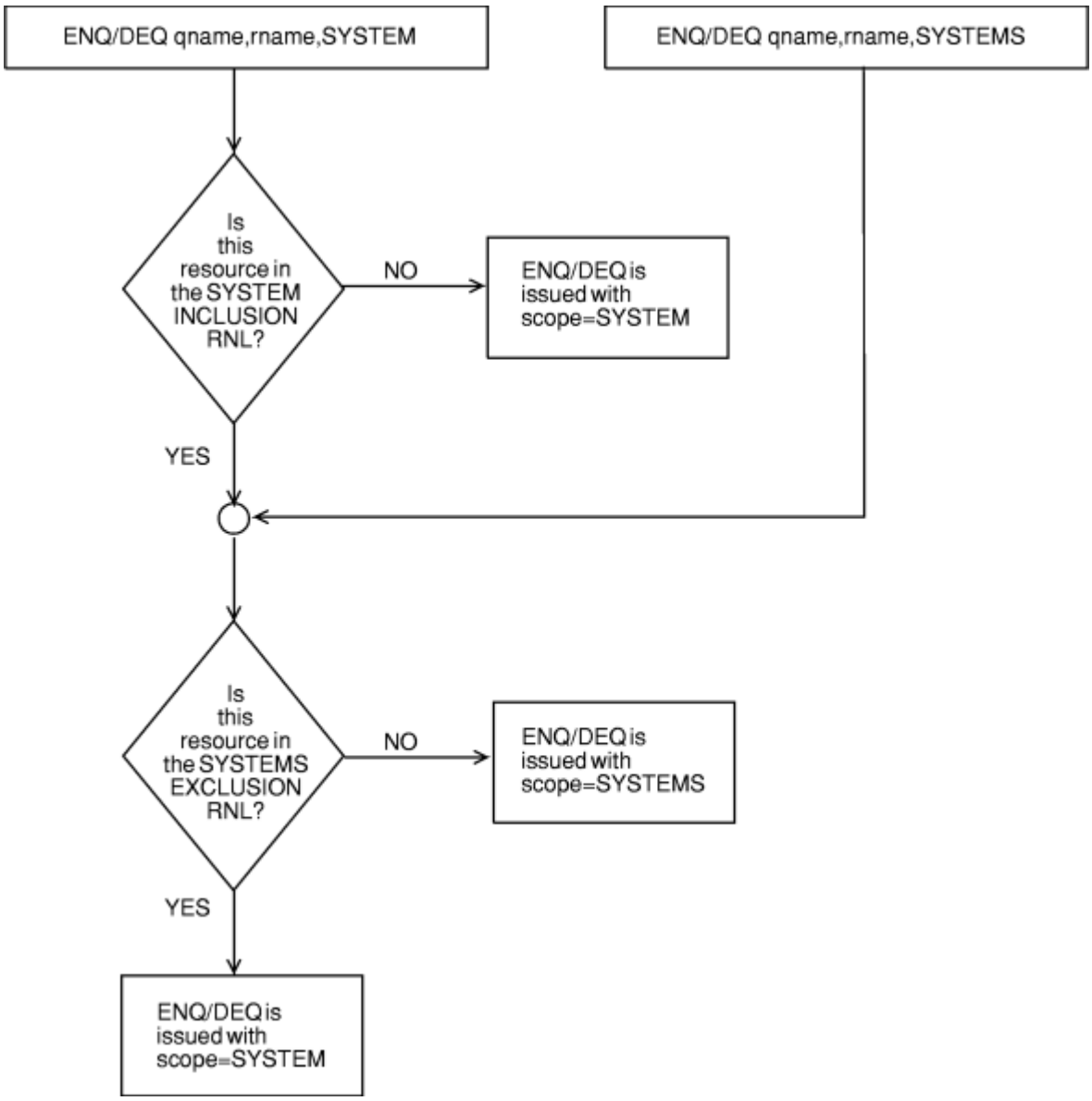


Figure 8. ENQ and DEQ processing summary

For a RESERVE request, global resource serialization first scans the SYSTEMS exclusion RNL to see if it should treat the resource as a local resource or as a global resource. If the resource is not named in the RNL and is thus a global resource, then global resource serialization scans the RESERVE conversion RNL to see if the global resource requires a reserve. If the resource is named in the SYSTEMS exclusion RNL, global resource serialization does not search the RESERVE conversion RNL; it treats the resource as a local resource and allows the system to issue the reserve.

Figure 9 on page 31 summarizes the processing done for a resource requested by a RESERVE macro. The figure shows that global resource serialization first scans the SYSTEMS exclusion RNL to determine whether or not the resource is global. If there is no match, it scans the RESERVE conversion RNL to determine whether or not the system is to issue a reserve. Figure 9 on page 31 also shows when global resource serialization internally changes the scope of a resource requested by a RESERVE macro.

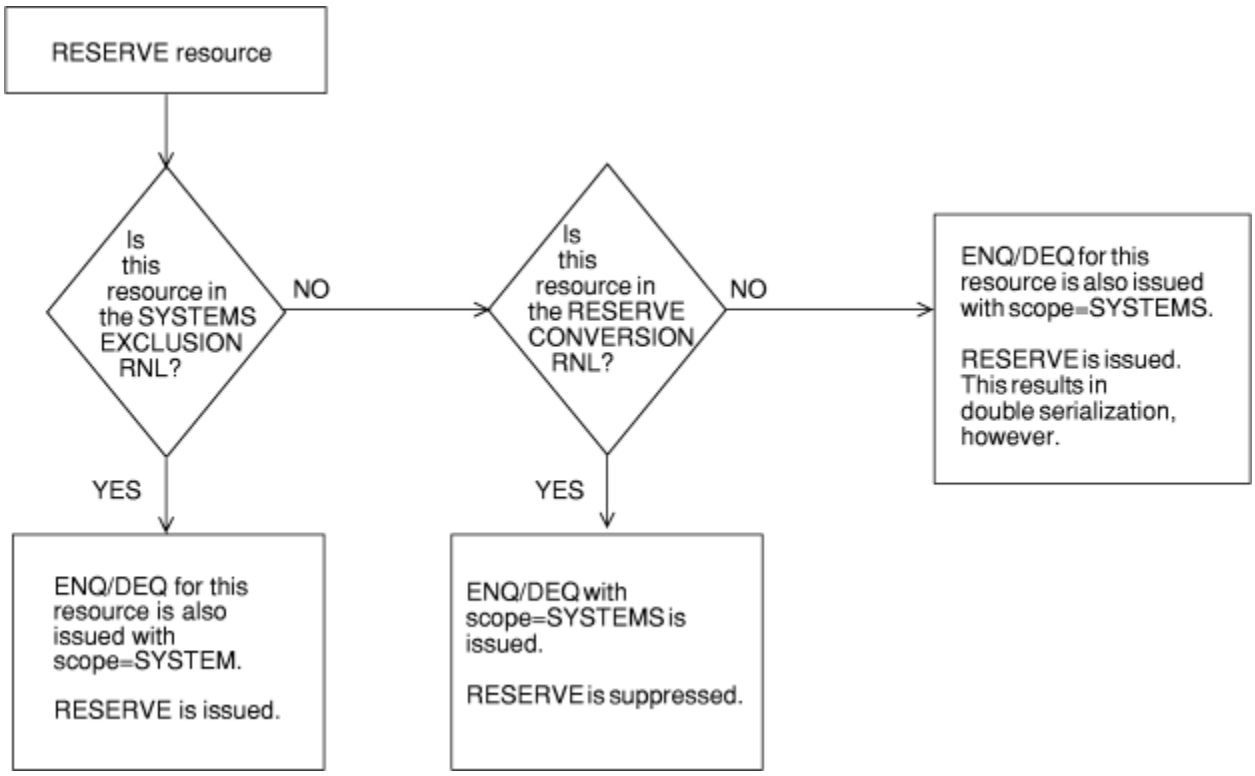


Figure 9. RESERVE Processing Summary

RESERVE conversion

One major purpose of global resource serialization is to eliminate the need to protect data sets on shared DASD volumes by issuing a RESERVE macro that causes a reserve of the entire volume. RESERVE can cause some critical problems, so keep in mind the restrictions on their conversion to global ENQs. Be sure to review "Potential problems when using RESERVE" on page 7 and "Solving the RESERVE problems" on page 8.

For each application's RESERVE request the installation must select one of the following choices to avoid double serialization:

- Convert the RESERVE for a resource. Place the resource name in the RESERVE conversion RNL and do not place its name in the SYSTEMS exclusion RNL. Global resource serialization suppresses the RESERVE and treats the requested resource as a global resource.
- Issue the RESERVE for the resource. Place the resource name in the SYSTEMS exclusion RNL. The system issues the reserve, and global resource serialization treats the requested resource as a local resource. It internally converts the scope of the request to SYSTEM to serialize access to the resource within the originating system. Global resource serialization then does not search the RESERVE conversion RNL, so the system issues the reserve to serialize access to the resource among multiple systems.

In a Parallel Sysplex® environment, for improved performance, availability, and serviceability, IBM recommends that a star complex using sysplex-wide RESERVE serialization convert to global ENQ serialization. Use the PATTERN specification to implement this conversion and include the following statement as the only reserve conversion entry in the GRSRNLxx member of parmlib.

```
RNLDEF RNL(CON) TYPE(PATTERN) QNAME(*)
```

Note however that the majority of restrictions detailed in "Solving the RESERVE problems" on page 8 are still applicable to star mode, with ring performance being the exception. The installation may need to make several changes in their process and their programs before all RESERVES can be converted.

If you are using the GDPS/PPRC product or the GDPS/PPRC HyperSwap Manager product, see the following publications for GDPS specific requirements:

- *GDPS/PPRC Installation and Customization Guide, ZG246703*
- *GDPS/PPRC HyperSwap Manager Installation and Customization Guide, ZG246746*

Migrating from GRSRNL=EXCLUDE to a set of RNLs

Migrating from a GRSRNL=EXCLUDE environment to full RNLs does not require a sysplex-wide outage for certain environments. For details about this, see the information later. When the FORCE option is specified, the SET GRSRNL=*xx* processing dynamically switches to the specified RNLs and moves all previous requests to the appropriate scope (SYSTEM or SYSTEMS), as if the RNLs were in place when the original ENQ requests were issued.

To ensure data integrity, the requirements for this function are listed as follows. The migration is canceled if any of the following requirements are not met and GRSRNL=EXCLUDE remains in effect.

1. The function must be used in a single system sysplex in GRS Star mode with GRSRNL=EXCLUDE.
2. No ISGNQXIT or ISGNQXITFAST exit routines can be active nor requests that were affected by either of the exits.
3. No ISGNQXITBATCH or ISGNQXITBATCHCND exit routines can be active.
4. No local resource requests can exist and become global resource requests if the global resource is already owned. This might be caused by some requests being issued with RNL=NO.
5. No resources with a MASID request can exist if only some of the requesters become global. This might be caused by some requests being issued with RNL=NO or some requests being issued with different scopes.
6. No RESERVE requests that are converted by the ISGNQXITBATCH or ISGNQXITBATCHCND exit and do not get converted by the new RNLs can hold a resource. This might be caused by a resource being managed globally by an alternative serialization product before the migration, but afterwards being serialized by device RESERVE.

- When in GRS Ring mode, the ISG248I message is issued, indicating that the SET GRSRNL command is not accepted in a GRSRNL=EXCLUDE environment.
- When in GRS Star mode, before the SET GRSRNL=*xx* migration begins, message ISG880D prompts out, asking the user to confirm the use of the FORCE option.

Note: Any errors in changing the RNL configuration can lead to deadlocks or data integrity errors. When doing migration to the specified RNLs, the only way to move back to GRSRNL=EXCLUDE is a sysplex-wide IPL. The FORCE option cannot be specified from the existing RNLs to another set of RNLs. Long-held resources can delay such a change infinitely, and therefore a cancelation of jobs or even a sysplex-wide outage is required to end the job.

A rebuild of the ISGLOCK structure is initiated as the final step in the migration. Test a rebuild of the ISGLOCK structure before you issue the command SETXCF START,REBUILD,STRNAME=ISGLOCK to initiate the migration.

Changing the RNL

In certain circumstances, you might need to change the scope of one or more resources. If the sysplex matches the complex, you can use the SET GRSRNL command to make the change dynamically; no system needs to reIPL. If, however, your ring complex includes any systems that are not a part of the sysplex, you must remove those non-sysplex systems from the global resource serialization complex before initiating the change. After the change completes, you can then reIPL the non-sysplex systems back into the global resource serialization complex. Plan your use of the RNL carefully when operating a mixed complex to avoid an unnecessary IPL.

Changing the RNL dynamically would be useful in the following examples:

- Add new applications that introduce new resources.

- Tune a complex by suppressing reserves, especially against such resources as catalogs.
- Reverse an RNL change that might be degrading the performance of your complex.
- Change your shared resources (changing local resources into global resources, or changing global resources into local resources).
- Run a job (such as DFDSS DEFRAG programs) that performs better with different RNL than the ones used for your installation's normal operations.

After the operator issues the SET GRSRNL command, global resource serialization suspends any job that requests any *affected resource* (a resource that is not the same in the current RNL as in the new ones) until the change completes or until the operator cancels the change. If a job currently holds one or more of the affected resources, the change is delayed until that job frees any affected resources. For more details see:

- "Changing the RNLs for a ring" on page 149 for the ring complex, and
- "Use of the sysplex couple data set" on page 97 and "Changing RNLs for a star complex" on page 107 for the star complex.

Some restrictions apply while global resource serialization is processing an RNL change.

- Global resource serialization can process only one change command at a time.
- The total amount of new RNLs that can be processed during any single change request cannot be greater than 61 KB.
- A new system cannot join the complex until after the change completes or is cancelled; however, a system can be removed from the sysplex during an RNL change. If the removed system is the system on which the change command was issued, the change is cancelled.

Note:

1. After a change completes, you must be sure to update the GRSRNL system parameter (in IEASYSXX) on every system in the sysplex to preserve the changes beyond the next reIPL.
2. You cannot migrate a global resource serialization ring complex to a star complex during an RNL change.

RNL changes are permanently recorded in SMF records, and RNL change events can be traced using the TRACE CT command with COMP=SYSGRS.

RNL defaults

Because the resources to include in each RNL depend on the needs of your installation, it is not possible to supply default RNLs that will work well in every environment. IBM does supply a default for each RNL; Figure 10 on page 34 shows these defaults.

The generic qname entry for SYSDSN in the SYSTEM inclusion RNL indicates that all data sets that go through MVS allocation (except for VIO and subsystem data sets, such as SYSIN, SYSOUT, and SUBSYS data sets) are to be global resources. The entries in the SYSTEMS exclusion RNL identify the system data sets with a qname of SYSDSN that specifically cannot be global resources.

SYSTEM inclusion RNL:

SYSDSN

SYSTEMS exclusion RNL:

SYSDSN SYS1.BRODCAST
SYSDSN SYS1.DAE
SYSDSN SYS1.DCMLIB
SYSDSN SYS1.DUMP (generic — all dump data sets)
SYSDSN SYS1.LOGREC
SYSDSN SYS1.MAN (generic — all SMF data sets)
SYSDSN SYS1.PAGE (generic -- all page data sets)
SYSDSN SYS1.STGINDEX
SYSDSN SYS1.UADS

RESERVE conversion RNL:

The RESERVE conversion list is empty.

Figure 10. Contents of the default RNLs

You can IPL the systems in your GRS complex and build a ring using the default RNLs, but an IPL with the default RNLs might not reflect all the goals your installation has for the global resource serialization complex.

For example, you might want to change the contents of the default RNLs to emphasize RESERVE conversion and avoid potential interlocks. The DISPLAY GRS,CONTENTION command provides information about resources that are causing contention, and RMF reports can also help. The most useful RMF reports related to the system delay activity are:

- Monitor I enqueue activity report
- Monitor II system enqueue contention (SENQ) and system enqueue reserve (SENQR) reports
- Monitor III (workload delay monitor) reports on resource-oriented enqueue delays and resource-oriented device delays

Guideline: Use the ENQ/RESERVE/DEQ monitor to help plan RNL use. See Chapter 3, “Using the ENQ/RESERVE/DEQ monitor tool,” on page 49.

Tip: To collect the same data but with less impact to the ENQ/DEQ path length, instead use SMF 87 subtype 2 records. See “Using SMF 87 records” on page 185 for more information.

Because a global resource serialization complex can consist of multiple levels of z/OS systems, and because each system control program usually works with multiple levels of other products, it is not possible to compile a complete and exhaustive list of recommended treatment for resources in all possible situations. There are, however, some general guidelines as well as some specific suggestions on known resources that are good candidates for the SYSTEM inclusion RNL, the SYSTEMS exclusion RNL, and the RESERVE conversion RNL. These guidelines and suggestions cover the following topics:

“CICS” on page 35
“DAE” on page 35
“DB2” on page 36
“DFSMShsm” on page 36
“SMS” on page 36
“IMS” on page 36
“ISPF or ISPF/PDF” on page 36
“JES2” on page 37
“JES3” on page 37

"System logger" on page 38
"RACF" on page 38
"Tape volumes" on page 39
"Temporary data sets" on page 39
"TSO/E" on page 39
"SYS1.UADS and SYS1.BRODCAST" on page 39
"TSO/E user data sets" on page 40
"VIO journaling data set" on page 40
"VSAM data sets" on page 41
"Catalogs" on page 41

The information is meant to give you an idea of the kind of decisions you might need to make, and it does not deal with the specifics of every level of every product.

If you need more information about the macros mentioned in the following explanation, see:

- *z/OS MVS Programming: Authorized Assembler Services Guide*
- *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*

CICS

If you are using CICS/XRF in a multi-MVS environment with DB2, you should use global resource serialization to ensure the integrity of DB2. In the case of an alternate CICS system taking over from an active CICS system, the original DB2 region that the failed CICS was using must be completely terminated before a new DB2 can be restarted with the newly active CICS. Global resource serialization can reduce the risk of data integrity problems caused by concurrent execution of DB2 on both MVS systems. One recommendation is to set up global resource serialization to control key DB2 data sets so that only one DB2 region is allowed to update them at the same time.

DAE

Dump analysis and elimination (DAE) allows an installation to suppress SVC dumps and SYSMDUMP ABEND dumps that are not needed because they duplicate previously written dumps. To identify the cause of previous and requested dumps, DAE uses symptom strings, which contain data that describes a problem. DAE stores these symptom strings in a DAE data set that you provide.

You can use the DAE data set in a single-system environment, or the systems in a sysplex can share a single DAE data set. The way you use the DAE data set determines how you specify the data set to global resource serialization to avoid contention for that data set.

Using DAE in a Single-System Environment

The default DAE data set, SYS1.DAE, is already defined in the default RNLs as a local resource. IBM suggests that you name the DAE data set SYS1.DAE in a single system because you do not need to change the data set definition for global resource serialization.

For a single system, the default RNLs contain the generic qname entry SYSDSN in the SYSTEM inclusion RNL and a specific rname entry for SYS1.DAE in the SYSTEMS exclusion RNL.

Note: If you intend to use a DAE data set name other than SYS1.DAE in a single system environment, you must add that DAE data set name as a specific rname entry in the SYSTEMS exclusion RNL to avoid contention when more than one system uses the same DAE data set name for different physical data sets.

Using DAE in a Sysplex

The shared DAE data set, if it is not named SYS1.DAE, is already set up as a global resource. IBM suggests that you provide a name other than SYS1.DAE for the DAE data set to be shared in the sysplex because then you do not need to change the data set definition for global resource serialization.

For a sysplex, the default RNL contains the generic qname entry SYSDSN in the SYSTEM inclusion RNL. An entry is not needed in the SYSTEMS exclusion RNL for a global resource.

Note: If you intend to use SYS1.DAE as the shared DAE data set for a sysplex, you must remove the specific rname entry for the SYS1.DAE data set from the default SYSTEMS exclusion RNL.

DB2

DFSMShsm

If you are using Hierarchical Storage Manager (DFSMShsm) on all systems in the complex that run DFSMShsm, no action is necessary for user data set serialization, as long as SYSDSN is named in the SYSTEM inclusion RNL.

For information about protecting DFSMShsm authorized queue names, see "Authorized qnames" on page 18.

Note: When using SMS-managed DASD volumes with DFSMShsm and USERDATASETSERIALIZATION active, DFSMShsm tries to delete temporary data sets during automatic primary space management. If no global serialization exists, or if global serialization is not performed for temporary data sets, you can delete an in-use temporary data set. You can also convert some of the reserves that DFHSM issues, such as ARCBACV and ARCMIGV. See *z/OS DFSMShsm Implementation and Customization Guide* for information about DFSMShsm reserves and how to decide whether or not to convert them.

SMS

You should place resource name IGDCDSXS in the RESERVE conversion RNL as a generic entry to serialize the access to SMS control data sets, ACDS, SCDS and COMMDS, across multiple systems. This will minimize delays due to contention for resources and prevent deadlocks associated with the VARY SMS command. For more information, see *z/OS MVS System Commands*.

Note: If there are multiple SMS complexes within a global resource serialization complex, be sure to use unique COMMDS, ACDS and SCDS data set names to prevent false contention. For information on allocating COMMDS, ACDS and SCDS data set names, see *z/OS DFSMS Implementing System-Managed Storage*.

IMS

For IMS, the qname of DSPURI01 can be managed either as always:

- being forced to use a RESERVE (by placing an entry in the SYSTEMS exclusion list).
- having a global resource serialization ENQ manage serialization to the resource.

ISPF or ISPF/PDF

To serialize access to resources with concurrent batch or TSO/E use of the resources, ISPF relies on MVS allocation (qname of SYSDSN).

To ensure the integrity of shared data, batch, or TSO/E, users who are updating a data set must allocate it with DISP=OLD.

Because MVS allocation does not satisfy an exclusive request and a shared request for the same resource at the same time, data set integrity is maintained between ISPF users and batch or TSO/E users.

To serialize access to partitioned data sets among multiple ISPF users, ISPF also issues its own ENQ, DEQ, and RESERVE macros.

To allow users to update a data set that has a record format of "U", ISPF serializes with the linkage editor to protect the entire partitioned data set.

Note:

1. If both ISPF and SPF (meaning a pre-ISPF product, such as 5668-009 or 5740-XT8) are installed on the same system, there is a danger of destroying partitioned data sets that are being updated. This problem can occur when ISPF and SPF update the same data set at the same time.

2. A partitioned data set extended (PDSE) is a data set that, like a PDS, allows you to partition data into members. Users can access PDSE members more easily and quickly than they can PDS members. For information about PDSE data sharing, see *z/OS DFSMS Using Data Sets*.

If you use ISPF on more than one system in the complex and do not use SPF on any system in the complex:

1. Place entries in the SYSTEMS exclusion RNL for the SPFEDIT data sets that cannot be global resources. For example, create SPFEDIT entries for the system data sets that cannot be shared. These data sets have the same rnames as those specified for SYSDSN in the default SYSTEMS exclusion RNL (shown earlier in Figure 10 on page 34).
2. Place entries in the RESERVE conversion RNL to convert the SPFEDIT reserves ISPF issues for data sets that you want global resource serialization to protect as global resources.

   Note: If your complex includes both a system running ISPF and a system running SPF, you cannot use global resource serialization to serialize access to global resources among ISPF and SPF users. You must not include entries for either SPFEDIT or SPFDSN in the RESERVE conversion RNL.
3. Resources that ISPF users on more than one system might share with batch users or TSO/E users must be global resources defined with both a qname of SYSDSN and a qname of SPFEDIT. That is, you must define, either explicitly or by default, an entry in the SYSTEM inclusion RNL for SYSDSN,dsname.

JES2

Global resource serialization can work well in a JES2 environment because it can replace job scheduling as a method of preserving the integrity of the data on shared DASD volumes. All of the considerations on temporary data sets and TSO/E, however, apply to a JES2 environment, and there is also an additional and important consideration.

Handling the reserve for the checkpoint data set in a JES2 multi-access spool configuration is a particularly critical decision. See *z/OS JES2 Initialization and Tuning Guide* for more information.

The location of the checkpoint data set, however, is another factor that affects your decision. There are two basic situations:

1. The checkpoint data set is the only data set on a volume or resides on a volume that contains no data sets that are ever serialized by reserves. In this case, do not convert the reserve; include the name of the checkpoint data set in the SYSTEMS exclusion RNL.
2. The checkpoint data set resides on a volume that contains other data sets that are serialized by reserves. Such use of a volume contradicts suggestions for placement of the JES2 checkpoint data set. If your installation requires such use, then you must convert the reserves for all resources on the volume, including the checkpoint data set. In this case, include the names of all data sets in the RESERVE conversion RNL.

Note: If your installation also uses an alternate checkpoint data set, the reserve JES2 issues for the primary checkpoint data set also provides serialization for the alternate. Checkpoint duplexing thus does not require additional action.

JES3

The major advantage of global resource serialization in a JES3 environment lies in the conversion of reserves to avoid interlocks and reduce contention for data sets on shared DASD volumes. Global resource serialization also provides the only way to serialize access across multiple systems to new nonspecific DASD data sets that the Storage Management Subsystem (SMS) does not manage.

JES3 cannot tolerate the processing delay required to protect its checkpoint data set as a global resource rather than with a reserve. The possible actions are the same as those described under "JES2" on page 37.

System logger

If you are using system logger in a sysplex environment, and you are not using single-system scope couple data set (CDS) types, then no further action is necessary for logger data set serialization, as long as SYSDSN is named in the SYSTEM inclusion RNL.However, if you are using single-system scope CDS types, then you must ensure that the SYSDSN related allocations for the system logger dataset allocations on the systems using either the CDS data type LOGRY or LOGRZ are in the SYSTEMS exclusion RNL so they do not result in GRS Complex wide SYSTEMS scope ENQs.

For more information on system logger requirements and guidelines for handling log stream data sets, see "Add the DASD data sets to the GRSRNL inclusion and exclusion list" in *z/OS MVS Setting Up a Sysplex*.

hlq
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.

lsname
Specifies the portion of the log stream name that limits the scope of the RNLDEF statement to the System Logger-created data sets.

Using the logrec recording medium

The logrec recording medium can be either a logrec data set or a logrec log stream. The logrec recording medium contains data about machine failures, as well as records for program error recording, missing-interrupt information, dynamic device reconfiguration (DDR) routines, and other related data.

If you are using a logrec log stream, follow the instructions above for specifying RNLDEF statements with the high-level qualifier.

If you are using a logrec data set, the default logrec data set name, SYS1.LOGREC, is defined in the default SYSTEMS exclusion RNL as a local resource.

For a single system, if you name the logrec data set SYS1.LOGREC, you do not need to change the data set definition for global resource serialization. If you name the data set something other than SYS1.LOGREC, then there is no need to add the name to the SYSTEMS exclusion RNL.

In a global resource serialization complex environment, IBM suggests you define a unique name for the logrec data set on each system. And, if you do, IBM also suggests that you *not* put this name in the SYSTEMS exclusion RNL.

RACF

If you are sharing a RACF® database across systems through DASD, you should consider converting the reserves used to serialize the volume, as long as all systems that access the RACF database are MVS systems that are part of the same Global Resource Serialization complex. For example, if a VM system is sharing access to the RACF database, you cannot convert the reserves.

To convert the reserves, place a generic entry for SYSZRACF in the reserve conversion RNL. In a mixed complex (meaning at least one system in the global resource serialization complex is outside of the multisystem sysplex), when you convert the SYSZRACF reserves you **must** reply to any global resource serialization messages related to ring disruptions or restarts from an MCS console. Failure to reply or replying to the message after issuing an operator command might result in a system hang condition.

In any case, do not put an entry that begins with 'SYSZRACF' or 'SYSZRAC2' in the SYSTEM inclusion RNL.

Guideline: IBM recommends moving all catalog and VSAM data sets off any volume that holds a RACF database.

Tape volumes

If your installation uses tape drives across multiple MVS images and volume serial numbers are unique, add a generic entry for SYSZVOLS to the SYSTEM inclusion RNL to ensure that a tape volume is used by only one job at a time. If you are using automatic tape switching, add a generic entry for SYSZVOLS to the SYSTEM inclusion RNL to prevent a system from holding a tape device while it waits for a mount of a volume that is being used by another system.

If tape drives are not being used by multiple MVS images, no RNL entry is required.

Temporary data sets

If you take no action, global resource serialization treats non-VIO temporary data sets as global resources. One reason for letting temporary data sets be global resources is that your installation can then run scratch functions (such as SCRATCH VTOC,SYS) safely at any time against shared volumes.

If your installation wants global resource serialization to treat temporary data sets as local resources, the data set names must appear in the SYSTEMS exclusion RNL. However, the format of a temporary data set name is not compatible with the format of a generic RNL entry. So a pattern entry that contains wildcard characters is necessary to extend the matching specification. The format of a temporary data set name is one of the following:

SYSyyddd.Thhmmss.RA000.jobname.Rggnnnnn
SYSyyddd.Thhmmss.RA000.jobname.dsname.Hgg

If you want GRS to treat temporary data sets as local resources, use the following RNLDEF in the appropriate GRSRNLxx parm member to create a pattern RNL entry.

- RNLDEF RNL(EXCL) TYPE(PATTERN)
- QNAME(*)
- RNAME(SYS?????.T??????.RA000.*)

For more information on a pattern resource name, see “How the RNL is scanned” on page 21.

TSO/E

When one or more of the systems in your global resource serialization complex runs TSO/E, you must decide whether or not to treat SYS1.UADS and SYS1.BRODCAST as global resources and how to handle user data sets. If your installation uses the RACF database in place of SYS1.UADS, see “RACF” on page 38.

If your installation runs ISPF, see the considerations identified under “ISPF or ISPF/PDF” on page 36. How you handle user data sets is especially critical in a JES2 environment.

SYS1.UADS and SYS1.BRODCAST

The default SYSTEMS exclusion RNL includes entries for SYS1.UADS and SYS1.BRODCAST, causing them to be local resources while you decide whether your installation wants them to be global resources. To make this decision, your installation must investigate and measure:

- Resource requirements (that is, the resources required to merge multiple versions of the two data sets into a single version of each and test the new versions)
- Performance implications (that is, the performance of one version of each data set accessed by all users in contrast to multiple versions of the same data sets each accessed by a subset of those users)

There are, however, significant advantages to treating SYS1.UADS and SYS1.BRODCAST as global resources:

- Your installation has only two data sets to maintain, rather than two data sets for each TSO/E system in the complex.
- A user can logon from any system in the complex, allowing a better workload balance.

- For foreground-initiated background jobs, a user who specifies NOTIFY will always receive the job-ended message regardless of which system in the complex processed the job.

To cause global resource serialization to treat SYS1.UADS and SYS1.BRODCAST as global resources, you must:

1. Merge all existing versions of SYS1.UADS and SYS1.BRODCAST into a single version of each data set.
2. Modify the default RNLs:
   a. Delete the entries for SYS1.UADS and SYS1.BRODCAST from the SYSTEMS exclusion RNL.
   b. Add SYSIKJUA as a generic qname entry in the SYSTEM inclusion RNL to make SYS1.UADS a global resource.
   c. Add SYSIKJBC as a generic qname entry in the SYSTEM inclusion RNL to make SYS1.BRODCAST a global resource.

Note: This procedure prohibits you from logging on more than once in a SYSPLEX if the scope is converted to SYSTEMS.

Related information appears in *z/OS TSO/E Customization*. Some of the information is repeated here for your convenience.

TSO/E user data sets

The data sets that are used only by TSO/E users include:

- Private user data sets that are not shared by other users or by batch jobs
- Temporary user-related data sets, such as ISPF, log, or recovery data sets
- Shared data sets, such as program libraries

If you use the default RNLs, the SYSDSN entry in the SYSTEM inclusion RNL defines all of these data sets as global resources.

If your installation wants all of them or some of them to be local resources, you must exclude them from global serialization. You can, if the structure of your userid allows, place a generic entry in the SYSTEMS exclusion RNL to define all TSO/E user data sets as local resources. If, however, a user might logon to different systems at different times, that user's data sets must be global resources. Place a generic entry for the userid in the SYSTEM inclusion RNL, and do not place an entry for the user in the SYSTEMS exclusion RNL.

If the structure of your userid does not allow you to create a generic entry to define all TSO/E user data sets as local resources, you can place a generic entry in the SYSTEMS exclusion RNL for the userid of each user whose TSO/E data sets are to be local resources. Omit the entry for a user whose TSO/E data sets are to be global resources. This method, however, might cause a very long RNL that you would have to change frequently. If the problem is significant at your installation, you might want to create an ISGNQXIT or ISGNQXITFAST exit routine to recognize the TSO/E user data sets that are to be local resources and exclude them from global serialization. See *z/OS MVS Installation Exits*.

VIO journaling data set

The VIO journaling data set is an optional VSAM data set that contains auxiliary storage management records for virtual I/O (VIO) data sets that the system saves across job steps and between IPLs. The default journaling data set name, SYS1.STGINDEX, is defined in the default SYSTEMS exclusion RNL as a local resource.

VIO Journaling Data Set in a Single System Environment:

For a single system, if you name the journaling data set SYS1.STGINDEX, you do not need to change the data set definition for global resource serialization. If you name the data set something other than SYS1.STGINDEX, there is no need to add the name to the SYSTEMS exclusion RNL. If your installation does not use the journaling data set, then no action is required in the SYSTEMS exclusion RNL.

VIO Journaling Data Set in a Multisystem Environment:

In a global resource serialization complex environment, IBM suggests that you define a unique name for the journaling data set on each system. And, if you do, IBM also suggests that you **not** put this name in the SYSTEMS exclusion RNL.

VSAM data sets

If you take no action, SYSVSAM data set serialization is global.

Do not place an entry for SYSVSAM in the SYSTEM inclusion RNL. Every SYSVSAM request that needs the SYSTEMS attribute already has it. (Placing an entry for SYSVSAM in the SYSTEM inclusion RNL can degrade performance.

Catalogs

Catalogs are VSAM data sets, but VSAM recognizes, catalogs, and manages access to them in a special way. Converting catalog reserves can reduce contention caused by catalog activity. Depending on your system configuration, you need to do the following to convert catalog reserves:

1. To avoid a data integrity exposure, verify that all systems using a catalog are part of the complex.
2. Place a generic entry for SYSIGGV2 in the RESERVE conversion RNL.

   Do not place an entry for SYSIGGV2 in the SYSTEM inclusion RNL. Every SYSIGGV2 request that needs the SYSTEMS attribute already has it. Placing an entry for SYSIGGV2 in the SYSTEM inclusion RNL can degrade performance.
3. If you have catalogs used by any systems that are not part of the complex, you must place specific SYSIGGV2 entries for these catalogs in the SYSTEMS exclusion RNL. The catalog name must be either 20 bytes or 44 bytes in length depending on the name of the catalog. When the catalog name is 20 bytes or less, the RNAME must be 20 characters in length. When the catalog name is greater than 20 bytes, the RNAME must be 44 bytes. The RNL TYPE must be SPECIFIC.

   For example, specify one of the following depending on the length of the catalog name:

   ```
   RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSIGGV2)
   RNAME('your.short.name.cat ')
   ```

   ```
   RNLDEF RNL(EXCL) TYPE(SPECIFIC) QNAME(SYSIGGV2)
   RNAME('your.longer.than.twenty.byte.catalog.name ')
   ```

   The second entry in the exclusion RNL for each catalog that is used by systems that are not part of the complex must be 88 bytes in length, a pattern entry with the first 44 bytes entered as question marks followed by the catalog name padded with spaces for the second 44 bytes of the entry. For example, the second entry using the catalog names from the previous example:

   ```
   RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSIGGV2)
   RNAME('????????????????????????????????????????????your.short.cat.name')
   ```

   ```
   RNLDEF RNL(EXCL) TYPE(PATTERN) QNAME(SYSIGGV2)
   RNAME('????????????????????????????????????????????
   your.longer.than.twenty.byte.catalog.name')
   ```

   See the GRSRNLxx parmlib member in *z/OS MVS Initialization and Tuning Reference* for more information on defining the RNLDEF statement.
4. IBM recommends SYSZVVDS and SYSVTOC entries be in the same global resource serialization RNL as either both converted, or both excluded. They must be converted in order to utilize the re-indexing of online volumes function.
5. SYSZBNDX uses the following rnames:
   - volser
   - ICKDSF HELD
   - volserRESERVE.

Rnames volser and ICKDSF HELD are regarding global ENQs and should not be in the SYSTEMS exclusion RNL. The volserRESERVE rname is regarding a RESERVE, and so normal RESERVE considerations apply; the installation should determine if the re-indexing of online volumes function could apply outside the complex and either add an entry to the SYSTEMS exclusion RNL or the conversion RNL to avoid double serialization.

RNL candidates

Based on these general guidelines, there are certain resources that are good candidates for a particular RNL. Table 3 on page 42 shows suggested resources for the SYSTEM inclusion RNL Table 4 on page 42 shows suggested resources for the RESERVE conversion RNL and, Table 5 on page 44 shows suggested resources for the SYSTEMS exclusion RNL.

Some of the examples below recommend a RESERVE over a global ENQ. This is because there are cases where the resource is only held for a short duration of time; a RESERVE request can perform better than a ring mode global ENQ. This is not the recommendation for star mode. There are also examples below for specific RESERVE conversions. In star mode, make sure that the conversion of all RESERVEs to global ENQs is with a single pattern entry if possible. See the restrictions listed in "Solving the RESERVE problems" on page 8.

For each resource shown, the tables include information on the resource name and a brief description of why you should consider placing the particular resource in the RNL.

Note:

You must specify the parts of the resource name shown in upper case letters exactly as shown, and you must replace the parts of the resource name shown in lower case letters with your installation-specific information.

Table 3. SYSTEM inclusion RNL guidelines

Qname	Rname	Notes
SYSDSN	dsname (optional)	Include a generic qname entry for SYSDSN to make data sets that go through MVS allocation global resources; use entries in the SYSTEMS exclusion RNL to make specific data sets local resources. The default RNL contains a generic entry for SYSDSN. A generic name is most useful, but the name can also be specific.
SYSIKJUA	none	You must include this entry if the TSO/E data set SYS1.UADS is to be a global resource. The name must be generic. Remember to delete the entry for SYSDSN,SYS1.UADS from the default SYSTEMS exclusion RNL.
SYSIKJBC	none	You must include this entry if the TSO/E data set SYS1.BRODCAST is to be a global resource. The name must be generic. Remember to delete the entry for SYSDSN,SYS1.BRODCAST from the default SYSTEMS exclusion RNL.
SYSZVOLS	none	Include a generic qname entry for SYSZVOLS to insure that a tape volume is only used by one job at a time if multiple MVS images are sharing tape drives and volume-serial numbers are unique. If tape drive(s) are not being used by multiple MVS images, this entry is not required.

Table 4. RESERVE conversion RNL Guidelines

Qname	Rname	Notes
SYSIEWLP	dsname (optional and padded to 44 bytes)	Include an entry to convert this reserve if all systems that share the resource are part of the complex. The entry serializes access between ISPF and the linkage editor. The name can be specific or generic.
SPZAPLIB	dsname	Include an entry to convert this reserve if all systems that share the resource are part of the complex. The name can be specific or generic.

Table 4. RESERVE conversion RNL Guidelines (continued)

Qname	Rname	Notes
SYSIGGV2	catalog dsname (optional and padded to 20 or 44 bytes)	Include this entry to convert the reserves for catalogs. Because the RNAME contains more than just the catalog name, you should use a generic specification. If you do not, updates to specific records in the catalog will not be properly serialized. **Requirement**: If you are sharing catalogs within a GRS complex, you must have a generic entry for SYSIGGV2 in the conversion RNL. Failure to do so can cause lockouts. See "Preventing Lockouts on Shared Volumes" in *z/OS DFSMS Managing Catalogs* for more information about the requirement. **Note:** Unless you have a single system with no sharing of catalogs and DASD volumes, you should use a generic entry for SYSIGGV2, SYSZVVDS and SYSVTOC in your GRSRNLxx member. These entries prevent having both a SYSTEMS level enqueue and hardware reserve issued for requests for these resources
SYSZRACF	none	Include this entry to convert the reserves for the RACF database. The name can be generic or specific.
SYSVTOC	volser	If you plan to run DFDSS DEFRAG or Softek/zDMF programs in a multisystem environment, you can convert SYSVTOC reserves by including a generic entry. Otherwise, SYSVTOC is generally not a good candidate for reserve conversion because reserves are of short duration and I/O intensive. **Requirement**: • The SYSZVVDS and SYSVTOC should be in the same RNL (both should go in the SYSTEMS Exclusion RNL or both should go in the RESERVE conversion RNL). • SYSZVVDS and SYSVTOC must be converted to use the re-indexing of online volumes function. See "Preventing Lockouts on Shared Volumes" in *z/OS DFSMS Managing Catalogs* for more information about the requirement.
SYSZVVDS	volser	Include this entry to treat SYSZVVDS reserves, which are of short duration and I/O intensive, as local resources. The name is generic. **Requirement**: • IBM recommends that SYSZVVDS and SYSVTOC be in the same RNL (both should go in the SYSTEMS Exclusion RNL or both should go in the RESERVE conversion RNL). • SYSZVVDS and SYSVTOC must be converted to use the re-indexing of online volumes function. See "Preventing Lockouts on Shared Volumes" in *z/OS DFSMS Managing Catalogs* for more information about the requirement.
SYSZBNDX	volserRESERVE	Use RNAME volserRESERVE. **Requirement**: This resource must be converted if you are using the re-indexing of online volumes function.
any	any	Include an entry for any reserve that you want to convert. Global resource serialization will treat the resource as a global resource, and the system will not issue the reserve. The name can be specific or generic.

Table 5. SYSTEMS exclusion RNL Guidelines

Qname	Rname	Notes
SYSVTOC	volser	Include this entry to treat SYSVTOC reserves, which are of short duration and I/O intensive, as local resources. The name is generic. **Requirement**: • IBM recommends that SYSZVVDS and SYSVTOC be in the same RNL (both should go in the SYSTEMS Exclusion RNL or both should go in the RESERVE conversion RNL). • SYSZVVDS and SYSVTOC must be converted to use the re-indexing of online volumes function. • In star mode, IBM recommends the conversion of all RESERVEs to global ENQs with a single pattern entry, when possible. Be sure to review the restrictions listed in "Solving the RESERVE problems" on page 8.
SYSZVVDS	volser	Include this entry to treat SYSZVVDS reserves, which are of short duration and I/O intensive, as local resources. The name is generic. **Requirement**: • IBM recommends that SYSZVVDS and SYSVTOC be in the same RNL (both should go in the SYSTEMS Exclusion RNL or both should go in the RESERVE conversion RNL). • SYSZVVDS and SYSVTOC must be converted to use the re-indexing of online volumes function. • In star mode, IBM recommends the conversion of all RESERVEs to global ENQs with a single pattern entry, when possible. Be sure to review the restrictions listed in "Solving the RESERVE problems" on page 8.
SYSZJES2		In a JES2 environment, include an entry for each checkpoint data set specified in the CKPTDEF initialization statement. The name should be generic. The qname is always SYSZJES2 See *z/OS JES2 Initialization and Tuning Guide* for more information. In star mode, IBM recommends the conversion of all RESERVEs to global ENQs with a single pattern entry, when possible. Be sure to review the restrictions listed in "Solving the RESERVE problems" on page 8.
	volser dsname	For MVS/SP Version 1.3.6 or MVS/SP Version 2 Release 1.5, use the volser and dsname specified for PRIMARY and DSNAME in CKPTDEF; for MVS/SP Version 2.2 or later releases, include a separate entry for each checkpoint data set and use the volser and dsname specified for CKPT1, CKPT2, NEWCKPT1, and NEWCKPT2 in CKPTDEF. See *z/OS JES2 Initialization and Tuning Reference*.
SYSZIAT	volser dsname	In a JES3 environment, include an entry for the checkpoint data set. The name must be generic. In star mode, IBM recommends the conversion of all RESERVEs to global ENQs with a single pattern entry, when possible. Be sure to review the restrictions listed in "Solving the RESERVE problems" on page 8.
SYSDSN	dsname	Include an entry for every system data set that is to be a local resource. The default RNLs contain several (see Figure 10 on page 34). The name can be specific or generic. In star mode, IBM recommends the conversion of all RESERVEs to global ENQs with a single pattern entry, when possible. Be sure to review the restrictions listed in "Solving the RESERVE problems" on page 8.
SYSVSAM	dsname (optional)	To make a VSAM data set a local resource, include an entry for the VSAM data set that is to be a local resource. To make all VSAM data sets local resources, use a generic qname entry for SYSVSAM. SYSVSAM and SYSDSN data set serialization must be consistent. In star mode, IBM recommends the conversion of all RESERVEs to global ENQs with a single pattern entry, when possible. Be sure to review the restrictions listed in "Solving the RESERVE problems" on page 8.

Table 5. SYSTEMS exclusion RNL Guidelines (continued)

Qname	Rname	Notes
SYSIGGV2		If you are excluding some but not all catalogs, see "Catalogs" on page 41 for information on how to exclude specific catalogs. If you are excluding all catalogs, then use a generic entry for SYSIGGV2 in the exclusion RNL. In star mode, IBM recommends the conversion of all RESERVEs to global ENQs with a single pattern entry, when possible. Be sure to review the restrictions listed in "Solving the RESERVE problems" on page 8.

Defining the RNLs

The RNLs on all systems in the complex must be the same; that is, each RNL on each system must contain the same resource name entries, and these resource name entries must appear in the same order.

The GRSRNLxx parmlib member is used to define the RNLs. During IPL, specifying the desired member(s) on the GRSRNL system parameter tells global resource serialization where it can find the RNLs. When the complex matches the sysplex, issuing the SET GRSRNL command tells global resource serialization where it can find the updated RNLs. However, the SET GRSRNL command does not work on any system in a mixed complex. Once you remove non-sysplex systems from the complex, so the sysplex matches the complex, you can use the SET GRSRNL command. See "Dynamic RNL processing" on page 97 for further information about using the SET GRSRNL command.

IBM supplies a default, member GRSRNL00 of SYS1.PARMLIB, that defines the default RNLs. See Figure 10 on page 34.

You can either use GRSRNL00 as is, modify it, or create additional GRSRNLxx members. You might find it useful to leave GRSRNL00 as is and use one or more separate members to define your own entries. Separating the IBM-supplied entries from your installation-dependent entries can make future migration easier. You might also find it useful to set up one member for generic entries and another member for specific entries.

z/OS MVS Initialization and Tuning Reference contains detailed information about specifying the entries in GRSRNLxx. The worksheet shown in Table 6 on page 46 lists the RNL entries IBM supplies in GRSRNL00 and includes blank slots for your own entries.

Each resource entry consists of:

- An RNL identifier
  - RNL(EXCL) for the SYSTEMS exclusion list
  - RNL(INCL) for the SYSTEM inclusion list
  - RNL(CON) for the RESERVE conversion list
- An entry type indicator

  TYPE(SPECIFIC)
  The entry is a specific name.

  TYPE(GENERIC)
  The entry is a generic name.

  TYPE(PATTERN)
  The entry is a pattern name.

- A qname, in the form QNAME(name).
- An rname, in the form RNAME(name). The rname is only required for TYPE(SPECIFIC)

Test GRSRNLxx carefully to ensure that it does not include syntax errors. In SYS1.SAMPLIB, IBM supplies a syntax checker that can detect such errors. Member SPPINST of SYS1.SAMPLIB contains information about using the syntax checker. To minimize the testing, create a single GRSRNLxx member, test it, and then copy it to the parmlib of all systems.

Table 6. GRSRNLxx Worksheet

List Identifier	Type Indicator	Qname	Rname
RNLDEF RNL(EXCL)	TYPE(SPECIFIC)	QNAME(SYSDSN)	RNAME(SYS1.BRODCAST)
RNLDEF RNL(EXCL)	TYPE(SPECIFIC)	QNAME(SYSDSN)	RNAME(SYS1.DAE)
RNLDEF RNL(EXCL)	TYPE(SPECIFIC)	QNAME(SYSDSN)	RNAME(SYS1.DCMLIB)
RNLDEF RNL(EXCL)	TYPE(GENERIC)	QNAME(SYSDSN)	RNAME(SYS1.DUMP)
RNLDEF RNL(EXCL)	TYPE(SPECIFIC)	QNAME(SYSDSN)	RNAME(SYS1.LOGREC)
RNLDEF RNL(EXCL)	TYPE(GENERIC)	QNAME(SYSDSN)	RNAME(SYS1.MAN)
RNLDEF RNL(EXCL)	TYPE(GENERIC)	QNAME(SYSDSN)	RNAME(SYS1.PAGE)
RNLDEF RNL(EXCL)	TYPE(SPECIFIC)	QNAME(SYSDSN)	RNAME(SYS1.STGINDEX)
RNLDEF RNL(EXCL)	TYPE(SPECIFIC)	QNAME(SYSDSN)	RNAME(SYS1.UADS)
RNLDEF RNL(INCL)	TYPE(GENERIC)	QNAME(SYSDSN)	

Table 6. GRSRNLxx Worksheet (continued)

List Identifier	Type Indicator	Qname	Rname

Chapter 3. Using the ENQ/RESERVE/DEQ monitor tool

The main objective of the global resource serialization monitor tool is to assist in planning the RNLs for global resource serialization implementation. The tool monitors ENQ, DEQ, and RESERVE requests, and collects data about the resources and the requesters.

Tip: To collect the same data but with less impact to the ENQ/DEQ path length, instead use SMF 87 subtype 2 records. See "Using SMF 87 records" on page 185 for more information.

The objectives of the tool are:

- Assist in planning the RNLs for global resource serialization implementation.
  - Find the name and the scope of ENQ.
  - Measure the rate and time of the RESERVEs.
- Reduce DASD contentions and interlock exposures by helping to identify RESERVEs for which changes to RNLs could reduce elapsed time for some jobs and to verify the results of RNL changes that are made.
  - Report RESERVE activities with total and maximum RESERVE time on shared DASD.
  - Report resources that contributed to the RESERVE time.
  - Verify the results when RNLs are implemented.
- Provide information to assist in the analysis of results of global resource serialization tuning.
  - Measure the global resource serialization ENQ delay for global enqueues, the actual RESMIL value, and the ENQ/DEQ requests rate.
  - Determine ENQ request patterns to help determine who the resource consumers are.
  - Trace global resource serialization ENQ delay using a major name SYSZENQM and a minor name ENQDELAY and minor name with delay value, date and time information (see Figure 14 on page 53, Figure 15 on page 54, and "ENQ/RESERVE/DEQ restrictions" on page 79).

An interactive ISPF application allows viewing and printing of the collected data. Alternatively, the data can be written to a sequential data set. Reports can be generated by modifying the sample procedures provided in SYS1.SAMPLIB. If the data is not written to datasets, it is collected in a dataspace associated with the monitor address space. The amount of data saved in the dataspace can be controlled by the DSP PARM (ses Figure 11 on page 51). The monitor stops collecting data and issues message ISGAU015 - DATA SPACE FULL when the dataspace is full.

Guideline: The monitor's intent is to assist with planning and tuning of RNLs for the implementation of global resource serialization. The monitor uses system resources in order to collect data. IBM recommends that you do not run the monitor continuously during normal production work. While running the monitor you might observe high CPU in address spaces that issue many ENQ/RESERVE/DEQ requests.

This contains the following topics:

- "Security considerations" on page 50
- "Setting up the monitor" on page 50
- "Monitor execution" on page 50
- "Starting the monitor" on page 52
- "Monitor control" on page 57
- "Messages - abends - return codes" on page 60
- "Filter facility" on page 62
- "Reports" on page 66

- “ENQ/RESERVE/DEQ restrictions” on page 79
- “Resources used and guidelines” on page 80
- “Monitor diagnostic information” on page 80
- “Monitor physical output record” on page 80
- “Monitor utilization hints” on page 81
- “ENQ/DEQ/RESERVE analysis aid reports” on page 82
- “Records layout” on page 85

Security considerations

The ENQ/RESERVE/DEQ monitor tool runs authorized as either a batch job or a started task. If the security administrator does not protect it, any unauthorized user can submit a job to start the monitor and gather ENQ and DEQ data. To prevent unauthorized users from doing this, use the RACF PROGRAM class to protect the ISGAUDIT program in SYS1.LINKLIB.

For detailed information about protecting global resource serialization services:

- *z/OS Planning for Multilevel Security and the Common Criteria*

For information about using the z/OS Security Server RACF program control and the PROGRAM class:

- *z/OS Security Server RACF Security Administrator's Guide*

Setting up the monitor

Use the following steps to set up the monitor tool.

1. Prepare the JCL to start the monitor. Sample JCL is ISGRUNAU in SYS1.SAMPLIB.
2. The ENQ Monitor supports the usage of one or two sequential data sets with required attributes **`BLKSIZE=3892,LRECL=3892 RECFM=F`** to collect the ENQ/DEQ data.
3. The data set names will be on the DD statements OUTPUT1 and OUTPUT2 in the JCL to start the monitor. Each block has room for 30 ENQ/DEQ SVCs. If the monitor collects 100 events per second, 12,000 blocks will be filled in one hour. One or both output data sets can be DUMMY. See “Monitor execution” on page 50 for additional information.
4. Allocate a data set used by the reports procedures with space allocation of TRK=(1,1), RECFM=FB, LRECL=132, BLKSIZE=18348 for DDNAME LOGR. The data set name is a variable in the reports procedures. See “Logs” on page 67 and Figure 38 on page 72.
5. Prepare the report procedures from the sample members ISGAJE1, ISGAJE1A, ISGAJE2, ISGAJE2A, ISGAJE3 and ISGAJE3A.

All the sample JCL is in SYS1.SAMPLIB, or see

- ISGAJE1 –Figure 36 on page 70
- ISGAJE1A – Figure 37 on page 71
- ISGAJE2 – Figure 38 on page 72
- ISGAJE2A – Figure 39 on page 74
- ISGAJE3 – Figure 40 on page 76
- ISGAJE3A – Figure 41 on page 78

For the procedure variables see “In-stream procedures variables” on page 68.

Monitor execution

The ENQ monitor can be started by either starting a procedure or submitting a batch job. This job should be considered as a non-ending job that is TIME=1440 or TIME=NONLIMIT.

Running the monitor on previous releases requires the systems programmer to ensure that the ENQ monitor is defined to run at the highest dispatch priority. As of z/OS V1R10, the ENQ monitor automatically runs at the right dispatching priority by joining a GRS Dependent Enclave. The enclave makes the monitor run as an extension of the GRS address space. This results in CPU time accumulated by the monitor to be added to the GRS address space's CPU time. Not running the monitor at the highest dispatching priority can affect ENQ/DEQ processing.

You should be aware that the monitor elongates the path length of every ENQ, DEQ, RESERVE, and ISGENQ. The elongation was substantially reduced in z/OS V1R10, but it still might not be acceptable. Before running the monitor in production or for a long time, make sure that you understand the impact of the elongation. Running the SYS1.SAMPLIB(ISGNQRSP) program with or without the active monitor and doing comparisons on the response times can help the user to make a decision.

```
//* Function: Run this job to start ENQ Monitor before starting
//*           ENQ Monitor ISPF interface or using the modify
//*           command.
//*
//* Operation: This JCL can be copied to the SYS1.PROCLIB and issue
//*            'S ISGRUNAU' from the MVS console to start the ENQ
//*            Monitor.
//*
//* Suggested modifications:
//*    1. Uncomment //*OUTPUTX if hardcopy output is needed.
//*    2. Modify //OUTPUTX DD DSN=USERID.AUDIT>OUTX,DISP=SHR to
//*       a correct pre-allocated dataset.
//*
//*       When allocates the dataset, it has to be in the following
//*       format:
//*              Organization  . . . : PS
//*              Record format . . . : F
//*              Record length . . . : 3892
//*              Block size  . . . . : 3892
//*
//* Recovery Operations:  None
//*
//* Distribution Library:  ASAMPLIB
//JOBNAME  JOB CLASS=A,MSGCLASS=A,TIME=NOLIMIT
//*
//*  JOB TO START THE MONITOR
//*
//*
//JOBLIB DD DSN=SYS1.LINKLIB,DISP=SHR
//STEP001 EXEC PGM=ISGAUDIT,PARM='DSP=32'
//OUTPUT1 DD DSN=USERID.AUDIT.OUT1,DISP=SHR
//OUTPUT2 DD DSN=USERID.AUDIT.OUT2,DISP=SHR
//*OUTPUT1 DD DUMMY
//*OUTPUT2 DD DUMMY
//SYSUDUMP DD SYSOUT=*
```

Figure 11. JCL to Run the Monitor

The optional keyword DSP on the PARM field can be used to specify the desired size of the dataspace in MB (megabytes). **DSP=nnn** which can range from **1 to 999**. The default dataspace size is 32 MB.

The two output sequential data sets must be different data sets or can be DUMMY data sets. If both data sets are DUMMY, data is collected in storage. When one data set becomes full, the monitor automatically switches to the alternate. If all specified data sets are full, the monitor terminates..

When the monitor is restarted, it will append the new data to the end of OUTPUT1 data set. It is possible to process the data collected while the monitor is active. This can be done either by switching to the alternate data set with the MODIFY option SW command (see “Monitor control” on page 57) or by selecting the data to process with the DATE keyword with the TO-TIME less than the actual time (see “Report programs parameters” on page 68).

If the events' rate causes loss of data, consider using a 3990-3 DASD FAST WRITE facility. Data can be lost if an unexpected error occurs during the process of copying information to the ENQ Monitor's address space. The MODIFY option L command and ISPF main panel display the number of lost events if loss of data occurs.

You can invoke an ISPF application to display and print the data saved in the dataspace. For additional information see “Starting the monitor” on page 52 and “Monitor control” on page 57.

When the dataspace is full, the monitor stops collecting data. The ISPF application can still display the dataspace information.

If the monitor is started as a procedure, it cannot be canceled or forced, it can only be stopped. If the monitor is submitted as a batch job, it can be cancelled, stopped, or forced.

Starting the monitor

To start the application, execute the SYS1.SBLSCLI0(ISGACLS0) EXEC from the ISPF/PDF Option 6 (Command).

Figure 12 on page 52 shows the Main Menu displayed when global resource serialization is in ring configuration.

```
                          ENQ/DEQ Monitor - Main Menu
Select an option:

__  1.  MAJOR Names                 Date &   Time      : 1995.160   06:00
    2.  Resource Name List          Monitor started at : 1995.159   10:5
    3.  Volume List                 Elapsed seconds    :            68790
    4.  Filter List                 SMF System ID      :             SC47
-------------------------------------------------------------------------------
GRS Ring -> From: SC49      To: SC52      This: SC47      NUMSYS: 7
-------------------------------------------------------------------------------
                                  Time of Delay High. . : 1995.160 04:05:28
Global Requests . . . :    24885  Enqueue Delay Hi - Low:     1260    12
Local  Requests . . . :   140708  Enqueue Delay     msec:       31

Major Names . . . . . :       39  ACCELSYS. . . . . . . :        2
Minor Names . . . . . :     1516  RESMIL  . . . . . msec:        0
Volumes . . . . . . . :       11  Data Space Used .bytes:   221900   10 %
Number of Events. . . :   344009  Active Filter. . . . .:       08
Lost Events . . . . . :        0  Events Rate  . . . . .:        3

Command ===> ____________________________________________________________
F1=Help   F2=Split  F3=Exit   F9=Swap   F12=Cancel
```

Figure 12. GRS Ring Main Menu - ISPF Application

If the monitor is active, the monitor automatically recognizes migration from global resource serialization ring to star configuration. Figure 13 on page 52 shows the Main Menu displayed when global resource serialization is in a star configuration.

```
                          ENQ/DEQ Monitor - Main Menu
Select an option:

__  1.  MAJOR Names                 Date &   Time      : 1996.241   06:04
    2.  Resource Name List          Monitor started at : 1996.241   04:22
    3.  Volume List                 Elapsed seconds    :           6165
    4.  Filter List                 SMF System ID      :           SC47
-------------------------------------------------------------------------------
GRS Star -> Number of Lock Entries: 1048576               NUMSYS: 10
-------------------------------------------------------------------------------
                                  Time of Delay High. . : 1996.241 05:40:00
Global Requests . . . :     4159  Enqueue Delay Hi - Low:   192484    302
Local  Requests . . . :    12194  Enqueue Delay  mic-sec:      405

Major Names . . . . . :       44
Minor Names . . . . . :      292
Volumes . . . . . . . :        2  Data Space Used .bytes:    47276    2 %
Number of Events. . . :    33762  Active Filter. . . . .:       08
Lost Events . . . . . :        0  Events Rate  . . . . .:        5

Command ===> ____________________________________________________________
F1=Help   F2=Split  F3=Exit   F9=Swap   F12=Cancel
```

Figure 13. GRS Star Main Menu - ISPF Application

Press **PF1** to display a help panel.

Option 1

Figure 14 on page 53 shows the panel displayed when Option 1 is selected.

```
                    ENQ/DEQ Monitor - Major Name List      ROW 1 TO 10 OF 65

Enter S to select a Major Name for details     .
      L major on command line to locate a Major.   Elapsed seconds:    6640

   Sel.   ----------   -------  ----   ----    -------  -average-   -Reserved-
   Field  Major Name   Scope    Exit   RNL     Counter     msec       seconds
   _      SYSZVVDS     SYSS             CONV       3554       23            86
   _      SYSZVVDS     SYS                         3152
   _      SYSZSDSF     SYSS             INCL         32
   _      SYSZRAC2     SYS                         3665
   _      SYSZRACF     RES              EXCL        338       39            13
   _      SYSZPSWD     SYS                          15
   _      SYSZMCS      SYS                         1053
   _      SYSZLOGR     SYS      YES                 37
   _      SYSZLLA1     SYS                         117
   _      SYSZJES2     SYS                         386
   _      SYSZJES2     RES                        1125     1101         1251
   _      SYSZISTH     SYS                          16
   _      QNAME        SYS                          17

Command ===> _____________________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=Print     F5=Sort_maj  F6=Sort_scp
F7=Backward  F8=Forward   F9=Swap     F10=Sort_cnt F11=Sort_avg F12=Sort_res
```

Figure 14. Option 1 - Major Name List

Option 1 shows the ENQ activities listed by major names, RNL action and the number of global and local ENQs if global resource serialization is active.

Press **PF1** to display a help panel.

Press **PF4** to print the entire major name list.

Press **PF5, PF6, PF10, PF11, PF12** to sort on major name, scope, count, average, and reserved seconds in that order.

Command **L major_name** locates the major name and position if it exists in the collected data.

Select a major name on the displayed row to get information about the minor names associated with that major name.

- Major name column displays the major name of the resource.
- Scope column displays the resource scope. An asterisk (*) to the left of the resource scope indicates a change of resource scope after scanning the RNLs.

  For example,

  – *SYSS on the scope column with INCL on the RNL column means the resource scope was changed from SYSTEM to SYSTEMS.
  – *SYS on the scope column with EXCL on the RNL column means the resource was changed from SYSTEMS to SYSTEM.
  – SYSS on the scope column with CONV on the RNL column means the resource scope was changed from a RESERVE macro to an ENQ with a scope of SYSTEMS. The RESERVE on the DASD will be suppressed.
  – RES on the scope column with EXCL on the RNL column means the resource scope was changed from SYSTEMS to SYSTEM. The RESERVE on the DASD will not be suppressed.
- Exit column indicates whether the request was altered by one of the ENQ/DEQ installation exit routines.
- RNL column displays the action taken by RNL processing.
  – Blank in the column means no change in the resource scope after scanning the RNLs.

  - CONV means the resource name matched an entry in the RESERVE Conversion List.
  - INCL means the resource name matched an entry in the SYSTEMS Inclusion List.
  - EXCL means the resource name matched an entry in the SYSTEM Exclusion List.
  - NO means RNL=NO keyword was specified with the resource name.
- Counter column displays the total number of requests on this major name.

Figure 15 on page 54 shows the panel displayed when you select a specific major.

```
                    ENQ/DEQ Monitor - Minor Name List       ROW 1 TO 9 OF 17


Minor Name list for:              Major Name  : QNAME
                                  RNL . . . . :
                                  Scope . . . : SYSTEM

Enter S to select a Minor Name for Jobnames    .
      L min.  on command line to locate a Minor.

  Sel.   ------------------------------------------------  --------
  Field  Minor Name:                                       Counter:
  _      01292787 95.160 04:05:28                                 1
  _      01120841 95.160 04:33:16                                 1
  _      01075989 95.160 04:31:58                                 1
  _      01056722 95.160 01:24:29                                 1
  _      00954460 95.160 01:58:21                                 1
  _      00893109 95.159 22:39:37                                 1
  _      00553288 95.160 06:06:09                                 1
  _      00526344 95.160 03:25:03                                 1
  _      00510494 95.160 05:37:46                                 1

Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=Print     F5=Sort_min  F6=Sort_cnt
F7=Backward  F8=Forward   F9=Swap     F10=Sort_tme
```

Figure 15. Selected Row - Minor Name List

Press **PF1** to display a help panel.

Press **PF4** to print the entire minor name list.

Press **PF5, PF6** to sort on minor name, counter.

Select a minor name on a displayed row to get information about jobnames, userids and program names involved in the ENQ or RESERVE processing for that resource.

Command **L minor_name** will locate the minor name if it exists in the collected data.

Press **PF10** to sort minor names by timestamp.

Selecting a specific minor name results in a display of the job and program names with indication of exclusive or shared use.

Press **PF5 and PF6** to sort on jobname, program name. Figure 16 on page 55 is an example of the panel the monitor displays when you select a specific minor name.

```
                      ENQ/DEQ Monitor - Jobname List    ROW 1 TO 1 OF 1


Jobname list for Major Name  : QNAME
                 Minor Name  : 01292787 95.160 04:05:28
                 Minor Length: 24
 --------  --------   ----------       -------- ---       ----------
 Job_name  User_ID    Enqs x Job       Pgm_name E/S       Enqs x PGM
 MONITOR   STC                 1       SVC-255   E                 1
****************************** BOTTOM OF DATA ************************

Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit    F4=Print   F5=S_enq_j   F6=S_enq_p
F7=Backward  F8=Forward   F9=Swap
```

Figure 16. Selected Row - Jobname List

Only 52 bytes of minor names will be displayed.

Press **PF1** to display a help panel.

Press **PF4** to print the entire jobname list.

Option 2

Option 2 of the main menu shows a selection panel to display the active RNL's entries. This option offers the same result as MVS D GRS,RNL=INCL (INCLUSION, EXCLUSION, CONVERSION).

```
                   ENQ/DEQ Monitor - SYSTEM Inclusion Table     Row 1 from 184

 RNL SYSTEM Inclusion Table:

 ------------------------------------------------------------------------------
 SPECIFIC    QNAME       RN252A
 SPECIFIC    QNAME       RN252C
 SPECIFIC    QNAME       RN255D
 SPECIFIC    QNAME       RN255E
 SPECIFIC    QNAME       RN255F
 SPECIFIC    QNAME       IQS301
 SPECIFIC    QNAME       IQS302
 SPECIFIC    QNAME       IQS082
 SPECIFIC    GRTEDR02    VAR#GRVNORNLR013
 SPECIFIC    GRTEDR02    VAR#GRVNORNLR014
 SPECIFIC    GRTEDR02    VAR#GRVNORNLR017
 SPECIFIC    GRTEDR02    VAR#GRVNORNLR018
 SPECIFIC    GRTEDR02    VAR#GRVNORNLR025
 SPECIFIC    GRTEDR02    VAR#GRVNORNLR026
 SPECIFIC    GRTEDR02    VAR#GRVNORNLR029

 Command ===> ____________________________________________________________
  F1=Help      F2=Split     F3=Exit      F7=Backward  F8=Forward   F9=Swap
 F12=Cancel
```

Figure 17. Option 2 - Resource Name List Table

Option 3

Option 3 of the main menu shows RESERVE activities.

```
                        ENQ/DEQ Monitor - VOLUME List

 Enter S to  select a Volume for details
       A for active Reserves on  Volume
       L volume on command line to locate a Volume

-- ------ --------- Dev. Max ---------- ------------ Reserve Time -------
S. Vol.   Total Res nbr  Res Elap.(sec)  Avg.(ms)  Min.(ms)  Max.(ms) Tot
 _ STSO04       201 02AD  03       1496       159        11       845
 _ STSO02       582 01A6  05       1485        21         6       506
 _ STSO01      1114 01AD  04       1496        43         6      2951
 _ RUBB01        61 0234  01       1477       105        34       488
 _ RES51B         1 02A0  01       1258     26784     26784     26784
 _ RES51A       129 0236  02       1499        31         8       629
 _ PR0003         1 02AF  01       1478        55        55        55
 _ PR0002         1 01AC  01       1478        44        44        44
 _ PR0001       261 01AB  03       1479        18         8       179
 _ HSMMG1        12 01A1  02       1129        82        17       364
 _ DB0003         1 02A9  01       1480        43        43        43
 _ DB0002         1 01AA  01       1481        46        46        46
 _ DB0001        12 01A9  04       1482        33        12       359
 _ CAT510      2136 0237  06       1499       146         6      1334

Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=Print     F5=Sort_vol  F6=Sort_cnt
F7=Backward  F8=Forward   F9=Swap     F10=Sort_avg F11=Sort_max F12=Sort_tot
```

Figure 18. Option 3 - Volume List

Figure 18 on page 56 shows the panel displayed when Option 3 is selected.

Select S on a single volume to display RESERVE activities on that volume. Figure 19 on page 56 shows the information displayed when S is selected on a single volume.

```
                    ENQ/DEQ Monitor - VOLUME Entry List      ROW 1 TO

Volume entry list for:

 Volser. . . . . : STSO01      Average Reserve Time (ms) : 32
 Tot.nr of Reserve : 487       Minimum Reserve Time (ms) : 6
 Dev.nr. . . . . : 01AD        Maximum Reserve Time (ms) : 1765
 Max Reserve Cnt. .: 04        Total   Reserve Time (sec): 15
 Elapsed Time (sec): 1594

Interval
- Rate - ------ -------- ----------------------  ------------ Time -----
S  min.   Count MajName  Minor name (max 22 ch)  Avg ms  Min ms  Max ms
_     0       5 SYSZVVDS STSO01 .                    31      30      32
_     0       2 SYSZVVDS STSO01 .                    22      21      23
_     0       4 SYSZVVDS STSO01 .                    29      27      31
_     8     214 SYSZVVDS STSO01                      19       4     133

Command ===> ____________________________________________________________
F1=Help      F2=Split     F3=Exit      F4=Print     F5=Sort_maj  F6=Sort_cnt
F7=Backward  F8=Forward   F9=Swap     F10=Sort_avg F11=Sort_max F12=Sort_tot
```

Figure 19. Option 3 - Volume Entry List (S is selected)

Select A on a single volume to display the active RESERVEs on the volume. Figure 20 on page 57 shows the information displayed when A is selected on a single volume with active RESERVEs.

```
                       ENQ/DEQ Monitor - Active Reserve List    Row 1 to 1 of 1

Active Reserve for :

 Volser. . . . . . : CAT510      Average Reserve Time (ms) : 288
 Tot.nr of Reserve : 5315        Minimum Reserve Time (ms) : 6
 Dev.nr. . . . . . : 0237        Maximum Reserve Time (ms) : 1438
 Max Reserve Cnt. .: 04          Total   Reserve Time (sec): 1553
 Elapsed Time (sec): 8083        Volume  Reserve Rate (min): 39

-------- ---------------------------------------- ----------- -------- --------
Major    Minor name (max 40 ch)                   Active msec Jobname  Pgmname
SYSZJES2 CAT510SYS1.HASPCKPT                              968 JES2     HASJES20
******************************* Bottom of data ********************************

Command ===> ____________________________________________________________
F1=Help    F2=Split   F3=Exit    F9=Swap   F12=Cancel
```

Figure 20. Option 3 - Volume Active Reserve List (A is selected)

Command **L volume_name** will locate the desired volume if it exists in the collected data.

Press **PF1** to display a help panel.

Press **PF4** to print the entire volume selection list.

Press **PF5, PF6, PF10, PF11, PF12** to sort on volume, total reserve count, average reserve time, maximum reserve time, and total reserve time, in that order.

Select a minor name to get the information about the jobnames and program names that issued RESERVE.

Note: The average, minimum, and maximum reserve time displayed in Figure 19 on page 56 and Figure 20 on page 57 can be out of sync if there is an active RESERVE request on that volume.

Option 4

Option 4 of the main menu shows the active filter.

Figure 21 on page 57 shows the panel displayed when you select option 4.

```
                         ENQ/DEQ Monitor - Filter  List         Row 1 to 4 of 4

Filter: 08                       DISCARD  SCOPE=STEP

Object:   Object name:                                                Action
--------  ------------------------------------------------------------ -------
MAJOR     SYSZTIOT                                                    DISCARD
MAJOR     SYSZJWTP                                                    DISCARD
RESOURCE  SYSDSN   USER.LINKLIB                                       DISCARD
RESOURCE  SYSDSN   SYS95                                              DISCARD
******************************* Bottom of data *******************************
```

Figure 21. Option 4 - Filter List

Monitor control

The monitor supports the MODIFY command with the following options:

- Change the filter table **(option I)**.
- List the active filter table, the active output data set, the number of events recorded, the number of lost events, the ENQ delay of a systems enqueue and the RESMIL value in milliseconds **(option L)**.

  Note: The monitor measures ENQ delay and RESMIL at the time of the command. The monitor encounters the same contention for the processor and storage as does any other address space in the system. Therefore, even after the system grants the global resource request, the monitor might not receive control immediately because of processor or storage contention, and the accuracy of the ENQ delay measurement.

RESMIL is not displayed if GRS=NONE is specified in IEASYSxx or global resource serialization is in a star complex.

- List the enqueue activities by MAJOR names, RNLs ACTION, if global resource serialization is active, and the approximate number of global and local requests (ENQs+DEQs) **(option T)**.
- List the RESERVE's activities by volume **(option V)**.
- Switch the active output data set **(option SW)**.
- List the active RNL's entries **(option R)**. This option offers a function similar to the MVS DISPLAY GRS,RNL=ALL command.
- Stop the monitor **(option STOP)**.
- Cancel the monitor if it is a batch job **(option C)**.

To stop the monitor the MVS P JOBNAME command can be used along with the MODIFY option STOP.

Figure 22 on page 58 shows all the supported options.

```
F  JOBNAME,I=xx          RE-INITIALIZE FILTER TABLE
                               xx RANGE FROM 00 TO 09

F  JOBNAME,L              DISPLAY INTERCEPTED SVCS
                                ACTIVE  DDNAME & FILTER-TABLE
                                NUMBER TRACED EVENTS
                                NUMBER LOST   EVENTS
                                ENQ  DELAY IN MICROSEC
                                RESMIL        IN MILLISEC

F  JOBNAME,T              DISPLAY THE ENQ/RES MAJOR-NAMES
                               ACTIVITIES

F  JOBNAME,T=MAJOR     DISPLAY THE ENQ/RES MAJOR-MINOR
                                 NAMES ACTIVITIES

F  JOBNAME,V                DISPLAY THE VOLUMES WITH RESERVE
                                 ACTIVITIES

F  JOBNAME,V=VOLSER     DISPLAY THE VOLUME's RESERVES


F  JOBNAME,T=FREEMAIN   RESET THE DATASPACE's CONTENT
                V=FREEMAIN

F  JOBNAME,R                 LIST THE ACTIVE RNLs

F  JOBNAME,SW              SWITCH OUTPUT DATA SET

F  JOBNAME,STOP           STOP THE MONITOR. PARTIALLY
                                FILLED BUFFER IS WRITTEN.

P  JOBNAME                   STOP THE MONITOR. PARTIALLY
                                 FILLED BUFFER IS WRITTEN.

C  JOBNAME                   ONLY IF NOT STARTED. PARTIALLY
                                 FILLED BUFFER IS NOT WRITTEN
```

Figure 22. AUDIT MODIFY Command Options

Figure 23 on page 58 shows sample output for option L if global resource serialization is in a ring configuration.

```
F JOBNAME,L

ISGAU008 - ISGAMF08 ACTIVE
ISGAU008 - OUTPUT1  ACTIVE
ISGAU008 - I=00004718 L=00000000 ENQ-DELAY=00009274 RESMIL=0010

I=xxx         number of events traced     L=yyy  number of lost events
ENQ-DELAY=zzz time in microsec required by a SYSTEMS ENQ to complete
```

Figure 23. AUDIT MODIFY Option L Output Example

Figure 24 on page 59 shows sample output for option L if global resource serialization is in a star complex or GRS=NONE is specified in IEASYSxx.

```
F JOBNAME,L

ISGAU008 - ISGAMF08 ACTIVE
ISGAU008 - OUTPUT1  ACTIVE
ISGAU008 - I=00004718 L=00000000 ENQ-DELAY=00009274

I=xxx         number of events traced     L=yyy  number of lost events
ENQ-DELAY=zzz time in microsec required by a SYSTEMS ENQ to complete
RESMIL=wwww   not displayed if GRS=NONE is specified in IEASYSxx or
              global resource serialization is in a star complex.
```

Figure 24. AUDIT MODIFY Option L Output Example (GRS=NONE or GRS in Star)

You can use the MODIFY option T command to find out the **major names** used in the system and to verify the **RNLs action**. For a sample output see Figure 25 on page 59 and Figure 26 on page 59.

```
F ISGRUNAU,T

  DATA SPACE USED=00024906 number in bytes
  ENQDELAY=00001245 result of the command being issued.
                    A single SYSTEMS scope ENQ/DEQ pair time is
                    measured and this how long it took in micro seconds.

  GLOBAL=00000006 LOCAL=00000979
  MINORS=00000170 number of unique RNAMEs
  VOLUMES=00000003  number of unique volumes specified on RESERVEs
  MAJORS=00000029 RNL number of unique QNAMEs


00000025 IGWLHANZ SYSTEM
00000013 SPFEDIT E*SYSTEM
00000065 SYSDSN SYSTEM
00000061 SYSIEFSD SYSTEM
00000141 SYSIGGV2 E*RESERVE
00000005 SYSIKJPL SYSTEM
00000002 SYSIKJUA SYSTEM
00000001 SYSVTOC E*SYSTEM
00000001 SYSZ#SSI N SYSTEM
00000004 SYSZAUDT N SYSTEMS

C*  ->  RESERVE-converted       SYSTEMS   HW-RESERVE eliminated
E*  ->  RESERVE-excluded        SYSTEM    HW-RESERVE issued
I*  ->  system -included   from SYSTEM    to SYSTEMS
E*  ->  systems-excluded   from SYSTEMS   to SYSTEM
N   ->  RNL=NO     request
```

Figure 25. AUDIT MODIFY Option T Example

The command MODIFY option T=major lists the minor names used by the selected major name. For a sample output see Figure 26 on page 59.

```
F JOBNAME,T=SYSVTOC

Rt COUNT VOLSER  DEVNO  MAX   ELAPSED SEC  AVE    MIN    MAX   TOTAL
min.                    RES.               msec.msec.msec.sec.
--------------------------------------
00000047 SYSVTOC      RESERVE
00 00010 STSO02   STSO02                  000183 000071 000325 000001
00 00014 RUBB01   RUBB01                  000114 000046 000153 000001
00 00003 PR0002   PR0002                  000108 000057 000141 000000
00 00007 PR0001   PR0001                  000130 000068 000147 000000
00 00001 HSMMG5   HSMMG5                  000067 000067 000067 000000
00 00002 HSMMG4   HSMMG4                  000102 000047 000158 000000
00 00002 HSMMG3   HSMMG3                  000122 000061 000184 000000
00 00004 HSMMG2   HSMMG2                  000216 000166 000306 000000
00 00002 HSMMG1   HSMMG1                  000093 000050 000137 000000
00 00002 DB0002   DB0002                  000156 000152 000161 000000
```

Figure 26. AUDIT MODIFY Option T=Major Example

The MODIFY command option V lists the volumes with RESERVE activities. For a sample output see Figure 27 on page 60 and Figure 28 on page 60.

```
F JOBNAME,V

COUNT    VOLSER  DEVNO  MAX    ELAPSED SEC  AVE    MIN    MAX    TOTAL
                        RES.                msec.msec.msec.sec.
00000125 STSO04  02AD   04   *00002102*   000047 000012 001546 000005
00000650 STSO02  01A6   04   *00002692*   000041 000008 002156 000027
00000773 STSO01  01AD   06   *00002626*   000046 000006 002558 000035
00000205 RUBB01  0234   03   *00002638*   000021 000006 000163 000004
00000002 RES43B  023B   01   *00001592*   000097 000089 000105 000000
00000004 PR0003  02AF   01   *00000213*   000197 000143 000231 000000
00000003 HSMMG2  01A3   01   *00001032*   000232 000183 000306 000000
00000013 HSMMG1  01A1   02   *00001032*   000089 000018 000342 000001
00000110 DB0002  01AA   02   *00001821*   000018 000006 000152 000001
00000039 DB0001  01A9   02   *00000615*   000012 000006 000119 000000
00002469 CAT430  023C   03   *00002698*   000214 000006 001243 000529
```

Figure 27. AUDIT MODIFY Option V Example

The MODIFY command option V=volser lists the RESERVE requests issued against the selected volume. For a sample output see Figure 28 on page 60.

```
F JOBNAME,V=CAT430

Rt COUNT VOLSER  DEVNO  MAX    ELAPSED SEC  AVE    MIN    MAX    TOTAL
min.                    RES.                msec.msec.msec.sec.

00001352 CAT430  023C   03   *00001922*   000272 000006 001243 000369
-------------------------------------------
00 00479 SYSZVVDS CAT430                   000016 000004 000187 000007
02 00065 SYSZRACF SYS1.RACF                000033 000016 000140 000002
10 00329 SYSZJES2 CAT430SYS1.HASPCKPT      001096 001064 001243 000359
00 00479 SYSIGGV2 CATALOG.CAT430           000019 000005 000188 000009

First two digits = Rate per minute
```

Figure 28. AUDIT MODIFY Option V=Volser Example

The MODIFY command options T=FREEMAIN and V=FREEMAIN can be used to reinitialize the dataspace without stopping and restarting the monitor.

The MODIFY option R command shows the active RNL's entries used.

Messages - abends - return codes

Figure 29 on page 62 lists the messages issued by ENQ Monitor. They are self-explanatory.

For a complete list of all ISG messages, see *z/OS MVS System Messages, Vol 9 (IGF-IWM)*.

AUDIT Messages

```
 - ISGAU001  -  INVALID REQUEST

                modify command with wrong parameter

 - ISGAU002  -  FILTER TABLE NOT FOUND

                program continues with the old filter table

- ISGAU002  -  FILTER TABLE UNCHANGED

                program continues with the old filter table

- ISGAU002  -  SPECIFIED TABLE IS THE SAME AS THE
                     CURRENT TABLE

                program continues with the old filter table

 - ISGAU003  -  MODIFY REQUEST DONE

 - ISGAU004  -  PARM FIELD NOT VALID

                program ends during initialization

 - ISGAU005  -  AUDIT ALREADY ACTIVE

                job submitted/started twice


 - ISGAU007  -  DDNAME  FULL, PLEASE SAVE

                dynamic switch to the other output data set

 - ISGAU008  -  FILTERxx ACTIVE                - response to MODIFY,L
             -  ddname   ACTIVE
             -  I=xxxxxx L=yyyyyyyy ENQ-DELAY=zzzzzzzz
             -  I=xxxxxx L=yyyyyyyy ENQ-DELAY=zzzzzzzz RESMIL=wwww

 - ISGAU009  -  DATA SET SWITCHED            - response to MODIFY,S

 - ISGAU009  -  DATA SET NOT SWITCHED -
                      DATA SET MAY BE FULL

                Output dataset will not switch

 - ISGAU010  -  FILTER RE-INITIALIZED        - response to MODIFY,I=xx

 - ISGAU012  -  BOTH OUTPUT DATA SETS ARE FULL

 - ISGAU013  -  ISGAMCST NOT AVAILABLE
```

```
  - ISGAU014  -  CANNOT START MONITOR -
                      SYSTEM LX NOT AVAILABLE

                    ENQ Monitor cannot be started

-  ISGAU014  -  CANNOT START MONITOR -
                       NO COMMON STORAGE

                    ENQ Monitor cannot be started

-  ISGAU014  -  CANNOT START MONITOR -
                       NO PRIVATE STORAGE

                    ENQ Monitor cannot be started
-  ISGAU014 - CANNOT START MONITOR - IWMEJOIN RSN = xxxxrcrs:
                    ENQ Monitor cannot be started because it can
                    not join the GRS Dependent WLM Enclave. Not
                    doing so can cause system problems so the
                    monitor is not allowed to be started. As this
                    is an internal error, search problem reporting
                    databases for a fix for the problem. If no fix
                    exists, contact the IBM Support Center and
                    provide the WLM IWMEJOIN service reason code.
                    The reason code code contains both the return
                    code (RC) and the reason code (RS). A dump of
                    the GRS address space may also be required.
-  ISGAU014 - TRANSWAP RC = xx
                    The monitor could not be started because
                    the SYSEVENT TRANSWAP serviced returned an
                    unexpected return code of xx.

-  ISGAU015  -  DATA SPACE FULL

-  ISGAU016  -  MONITOR STOPPED - QUEUE THRESHOLD EXCEEDED

-  ISGAU017  -  MONITOR STOPPED - DATASET(S) NO LONGER ACCESSIBLE
```

Figure 29. AUDIT Messages

Figure 30 on page 62 shows the abend the monitor issues.

```
 - ABEND 1FF

           RC=8    STOP/MODIFY PROBLEMS
```

Figure 30. AUDIT Abends

The monitor terminates if both output data sets are full or an unexpected error occurs. For any abend condition, the monitor will writes a logrec record for diagnostic information.

Filter facility

This monitor uses a filter facility to determine what resources to monitor and what data to collect. The default filter is ISGAMF08. Example of a filter is shown in Figure 33 on page 66.

You can display the name of the active filter with command:

```
F JOBNAME,L
```

and you can dynamically change the active filter with command:

```
F JOBNAME,I=xx
```

Note that changing the filter options does not purge data that was already collected by the previous filter options.

The name of the filter should be ISGAMFxx where xx can range from 00 through 09.

Sample filter ISGAMF00 in SYS1.SAMPLIB is an exact copy of the default member ISGAMF08. If the filter needs to be changed, it should be assembled and linkedited in an authorized library or SYS1.LINKLIB with AC(0).

Build the filter option using the following two macros:

GFLG
Generates the flag byte to control the monitor.

You must only use this macro once in the table.

```
      GFLG  FILTER=Y/N,MATCH=Y/N

            REQTYPE = ALL (REQUESTs that pass other filtering)
            REQTYPE = NCRESERVE (Only gather Non-converted Reserves. Only
                                 gather requests that result in actual hardware
                                 reserves and pass other filtering)

                      DEFAULT=ALL

           FILTER  =  N (TRACE ALL)

                      DEFAULT=Y

           MATCH   =  Y (TRACE IF A NAME ENTRY MATCHES COLLECTED DATA)
           MATCH   =  N (DO NOT TRACE IF A NAME ENTRY MATCHES
                               COLLECTED DATA)

                      DEFAULT=Y
```

Figure 31. GFLG Filter Example

NAME
The NAME keyword is used to generate the name list. The NAME types can be MAJOR, MINOR, and JOB/STC/TSU. Note that a names list is not required. When the NAME keyword is not provided, only the FILTER and REQTYPE keywords are used to determine if an event should be traced.

```
         NAME  N=,M=,T=,[L=]

                    N= Major name        1 to  8 chars

                    M= Minor name        1 to 52 chars

                    T=type on entry   U=Resource - Major & Minor names
                                      M=Major name
                                      J=jobname
                                      S=started-task
                                      T=tso-user

      OPTIONAL ==>  L=length of compare - valid values 1-60
                      default=length of character string (except
                      for Major name which is considered 8 byte).
                      If L is higher than character string, the NAME
                      is padded with blanks.
                      Used only with T=U,J,S,T
```

Figure 32. NAME Filter Example

The trace control operates at **two hierarchical levels**.

Level one:
The resources whose MAJOR name is specified in a name entry with keyword **T=M** are not traced. For example, if the resources with major name SYSDSN have to be discarded, code the following:

```
      GFLG  FILTER=Y

      SVCF                  /*this statement is required*/

      NAME  N=SYSDSN,T=M
```

Level two:

The resources whose MAJOR and MINOR is specified in a NAME entry with keyword **T=U** are processed according to the GFLAG setup:

- **GFLAG MATCH=Y** writes a trace entry if the ENQ/DEQ resource matches a name entry type=U
- **GFLAG MATCH=N** writes a trace entry if the ENQ/DEQ resource does not match a name entry type=U

Example 1: If the resources representing the temporary data sets for year '96 (resource major SYSDSN minor SYS96039.xxxx.yyyy) have to be discarded, code the following:

```
       GFLG  FILTER=Y,MATCH=N

       SVCF                  /*this statement is required*/

       NAME  N=SYSDSN,M=SYS96,T=U,L=13

           note: checked with length of 13 bytes
                 (8 major name and first 5 minor name bytes)
```

Example 2: If only the resources with major names SYSZVVDS and SYSIGGV2 have to be traced, code the following:

```
       GFLG  FILTER=Y,MATCH=Y

       SVCF                  /*this statement is required*/

       NAME  N=SYSZVVDS,T=U,L=8        NAME  N=SYSIGGV2,T=U,L=8

           note: checked with length of 8 bytes
                 (8 major name, any minor name)
```

The resources used by JOB/STC/TSU whose name is specified in a NAME entry with keyword **T=J/S/T** are processed according to the GFLAG setup:

- **GFLAG MATCH=Y** writes a trace entry if the requester name matches a name entry type=J/S/T
- **GFLAG MATCH=N** writes a trace entry if the requester name does not match a name entry type=J/S/T

Example 3: To see all the resources used by started task TSO, NET and TSO users starting with IPO1, (the resources with scope STEP are discarded by default), code the following:

```
       GFLG  FILTER=Y,MATCH=Y

       SVCF                  /*this statement is required*/

NAME  N=TSO,T=S,L=8
       NAME  N=NET,T=S,L=8
       NAME  N=IPO1,T=T,L=4

       note: stcs TSO end NET checked with length of 8, name padded
             with five blanks.
```

Example 4: Only collect events that result in a hardware reserve for RESERVEs issued against the SYSZVVDS resource.

```
       GFLG FILTER=Y,REQTYPE=NCRESERVE,MATCH=Y

       SVCF                  /*this statement is required*/

NAME N=SYSZVVDS,T=M,L=8
```

Example 5: Collect all events that result in a hardware reserve.

```
GFLG FILTER=Y,REQTYPE=NCRESERVE

SVCF                  /*this statement is required*/


note: No name entries are provided so only the REQTYPE=NCRESERVE
      filter is applied.
```

The following table summarizes the possible combinations and actions of the filter REQTYPE=NCRESERVE:

GFLG	No-Match	Match (*)
FILTER=Y, MATCH=Y	No-Trace	Trace
FILTER=Y, MATCH=N	Trace	No-Trace
FILTER=N	Trace	Trace
REQTYPE=NCRESERVE		

(*)MATCH means that the resource name specified in a NAME entry type T=U matches the MAJOR and MINOR names of the ENQ/DEQ request, or for NAME entries type J/S/T, the job/stc/tso name matches the requester.

Example 6: Collect information about unauthorized ENQ requests (from all ENQ services) that fail when AUTHQLVL=2. AUTHQLVL indicates the list of authorized qnames in effect for the system in the global resource serialization complex. For information about authroized qnames and AUTHQLVL, see “Authorized qnames” on page 18.

```
GFLG FILTER=Y,REQTYPE=AUTHQ2,MATCH=N

SVCF                  /*this statement is required*/


note: No name entries are provided so only the REQTYPE=AUTHQ2
      filter is applied.
```

The following table summarizes the possible combinations and actions of the filter REQTYPE=AUTHQ2:

GFLG	No-Match	Match (*)
FILTER=Y, MATCH=Y	No-Trace	Trace
FILTER=Y, MATCH=N	Trace	No-Trace
FILTER=N	Trace	Trace
REQTYPE=AUTHQ2		

(*)MATCH means that the resource name specified in a NAME entry type T=U matches the MAJOR and MINOR names of the ENQ/DEQ request, or for NAME entries type J/S/T, the job/stc/tso name matches the requester.

The following is a filter coding sample.

```
*********************************************************************
*       SELECT ONE OF THE FOLLOWING
*       GFLAG DEFAULT FILTER=Y,MATCH=Y

GFLAG   EQU   *
*       GFLG  FILTER=N                      NO TRACE AT ALL
*
        GFLG  MATCH=N,FILTER=Y              NO MATCH -> TRACE
*
*       GFLG  MATCH=Y,FILTER=Y              MATCH    -> TRACE
*
*********************************************************************
*
        SVCF                       THIS STATEMENT IS REQUIRED
*
*********************************************************************
NAME1   EQU   *                    THIS STATEMENT IS REQUIRED
*********************************************************************
*       GENERATE THE NAME TABLE ENTRY WITH THE MACRO NAME
*
        NAME  N=SYSZTIOT,T=M        MAJOR NAME 'SYSZTIOT' NOT TRACED
        NAME  N=SYSZJWTP,T=M        MAJOR NAME 'SYSZJWTP'     "
        NAME  N=SYSZJES2,M=CVCB,T=U,L=12
        NAME  N=SYSDSN,M=SYS96,L=13,T=U
*
* Note:
* The GFLAG requests TRACE for NO-MATCH.
* The resources with the following MAJOR & MINOR are NOT TRACED
*
*       SYSZTIOT               all major SYSZTIOT
*       SYSZJWTP               all major SYSZJWTP
*       SYSZJES2 CVCB          checked with length of 12 chars.(8+4)
*       SYSDSN SYS96           checked with length of 13 chars.(8+5)
* (temp dsn for year '96)
*
*********************************************************************
```

Figure 33. Filter Coding Example

Reports

Programs and JCL procedures can generate three reports from the collected data. Three procedures collect data for single MVS system input, and three procedures collect data for multiple MVS systems input to support sysplex. For multiple systems, input to the procedures is concatenated. These programs do not require authorization.

They are:

- Procedure ISGAJE1 for **sequential trace report** with single MVS system input. Program used is ISGAMED1. The output generated is shown in Figure 42 on page 82.
- Procedure ISGAJE1A for **sequential trace report** with multiple MVS systems input. Programs used are ISGAMED2, ISGAMED3 and SORT. The output generated is shown in Figure 43 on page 83.

  The reports contain the following information:

  - Date and time of the request.
  - SMF system id.
  - Jobname of the requester.
  - Program name of the requester.
  - Module authorization information (only for ISGAJE1).
  - SVC type.
  - RNL action.
  - Scope of the request.
  - Volume serial for reserve requests.
  - Major and minor name

  - Device number for reserve requests.
  - Minor name length.
- Procedure ISGAJE2 for **volume reserve time report** with single MVS system input.
- Procedure ISGAJE2A for **volume reserve time report** with multiple MVS systems input. The output generated is shown in Figure 44 on page 83.

  Programs used are ISGAMEDM - ISGAMVOL - ISGAMCTM - ISGAMMRT and SORT.

  For every volume with reserve activity the following is given:

  - Count of reserve requests.
  - Time when the volume had the longest reserve.
  - Highest volume reserve count.
  - Average, minimum and maximum reserve time in msec, milliseconds, used for times up to 999.999 milliseconds.
  - Total reserve time in sec.
  - Reserve rate per minute.
  - The list of resources that reserved the volume with the timing information.

  The output is sorted by volume reserve time for ISGAJE2 and by SMF system id and volser for JCL652A.
- Procedure ISGAJE3 for **resource report** with single MVS system input.
- Procedure ISGAJE3A for **resource report** with multiple MVS systems input. The output generated is shown in Figure 45 on page 84.

  Programs used are ISGAMEDM - ISGAMCNT - ISGAMSRT - ISGAMERN and SORT.

  For each major name the following is given:

  - Count of ENQ requests.

    Note: If DEQ is issued for each ENQ, the total count of requests for the resource is doubled.
  - Scope of the request.
  - SMF system id.
  - List of minor names used with the RNL action:
    1. *C reserve-converted --> SYSTEMS and NO-HW-RESERVE
    2. *E reserve-excluded --> SYSTEM and HW-RESERVE
    3. *E systems excluded --> SYSTEM
    4. *I system included --> SYSTEMS
    5. N keyword RNL=NO
    6. 'blank' no action
    7. Length of the minor name (ML).
    8. Timing information for reserve requests.

Logs

The volume-reserve-time procedure on STEP002 generates on LOG2 DD statement the list of RESERVE/RELEASE activity from which the report is generated. With the time of the longest continuous reserve given in the volume report, it is possible to find in the LOG2 data set the names of the resources involved. If the above information is not required, the LOG2 DD statement can be specified as DUMMY. Figure 46 on page 85 gives an example for Reserve/Release trace at time of Max.

On STEP007 the ISGAMMRT program on the LOGV DD statement, logs the VOLUME list from which the report is created. The ISGAMMRT program also creates on the LOGR DD statement a data set with the reserve resource names with the time statistics.

NOTE

Because the LOGR data set is required by the resources-usage-procedure, the procedure for volume-reserve-time should be run **before** the resources-usage-procedure.

If the LOGR data set is specified as DUMMY, the resources-usage will not have the timing information.

The resources-usage procedure on STEP007, generates on LOG1 output DD statement the MAJOR by SCOPE names from which the report is created. If the above information is not required, you can specify the LOG1 DD statement as DUMMY.

All the reports and logs can be spool, tape, or disk data sets.

ATTENTION

The reserve timing are incorrect if reserve requests are discarded with the filter, if data are lost, and if a job is cancelled or ABEND with an active reserve.

Temporary data sets

Change the temporary data sets used by the procedures according to the size of the trace data set. You can change the size by modifying the variable in the sample in-stream procedures.

In-stream procedures variables

```
VOLUME RESERVE TIME REPORT   -  procedure JCL2
INPUT -->  input data set name
OUT1  -->  Intermediate output  = 1.2 the size of input - CYLS
OUT2  -->  Intermediate output  = 2/3 the size of OUT1   - CYLS
OUT3  -->  Intermediate output  = 1 cylinder
WORK1 -->  sort workfile        = for OUT1 sorting       - CYLS
WORK2 -->  sort workfile        = 1 cylinder             - CYLS
LOGR  -->  data set name for LOGR

RESOURCES USAGE REPORT       -  procedure JCL3
INPUT -->  input data set name
OUT1  -->  Intermediate output  = 1.2 the size of input  - CYLS
WORK1 -->  sort workfile        = for OUT1 sorting       - CYLS
LOGR  -->  data set name for LOGR
```

Figure 34. In-stream Procedures Variables

Report programs parameters

The PARM field is supported by ISGAMEDM, ISGAMED1, ISGAMVOL, ISGAMMRT, ISGAMCNT, and ISGAMERN and they allow:

- Selection by date and time with LOCAL TIME values. **ISGAMEDM-ISGAMED1 'PARM=DATE'**
- Difference between LOCAL TIME and the TOD (GMT TIME) that is collected by the monitor. **ISGAMEDM-ISGAMED1 PARM=DELTA**
- Clear the input data set. **ISGAMED1 PARM=CLEAR** supported only with procedure ISGAJE1 (single MVS system input).
- Perform volume reserve time analysis for selected volumes. **ISGAMVOL PARM=volser**

- Limit the volume reserve time report to top xx volumes. **ISGAMMRT PARM=TOP**
- Limit the resources report to the selected MAJOR names. **ISGAMCNT PARM=major1,major2,majorx**
- Limit the resources report to resources with count higher than yy. **ISGAMERN PARM=COUNT**

The supported PARM values are shown in Figure 35 on page 69.

```
ISGAMEDM - ISGAMED1

  DATE=(yyddd.hhmm-yyddd.hhmm)   local-time
                      or

  DATE=(yyyyddd.hhmm-yyyyddd.hhmm)

  select   from   :     to
             (both must be specified)

   yyyy up to 9999
   yy    up to 99   Note: treated as 19yy
   ddd  up to 366
   hh    up to 23
   mm   up to 59


  DELTA=(+]-h)

   local time delta from TOD

   h   from 0 to 12


  CLEAR

   input data set is cleared if it is not currently active
   with AUDIT. Supported only with procedure ISGAJE1

ISGAMVOL

 volume1,volume2,volumex

  volumes to process for reserve time report

ISGAMMRT

 TOP=xxxxxxxx          from  1 to 99999999

  list the top xx-xx volumes.

ISGAMCNT

 major1,major2,majorx

  major names to select for resources report

ISGAMERN

 COUNT=xxxxxxxx          from  1 to 99999999

  list the resources with use count high or equal to xx-xx.
```

Figure 35. PARM Supported by Report Programs

If a PARM keyword or a PARM value is not valid, the program terminates.

Region requirements

The procedures for volume-reserve-time report and for resource-usage report require a region size below 16MB large enough to hold an internal table created by the programs. The table requires 144 byte of virtual storage for each resource name that has been requested by the RESERVE macro and 56 byte for each DASD volume with reserve activity. For example, to process 10 thousand different resources and 300 volumes, GRS uses 1.5MB of virtual storage.

The ISGAMVOL program at the end of its processing issues messages about the number of table entries used.

```
ELEMENTS=00000988      (resource name entries)
     VOLUMES =00000105      (volumes  entries)
```

Use different selections during the sort steps to get different aggregation of the data collected. The layout of the output records generated by the programs are shown in “ISGAMEDM ISGAMED1 ISGAMVOL output record fields” on page 86.

The following is the JCL for the reports' procedures:

```
//ISGAJE1  JOB CLASS=A,MSGCLASS=X
//*
//*        SEQUENTIAL TRACE REPORT
//*
//JOBLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//STEP001  EXEC PGM=ISGAMED1,
//*
//*    TRACE REPORT
//*
//*Note: DNS=USERID.AUDIT.OUT1,DISP=SHR has to be modified to get the
//*      correct dataset.
//*
//INPUT  DD DSN=USERID.AUDIT.OUT1,DISP=SHR
//OUTPUT DD SYSOUT=*,
//       DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSUDUMP DD SYSOUT=*
```

Figure 36. JCL for Trace Report for Single MVS System Input

```
//ISGAJE1A JOB CLASS=A,MSGCLASS=X
//*
//*        SEQUENTIAL TRACE REPORT
//*
//*****************************************************************
//JOBLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//JCL1A     PROC  WORK1=90,OUT1=300,
//             INPUT1=USERID.AUDIT.OUT1,
//             INPUT2=USERID.AUDIT.OUT2,
//             INPUT3=USERID.AUDIT.OUT3,
//             INPUT4=USERID.AUDIT.OUT4
//*            UP TO 32 INPUT DATA SETS
//*****************************************************************
//STEP001 EXEC PGM=ISGAMED2,
// PARM='DATE=(95215.0800:95215.1100),DELTA=(+2)'
//*****************************************************************
//*
//*   TRACE REPORT
//*
//*****************************************************************
//INPUT     DD DSN=&INPUT1,DISP=SHR
//          DD DSN=&INPUT2,DISP=SHR
//          DD DSN=&INPUT3,DISP=SHR
//          DD DSN=&INPUT4,DISP=SHR
//*         UP TO 32 INPUT DATA SETS
//OUTPUT    DD DSN=&&OUT2,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSUDUMP DD SYSOUT=*
//*
//STEP002   EXEC PGM=SORT,
//          COND=(0,LT,STEP001)
//*
//*         SORT  TOD
//*
//SORTIN    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD=DSN=&&OUT3,DISP=(,PASS), UNIT=SYSDA, SPACE=(CYL, &OUT1),
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP003   EXEC PGM=ISGAMED3,
//          COND=(0,LT,STEP001)
//*
//*         TRANSLATE TOD
//*
//INPUT     DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD SYSOUT=*,DCB=(LRECL=132,RECFM=FB,BLKSIZE=13200)
//SYSUDUMP  DD SYSOUT=*
//          PEND
//          EXEC  JCL1A
//STEP002.SYSIN DD *
 SORT FIELDS=(1,8,A),FORMAT=BI,FILSZ=E5000,EQUALS
```

Figure 37. JCL for Trace Report for Multiple MVS Systems Input

```
//ISGAJE2  JOB CLASS=A,MSGCLASS=X
//*
//*       VOLUMES RESERVE TIME REPORT
//*********************************************************************
//JCL2      PROC WORK1=30,WORK2=1,OUT1=90,OUT2=60,OUT3=1,
//            INPUT=USERID.AUDIT.OUT1,LOGR=USERID.LOGR
//*********************************************************************
//*
//*       VOLUMES RESERVE TIME REPORT
//*
//STEP001   EXEC PGM=ISGAMEDM,
//*         PARM='DELTA=(+2),DATE=(94011.0800:94012.0900)'
//INPUT     DD DSN=&INPUT,DISP=SHR
//OUTPUT    DD DSN=&&OUT2,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//STEP002   EXEC PGM=ISGAMVOL,
//          COND=(0,LT,STEP001)
//*         PARM='TPVOL ,XA9RES'
//*
//*       INSERTS VOLID AND RESERVE COUNT IN PROBABLE RELEASE
//*
//INPUT     DD DSN=&&OUT2,DISP=(OLD,PASS)
//OUTPUT    DD DSN=&&OUT3,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//OUTPUT1   DD DSN=&&OUT4,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT2),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//*OG2      DD DUMMY
//LOG2      DD SYSOUT=*,
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSDUMP   DD SYSOUT=*
//*
//STEP003   EXEC PGM=SORT,
//          COND=(0,LT,STEP002)
//*
//*      SORT  VOLID - SMFID
//*
//SORTIN    DD DSN=&&OUT3,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD DSN=&&OUT2,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP004   EXEC PGM=ISGAMCTM,
//          COND=(0,LT,STEP001)
//*
//* RESERVE TIME AND FREQUENCY
//*
//INPUT     DD DSN=&&OUT2,DISP=(OLD,PASS)
//*UTPUT    DD SYSOUT=*,DCB=(LRECL=132,RECFM=FB,BLKSIZE=13200)
//OUTPUT    DD DSN=&&OUT5,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT3),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//WORK      DD DSN=&&OUT3,DISP=(OLD,PASS)
//SYSUDUMP  DD SYSOUT=*
//STEP005   EXEC PGM=SORT,
//          COND=(0,LT,STEP001)
//*
//*     SORT  TOTAL VOLUME RESERVE TIME
//*
```

Figure 38. JCL for Volume Reserve Report for Single MVS System Input

```
//SORTIN    DD DSN=&&OUT5,DISP=(OLD,PASS)
//SORTOUT   DD DSN=&&OUT3,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//STEP006   EXEC PGM=SORT,
//          COND=(0,LT,STEP005)
//*
//*     SORT  TOTAL RESOURCE RESERVE TIME
//*
//SORTIN    DD DSN=&&OUT4,DISP=(OLD,PASS)
//SORTOUT   DD DSN=&&OUT2,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//STEP007   EXEC PGM=ISGAMMRT,
//          COND=(0,LT,STEP001),
//          PARM='TOP=20'
//*
//*     MERGE VOLUME AND RESOURCE DATA
//*
//INPUT1    DD DSN=&&OUT2,DISP=(OLD,PASS)
//INPUT2    DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD SYSOUT=*
//LOGV      DD SYSOUT=*
//LOGR      DD DSN=&LOGR,DISP=SHR
//SYSDUMP   DD SYSOUT=*
//          PEND
//          EXEC  JCL2
//STEP003.SYSIN DD *
 SORT FIELDS=(53,6,A,126,4,A),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP005.SYSIN DD *
 SORT FIELDS=(37,8,D),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP006.SYSIN DD  *
 SORT FIELDS=(110,10,D),FORMAT=BI,FILSZ=E5000,EQUALS
```

```
//ISGAJE2A JOB CLASS=A,MSGCLASS=X
//*
//*        VOLUME RESERVE TIME REPORT
//*****************************************************************
//JOBLIB DD DNS=SYS1.MIGLIB,DISP=SHR
//JCL2A     PROC  WORK1=90,WORK2=30,OUT1=300,OUT2=100,OUT3=1,
//           INPUT1='USERID.AUDIT.OUT1',LOGR='USERID.LOGR',
//           INPUT2='USERID.AUDIT.OUT2',
//           INPUT3='USERID.AUDIT.OUT3',
//           INPUT4='USERID.AUDIT.OUT4'
//*          UP TO 32 INPUT DATA SETS
//*****************************************************************
//*
//*        VOLUME RESERVES TIME REPORT
//*
//*****************************************************************
//STEP001   EXEC PGM=ISGAMEDM,
//          PARM='DELTA=(+2),DATE=(93202.1400:93203.0800)'
//INPUT     DD DSN=&INPUT1,DISP=SHR
//          DD DSN=&INPUT2,DISP=SHR
//          DD DSN=&INPUT3,DISP=SHR
//          DD DSN=&INPUT4,DISP=SHR
//*          UP TO 32 INPUT DATA SETS
//OUTPUT    DD DSN=&&OUT2,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LREC=132,RECFM=FB,BLKSIZE=18348)
//STEP002   EXEC PGM=ISGAMVOL,
//          COND=(0,LT,STEP001)
//*         PARM='TPVOL ,XA9RES'
//*
//*       INSERTS VOLID AND RESERVE COUNT IN PROBABLE RELEASES
//*
//INPUT     DD DSN=&&OUT2,DISP=(OLD,PASS)
//OUTPUT    DD DSN=&&OUT3,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LRECL=132,RECRM=FB,BLKSIZE=18348)
//OUTPUT1   DD DSN=&&OUT4,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&&OUT2),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//*OG2      DD DUMMY
//LOG2      DD SYSOUT=*,
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSUDUMP  DD SYSOUT=*
//*
//STEP003   EXEC PGM=SORT,
//          COND=(0,LT,STEP002)
//*
//*      SORT  VOLID AND SMFID
//*
//SORTIN    DD DSN=&&OUT3,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*,
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SUSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SUSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SUSDA,SPACE=(CYL,&WORK1)
//STEP004   EXEC PGM=ISAMCTM,
//          COND=(0,LT,STEP003)
//*
//* RESERVE TIME AND FREQUENCY
//*
```

Figure 39. JCL for Volume Reserve Report for Multiple MVS Systems Input

```
//INPUT     DD DSN=&&OUT2,DISP=(OLD,PASS)
//*UTPUT    DD SYSOUT=*,DCB=(LRECL=132,RECFM=FB,BLKSIZE=13200)
//OUTPUT    DD DSN=&&OUT5,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT3),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//WORK      DD DSN=&&OUT3,DISP=(OLD,PASS)
//SYSDUMP   DD SYSOUT=*
//STEP005   EXEC PGM=SHORT,
//          COND=(0,LT,STEP004)
//*
//*     SORT  VOLUME, SMFID, TOTAL VOLUME RESERVE TIME
//*
//SORTIN    DD DSN=&&OUT5,DISP=(OLD,PASS)
//SORTOUT   DD DSN=&&OUT5,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//STEP006   EXEC PGM=SHORT,
//          COND=(0,LT,STEP005)
//*
//*     SORT  TOTAL RESOURCE RESERVE TIME
//*
//SORTIN    DD DSN=&&OUT4,DISP=(OLD,PASS)
//SORTOUT   DD DSN=&&OUT2,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK2)
//STEP007   EXEC PGM=ISAMMRT,
//          COND=(0,LT,STEP006)
//*         PARM='TOP=20'
//*     MERGE VOLUME AND RESOURCE DATA
//*
//INPUT1    DD DSN=&&OUT2,DISP=(OLD,PASS)
//INPUT2    DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD SYSOUT=*
//LOGV      DD SYSOUT=*
//LOGR      DD DSN=&LOGR,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//          PEND
//          EXEC JCL2A
//STEP003.SYSIN DD *
 SORT FIELDS=(53,6,A,126,4,A),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP005.SYSIN DD *
 SORT FIELDS=(25,6,D,32,4,A,37,8,D),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP006.SYSIN DD *
 SORT FIELDS=(110,10,D),FORMAT=BI,FILSZ=E5000,EQUALS
```

```
//ISGAJE3  JOB CLASS=A,MSGCLASS=X
//*      RESOURCES  REPORT
//*********************************************************************
//JCL3      PROC   WORK1=30,OUT1=180,INPUT=USERID.AUDIT.OUT1,
//          LOGR=USERID.LOGR
//*********************************************************************
//*
//*      RESOURCES REPORT
//*
//STEP001   EXEC PGM=ISGAMEDM,
//*         PARM='DATE=(93274.0800:93274.1300),DELTA=(+2)'
//INPUT     DD DSN=&INPUT,DISP=SHR
//OUTPUT    DD DSN=&&OUT2,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//STEP002   EXEC PGM=SORT,
//          COND=(0,LT,STEP001)
//*
//*      SORT SCOPE,  MAJOR/MINOR NAME,  SMFID
//*
//SORTIN    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD DSN=&&OUT3,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP003   EXEC PGM=ISGAMCNT,
//*         PARM=('SYSZRACF,SYSZJES2,SYSDSN')
//          COND=(0,LT,STEP002)
//*
//*      COUNT RESOURCE
//*
//INPUT     DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD DSN=&&OUT2,DISP=(OLD,PASS)
//SYSUDUMP  DD SYSOUT=*
//STEP004   EXEC PGM=SORT,
//          COND=(0,LT,STEP003)
//*
//*      SORT  MAJOR/SCOPE AND NUMBER OF REQUEST
//*
//SORTIN    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD DSN=&&OUT3,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP005   EXEC PGM=ISGAMSRT,
//          COND=(0,LT,STEP001)
//*
//*      INSERT COUNT FOR SORT
//*
//INPUT     DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*UTPUT    DD SYSOUT=*
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//STEP006   EXEC PGM=SORT,
//          COND=(0,LT,STEP005)
```

Figure 40. JCL for Resource Usage Report for Single MVS System Input

```
//*
//*      SORT  SCOPE AND NUMER OF MAJOR REQUEST
//*
//SORTIN    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD DSN=&&OUT3,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP007   EXEC PGM=ISGAMERN,
//          COND=(0,LT,STEP006),
//          PARM='COUNT=00000001'
//*
//*    COUNT RESOURCE WITH SAME SCOPE
//*
//INPUT    DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT   DD SYSOUT=*,
//         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//LOG1     DD SYSOUT=*,
//         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//*    LOGR IS CREATED BY  VOLUMES-RESERVE-TIME  REPORT
//LOGR     DD DSN=&LOGR,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//         PEND
//         EXEC  JCL3
//STEP002.SYSIN     DD *
 SORT FIELDS=(42,7,A,60,61,A,126,4,A),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP004.SYSIN     DD *
 SORT FIELDS=(61,8,A,42,7,A,11,8,D),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP006.SYSIN DD *
 SORT FIELDS=(42,7,A,20,8,D),FORMAT=BI,FILSZ=E5000,EQUALS
```

```
//ISGAJE3A JOB CLASS=A,MSGCLASS=X
//*
//*        RESOURCES REPORT
//*****************************************************************
//JOBLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//JCL3A     PROC  WORK1=90,WORK2=30,OUT1=300,OUT2=150,OUT3=31,
//          INPUT1='USERID.AUDIT.OUT1',LOGR='USERID.LOGR',
//          INPUT2='USERID.AUDIT.OUT2',
//          INPUT3='USERID.AUDIT.OUT3',
//          INPUT4='USERID.AUDIT.OUT4'
//*         UP TO 32 INPUT DATA SETS
//*****************************************************************
//*
//*      RESOURCES REPORT
//*
//*****************************************************************
//STEP001   EXEC PGM=ISGAMEDM,
//*         PARM='DARE=(93202.1400:93203.0800) ,DELTA=(+2)'
//INPUT     DD DSN=&INPUT1,DISP=SHR
//          DD DSN=&INPUT2,DISP=SHR
//          DD DSN=&INPUT3,DISP=SHR
//          DD DSN=&INPUT4,DISP=SHR
//*         UP TO 32 INPUT DATA SETS
//OUTPUT    DD DSN=&&OUT2,DISP=(,PASS) ,UNIT=SYSDA,SPACE=(CYL,&OUT1),
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//STEP002   EXEC PGM=SORT,
//          COND=(0,LT,STEP001)
//*
//*      SORT SCOPE,  MAJOR, SMF-ID, MINOR
//*
//SORTIN    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*,
//*         DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD DSN=&&OUT3,DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,&OUT1)
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP003   EXEC PGM=ISGAMCNT,
//          COND=(0,LT,STEP002)
//*         PARM=('SYSZRACF,SYSZJES2,SYSDSN')
//*
//*      COUNT RESOURCE
//*
//INPUT     DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD DSN=&& OUT2,DISP=(OLD,PASS)
//SYSUDUMP  DD SYSOUT=*
//STEP004   EXEC PGM=SORT,
//          COND=0,LT,STEP003)
//*
//*      SORT  MAJOR/SCOPE AND NUMBER OF REQUEST
//*
```

Figure 41. JCL for Resource Usage Report for Multiple MVS Systems Input

```
//SORTIN    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*,
//*          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD DSN=&&OUT3,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP005   EXEC PGM=ISGAMSRT
//          COND=(0,LT,STEP004)
//*
//*      INSERT COUNT FOR SORT
//*
//INPUT     DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*UTPUT    DD SYSOUT=*
//*          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//STEP006   EXEC PGM=SORT,
//          COND=(0,LT,STEP005)
//*
//*      SORT  SCOPE, MAJOR, SMFID, NUMBER OF REQUESTS
//*
//SORTIN    DD DSN=&&OUT2,DISP=(OLD,PASS)
//*ORTOUT   DD SYSOUT=*
//*          DCB=(LREC=132,RECFM=FB,BLKSIZE=18348)
//SORTOUT   DD DSN=&&OUT3,DISP=(OLD,PASS)
//SYSOUT    DD DUMMY
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,&WORK1)
//STEP007   EXEC PGM=ISGAMERN,
//          COND=(0,LT,STEP006),
//          PARM='COUNT=00000001'
//*
//*    COUNT RESORCE WITH SAME SCOPE
//*
//INPUT     DD DSN=&&OUT3,DISP=(OLD,PASS)
//OUTPUT    DD SYSOUT=*,
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//LOG1      DD SYSOUT=*
//          DCB=(LRECL=132,RECFM=FB,BLKSIZE=18348)
//LOGR      DD DSN=&LOGR,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//          PEND
//          EXEC  JCL3A
//STEP002.SYSIN     DD *
SORT FIELDS=(42,7,A,126,4,A,60,61,A),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP004.SYSIN     DD *
SORT FIELDS=(61,8,A,42,7,A,11,8,D),FORMAT=BI,FILSZ=E5000,EQUALS
//STEP006.SYSIN DD *
SORT FIELDS=(42,7,A,60,8,A,126,4,A,20,8,D),FORMAT=BI,                X
              FILSZ=E5000,EQUALS
```

ENQ/RESERVE/DEQ restrictions

The ENQ/DEQ/RESERVE monitor has some restrictions:

- The ENQ RET=TEST requests are traced, but they are discarded during data reduction.
- The maximum length of the minor name traced is 52 bytes (data set name length plus member name). The reported minor name length can be used to see if the minor name has been truncated.
- Data is collected at the entry of the SVC. If RNLs are active, GRS simulates data collection.
- GRS traces only the first parameter list entry for RESERVE requests.
- The maximum RESERVE time measurable is around 4 hours 45 minutes, the lowest reserve time is rounded to 1 millisecond.
- If you specify the same RESERVE resource name (major and minor) for different volumes, GRS reports incorrect data.
- If GRS discards RESERVE requests with the filter, the timings given in the volume-reserve-time procedure are incorrect.
- If a job is cancelled or ABENDs with an active RESERVE, the timings given in the volume-reserve-time procedure are incorrect.

- The maximum ENQ delay-time measurable is 99.99 seconds.
- GRS registers the RESERVE count before issuing the hardware reserve. Therefore for volumes with high contention, the count might be incorrect along with the measured reserve time.
- If you issue DEQ requests without an outstanding reserve, the measured reserve time might be incorrect. Likely candidates include CATALOG address space and HSM subsystem.
- In a global resource serialization complex, only the ENQ/DEQ issued in the system where the monitor is active are traced.
- The monitor encounters the same contention for the processor and storage as does any other address space in the system. Thus, even after the system grants the global resource request, the monitor might not receive control immediately because of processor or storage contention, and the ENQ delay measurement might not be accurate. Experience shows that the tool can lose data if the global resource serialization ENQ delay exceeds 500 msec. or more.
- The NUMSYS (number of systems in the global resource serialization complex) value is updated periodically through an internal command or when you enter a Modify command.

Resources used and guidelines

The ENQ/RESERVE/DEQ monitor does not use common virtual storage below 16M. The filter table and main table are in ESQA.

The monitor uses system resources while collecting data. IBM recommends that you do not run the monitor continuously during normal production work. While running the monitor you might observe high CPU in address spaces that issue many ENQ/RESERVE/DEQ requests.

The monitor starts with 200 buffers in private storage. It can be expanded if more storage is required. Each buffer is 4K is size. It contains a header and 30 logical records.

The number of buffers depends on the numbers of events to trace and on the speed of the output device. A counter of lost events is maintained by the monitor and is displayed with the MODIFY L command. An indication of lost event is also written in the output data set. See “Monitor physical output record” on page 80.

The main program writes the buffers on the sequential data sets using the macros WRITE - CHECK for each physical record and executes a TCLOSE after each buffer is written to the sequential data set. If 300 ENQ/DEQs are generated per second, 10 buffers will be written out in one second. This causes 10 I/O requests to the output data set. The data set can be allocated on a volume with DFW (DASD FAST WRITE) option active if the device cannot keep up with the I/O requests.

If the data collected in the dataspace is good enough for the analysis, specify the output data sets as DUMMY.

Monitor diagnostic information

The main program is protected with an ESTAE that retries for abends X'B37', X'D37', and X'E37' unless the two output data sets are full.

If the monitor abends, a software logrec is written for diagnostic information.

An FRR protects the data collection routine. If a problem happens, the FRR routine writes a record in the SYS1.LOGREC and flags the logical record header with the word ERRO.

If the ISPF application abends, the panel displays the ABEND code and the system takes a dump if the logon procedure contains a SYSUDUMP statement.

Monitor physical output record

The record written to the output data set is a physical buffer containing an header and 30 audit records.

Each AUDIT LOGICAL RECORD is 128 bytes long. The layout is given in “DSECT monitor physical and logical record” on page 85. The HEADER can have different contents, that is:

```
C'AREC'            NORMAL RECORD
c'ERRO'           ERROR DURING DATA COLLECTION - RETRIED
```

The record contains the TOD (GMT) when the SVC was issued, the jobname, the program name, the type of request, the scope, the volume serial (for reserve), the major name, the minor name up to 52 characters, the minor name length, the device number (reserve), the RNL action, and the SMF system identifier.

Monitor utilization hints

The following table lists monitors scope with reference pages.

Scope	Monitor facility	Reference
Monitor Installation		“Setting up the monitor” on page 50
Monitor Execution		“Monitor execution” on page 50
Monitor Control		“Monitor control” on page 57
Report the resources names used	RESOURCES REPORT and ISPF APPLICATION	Figure 40 on page 76 Figure 14 on page 53
Measure the volume reserve time(*)	VOLUME REPORT and ISPF APPLICATION	Figure 38 on page 72 Figure 18 on page 56
Measure the rate and time of RESERVE requests(**)	RESOURCES REPORT and ISPF APPLICATION	Figure 40 on page 76 Figure 19 on page 56
Find programs/applications using a resource	TRACE REPORT and ISPF APPLICATION	Figure 36 on page 70 Figure 16 on page 55
Find nested reserves causing high volume reserve	VOLUME REPORT and LOG2	“Logs” on page 67
Monitor the resources used by an application	FILTER MATCH with keywords J/S/T	“Filter facility” on page 62
Monitor the unknown resources	FILTER NOMATCH with keywords M/U	“Filter facility” on page 62
Interactively see the ENQ delay and the current RESMIL value	MODIFY option L and ISPF APPLICATION	Figure 24 on page 59 Figure 12 on page 52 Figure 15 on page 54
Interactively see the major/scope counts	MODIFY option T and ISPF APPLICATION	Figure 25 on page 59 Figure 12 on page 52
Interactively see the RNL's tables action	MODIFY option T and ISPF APPLICATION	Figure 25 on page 59 Figure 14 on page 53
Trace non-converted reserves	REQTYPE=NCRESERVE filter	Figure 31 on page 63

(*) The volume reserve time is incorrect if reserve requests are discarded with the filter facility or data is lost.

(**) Based on knowing the RESERVE rate and time you can choose between including the RESERVE requests in the CONVERSION or EXCLUSION RNL list.

ENQ/DEQ/RESERVE analysis aid reports

```
-------------------------------------------------------------------------------------------------------------------------------
PROGRAM = NOT-PRB, no CDE associated            SCOPE -> RESERVE  XX = DEVICE RESERVE COUNT

AUT     = A, module  authorized                       -> RELEASE  XX = DEVICE RESERVE COUNT
        = L, from auth. library                       -> REL/SYS  XX = DEQ ASSOCIATED WITH RESER
                                                      -> SYSTEM(S)SH = SHARED
RNL     = N, keyword RNL=NO                           ->          EX = EXCLUSIVE
        = C* converted    SYSTEMS no HW-RESERVE       ->          G  = GENERIC DEQ
        = E* excluded     SYSTEM  +  HW-RESERVE       ->        ' ' = SYSTEM  DEQ
        = E* excluded     SYSTEM  from SYSTEMS        -> REL/SYS ' ' = SYSTEMS DEQ
        = I* included     SYSTEMS from SYSTEM

SMF     = SMF-ID

ML      = Minor Name Length

SVC     = ENQT, keyword RET=TEST
-------------------------------------------------------------------------------------------------------------------------------

     TIME     JOBNAME PROGRAM  AUT SVC RNL SCOPE     VOLSER   MAJOR     MINOR                                          DEV SMF  ML

DATE  1994 217

13 53.41.96  $FRANCOE SVC-255     ENQ  N SYSTEMS EX         AUDITCOD ENQDELAY                                            ISP1 08
13 53.41.96  $FRANCOE SVC-255     DEQ  N REL/SYS            AUDITCOD ENQDELAY                                            ISP1 08
13 53.41.96  $FRANCOE SVC-034     ENQ    SYSTEM  EX         SYSIEFSD Q10                                                 ISP1 03
13 53.41.97  $FRANCOE SVC-034     DEQ    SYSTEM            SYSIEFSD Q10                                                 ISP1 03
13 53.41.97  CATALOG  IGGPACDV AL DEQ    RELEASE 03 STSO01  SYSZVVDS STSO01                                            ISP1 08
13 53.41.97  CATALOG  IGGPACDV AL DEQ    RELEASE 02 STSO01  SYSZVVDS STSO01                                            ISP1 06
13 53.41.97  CATALOG  IGGPACDV AL DEQ    SYSTEM             SYSZVVDS CATALOG.TSOCAT1                                  ISP1 44
13 53.41.97  CATALOG  IGGPACDV AL DEQ    REL/SYS            SYSIGGV2 CATALOG.TSOCAT1                                  ISP1 20
13 53.41.97  CATALOG  IGGPACDV AL DEQ    REL/SYS            SYSIGGV2 ROC.SPAUL.SCRIPT                 CATALOG.        ISP1 88
13 53.43.62  $FRANCO  ISFMAIN   L ENQ    SYSTEM  EX         SDSFINDX ISF.HASPINDX                     RES42B          ISP1 50
13 53.43.73  $FRANCO  ISFMAIN   L DEQ    SYSTEM             SDSFINDX ISF.HASPINDX                     RES42B          ISP1 50
13 53.46.38  JES2     HASJES20 AL ENQ  E*RESERVE 01 CAT422  SYSZJES2 CAT422MVS422.SYS1.HASPCKPT                 0222ISP1 50
13 53.46.81  ROC      ISREDIT   L ENQ    RESERVE 01 STSO02  SPFEDIT  ROC.CICSCO.SCRIPT                               01A6ISP1 44
13 53.46.81  ROC      SVC-019     ENQ    SYSTEMS EX         SYSZDSCB STSO02AROC.CICSCO.SCRIPT                        ISP1 24
13 53.47.13  $FRANCO  ISFMAIN   L ENQ    SYSTEM  EX         SDSFINDX ISF.HASPINDX                     RES42B          ISP1 50
13 53.47.22  $FRANCO  ISFMAIN   L DEQ    SYSTEM             SDSFINDX ISF.HASPINDX                     RES42B          ISP1 50
13 53.47.42  ROC      SVC-019     ENQ    SYSTEMS EX         SYSZDSCB STSO02SROC.CICSCO.SCRIPT                        ISP1 24
13 53.47.45  ROC      SVC-019     DEQ    REL/SYS            SYSZDSCB STSO02SROC.CICSCO.SCRIPT                        ISP1 24
13 53.47.45  ROC      SVC-019     DEQ    REL/SYS            SYSZDSCB STSO02AROC.CICSCO.SCRIPT                        ISP1 24
13 53.47.45  ROC      ISREDIT   L DEQ    REL/SYS            SPFEDIT  ROC.CICSCO.SCRIPT                               ISP1 44
13 53.47.50  JES2     HASJES20 AL DEQ    RELEASE 00 CAT422  SYSZJES2 CAT422MVS422.SYS1.HASPCKPT                      ISP1 50
13 53.48.13  ROC      ISREDIT   L DEQ    REL/SYS            SPFEDIT  ROC.CICSCO.SCRIPT                 ESE1          ISP1 52
13 53.50.68  SMS      IGDICMT0 AL ENQ    SYSTEM  EX         SYSZIGDI ICMRT.CMDSADDR_LOCKED                           ISP1 21
13 53.50.68  SMS      IGDICMT0 AL ENQ    RESERVE 01 STSO02  IGDCDSXS SMS.IPC1.COMMDS                                 01A6ISP1 44
13 53.50.73  SMS      IGDICMT0 AL DEQ    RELEASE 00 STSO02  IGDCDSXS SMS.IPC1.COMMDS                                 ISP1 44
13 53.50.74  SMS      IGDICMT0 AL DEQ    SYSTEM             SYSZIGDI ICMRT.CMDSADDR_LOCKED                           ISP1 21
13 53.51.15  $FRANCO4 SVC-099     ENQ  I*SYSTEMS EX         SYSDSN   PROVA                                           ISP1 05
13 53.51.18  $FRANCO4 SVC-099     ENQ    SYSTEM  EX         SYSIEFSD Q4                                                  ISP1 02
13 53.51.19  $FRANCO4 SVC-026     ENQ  C*SYSTEMS EX MVS4R1  SYSVTOC  MVS4R1                                           ISP1 06
13 53.51.20  $FRANCO4 SVC-026     DEQ    REL/SYS            SYSVTOC  MVS4R1                                           ISP1 06
13 53.51.20  ELOESA2  SVC-216     ENQ    SYSTEM  EX         SYSZLLA1 UPDATE                                          ISP1 06
13 53.51.25  ELOESA2  SVC-216     DEQ    SYSTEM             SYSZLLA1 UPDATE                                          ISP1 06
13 53.51.25  LLA      CSVLLCRE AL ENQ    SYSTEM  EX         SYSZLLA1 UPDATE                                          ISP1 06
13 53.51.31  LLA      CSVLLCRE AL DEQ    SYSTEM             SYSZLLA1 UPDATE                                          ISP1 06
13 53.51.79  PER      ISREDIT   L ENQ    RESERVE 01 PR0001  SPFEDIT  DISO34.INSTLIB                                  01ABISP1 44
13 53.52.00  PER      SVC-019     ENQ    REL/SYS            SYSZDSCB PR0001SDISO34.INSTLIB                           ISP1 21
13 53.52.00  PER      SVC-019     DEQ    REL/SYS            SYSZDSCB PR0001ADISO34.INSTLIB                           ISP1 21
13 53.52.00  PER      ISREDIT   L DEQ    REL/SYS            SPFEDIT  DISO34.INSTLIB                                  ISP1 44
13 53.52.07  PER      ISREDIT   L DEQ    REL/SYS            SPFEDIT  DISO34.INSTLIB                 DSVFPM           ISP1 52
13 53.52.53  JES2     HASJES20 AL ENQ    RESERVE 01 CAT422  SYSZJES2 CAT422MVS422.SYS1.HASPCKPT                      ISP1 50
13 53.53.24  $FRANCO  SVC-034     ENQ    SYSTEM  EX         SYSIEFSD Q10                                                 ISP1 03
13 53.53.24  $FRANCO  SVC-034     DEQ    SYSTEM             SYSIEFSD Q10                                                 ISP1 03
13 53.53.30  $FRANCOE SVC-034     ENQ    SYSTEM  EX         SYSIEFSD Q10                                                 ISP1 03
13 53.53.30  $FRANCOE SVC-034     DEQ    SYSTEM             SYSIEFSD Q10                                                 ISP1 03
13 53.53.63  JES2     HASJES20 AL DEQ    RELEASE 00 CAT422  SYSZJES2 CAT422MVS422.SYS1.HASPCKPT                      ISP1 50
13 53.53.83  ROC      ISREDIT   L ENQ    SYSTEMS EX         SPFEDIT  ROC.CICSCO.SCRIPT                 HEAD1         ISP1 52
13 53.55.12  $FRANCO  ISFMAIN   L ENQ    SYSTEM  EX         SDSFINDX ISF.HASPINDX                     RES42B          ISP1 50
13 53.55.24  $FRANCO  ISFMAIN   L DEQ    SYSTEM             SDSFINDX ISF.HASPINDX                     RES42B          ISP1 50
13 53.56.32  ELOESA2  SVC-216     ENQ    SYSTEM  EX         SYSZLLA1 UPDATE                                          ISP1 06
```

Figure 42. Trace Report for Single System

```
------------------------------------------------------------------------------------------------------------------------------------------
PROGRAM = NOT-PRB, no CDE associated            SCOPE -> RESERVE  XX = DEVICE RESERVE COUNT

                                                      -> RELEASE  XX = DEVICE RESERVE COUNT
                                                      -> REL/SYS  XX = DEQ ASSOCIATED WITH RESER
                                                      -> SYSTEM(S)SH = SHARED
RNL     = N, keyword RNL=NO                           ->          EX = EXCLUSIVE
        = C* converted    SYSTEMS no HW-RESERVE       ->          G  = GENERIC DEQ
        = E* excluded     SYSTEM  +  HW-RESERVE       ->         ' ' = SYSTEM  DEQ
        = E* excluded     SYSTEM  from SYSTEMS        -> REL/SYS ' ' = SYSTEMS DEQ
        = I* included     SYSTEMS from SYSTEM

SMF     = SMF-ID

ML      = Minor Name Length

SVC     = ENQT, keyword RET=TEST
------------------------------------------------------------------------------------------------------------------------------------------

REPORTING SMFID=SC47       START=(1995.248 - 07:37.59.120)       END=(1995.248 - 07:39.25.768)
REPORTING SMFID=SC49       START=(1995.248 - 07:38.09.870)       END=(1995.248 - 07:39.23.449)
REPORTING SMFID=SC52       START=(1995.248 - 07:37.50.468)       END=(1995.248 - 07:38.46.077)
SELECTION DATE=(95240.0800:95255.1100)
------------------------------------------------------------------------------------------------------------------------------------------
     TIME   SMFID JOBNAME  PGMNAME SVC RNL SCOPE     VOLSER   MAJOR     MINOR                                                 DEV      ML

DATE  1995 248

07 38.15.89 SC54 SMS      IGDICMT0 ENQ   SYSTEM  EX         SYSZIGDI  ICMRT.CMDSADDR_LOCKED                                          2
07 38.15.89 SC54 SMS      IGDICMT0 ENQ C*SYSTEMS EX TOTCAT  IGDCDSXS  SYS1.COMMDS10                                         0FC7     4
07 38.15.93 SC54 SMS      IGDICMT0 DEQ C*SYSTEMS    TOTCAT  IGDCDSXS  SYS1.COMMDS10                                         0FC7     4
07 38.15.94 SC54 SMS      IGDICMT0 DEQ   SYSTEM             SYSZIGDI  ICMRT.CMDSADDR_LOCKED                                          2
07 38.15.98 SC47 SMS      IGDICMT0 ENQ   SYSTEM  EX         SYSZIGDI  ICMRT.CMDSADDR_LOCKED                                          2
07 38.15.98 SC47 SMS      IGDICMT0 ENQ C*SYSTEMS EX TOTCAT  IGDCDSXS  SYS1.COMMDS10                                         0FC7     4
07 38.16.05 SC47 SMS      IGDICMT0 DEQ C*SYSTEMS    TOTCAT  IGDCDSXS  SYS1.COMMDS10                                         0FC7     4
07 38.16.05 SC47 SMS      IGDICMT0 DEQ   SYSTEM             SYSZIGDI  ICMRT.CMDSADDR_LOCKED                                          2
07 38.16.14 SC52 *MASTER* SVC-076  ENQ   SYSTEM  EX         SYSZLOGR  RECORDER                                                       0
07 38.16.14 SC52 DUMPSRV  IEAVTSDS ENQ   SYSTEM  EX         SYSIEA01  SDPOSTEX                                                       0
07 38.16.14 SC52 DUMPSRV  IEAVTSDS DEQ   SYSTEM             SYSIEA01  SDPOSTEX                                                       0
07 38.16.14 SC52 *MASTER* SVC-076  DEQ   SYSTEM             SYSZLOGR  RECORDER                                                       0
//                        //              //                //
07 38.25.78 SC47 DUMPSRV  SVC-083  DEQ   SYSTEM             SYSZDRK      U83                                                      08
07 38.25.94 SC54 SMS      IGDICMT0 ENQ   SYSTEM  EX         SYSZIGDI ICMRT.CMDSADDR_LOCKED                                         21
07 38.25.94 SC54 SMS      IGDICMT0 ENQ C*SYSTEMS EX TOTCAT  IGDCDSXS SYS1.COMMDS10                                         0FC7     44
07 38.26.01 SC54 SMS      IGDICMT0 DEQ C*SYSTEMS    TOTCAT  IGDCDSXS SYS1.COMMDS10                                                  44
07 38.26.01 SC54 SMS      IGDICMT0 DEQ   SYSTEM             SYSZIGDI ICMRT.CMDSADDR_LOCKED                                         21
07 38.26.04 SC52 *MASTER* SVC-076  ENQ   SYSTEM  EX         SYSZLOGR RECORDER                                                      08
07 38.26.04 SC52 DUMPSRV  IEAVTSDS ENQ   SYSTEM  EX         SYSIEA01 SDPOSTEX                                                      08
07 38.26.04 SC52 DUMPSRV  IEAVTSDS DEQ   SYSTEM             SYSIEA01 SDPOSTEX                                                      08
```

Figure 43. Trace Report for Multiple MVS Systems

```
------------------------------------------------------------------------------------------------------------------------------------------
** VOLUME CAT410 **
-------------------
START=1992.26610.15.28   END=1992.26710.45.26   VOLUME=CAT410    SMFID=ISP1     ELAPSED SECONDS=00000120

 COUNT   TIME OF MAX                            DEV  MAX-COUNT SMFID  VOLUME  AV-MSEC   MIN-MSEC  MAX-MSEC  TOTAL-SEC RATE/MIN

0002694592.266 13.05.00.36                     02AB       05   ISP1  CAT410  00000600  00000005  00003218  00016157  00000019

00014535 SYSZJES2 CAT410SYS1.HASPCKPT                                 CAT410  00001106  00001076  00007264  00016059  00000010
00006032 SYSIGGV2 CATALOG.CAT410                                      CAT410  00000021  00000000  00001787  00000131  00000004
00006048 SYSZVVDS CAT410                                              CAT410  00000017  00000004  00000695  00000104  00000004
00000307 SYSZRACF SYS1.RACF.BKUP1                                     CAT410  00000084  00000042  00000228  00000025  00000000
00000001 SYSIGGV2 CATALOG.TSOCAT2                            CATALOG.  CAT410  00000517  00000517  00000517  00000000  00000000
00000002 SYSIGGV2 DSMUTMSG.TEXT                              CATALOG.  CAT410  00000370  00000247  00000493  00000000  00000000
00000001 SYSIGGV2 DSMAPA1.TEXT                               CATALOG.  CAT410  00000132  00000132  00000132  00000000  00000000
00000010 SYSZVVDS CAT410                                              CAT410  00000045  00000026  00000127  00000000  00000000
00000008 SYSZVVDS CAT410                                              CAT410  00000031  00000031  00000032  00000000  00000000
00000002 SYSIGGV2 @PL.@W000002.DZJOPTWS                      CATALOG.  CAT410  00000013  00000012  00000015  00000000  00000000

------------------------------------------------------------------------------------------------------------------------------------------
** VOLUME CICS01 **
-------------------
START=1992.26611.00.01   END=1992.26710.00.02   VOLUME=CICS01    SMFID=ISP1     ELAPSED SECONDS=00000120

 COUNT   TIME OF MAX                            DEV  MAX-COUNT SMFID  VOLUME  AV-MSEC   MIN-MSEC  MAX-MSEC  TOTAL-SEC RATE/MIN

0000026392.266 16.35.26.62                     02A6       03   ISP1  CICS01  00000021  00000005  00000364  00000005  00000000

00000181 SYSZVVDS CICS01                                              CICS01  00000017  00000005  00000080  00000003  00000000
00000024 SYSVTOC  CICS01                                              CICS01  00000080  00000055  00000141  00000001  00000000
00000050 SYSIGGV2 CATALOG.CIC210                                      CICS01  00000028  00000008  00000364  00000001  00000000
00000004 SYSIGGV2 CATALOG.CIC161                                      CICS01  00000048  00000027  00000092  00000000  00000000
00000002 SYSZVVDS CICS01                                              CICS01  00000039  00000025  00000053  00000000  00000000
00000001 SYSIGGV2 NVAS2.AEMSAMP0                             CATALOG.  CICS01  00000021  00000021  00000021  00000000  00000000

------------------------------------------------------------------------------------------------------------------------------------------
** VOLUME DB0001 **
-------------------
START=1992.26611.00.03   END=1992.26710.00.04   VOLUME=DB0001    SMFID=ISP1     ELAPSED SECONDS=00000120

 COUNT   TIME OF MAX                            DEV  MAX-COUNT SMFID  VOLUME  AV-MSEC   MIN-MSEC  MAX-MSEC  TOTAL-SEC RATE/MIN

0000017692.266 16.33.13.83                     01A9       01   ISP1  DB0001  00000024  00000005  00000211  00000004  00000000

00000149 SYSZVVDS DB0001                                              DB0001  00000017  00000005  00000211  00000002  00000000
00000027 SYSVTOC  DB0001                                              DB0001  00000058  00000045  00000112  00000001  00000000

------------------------------------------------------------------------------------------------------------------------------------------
** VOLUME DB0001 **
-------------------
START=1992.26611150.35   END=1992.26710.23.34   VOLUME=DB0001    SMFID=ISP2     ELAPSED SECONDS=00000120

 COUNT   TIME OF MAX                            DEV  MAX-COUNT SMFID  VOLUME  AV-MSEC   MIN-MSEC  MAX-MSEC  TOTAL-SEC RATE/MIN

0000012092.267 12.12.33.47                     01A9       01   ISP2  DB0001  00000020  00000010  00000309  00000002  00000000

00000101 SYSZVVDS  DB0001                                             DB0001  00000017  00000010  00000309  00000002  00000000
00000019 SYSVTOC  DB0001                                              DB0001  00000068  00000045  00000080  00000001  00000000

------------------------------------------------------------------------------------------------------------------------------------------
** VOLUME DB0002 **
-------------------
START=1992.26611.00.04   END=1992.26710.00.05   VOLUME=DB0002

 COUNT   TIME OF MAX                            DEV  MAX-COUNT SMFID  VOLUME  AV-MSEC   MIN-MSEC  MAX-MSEC  TOTAL-SEC RATE/MIN

0000013192.266 16.33.25.85                     01AA       01   ISP1  DB0002  00000025  00000005  00000143  00000003  00000000

00000103 SYSZVVDS DB0002                                              DB0002  00000018  00000005  00000143  00000001  00000000
00000028 SYSVTOC  DB0002                                              DB0002  00000052  00000039  00000121  00000001  00000000
```

Figure 44. Volumes Reserve Time Report

```
-------------------------------------------------------------------------------------------------------------------------------------------------------
-
** VOLUME PR0002 **
-------------------
START=1992.26611.00.03   END=1992.26710.00.04   VOLUME=ISPDB4    SMFID=ISP1     ELAPSED SECONDS=00000120

 COUNT   TIME OF MAX                              DEV  MAX-COUNT SMFID  VOLUME  AV-MSEC   MIN-MSEC  MAX-MSEC  TOTAL-SEC RATE/MIN

0000010792.266 14.00.03.71                       02A5        02  ISP1   ISPDB4  00000030  00000005  00000135  00000003  00000000

00000076 SYSZVVDS ISPDB4                                                ISPDB4  00000016  00000005  00000059  00000001  00000000
00000024 SYSVTOC  ISPDB4                                                ISPDB4  00000078  00000058  00000135  00000001  00000000
00000007 SYSIGGV2 CATALOG.CSPDB4                                        ISPDB4  00000036  00000007  00000106  00000000  00000000

-------------------------------------------------------------------------------------------------------------------------------------------------------
-
```

```
-------------------------------------------------------------------------------------------------------------------------------------------------------
** MAJOR  SYSZJES2 **
--------------------
 COUNT              MAJOR                                                VOLUME  AV-MSEC MIN-MSEC MAX-MSEC  TOT-SEC  RATE/MIN
00001901            SYSZJES2            SCOPE=RESERVE                SMFID=ISP1

RNL (ML)   COUNT               MINOR

*E   50   00001901  SYSZJES2 CAT422MVS422.SYS1.HASPCKPT                       CAT422   001112  001077  00001391  00002034  000011
-------------------------------------------------------------------------------------------------------------------------------------------------------
** MAJOR  SYSZRACF **
--------------------
 COUNT              MAJOR                                                VOLUME  AV-MSEC MIN-MSEC MAX-MSEC  TOT-SEC  RATE/MIN
00000968            SYSZRACF            SCOPE=RESERVE                SMFID=ISP1

RNL (ML)   COUNT               MINOR

09   00000966  SYSZRACF SYS1.RACF                                         CAT422   000049  000001  00000246  00000046  000005
15   00000002  SYSZRACF SYS1.RACF.BKUP1                                   RES42A   000130  000125  00000135  00000000  000000
-------------------------------------------------------------------------------------------------------------------------------------------------------
** MAJOR  SYSVTOC  **
--------------------
 COUNT              MAJOR                                                VOLUME  AV-MSEC MIN-MSEC MAX-MSEC  TOT-SEC  RATE/MIN
00000750            SYSVTOC             SCOPE=RESERVE                SMFID=ISP1

RNL (ML)   COUNT               MINOR

06   00000490  SYSVTOC  RUBB01                                             RUBB01   000148  000005  00001723  00000059  000002
         //        //
-------------------------------------------------------------------------------------------------------------------------------------------------------
 COUNT              MAJOR                                                VOLUME  AV-MSEC MIN-MSEC MAX-MSEC  TOT-SEC  RATE/MIN
00000120            SYSVTOC             SCOPE=RESERVE                SMFID=ISP2

RNL (ML)   COUNT               MINOR

06   00000120  SYSVTOC  RUBB01                                             RUBB01   000150  000010  00000756  00000018  000002
         //        //
-------------------------------------------------------------------------------------------------------------------------------------------------------
** MAJOR  SPFEDIT  **
--------------------
 COUNT              MAJOR                                                VOLUME  AV-MSEC MIN-MSEC MAX-MSEC  TOT-SEC  RATE/MIN
00000358            SPFEDIT             SCOPE=RESERVE                SMFID=ISP1


RNL (ML)   COUNT               MINOR

44   00000107  SPFEDIT  DBR.PROFILE                                        PR0002   000174  000131  00000373  00000017  000000
44   00000050  SPFEDIT  ROC.PROFILE                                        STSO01   000224  000150  00000352  00000010  000000
         //          //
-------------------------------------------------------------------------------------------------------------------------------------------------------
** MAJOR  SYSIEFSD **
--------------------
 COUNT              MAJOR
00007167            SYSIEFSD            SCOPE=SYSTEM                 SMFID=ISP1

RNL (ML)   COUNT               MINOR

02   00002189  SYSIEFSD Q4
         //       //
-------------------------------------------------------------------------------------------------------------------------------------------------------
** MAJOR  SYSDSN   **
--------------------
 COUNT              MAJOR
00000042            SYSDSN              SCOPE=SYSTEM                 SMFID=ISP1

RNL (ML)   COUNT               MINOR

09   00000041  SYSDSN   SYS1.UADS
08   00000001  SYSDSN   SYS1.DAE
-------------------------------------------------------------------------------------------------------------------------------------------------------
** MAJOR  SYSDSN   **
--------------------
 COUNT              MAJOR
00001452            SYSDSN              SCOPE=SYSTEMS                SMFID=ISP1

RNL (ML)   COUNT               MINOR

I*   18   00000049  SYSDSN   DBR.ISR0001.BACKUP
I*   18   00000047  SYSDSN   FUM.ISR0001.BACKUP
         //           //
-------------------------------------------------------------------------------------------------------------------------------------------------------
```

Figure 45. ENQ/RESERVE Resources Report

```
16 58.59.98  IPOUSR4  NOT-PRB     DEQ    REL/SYS 00 XA9RES  SYSVTOC  XA9RES                                        IPO1

16 59.02.48  JES2     HASJES20 AL ENQ    RESERVE 01 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                  01C5IPO1

16 59.09.91  CATALOG  IGGPACDV AL ENQ    RESERVE 02 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                01C5IPO1
16 59.09.91  CATALOG  IGGPACDV AL ENQ    RESERVE 03 XA9RES  SYSZVVDS XA9RES                               01C5IPO1
16 59.09.94  CATALOG  IGGPACDV AL DEQ    RELEASE 02 XA9RES  SYSZVVDS XA9RES                                   IPO1
16 59.10.03  CATALOG  IGGPACDV AL DEQ    REL/SYS 01 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                   IPO1
16 59.10.69  CATALOG  IGGPACDV AL ENQ    RESERVE 02 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                01C5IPO1
16 59.10.70  CATALOG  IGGPACDV AL ENQ    RESERVE 03 XA9RES  SYSZVVDS XA9RES                               01C5IPO1
16 59.10.72  CATALOG  IGGPACDV AL DEQ    RELEASE 02 XA9RES  SYSZVVDS XA9RES                                   IPO1
16 59.10.76  CATALOG  IGGPACDV AL DEQ    REL/SYS 01 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                   IPO1
16 59.10.91  CATALOG  IGGPACDV AL ENQ    RESERVE 02 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                01C5IPO1
16 59.10.91  CATALOG  IGGPACDV AL ENQ    RESERVE 03 XA9RES  SYSZVVDS XA9RES                               01C5IPO1
16 59.10.96  CATALOG  IGGPACDV AL DEQ    RELEASE 02 XA9RES  SYSZVVDS XA9RES                                   IPO1
16 59.10.98  CATALOG  IGGPACDV AL DEQ    REL/SYS 01 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                   IPO1
16 59.11.13  CATALOG  IGGPACDV AL ENQ    RESERVE 02 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                01C5IPO1
16 59.11.13  CATALOG  IGGPACDV AL ENQ    RESERVE 03 XA9RES  SYSZVVDS XA9RES                               01C5IPO1
16 59.11.15  CATALOG  IGGPACDV AL DEQ    RELEASE 02 XA9RES  SYSZVVDS XA9RES                                   IPO1
16 59.11.17  CATALOG  IGGPACDV AL DEQ    REL/SYS 01 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                   IPO1
16 59.11.35  CATALOG  IGGPACDV AL ENQ    RESERVE 02 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                01C5IPO1
16 59.11.35  CATALOG  IGGPACDV AL ENQ    RESERVE 03 XA9RES  SYSZVVDS XA9RES                               01C5IPO1
16 59.11.43  CATALOG  IGGPACDV AL DEQ    RELEASE 02 XA9RES  SYSZVVDS XA9RES                                   IPO1
16 59.11.51  CATALOG  IGGPACDV AL DEQ    REL/SYS 01 XA9RES  SYSIGGV2 CATALOG.MASTER.XA9RES                   IPO1
16 59.11.89  IPOUSR4  ISRUDA      ENQ    RESERVE 02 XA9RES  SPFEDIT  TOOLS.PRC                           01C5IPO1
16 59.12.65  JES2     HASJES20 AL DEQ    RELEASE 01 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                     IPO1
16 59.17.66  JES2     HASJES20 AL ENQ    RESERVE 02 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                  01C5IPO1
16 59.27.96  JES2     HASJES20 AL DEQ    RELEASE 01 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                     IPO1
16 59.32.97  JES2     HASJES20 AL ENQ    RESERVE 02 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                  01C5IPO1
16 59.43.22  JES2     HASJES20 AL DEQ    RELEASE 01 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                     IPO1

16 59.44.76  IPOUSR4  ISRUDA      DEQ    REL/SYS 00 XA9RES  SPFEDIT  TOOLS.PRC                               IPO1

16 59.48.23  JES2     HASJES20 AL ENQ    RESERVE 01 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                  01C5IPO1
16 59.58.37  JES2     HASJES20 AL DEQ    RELEASE 00 XA9RES  SYSZJES2 XA9RESSYS1.HASPCKPT                     IPO1
```

Figure 46. Reserve/Release Trace at Time of Max

The trace at time of max is the output of the **LOG2** DD statement of the ISGAJE2 procedure. See Figure 38 on page 72.

Records layout

DSECT monitor physical and logical record

```
BUFF     DSECT                     SP 229
ACPB         DC    4C'ACPB'      Buffer eye catcher
NEXT     DC    1F'0'         4
SRBOFFB  EQU   *-LOCK             OFFSET SRB IN BUFFER
SRBAREA  DC    44X'00'       8    SRB FOR BUFFER SCHEDULE
OFFDATA  EQU   *-LOCK
DATA     EQU   BUFFSIZ-OFFDATA

RECORD   DSECT                     30 RECORDS PER BUFFER
HEADER   DC    4C'0'
NORMAL   EQU   'AREC'              NORMAL
ERROR    EQU   C'ERRO'             ERROR DURING DATA COLLECTION
LOST     EQU   C'LOST'             DATA LOST PREVIOUSLY
RFLAG    DC    X'00'
JOBINIT  EQU   X'80'
STCTSU   EQU   X'40'
SVCNUMB  DC    X'00'
TOD      DC    8X'00'
JOBNAME  DC    8C'0'
DEVNO    DC    2X'00'            DEVICE NUMBER BINARY
         DC    1X'00'            SPARE
CONRNL   DC    1C'0'             RNL ACTION
SMFID    DC    4C'0'             SMFID
ENQFLG1  DC    X'00'             ENQ/DEQ FLAGS
ENQFLG2  DC    X'00'
VOLSER   DC    6C'0'             VOLSER
PGMNAME1 DC    8C'0'             NAME
ATTR1    DC    X'00'             ATTRIBUTE
SUBRECL  EQU   *-PGMNAME1
MAJOR    DC    8C' '             MAJOR NAME
MINOR    DC    52C' '            MINOR NAME
MINORLEN DC    1X'00'            MINOR LENGTH
USERID   DC    8c' '             Userid
         DC    11X'00'
RESCNT   DC    X'00'             VOLUME RESERVE COUNT
ENDREC   EQU   *
```

ISGAMEDM ISGAMED1 ISGAMVOL output record fields

```
*
*  Output record of data reduction programs ISGAMEDM-ISGAMED1-ISGAMVOL
-------------------------------------------------------------------
start length     contents         format      comments
-------------------------------------------------------------------

  1 -  8         TOD            8 BIN       ISGAMEDM-ISGAMVOL

  1 - 13         DATE          13 CHAR    ISGAMED1

  1 - 11         TIME          11 CHAR      ISGAMED1

 14 -  8        JOBNAME         8 CHAR

 23 -  8        PGNAME          8 CHAR

 32 -  2        PGM AUTH        2 CHAR       A L

 35 -  3        REQUEST         3 CHAR       ENQ/DEQ

 40 -  2        RNL             2 CHAR

 42 -  7        SCOPE           8 CHAR       SYSTEM(S)/RESERVE

 50 -  2        SCOPE1          2 CHAR       EX-SH  RES/CNT

 53 -  6        VOLID           6 CHAR

 61 -  8        MAJOR NAME      8 CHAR

 69 -  1        MINOR NAME LEN   1 BIN       LEN OF PRINTABLE MINOR

 70 - 52        MINOR NAME      52 CHAR

122 -  4       DEVICE NUMBER    4 CHAR

126 -  4       SMF ID           4 CHAR

130 -  3       MINOR NAME LEN   3 CHAR
```

EDSORTED output record fields

```
*
*  Output record of data reduction program ISGAMCNT
-------------------------------------------------------------
start length     contents         format      comments
-------------------------------------------------------------

  1 -  8        TOTAL COUNT      8 CHAR

 35 -  3        REQUEST          3 CHAR       ENQ/DEQ

 40 -  2        RNL              2 CHAR

 42 -  7        SCOPE            8 CHAR       SYSTEM(S)/RESERVE

 50 -  2        SCOPE1           2 CHAR       EX-SH  RES/CNT

 53 -  6        VOLID            6 CHAR

 61 -  8        MAJOR NAME       8 CHAR

 69 -  1        MINOR NAME LENGTH 1 BIN       LEN OF PRINTABLE MINOR

 70 - 52        MINOR NAME       52 CHAR

122 -  4        DEVICE NUMBER    4 CHAR

126 -  4        SMF ID           4 CHAR

130 -  3        MINOR NAME LENGTH 3 CHAR
```

ISGAMCTM output record fields

```
*
*  Output record of data reduction program ISGAMCTM
*
-----------------------------------------------------------------
start length     contents         format      comments
-----------------------------------------------------------------

 1 -  8         YYYY DDD         6 CHAR      date of max

 9 - 11         HH.MM.SS.mm     11 CHAR      time of max

21 -  4         DEVICE NUMBER    4 CHAR

25 -  6         VOLID            6 CHAR

32 -  4         SMFID            4 CHAR

37 -  8         TOT RES TIME     8 CHAR      seconds

46 -  8         AVERAGE          8 CHAR      msec

55 -  8         MINIMUN          8 CHAR      msec

64 -  8         MAXIMUM          8 CHAR      msec

73 -  8         NUM OF REQUEST   8 CHAR

82 -  2         MAX CONC RES     2 CHAR
```

Part 2. Global Resource Serialization Star

Chapter 4. Star processing

In a star complex, global resource serialization uses a XES lock structure to serialize requests for global resources.

Figure 47 on page 91 shows an example overview of a three system star complex. All systems in a star complex must be members of the same sysplex and be connected to a coupling facility containing the global resource serialization lock structure, ISGLOCK, to manage contention for global resources. For practical purposes, where global resource serialization star complex is concerned, the terms sysplex and complex are synonymous. No channel-to-channel (CTC) connection of systems, other than those managed by XCF, are supported by global resource serialization in a star complex.

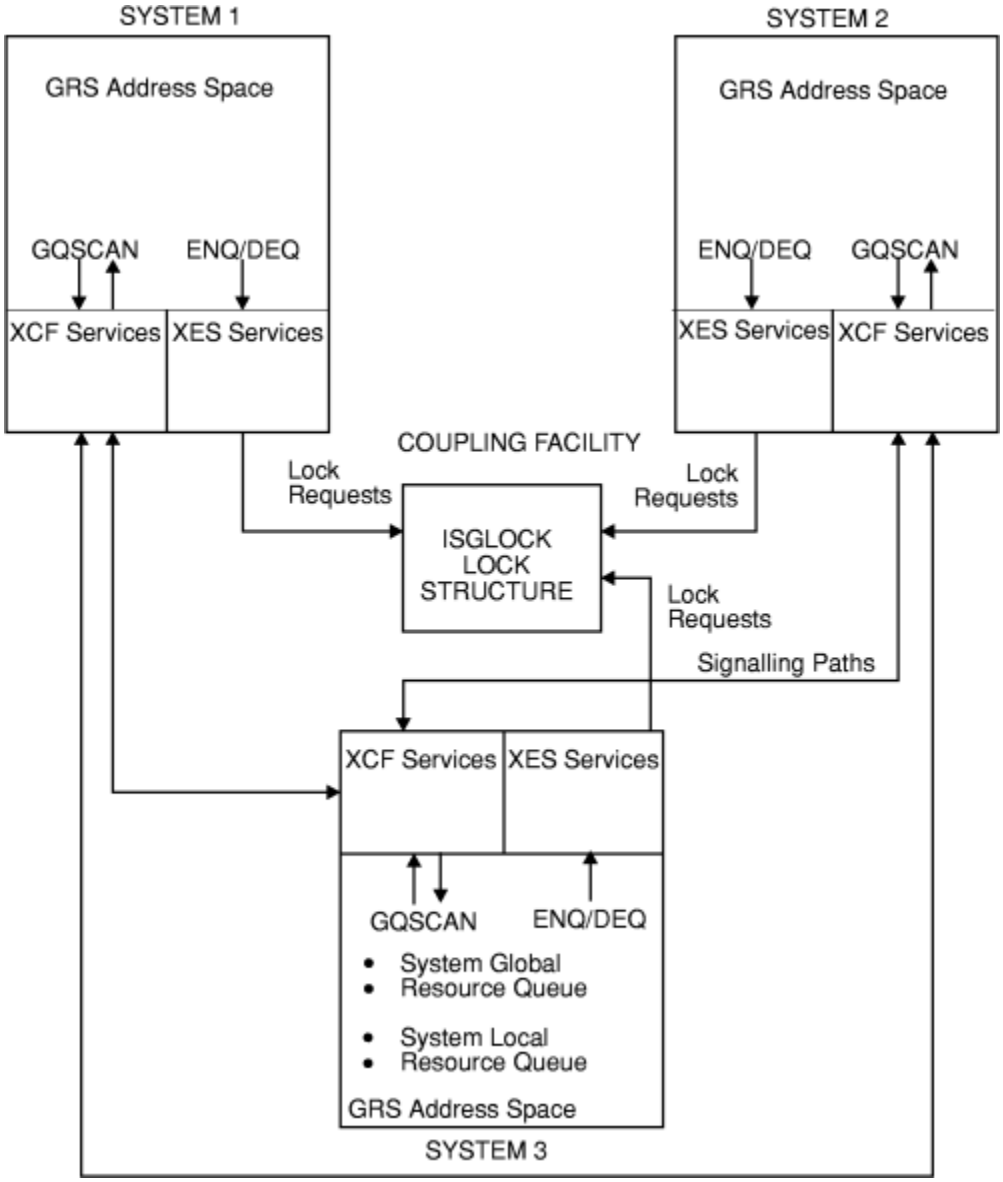


Figure 47. Overview of the global resource serialization star complex

The ISGLOCK Lock Structure: When a system in a star complex issues an ENQ, DEQ, ISGENQ, or RESERVE request for a global resource, global resource serialization converts the request to a lock request against the ISGLOCK lock structure. Global resource serialization uses the ISGLOCK lock structure to coordinate the requests to ensure proper resource serialization across all systems in the

complex. The status of each request is returned to the system that originated the request. Based on the results of these lock requests, global resource serialization will respond to the requester with the outcome of the serialization request.

Global ENQ/DEQ processing overview

In a star complex, requests for ownership of global resources will be handled through ISGLOCK, the lock structure, on a coupling facility that is fully connected with all the systems in the sysplex. Global resource serialization uses the ISGLOCK lock structure to reflect a composite system level view of the interest in every global resource, for which there is at least one requester. In general, global resource serialization alters this composite view each time you change the set of requesters for the resource.

Each time an ENQ request is received, global resource serialization processing analyzes the state of the resource request queue for the resource. If the new request alters the composite state of the queue, an IXLLOCK macro request is made to reflect the changed state of the resource for the requesting system. If the resource is immediately available, the requester is then granted ownership of the resource.

If the resource is not immediately available, global resource serialization will maintain the request in the waiting state. When the appropriate DEQ request is received, global resource serialization will either resume or post the requester (depending on the ENQ options).

ISGQUERY and GQSCAN processing overview

In a global resource serialization star sysplex, the basic flow of a GQSCAN/ISGQUERY request for global resource information is illustrated by the high level diagram in Figure 48 on page 92.

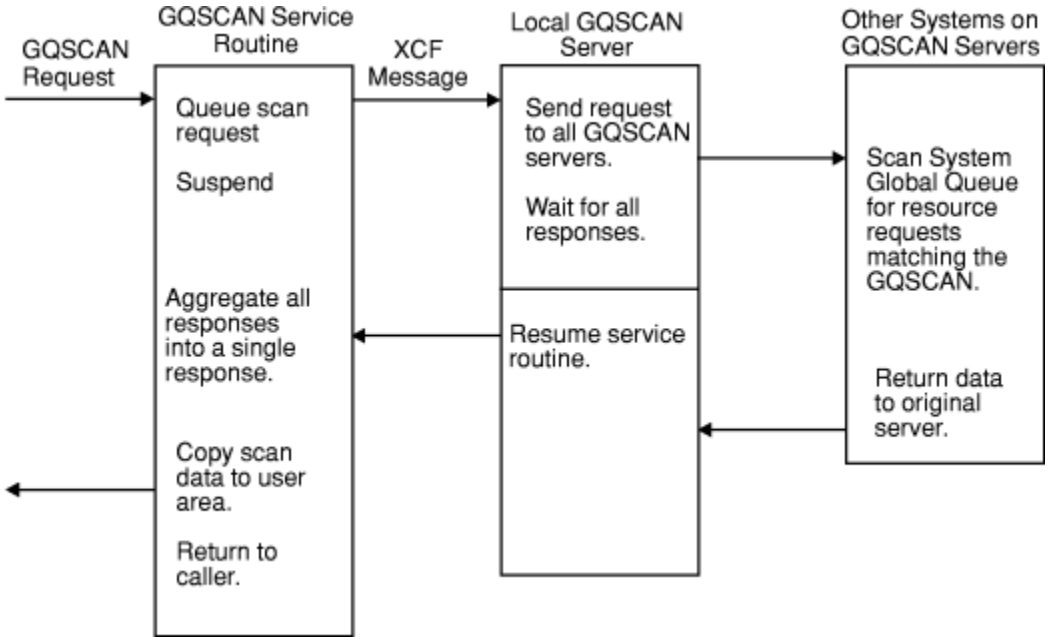


Figure 48. Overview of GQSCAN/ISGQUERY request for global resource data

The flow, as illustrated in Figure 48 on page 92, is as follows:

- The request for global resource information is queued to the GQSCAN/ISGQUERY server on that system.
- Global resource serialization suspends the GQSCAN/ISGQUERY requester. For GQSCAN/ISGQUERY users that do not need information about other systems and can not tolerate suspension, see "Cross-system processing option" on page 93.
- Global resource serialization processing will then package the request for transmission and send it to each of the other systems in the star complex.

- Each system in the complex scans its system global resource queue for global resource requests that match the GQSCAN/ISGQUERY selection criteria, and respond to the originating system server with the requested global resource information, if any, and a return code.
- The originating system waits for responses from all of the systems in the sysplex and builds a composite response, which is returned to the caller's output area through the GQSCAN/ISGQUERY back-end processing. Control is then returned to the caller with the appropriate return code.

Cross-system processing option

The XSYS option on the GQSCAN macro allows the GQSCAN issuer to indicate whether cross-system processing is required. The default is XSYS=YES. You can specify the option XSYS=NO to turn cross-system processing off for that particular GQSCAN request. If you specify the no cross-system option, XSYS=NO, only the global resource information for the caller's system is returned. This type of GQSCAN can run under the caller's task, without causing the unit of work to be suspended.

Specifying XSYS=NO benefits users of GQSCAN that cannot be suspended and do not require data about requesters on other systems in the complex.

For more information about the GQSCAN macro, see *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*.

Processing system failures

There is no disruption to the star complex when a failure occurs that requires a system be removed from the sysplex. Processing for removing the system from the sysplex amounts to a dequeueing of all requests for global resources that originated on that system. Global resource serialization continues processing global requests for the remaining systems.

See *z/OS MVS Setting Up a Sysplex* for information on planning sysplex availability and recovery.

Contention management

Resource contention can result in less important work holding resources that are needed by more important work. To prevent this, ENQ and Latch contention processing issue an SRM ENQHOLD SYSEVENT to promote the work holding the resource. The installation uses the enqueue residence value (ERV) on the IEAOPTxx parmlib member to alter the amount of CPU service units that an address space or an enclave is allowed to absorb when it is causing contention. During this time, the work unit runs with increased dispatching priority.

Because a DASD RESERVE can also cause hardware contention, but not an ENQ contention, global resource serialization always issues an SRM ENQHOLD SYSEVENT against authorized programs that reserve a DASD device. Global resource serialization cannot know if there is a hardware contention because the associated RESERVE ENQ is usually excluded to a SYSTEM scope and there could be a waiter on another system that is sharing the DASD.

Additional information about contention management can be found in the following documentation:

- *z/OS MVS Initialization and Tuning Guide*.
- *z/OS MVS Initialization and Tuning Reference*.
- *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*.
- *z/OS MVS Programming: Authorized Assembler Services Guide*.

See “Checking for ENQ contention problems” on page 180 to determine how to find ENQs and GRS-managed latches that are in contention.

Contention notification

Resource contention can cause poor system performance and resource contention that lasts over time can result in program starvation or deadlock conditions. Global resource serialization provides APIs to query for ENQ contention information:

- ISGECA - obtains waiter and blocker information about SYSTEM scope and SYSTEMS scope ENQ resources.
- ISGQUERY - obtains the status of resources and the requesters of STEP scope, SYSTEM scope, and SYSTEMS scope ENQ resources.

Global resource serialization also issues an event notification facility (ENF) signal 51 to notify monitoring programs to track contention for SYSTEM and SYSTEMS scope ENQ resources. This is useful in determining contention bottlenecks, preventing bottlenecks, and potentially automating correction of these conditions.

In ring mode, each system in the global resource serialization complex knows about the complex wide SYSTEMS scope ENQs that allow each system to issue the appropriate ENF 51 contention notification event. However, in star mode, each system only knows about the ENQs that are issued by the system. To ensure that the ENF 51 notification for SYSTEMS scope ENQs are issued on all systems in proper sequential order, one system in the sysplex is appointed as the sysplex wide SYSTEMS contention notification system (CNS). All SYSTEMS scope ENQ contention events are sent to the notifying system, which then issues a sysplex wide ENF 51. During IPL or if the notification system can no longer perform the duties, any system in the sysplex can act as the contention notification system. You can determine the current contention notification system through the DISPLAY GRS command.

For more information about monitoring contention changes and a complete description of ENF 51, see ENF event codes and meanings in *z/OS MVS Programming: Authorized Assembler Services Guide*.

Star configurations choose which system to act as the contention notifying system with the SETGRS CNS command. All systems in the global resource serialization star complex compatible with z/OS V1R7 require a PTF for the SETGRS CNS command to work. If any system in the complex is earlier than z/OS V1R7 or without the PTF installed, the command cannot be issued by any member of the complex.

If the contention notification system (CNS) fails, one of the remaining systems automatically becomes the new CNS. Your installation might consider automation for the rare case the system moves the CNS. If so, key off message ISG364I for the *SYSTEM INITIATED* cases and set the CNS through SETGRS to the system you choose.

Additional information about contention notification and listening for system events is found in:

- *z/OS MVS Programming: Authorized Assembler Services Guide*.
- *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*.
- *z/OS MVS Programming: Sysplex Services Guide*.
- *z/OS MVS Installation Exits*.

Chapter 5. Planning a star complex

Decisions you need to make

Decisions you make about how you want to process requests for various resources set your installation's goals for global resource serialization. The actual global resource serialization star complex that you design is one of the tools you use to achieve these goals.

A global resource serialization star complex consists of all the systems that are able to share global resources.

Designing the complex involves answering these basic questions:

- What resources does your installation want to share?
- Which systems use these resources?

The resources your systems need to share determine the systems in the complex. The most likely candidates, of course, are those systems that are already serializing access to resources on shared DASD volumes and, especially, those systems where interlocks, contention, or data protection by job scheduling are causing significant problems.

Notes

- In a star complex the terms, sysplex and complex are synonymous, in that the systems in the complex are the same as the systems in the sysplex.
- "Systems" refers to the number of MVS images, not the number of processors.
- Global resource serialization ring complexes can contain XCFLOCAL, MONOPLEX, or systems that are part of a multisystem sysplex. However, only one multisystem sysplex can exist in any global resource serialization complex.

It is possible for a single installation to have two or more global resource serialization complexes, each operating independently. However, the independent complexes cannot share resources. Also, be certain there are no common links available to global resource serialization on any two complexes.

To avoid a data integrity exposure, ensure that no system outside the complex can access the same shared DASD as any system in the complex. If that is unavoidable, however, you must serialize the data on the shared DASD with the RESERVE macro. You need to decide how many of these systems to combine in one complex.

In large complexes, the star mode works better than ring mode. See "Methods of serializing global resources" on page 12. Designing your star complex is relatively simple, because all of the systems in the star complex are part of the same sysplex.

Designing a star complex

The star method for processing requests for global resources operates in a sysplex like any other MVS component that uses the coupling facility. Unlike the ring method, which might require you to set up XCF signalling paths for the SYSGRS group, the star method will operate well without any special definitions. There is no need to set up XCF signalling paths between systems or to define CTC links directly to global resource serialization.

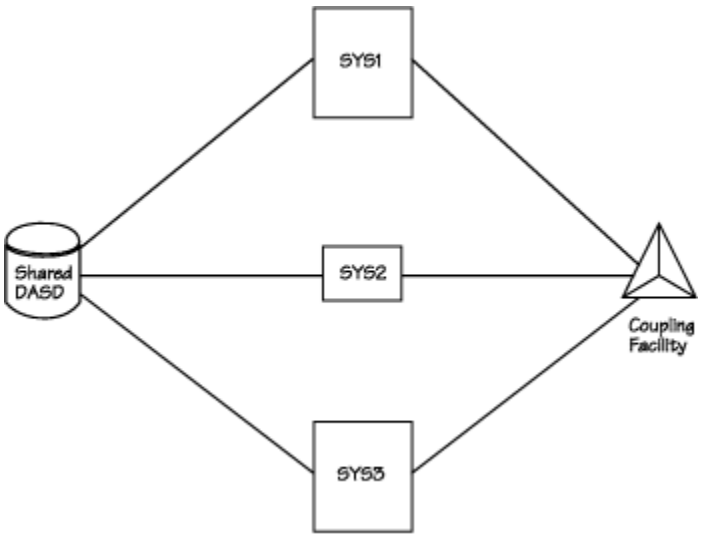


Figure 49. The Star Concept

XCF requires a DASD data set, called a sysplex couple data set, to be shared by all systems. An alternate data set can be used to facilitate migrating from a ring to a star complex. On the sysplex couple data set, MVS stores information related to the sysplex, systems, and XCF groups, such as global resource serialization.

The following policies are used to manage global resource serialization:

- Coupling facility resource management (CFRM) policy, which is required, defines how MVS manages coupling facility resources.
- Sysplex failure management (SFM) policy, which is optional, defines how MVS is to manage sysplex resources. GRS exploits both XCF and XES critical member support and indicates a TERMLEVEL of SYSTEM. For more information about defining SFM policy, see *z/OS MVS Setting Up a Sysplex*. IBM suggests using the SFM policy.

Defining the sysplex couple data set for the star complex

You must format the sysplex couple data set to accept global resource serialization before you can:

- Initialize a star complex, or
- Migrate to a star complex from a ring complex.

1. When preparing a star complex, use the IXCL1DSU utility to allocate the sysplex couple data set.
2. Under the DEFINEDS statement for the TYPE(SYSPLEX) couple data set, add the following parameters:

```
DEFINEDS
  DATA TYPE(SYSPLEX)
    ITEM NAME (GRS) NUMBER(1)
```

Figure 50. Syntax of the Global Resource Serialization Record for IXCL1DSU

Where:

ITEM NAME(GRS) NUMBER(1)
Allocates the couple data set storage for use by global resource serialization.

The above keyword specification is a requirement for a star complex. If the specification is omitted, no storage will be allocated for use by global resource serialization. For more information on the utility, see *z/OS MVS Setting Up a Sysplex*.

3. You can format and bring on-line a sysplex couple data set with the global resource serialization record any time prior to migrating to a star complex. Use the SETXCF command to:

- Make a newly formatted sysplex couple data set the alternate sysplex couple data set, then
- Specify the alternate sysplex couple data set as the primary one.

For information on the use of the SETXCF command, see:

- *z/OS MVS Setting Up a Sysplex*.
- *z/OS MVS System Commands*.

Use of the sysplex couple data set

In a star complex, the Resource Name Lists (RNLs) are maintained on the sysplex couple data set. The RNLs are initialized by the first system to join the sysplex. All subsequent systems will read this data from the sysplex couple data set and compare it to the version used during system initialization. RNLs are specified using the GRSRNL= parameter in the IEASYSxx member of parmlib. If the RNLs used by the system during initialization do not match those in the sysplex couple data set, MVS issues message ISG312W and the system is put into an X'0A3' wait state.

Dynamic RNL processing

For dynamic RNL changes in a global resource serialization star complex, change the RNLs on the sysplex couple data set. This method ensures that all new systems to join the complex use the proper set of RNLs. For more information on changing RNLs see "Changing RNLs for a star complex" on page 107.

The CFRM couple data set

Management of global resources in the star complex is performed through a coupling facility lock structure. The coupling facility is managed by the coupling facility resource management (CFRM) policy, which resides on the CFRM coupling facility data set.

The CFRM policy defines:

- The lock structure
- The amount of coupling facility storage to be used for the lock structure
- The preference list of coupling facilities in which the structure resides, and
- An unordered exclusion list of other structures that should not be allocated in the same coupling facility as the specified lock structure.

You must install and define at least one coupling facility to MVS before you can configure a star complex. However, IBM suggests that you **do not** use the star method of serializing global resources if there is only one coupling facility available for use in your installation. In this case, failure of the coupling facility will result in termination of all the systems in the sysplex with an X'0A3' wait state.

Global resource serialization uses a lock structure, named ISGLOCK, to serialize global resources across the sysplex and record which systems have requesters for particular resources. You must specify ISGLOCK **must** in the CFRM policy by using ISGLOCK as the structure name on the NAME parameter.

Following is an example of the structure definition for the ISGLOCK structure. See Figure 52 on page 98 to determine the size for ISGLOCK.

```
STRUCTURE
     NAME(ISGLOCK)
     SIZE(8448)
     PREFLIST(CFACIL01,CFACIL02)
     EXCLLIST(ABCSTR)
```

Figure 51. Specifying Global Resource Serialization Lock Structure in the CFRM Policy

Note that the optional CFRM policy parameter, INITSIZE, is not specified in the ISGLOCK structure definition. It is not necessary to specify INITSIZE because global resource serialization does not support structure alter.

For information on managing coupling facility resources, see *z/OS MVS Setting Up a Sysplex*. For a discussion of GRS rebuild processing, see "ISGLOCK rebuild processing" on page 102.

Sizing the ISGLOCK structure

As stated before, global resource serialization uses the lock structure, ISGLOCK, to manage resource serialization requests for the sysplex. You specify the size of ISGLOCK in your CFRM policy. The size is based on the following factors:

- The size and type of systems in the sysplex, and
- The type of workload being performed.

Global resource serialization requires that the lock structure contains at least 32767 (32K) locks. Typically, a small sysplex, made up of smaller processors, running a transaction processing workload will use a smaller lock size than a larger sysplex, composed of large processors, running a batch/TSO workload combination.

IBM suggests that you use the method described in Figure 52 on page 98 to determine the size for ISGLOCK or the procedure found in the z/OS Internet Library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

```
# lock table entries = peak # resources * 100

Round # lock table entries up to next power of 2

Lock table size (in bytes) = # lock table entries * 8

Structure Size (in K bytes) = (lock table size/1024) + 256
```

Figure 52. Formula for Determining SIZE Parameter for the CFRM Policy

Where:

Peak # Global Resources
Is the number of unique globally managed resources (SYSTEMS ENQs and converted RESERVEs) measured at a time of peak system utilization.

IBM has provided a program, ISGSCGRS in SYS1.LINKLIB, that runs GQSCAN. ISGSCGRS returns the number of global resources in the complex. The JCL for the program is in SYS1.SAMPLIB(ISGCGRS).

Getting the right structure size

When you create CFRM policy specify your INITSIZE parameter equal to the SIZE parameter (SIZE=INITSIZE) for the structure. Remember the number of lock entries cannot be altered.

When the first system joins the star, global resource serialization uses information about the structure size specified in the CFRM policy and then creates the largest lock structure that will fit within that size. Message ISG3371I is issued to document the size of the structure. If the size of the ISGLOCK lock structure created by global resource serialization is less than the CFRM policy size, global resource serialization will issue message ISG322A. If you receive this message, check the coupling facility configuration. If either the policy size or the amount of storage available in the coupling facility is insufficient, global resource serialization issues message ISG338W.

The size of the lock structure can be tuned by checking the number of false contention occurrences in the resource management facility (RMF) report. If the false contention rate is high (greater than one or two percent), use a larger lock structure (double the number of lock entries) to reduce the rate. You can increase the policy size for the structure and then rebuild the structure to reduce the amount of false contention. To rebuild a coupling facility structure, use the SETXCF STAR T, REBUILD command.

See "Rebuilding the ISGLOCK structure" on page 108 on the use of the SETXCF command.

Defining parmlib members for a star complex

Initializing a star complex requires specifying STAR on the GRS= system parameter in the IEASYS*xx* parmlib member, or in response to message `IEA101A SPECIFY SYSTEM PARAMETERS`. The definitions of the START, JOIN, and TRYJOIN options apply to a ring complex only. The NONE option applies when global resource serialization is not required.

For a star complex, the syntax of the GRS= parameter and the specification for initiating a system into a star complex is described below. The processing that occurs for the star option is described in "Bringing up a star complex" on page 101. This section also describes what occurs when systems with mismatched GRS= specifications are initialized into the same GRS complex.

GRS=STAR

Figure 53. Syntax for the GRS= System Parameter

STAR
Specifies that the system being IPLed is to join a global resource serialization star complex. If no global resource serialization complex has yet been established (that is, there are no systems active in either a ring or star complex), the system will be initialized as the first system in a global resource serialization star complex.

Note: The GRSRNL= parameter should also be specified if the GRS= parameter specifies STAR.

Global resource serialization star definition parameter (GRSDEF)

GRSCNF=xx identifies the GRSCNFxx parmlib member that contributes toward the definition of the complex. GRSCNF00 is the default member and contains ring-specific keywords, so another GRSCNFxx member should be defined by the installation for star. The GRSDEF syntax for keywords specific to star mode are shown in Figure 54 on page 99.

For further information on the parameters mentioned in this topic, see *z/OS MVS Initialization and Tuning Reference*.

```
GRSDEF  [MATCHSYS{(name)}]
                [{(*)}   ]
        [CTRACE (parmlib member name)]
        [SYNCHRES     {(YES)                        }]
        [             {(NO)                         }]
        [GRSQ         {(ALL)                        }]
        [             {(CONTENTION)                 }]
        [             {(LOCAL)                      }]
        [ENQMAXA(1-8 decimal digit)]
        [ENQMAXU(1-8 decimal digit)]
```

Figure 54. Star-relevant parameters for GRSCNFxx

Specifying the name of the system for GRSDEF parameters (MATCHSYS)

MATCHSYS({*name* | *}) specifies the name of the system the GRSDEF parameters are to be associated with. If MATCHSYS(*) is specified, the GRSDEF parameters are to be used by all systems for which there is no GRSDEF statement with a matching system name.

Specifying global resource serialization tracing options (CTRACE)

The CTRACE(*parmlib member name*) parameter in GRSCNFxx allows you to modify the default tracing options used by global resource serialization. Since the defaults provide adequate serviceability information, these should be changed only upon the recommendation of your IBM Service Representative

Activating synchronous reserve processing (SYNCHRES)

The SYNCHRES{(YES) | (NO)} parameter indicates whether the system defined in the GRSCNFxx MATCHSYS parameter is to have synchronous reserve processing activated. See "Understanding the synchronous RESERVE feature" on page 9 for more information.

Collecting global resource serialization information (GRSQ)

The GRSQ keyword is relevant only for star mode. It is ignored for ring and none mode. When the SDATA=GRSQ option is specified, the GRSQ keyword collects global resource serialization information during SYSMDUMP and SVC dump processing. ENQ resource information from all systems in the global resource serialization complex is gathered and placed into the dump. This can be time consuming when running in star mode because global resource serialization needs to collect and merge information from all systems in the sysplex. The time it takes increases as more systems are added to the sysplex or the number of concurrent ENQs increase, or both. The GRSQ keyword allows the installation to limit the data collected and therefore lessen the time needed to complete the dump.

- GRSQ(CONTENTION) causes GRSQ processing to collect all ENQ resources on the local system, and only collect information about global resources in contention in the rest of the GRS complex. This is the default, which we recommend.

Specifying the maximum ENQ requests (ENQMAXA, ENQMAXU)

The ENQMAXA and ENQMAXU parameters in GRSCNFxx identify the system-wide maximum of concurrent ENQ requests for authorized (ENQMAXA) or unauthorized (ENQMAXU) requesters.

The ENQMAXA range is 250,000 to 99,999,999. The default MAXVALUE is 250,000.

The ENQMAXU range is 16,384 to 99,999,999. The default MAXVALUE is 16,384.

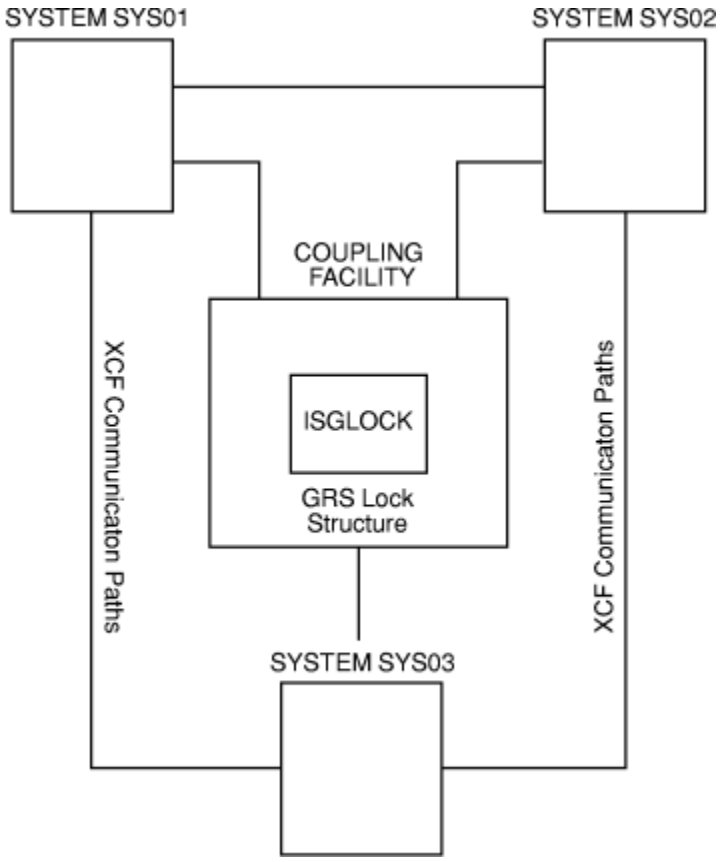
See "Limiting global resource serialization requests" on page 10 for more information.

GRSDEF considerations

Global resource serialization will ignore non-applicable parameters on the GRSDEF statement(s) when initializing systems into a star complex as well as when initializing systems into a ring complex. It is possible for an installation to create a single GRSCNFxx parmlib member that can be used for the initialization of either a star or a ring complex. This will help in the transition from a ring to a star complex if the installation elects to use the SETGRS MODE=STAR capability to make the transition. When ring-related parameters are ignored, global resource serialization will issue message ISG313I indicating the parameter and the reason.

When initializing a system into either a ring or a star complex, syntax or specification errors in the GRSCNFxx parmlib member will, in most cases, result in the initialization failing. The failure might occur whether or not the parameter in error was applicable to the particular type of complex being initialized. In these cases, an error message will always be issued and then if it is possible for system processing to continue without global resource serialization, the operator will be prompted to specify *NONE* or else re-IPL the system. If it is not possible to continue (for example, PLEXCFG=MULTISYSTEM was specified), then initialization will be terminated with an X'0A3' wait state with reason code X'0C'.

Figure 55 on page 101 shows an example configuration consisting of a three system sysplex (SYS01, SYS02, and SYS03) and a set of GRSCNFxx parameters that might be used to initialize the systems as a global resource serialization star complex. In this example, the global resource serialization lock structure (ISGLOCK) must have been previously defined to the CFRM policy.




```
GRSCNFxx parmlib member:
                              /* SYSTEM DEFINITIONS .........

   GRSDEF MATCHSYS(*)         /* GRSDEF FOR SYSTEMS SYS01 AND SYS02     */
          CTRACE(CTIGRS01)    /* PARMLIB MEMBER CTIGRS01 CONTAINS TRACE */
                              /* OPTIONS                                */

   GRSDEF MATCHSYS(SYS03)     /* GRSDEF FOR SYSTEM SYS03                */

          CTRACE(CTIGRS02)    /* PARMLIB MEMBER CTIGRS02 CONTAINS TRACE */
                              /* OPTIONS                                */
```

Figure 55. Example Global Resource Serialization Star Parmlib Definition

Bringing up a star complex

To initialize a star complex, specify GRS=STAR either in the IEASYS*xx* parmlib member, or during IPL when prompted for system parameters. Specify GRS=STAR, for each system that is part of a star complex.

If the system is the first system in the complex, it will initialize ISGLOCK on the coupling facility to start the star complex. If there is no previously-activated CFRM policy, you can specify the name of the CFRM policy to be started in the COUPLExx parmlib member 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 a previously-activated CFRM policy, the CFRMPOL specification is ignored and the system uses the name of the CFRM policy that was previously in effect.

Potential error scenarios

Because global resource serialization supports two types of complexes, a star or a ring, it is possible that due to an error in the specification of the GRS= parameter, the installation could accidentally try to IPL a system into the wrong type of complex. The possible error and the action taken by global resource serialization in each are as follows:

- If an MVS system is IPLed with one of the ring-related parameters (GRS=START, JOIN, or TRYJOIN) and a star complex already exists, the system will fail with a X'0A3' wait state with a X'84' reason code.
- If a down-level MVS system (at least an MVS/ESA SP 4.1.0 system, but not OS/390® release 2) is IPLed with one of the ring-related parameters (GRS=START, JOIN, or TRYJOIN), and a star complex already exists, the system will fail with a X'0A3' wait state with a X'84' reason code.
- If a down-level MVS system (prior to MVS/ESA SP 4.1.0) is IPLed with one of the ring-related parameters (START, JOIN, or TRYJOIN) and a star complex already exists, global resource serialization will not be able to detect this type of error. See "Error scenarios you can avoid" on page 102.
- If the installation is running a ring complex where all systems are interconnected using CTC support rather than XCF communication, and the operator or system programmer accidently specifies GRS=STAR when IPLing a system that should be part of the ring, global resource serialization will not be able to detect this type of error. See "Error scenarios you can avoid" on page 102.

Error scenarios you can avoid

In the preceding topic, the result of the last two scenarios is the creation of two separate global resource serialization complexes that will not correctly serialize resources across the two sets of systems. This problem will not occur if you correctly migrate the complex from a mixed-ring complex to a star complex.

- If your goal is to run in a global resource serialization star complex, make sure that the CTC definitions are removed from the GRSCNFxx member used to define the star complex, or
- If you desire to remain in a ring complex, do **not** add the global resource serialization record to the sysplex couple data set.

Initial processing environment

In addition to the IEASYSxx GRS= parameter indicating how GRS should initialize the system, the PLEXCFG= parameter allows the installation to indicate the processing environment into which the system is being IPLed. The GRS= system parameter is checked against this parameter to ensure that the system is being initialized properly for the particular environment the installation wants to establish. The relationship of the GRS=STAR parameter to the options that can be specified on the PLEXCFG= parameter is described in Table 7 on page 102. The PLEXCFG=options and their meanings can be found in *z/OS MVS Initialization and Tuning Reference*.

Table 7. GRS=STAR Relationship to PLEXCFG= Options

PLEXCFG=	Initialization Action Taken by MVS
XCFLOCAL	Prompt for NONE
MONOPLEX	Prompt for NONE
ANY	If XCF is in local mode, then prompt for NONE. If XCF is in sysplex mode, then initialize the system to run as a member of a global resource serialization star complex.
MULTISYSTEM	Initialize the system to run as a member of a global resource serialization star complex.

ISGLOCK rebuild processing

Global resource serialization will attempt rebuild processing in response to a policy decision made by CFRM or SFM, or in response to an operator-issued SETXCF START,REBUILD command. As global resource serialization on each system is notified of the rebuild or the structure or connectivity failure, message ISG323A is issued to indicate global resource serialization requesters will be suspended due to a rebuild in progress. Each system will then proceed to perform its part of the rebuild process.

For a structure or connectivity error, the first system in the star complex to be notified will respond by automatically initiating a rebuild of the lock structure, ISGLOCK. Depending on the active SFM policy, MVS determines whether a rebuild is required:

- If so, global resource serialization will continue with the rebuild process.

  Any systems without connectivity to the original lock structure is terminated with a X'0A3' wait state and a reason code X'D0'.
- If not, the rebuild will be stopped, message ISG236I will be issued and global resource serialization will attempt to return to and use the original lock structure.

Any systems without connectivity to the original lock structure will be terminated with a X'0A3' wait state and a reason code X'C8' or X'D0'.

If the rebuild completes successfully, message ISG325I will be issued to indicate the rebuild is complete and normal global resource serialization processing has been resumed.

Coupling facility structure failure

If the global resource serialization lock structure, ISGLOCK, fails at any time during global resource serialization initialization, the system waits for the other systems to do the rebuild and then continues with the IPL.

Following initialization, if a global resource serialization coupling facility structure failure occurs, global resource serialization will initiate a rebuild of the structure. For a discussion of global resource serialization rebuild processing, see "ISGLOCK rebuild processing" on page 102.

For a detailed recovery procedure and message examples, see System z Parallel Sysplex Recovery (publibfp.dhe.ibm.com/epubs/pdf/e0s1p702.pdf).

Loss of connectivity to a coupling facility structure

If at any time during global resource serialization initialization, connectivity to the global resource serialization coupling facility structure is lost, the system will finish initialization and then either participate in the rebuild or enter a wait state (SFM policy decision).

When loss of connectivity occurs early in global resource serialization initialization, a re-IPL of the system is the appropriate action to take.

Following initialization, if a system loses connectivity to the lock structure, ISGLOCK, global resource serialization initiates a rebuild of the structure.

Specifying global resource serialization tracing options (CTRACE)

Use the CTRACE parameter in GRSCNFxx to specify the component trace options you want for your installation. For more information about global resource serialization component trace, see *z/OS MVS Diagnosis: Tools and Service Aids*. For information about how to specify the options you want, see *z/OS MVS Initialization and Tuning Reference*.

The CTRACE parameter in GRSCNFxx allows you to modify the default tracing options used by global resource serialization. Because the defaults provide adequate serviceability information, you should change them only upon the recommendation of your IBM service representative.

Trace options are provided to select which events should be traced. The options which pertain to the star processing environment are CONTROL, REQUEST, MONITOR, SIGNAL, and FLOW. Each can be made more specific by specifying CONTROL*n*, REQUEST*n*, MONITOR*n*, and SIGNAL*n*, or FLOW*n* to provide greater flexibility and granularity in selecting which events to be traced. When viewing the trace data using the IPCS CTRACE subcommand, filtering options are provided to limit the scope of the data displayed, and subheaders are shown which briefly describe each trace key.

Minimum options (MINOPS)

Minimum options (MINOPS) are defined for GRS=STAR so that exceptional events are traced at all times, regardless of the options specified. CTRACE permits the trace state to be ON (with both the specified and minimum options in effect) or MIN (only the minimum options in effect). The trace will never be OFF. When viewing the trace data using the IPCS CTRACE subcommand, filtering options are available to limit the scope of the data displayed.

Chapter 6. Operating the star complex

Because the systems in a star complex must match the systems in the sysplex, building and operating the complex is straightforward. If you implement your design properly there is little need for operator intervention. However, the procedures for operating the systems in the complex are important.

As you read this topic and you develop your operator procedures, consult the following books:

- For sysplex operations involving the coupling facility, see *z/OS MVS Setting Up a Sysplex*.
- For exact text and responses to global resource serialization messages, see *z/OS MVS System Messages, Vol 9 (IGF-IWM)*, which contains a complete description of how to use the DISPLAY command with global resource serialization.
- For recovery procedures, see Chapter 11, "Diagnosing global resource serialization," on page 175.

Operating the star complex

In general, your operational planning focuses on three areas: building the complex, normal operations, and recovery operations. The design of your complex affects all three areas.

As previously discussed, systems in a star sysplex are connected to a coupling facility through communication links, and to each other through XCF managed CTCs, or XES signaling paths, or both.

Building the complex

The process of building a global resource serialization complex can have two phases: a configuration check and the IPL of the systems.

Configuration check

Before you IPL your system to start or join a global resource serialization star complex, your operator should verify that:

1. The lock structure, ISGLOCK, is specified in the coupling facility resource management (CFRM) policy.
2. All coupling facilities that can contain the lock structure are specified in the CFRM preference list.
3. All connections to shared resources (such as DASD) are correct.
4. The sysplex couple data set has been previously formatted with the global resource serialization parameter so that the global resource serialization record which contains the RNLs is available.

If the shared resource connections are incorrect, a serious data integrity exposure could occur. This exposure occurs when systems in the complex are serializing access to a global resource by means of an ENQ macro with a scope of SYSTEMS. For example, if the RESERVE conversion RNL for the complex contains an entry for a resource, this entry causes global resource serialization on each system in the complex to suppress the reserve for that resource. If a system outside the star complex can use a reserve to access the same resource at the same time, the resource could be damaged.

IPL

The GRS= system parameter indicates to MVS at IPL time that a system is to be part of a global resource serialization star complex. The GRS and GRSCNF parameters remain in effect for the duration of the IPL; the only way to change their value is to IPL the system again. You can change the RNLs without having to reIPL the entire complex. Use the SET GRSRNL command to accomplish this change. See "Changing RNLs for a star complex" on page 107 for more information.

Many values in the GRSCNF parmlib member can be changed without re-IPLing the system. See the SETGRS command in *z/OS MVS System Commands* for the complete list.

There are several ways you can specify the global resource serialization parameters. To minimize operator intervention during IPL, it is generally best to specify GRS, GRSRNL, and GRSCNF in the IEASYSxx parmlib member or accept the default values. See "Defining parmlib members for a star complex" on page 99.

You can specify GRS=STAR in IEASYSxx on each system in the sysplex. The first system in the star sysplex to initialize will initialize the global resource serialization lock structures on a coupling facility to start the star complex. See *z/OS MVS Setting Up a Sysplex*.

Normal operations in a sysplex

Once the complex is built, it requires little, if any, operator intervention. If a problem occurs, either global resource serialization or some other system component will detect the problem and issue messages that describe it before the operator could notice it.

For example, some of the error messages that global resource serialization issues indicate damage to resources or to the resource control blocks. These messages are ISG031E, ISG032E, ISG033E, ISG034E and ISG035E. The problem that causes any of these messages can also cause the job requesting the resource to terminate abnormally. If the damage is extensive, the problem can cause multiple jobs to terminate abnormally, requiring you to reIPL to restore the control blocks.

During normal processing, operators can use system commands to monitor and control global resource serialization. The system commands related to global resource serialization are:

- DISPLAY GRS, which displays the status of each system in the complex. For more information see "Displaying the status of systems in a star complex" on page 106.
- SET GRSRNL, which changes the RNLs dynamically. For more information see "Changing RNLs for a star complex" on page 107.
- SETGRS, which is used to migrate systems in an active ring sysplex to a star sysplex. You can alter the values specified in GRSCNFxx parmlib member by using the SETGRS command. For more information about using the SETGRS command, see "Using the SETGRS command to convert to a star complex" on page 110.
- VARY XCF,sysname,OFFLINE, which removes a system from the complex (and also the global resource serialization star complex). For more information, see *z/OS MVS Setting Up a Sysplex*.

Note: The VARY GRS command is not supported in a star complex. If the command is entered, it will be rejected with message ISG153I.

Displaying the status of systems in a star complex

The DISPLAY GRS (D GRS) command shows the state of each system in the complex. Note that D GRS shows system status only as it relates to the global resource serialization star. D GRS does not reflect how well a system is running.

You can also use D GRS to display the local and global resources requested by the systems in the star complex, the contents of the RNLs, and jobs that are delayed or suspended by a SET GRSRNL command. Issuing a D GRS command without any other parameters shows the same display as issuing a D GRS,SYSTEM command. These uses are described in *z/OS MVS System Commands*.

You can issue D GRS from any system in the star complex, and at any time after the star complex has been started. The D GRS display shows the status of the star from that system's point of view; thus, the displays issued from different systems might show different results. Figure 56 on page 107 shows an example of the information D GRS produces and explains the values that can appear in each field.

```
20.30.03  ISG343I  20:30:02  GRS STATUS 540
SYSTEM    STATE               SYSTEM    STATE
 SYS2      CONNECTED           SYS1      CONNECTED
 SYS3      CONNECTING          SYS4      CONNECTING
```

SYSTEM
The name of the system.

STATE
The state of the system at the time when the command was issued.

CONNECTING
The system is processing the GRS=STAR parameter. It is not yet a member of the star complex.

CONNECTED
The system is part of the star complex.

REBUILDING
The system is not part of a star complex, but is rebuilding the ISGLOCK lock structure. All tasks that are trying to obtain global resources are suspended by the system.

Figure 56. D GRS Star Explanation

In addition to the states defined for the star complex, the number of locks defined to the ISGLOCK lock structure is displayed. See *z/OS MVS System Messages, Vol 9 (IGF-IWM)* for a description of message ISG343I, which details the output of the DISPLAY GRS command.

Changing RNLs for a star complex

You can dynamically change the RNLs that global resource serialization uses in a star configuration, because the sysplex always matches the complex.

To change the RNLs currently being used by global resource serialization, set up the GRSRNLxx parmlib members with the new RNLs. Next, issue the SET GRSRNL command on a system that has access to those members. The new RNLs are then communicated to all systems in the complex. Keep in mind that if the complex is operating under a GRSRNL=EXCLUDE command, you cannot issue either the SET GRSRNL=EXCLUDE command or the SET GRSRNL= command (with one exception; see "Migrating from GRSRNL=EXCLUDE to a set of RNLs" on page 32 for more information). Note that even though only one system needs the updated parmlib members to start the change, you must copy the updated GRSRNLxx parmlib members to each system's parmlib. Any system that needs to can then reIPL into the same complex. For example, if you issue SET GRSRNL=(BN,K1) to change RNLs, and you use IEASYSK1 to IPL your systems, IEASYSK1 should contain GRSRNL=(BN,K1). Otherwise, the change will be in effect only for the duration of the current IPL.

Global resource serialization ensures that the integrity of all resources is maintained throughout the RNL change. In particular, before an RNL change can complete, special processing may be performed if any jobs are using the resources that are different in the old and new RNLs. These resources are known as *affected* resources. Jobs issuing new requests for these resources are suspended until the RNL change is complete. These are known as *suspended jobs*. The following message is issued on each system in the complex that has suspended one or more jobs:

```
ISG210E RNL CHANGE WAS INITIATED BY SYSTEM sysname
        SOME JOBS ARE BEING SUSPENDED UNTIL RNL CHANGE COMPLETES.
```

If any job currently holds one or more of the affected resources, the change is delayed until all of the affected resources are freed. Jobs holding an affected resource (and thereby delaying the RNL change) are *delaying jobs*. When jobs are holding affected resources and delaying the change, the following messages are issued on whichever console originated the RNL change:

```
ISG219E RNL CHANGE WAITING FOR RESOURCES TO BE FREED.
        TO LIST DELAYING JOBS, USE ROUTE SYSNAME,DISPLAY GRS,DELAY.
        TO LIST SUSPENDED JOBS, USE ROUTE SYSNAME,DISPLAY GRS,SUSPEND.
```

```
ISG220D REPLY C TO CANCEL RNL CHANGE COMMAND, OR S FOR SUMMARY OF RNL
        CHANGE PROGRESS.
```

Getting a list of delaying jobs

The DISPLAY GRS,DELAY (D GRS,DELAY) operator command lists the jobs that hold affected resources and are causing the change to be delayed. The jobs listed might release the affected resources normally, or you can cancel then at your discretion. Once these jobs release the affected resources, the RNL change completes.

Getting a list of suspended jobs

The DISPLAY GRS,SUSPEND (D GRS,SUSPEND) operator command lists the jobs that are being suspended due to the RNL change. The jobs listed remain suspended until the RNL change completes, or until you cancel the RNL change.

Responding to the ISG220D message

Replying to message ISG220D with an S produces a summary of the RNL change progress. This summary indicates the number of jobs on each system that are delaying or are suspended by the RNL change. Replying to message ISG220D with a C causes the RNL change to be cancelled.

If the operator chooses not to respond to message ISG220D with a C, the change will take place when all delaying jobs release the affected resources.

Cancelling the RNL changes or jobs

There might be instances where the operator must either cancel the RNL change or cancel jobs that hold the affected resources. For example consider cancelling the RNL if a job is:

1. Not cancellable is holding affected resources for a long time.
2. Holding an affected resource and cannot DEQ that resource because it is suspended by global resource serialization pending a new ENQ for another affected resource.
3. Waiting for some other work in the system that has issued an ENQ for an affected resource and has become suspended.
4. Suspended by the RNL change is considered more important than the RNL change.

See *z/OS MVS System Commands* for more information about DISPLAY GRS and SET GRSRNL commands.

Rebuilding the ISGLOCK structure

MVS services include a **rebuild** function that allows the global resource serialization coupling facility structure, ISGLOCK, to be resized. The operator can initiate a rebuild of the structure by entering the following command:

```
SETXCF START,REBUILD,STRNAME=ISGLOCK
```

An operator's rebuild request results in the structure being recreated on either the same coupling facility or a different coupling facility, depending on the current coupling facility resource management (CFRM) policy for the structure.

In conjunction with the rebuild function, there is a **stop rebuild** function that allows the termination of an in-process rebuild. If a rebuild of a structure is initiated and the operator wants to terminate it, the operator can enter **stop rebuild**. Global resource serialization always honors a **stop rebuild** request and attempts to return to and use, the original lock structure. Any systems that no longer have access to the original structure are terminated with a X'0A3' wait state and a reason code of X'C8' or X'D0'.

```
SETXCF STOP,REBUILD,STRNAME=ISGLOCK
```

Shutting down a coupling facility

You might, for whatever reason, want to shut down a coupling facility that contains the ISGLOCK lock structure. Before you shut down the original coupling facility, make sure that another coupling facility is available for use by the sysplex. The ISGLOCK lock structure has the following properties:

- The ISGLOCK lock structure cannot be deleted.
- Failure to connect to the new structure during rebuild, depending on the sysplex failure management (SFM) policy, cause one of the following conditions:
  - Systems that cannot connect are put into a wait state.
  - Rebuild is cancelled.
  - Global resource serialization tries continuously to rebuild the structure, if the original problem is a structure failure and either the new structure fails or the operator tries to stop the rebuild.

The procedures to shut down a coupling facility containing the ISGLOCK lock structure, when there is another coupling facility available, are as follows:

1. Reduce the workload (the less the workload, the faster rebuild will complete)
2. Issue SETXCF START,REBUILD,STRNAME=ISGLOCK,LOC=OTHER
3. Wait for message ISG325I on all systems:

   ```
   ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS COMPLETED ON
   sysname.
   ```

4. Shutdown the original coupling facility

Steps in converting to a star complex

There are two phases to migration that you need to consider. They are:

1. Migrating to the new release.

   If the installation does not plan to run a star complex after the new release is installed, re-IPL the system specifying GRS=JOIN to bring the system back into the ring complex. The only systems that should remain in the ring complex are systems connected to the coupling facility.
2. Converting from a ring complex to a star complex.

Converting to a star complex

If your installation is already running a ring, do the following to migrate to a star complex:

1. Set up new GRSCNFXX and IEASYSxx parmlib members to define the parameter information that will be needed later for the star complex. See *z/OS MVS Initialization and Tuning Reference*.
2. If the complex is a mixed complex, convert any non-sysplex systems to sysplex systems, or remove non-sysplex systems from the complex. Make sure that non-sysplex systems are not using resources that are shared by systems in the sysplex.
3. Do any hardware changes that need to be done to prepare for the use of the coupling facility hardware.
4. Define the ISGLOCK lock structure for global resource serialization. The structure must be accessible to all systems in the complex.
5. Create a new sysplex couple data set that contains the global resource serialization record, and make the new data set the active sysplex couple data set.
6. At this point, depending on what is best for your installation you can:
   - Take down the entire ring complex and re-IPL each system specifying GRS=STAR, or
   - Use the SETGRS MODE=STAR command to migrate from a ring to a star complex without a complex-wide IPL.

Using the SETGRS command to convert to a star complex

The SETGRS command is used to migrate an active ring to a star complex without requiring a complex-wide IPL. You can not use the SETGRS command to migrate from a star to a ring complex. (Returning to a ring complex requires an IPL of the complex.)

```
SETGRS MODE=STAR
```

MODE=STAR
Indicates that the ring complex is to be converted to a global resource serialization star complex.

If you elect to use the SETGRS command, to dynamically switch from a ring complex to a star complex, do the following:

1. Make sure that there are no ring disruptions or dynamic RNL changes in progress.
2. Notify the interactive user community that there might be a temporary delay in their activity for several minutes while the migration takes place.
3. Issue the SETGRS MODE=STAR command on any system in the complex.

It will take a few minutes for the migration to complete. As each system migrates to the star the following messages are issued:

-
```
ISG331I SYSTEM sysname INITIATED SYSPLEX-WIDE MIGRATION TO
GRS STAR MODE. THIS SYSTEM IS PARTICIPATING IN THE MIGRATION.
```

- `ISG337I GRS LOCK STRUCTURE (ISGLOCK) CONTAINS lockentries LOCKS.`
- `ISG300I GRS STAR COMPLEX INITIALIZATION COMPLETE.`

If the operator issues the D GRS command during migration, a variety of statuses are displayed. First, the status for all system in the ring changes to migrating. The systems will then disappear from the list of systems in the message display, and the display changes to the star format. When the systems reappear, their status displays as connecting. There is no cause for concern if some systems are missing from the message display at this point in the migration. Once migration is complete, all systems should be displayed as connected.

Conditions during dynamic conversion

While the system is processing a SETGRS MODE=STAR command, GQSCAN does not work (fails with return code X'0C' with reason code X'10'), and the following global resource serialization requests are suspended:

- ENQ
- DEQ
- RESERVE

ISGQUERY does not work during GRS=STAR migration (RC=X'0C' Rsn='xxxx0C03'X).

The length of time global resource serialization requesters are suspended might be a few minutes while the ISGLOCK lock structure and global resource serialization sysplex couple data set records are initialized with all of the complex-wide information, and significant changes are made to the internal control block structures. IBM recommends that you migrate at a time when the amount of global resource request activity is likely to be minimal.

- During the migration to a star complex, ISG377E and ISG378E might be issued, indicating a global resource serialization ring disruption is in progress, and might remain outstanding for a long time. This is normal. They will be removed before the ring complex has completed migration to a star complex.
- Because to the restructuring of global resource serialization control structures, some ABEND records might appear in LOGREC during a SETGRS MODE=STAR migration. However, no dumps will be taken for this class of expected abends.

Part 3. Global Resource Serialization Ring

Chapter 7. Ring processing

As stated earlier, a global resource serialization ring complex consists of one or more systems connected by communication links. Global resource serialization uses the links to pass information about requests for global resources from one system in the complex to another.

Regardless of the physical configuration of systems and links, the global resource serialization complex consists of every system that indicates at IPL time that it is to be part of the complex. For various reasons, such as a system or link failure, not all of the systems in the complex might be actively using global resource serialization at any particular time. Those systems that are actively using global resource serialization to serialize access to global resources make up the **global resource serialization ring**.

Figure 57 on page 115 shows a four-system global resource serialization ring complex. When all four systems in the complex are actively using global resource serialization, the complex and the ring are the same.

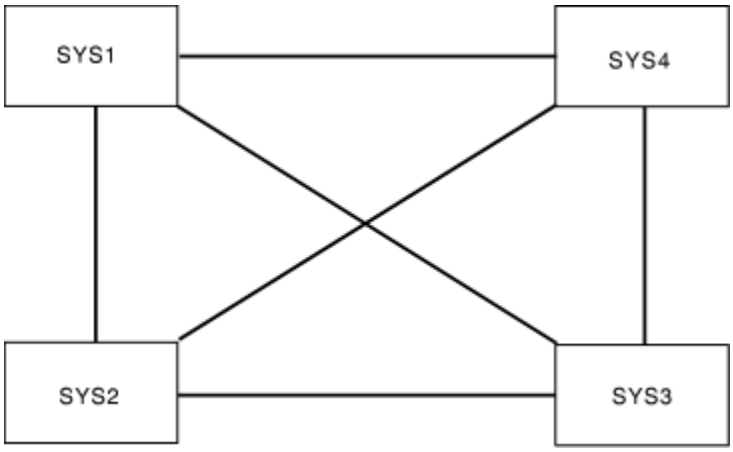


Figure 57. Fully-Connected Four-System Complex

The complex shown in Figure 57 on page 115 has a communication link between each system and every other system; such a complex is a **fully-connected complex**. A sysplex requires full connectivity between systems. Therefore, when the sysplex and the complex are the same, the complex has full connectivity. Although a mixed complex might not be fully connected, a fully-connected complex allows the systems to build the ring in any order and allows any system to withdraw from the ring without affecting the other systems. It also offers more options for recovery if a failure disrupts ring processing.

For example, if system SYS1 in Figure 57 on page 115 were to fail and end its active participation in serializing access to global resources, it would still be part of the complex, but it would not be part of the ring. Figure 58 on page 116 shows the ring that would continue processing after system SYS1 stopped serializing access to global resources.

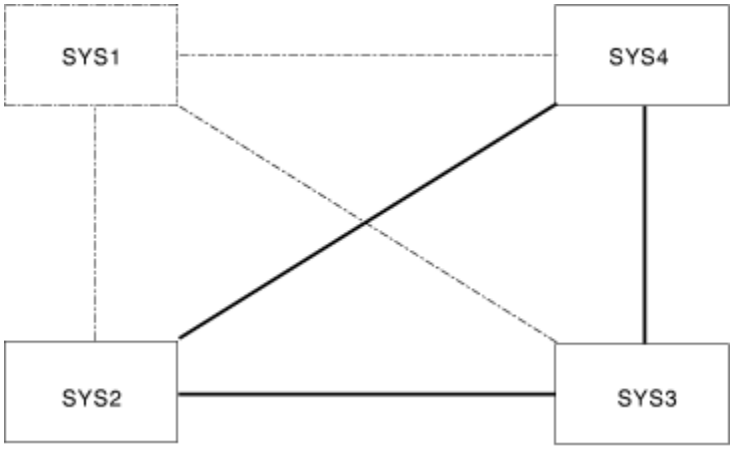


Figure 58. Three-System Ring — SYS1 Failed

Chapter 8, “Designing a ring complex,” on page 119 describes how to design the complex in detail. Chapter 9, “Operating the ring complex,” on page 147 describes how to plan operational procedures for the complex you design.

The concept of the global resource serialization ring is important because, regardless of the physical configuration of systems and links that make up the complex, global resource serialization uses a ring processing protocol to communicate information from one system to another. Once the ring is active, the primary means of communication is the ring system authority message (RSA-message).

The RSA-message

The RSA-message contains information about requests for global resources (as well as control information). It passes from one system in the ring to another. No system can grant a request for a global resource until other systems in the ring know about the request; your installation, however, can control how many systems must know about the request before a system can grant access to a resource. (See “Processing a request for a resource” on page 116.) The RSA-message contains the information each system needs to protect the integrity of resources; different systems cannot grant exclusive access to the same resource to different requesters at the same time.

When a system receives a request for a global resource, the system suspends the requester and, when the RSA message arrives, places the request in the RSA-message. Systems in the global resource serialization ring batch their requests for global resources. For example, a system might receive seven requests for global resources while waiting for the incoming RSA-message — the message it receives from the preceding system. It adds all seven requests to the outgoing RSA-message — the message that it sends on to the next system in the ring. Batching requests for resources minimizes the communication overhead for global resource serialization.

The order and direction of the RSA-message can change when systems enter or leave the ring. The amount of time that each system holds the RSA-message is called the **residency time**. Your installation sets the general limits for residency time. While it holds the RSA-message, each system processes the requests in the incoming RSA-message and adds its new requests to the outgoing RSA-message.

Processing a request for a resource

Global resource serialization processes requests for local resources and requests for global resources differently. When a task running on a system in the ring requests a local resource, that system handles the request on its own. Global resource serialization does not place a local resource request in the RSA-message.

When a task running on a system in the ring requests a global resource, however, the processing is very different because each system in the ring must keep track of all requests for global resources. How the ring processes a request for a global resource depends on whether or not your installation uses ring acceleration.

Request processing without ring acceleration

Figure 59 on page 117 summarizes the processing of a global resource request without ring acceleration. The steps include:

1. The originating system suspends the requesting task.
2. When the incoming RSA-message arrives, the system places the request in the outgoing RSA-message, then passes the RSA-message on to the next system in the ring.
3. The request in the RSA-message makes a cycle around the ring. Each system in the ring records the request.
4. When the RSA-message completes its cycle, the originating system recognizes that all systems know about the request. The originating system then actually processes the request; it grants access to the resource according to normal ENQ processing. That is, if the resource is available, the system grants the suspended task access to the resource and marks the task as ready to execute. If the resource is not available, the task continues to wait until it becomes available.

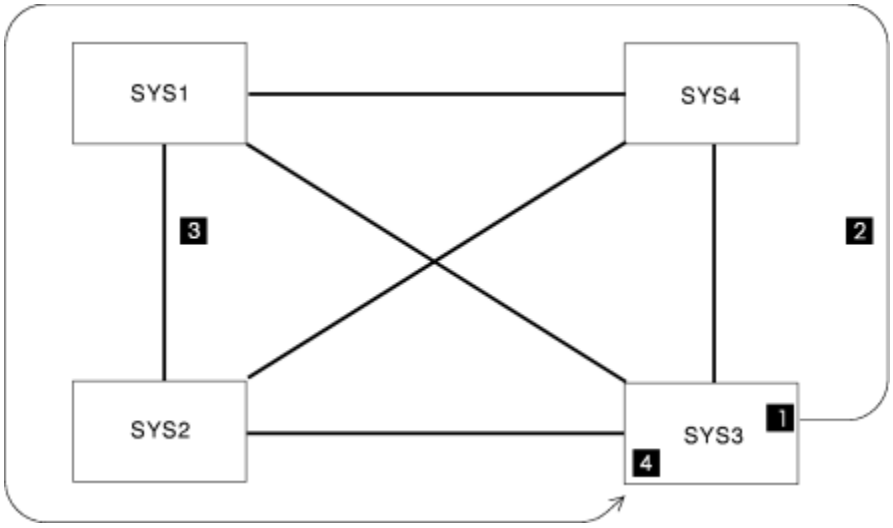


Figure 59. Global Resource Request Processing without Ring Acceleration

This processing ensures that each system in the ring has the same information about global resources at any particular time. The fact that each system has the same information ensures recovery from a failure that disrupts ring processing. Global resource serialization can maintain data integrity. Each system has accurate information about the state of all global resources granted at the time it last sent the RSA-message on to the next system in the ring. Thus, each system has the information it needs to:

1. Prevent different systems from allocating the same resource to different requesters
2. Allow the ring to continue processing without data integrity exposures even after multiple communication link or system failures

This processing, however, means that every task that requests access to a global resource is suspended for at least the time it requires for the RSA-message to complete its cycle around the ring. For installations that cannot tolerate this delay, global resource serialization provides an alternative ring processing technique called ring acceleration.

Request processing with ring acceleration

Ring acceleration is available only when all systems in the ring are running MVS/SP Version 3 or higher. Ring acceleration also requires alternate links, except between systems in a sysplex, and IBM recommends that the complex be a fully-connected complex. An installation where the complex is the

same as the sysplex does not need to perform any additional setup to use ring acceleration; multisystem sysplexes must be fully connected. To achieve full connectivity in a mixed complex, however, see Chapter 8, “Designing a ring complex,” on page 119 for details about how to define the necessary links.

Ring acceleration can significantly reduce the amount of time tasks spend waiting for global resources, especially in a large complex. It can, however, affect recovery. “Ring acceleration (ACCELSYS)” on page 134 describes the factors involved in determining how your installation should use ring acceleration.

Using ring acceleration changes the processing of a global resource request. Figure 60 on page 118 summarizes the processing of a global resource request with ring acceleration. The steps include:

1
The originating system suspends the requesting task.

2
When the incoming RSA-message arrives, the system places the request in the outgoing RSA-message, then passes the RSA-message on to the next system in the ring.

3
Your installation chooses the number of systems that must see the RSA-message before a system sends a “shoulder-tap”, an acknowledgement that it has received the RSA-message, to the originating system. Assuming that the number of systems is two, the next system in the ring sends the shoulder-tap to the originating system.

4
When the originating system receives the shoulder-tap, the ring acceleration signal, it grants access to the resource according to normal ENQ processing. That is, if the resource is available, the system grants the suspended task access to the resource and marks the task as ready to execute. If the resource is not available, the task continues to wait until it becomes available.

5
The RSA-message continues on its cycle around the ring so that each system in the ring knows about the request. The task that requested the resource, however, does not need to wait for the cycle to complete before obtaining access to the resource.

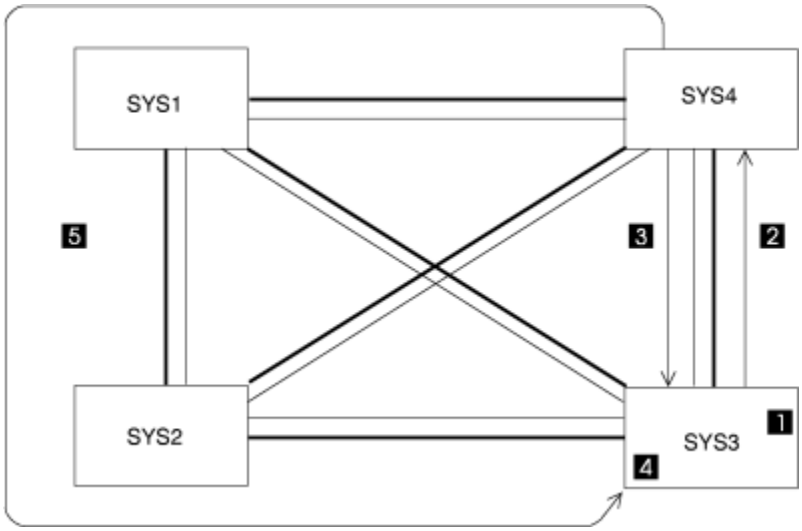


Figure 60. Global Resource Request Processing with Ring Acceleration

Chapter 8. Designing a ring complex

In a sense, the decisions you make about how you want to process requests for various resources are the decisions that set your installation's goals for global resource serialization. The actual global resource serialization complex that you design is one of the tools you use to achieve these goals.

As stated earlier, a global resource serialization complex consists of all the systems that are able to share global resources. There are links between the systems that enable them to communicate, primarily by passing the RSA-message from one system to another. Each system in the complex indicates at IPL time that it is to be part of the complex.

Designing the complex involves answering these basic questions:

- What resources does your installation want to share?
- Which systems use these resources?

The resources your systems need to share determine the systems in the complex. The most likely candidates, of course, are those systems that are already serializing access to resources on shared DASD volumes and, especially, those systems where interlocks, contention, or data protection by job scheduling are causing significant problems. In addition, systems operating in a multisystem sysplex must also be in the same global resource serialization complex.

Note: "Systems" refers to the number of MVS operating systems, not the number of processors. For example:

- In a PR/SM environment, many systems can run in the same processor complex if they are in Logical Partitions (LPARs). The number of systems allowed varies by processor. See *PR/SM Planning Guide* to determine the number supported for your processor.
- Under VM/ESA multiple MVS systems can run as second-level systems in the same processor complex.

It is possible for a single installation to have two or more global resource serialization complexes, each operating independently. However, the independent complexes should not share resources. Also, there should be no common links made available to global resource serialization on any two complexes.

To avoid a data integrity exposure, ensure that no system outside the complex can access the same shared DASD as any system in the complex. If that is unavoidable, however, you must serialize the data on the shared DASD with the RESERVE macro. You need to decide how many of these systems to combine in one complex. The design of global resource serialization can support up to 32 systems. The actual number of systems your particular installation can reasonably configure into a complex depends on a number of factors. You should, for example, consider the operations and performance implications of a large ring complex. See *z/OS MVS Setting Up a Sysplex* for more information.

Note:

1. Global resource serialization ring complexes can contain XCFLOCAL, MONOPLEX, or systems that are part of a multisystem sysplex. However, only one multisystem sysplex can exist in any global resource serialization complex.
2. Sysplex failure management (SFM) policy, which is optional, defines how MVS is to manage sysplex resources. GRS exploits both XCF and XES critical member support and indicates a TERMLEVEL of SYSTEM. For more information about defining SFM policy, see *z/OS MVS Setting Up a Sysplex*. IBM suggests using the SFM policy.

The way you approach the task of designing a global resource serialization complex depends on your system environment. If you are operating a global resource serialization complex where all the systems are part of the same sysplex, see "Designing a complex that matches a sysplex" on page 120. If you are operating a global resource serialization complex where not all of the systems in the complex are in the multisystem sysplex (a mixed complex), you have many explicit decisions to make. Section "Designing a mixed complex (sysplex does not match complex)" on page 125 explains the setup procedures for other

global resource serialization complex configurations. This section also describes concerns that might arise during migration from an existing global resource serialization complex to a sysplex.

Designing a complex that matches a sysplex

Global resource serialization uses XCF signalling paths to communicate between systems in a sysplex. You do not need to define links for global resource serialization. See Figure 61 on page 120.

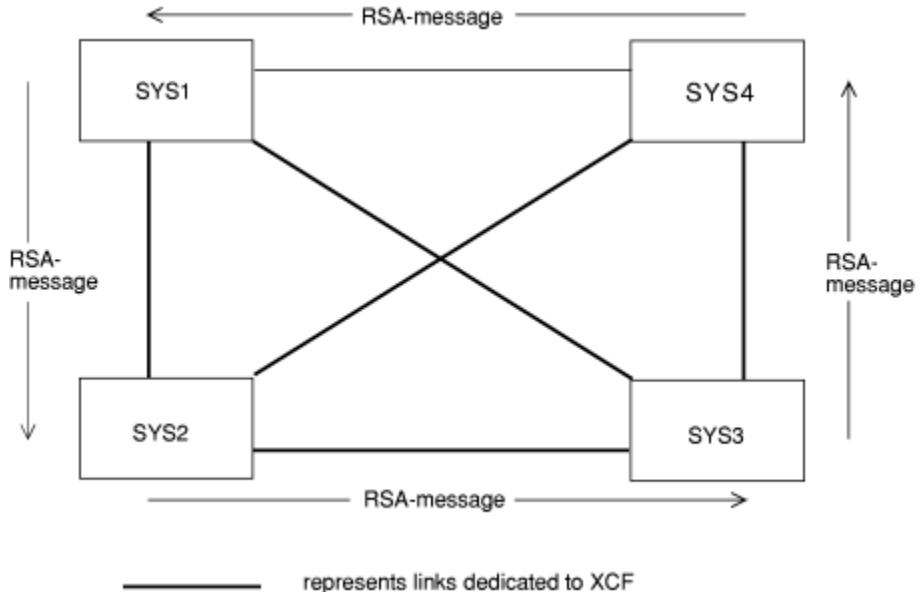


Figure 61. Links in a Sysplex

You must, however, define the global resource serialization processing options that your complex needs. For example, you must decide:

- The length of the residency time value, that is, the minimum length of time a system holds the RSA-message before sending it on to the next system in the ring
- The length of the tolerance interval, that is, the maximum length of time that global resource serialization is to wait for the incoming RSA-message before it signals a disruption
- Whether ring acceleration is to be used, and if so, the number of systems in the ring that need to know about global ENQ requests before they can be granted.

For more information, see *z/OS MVS Setting Up a Sysplex*.

Use the information in "Processing options in a sysplex" on page 120 to make these decisions.

Processing options in a sysplex

Processing options provide the information that global resource serialization needs about the systems in the complex. Some of this information comes from the system parameters specified at IPL time. These parameters are:

- SYSNAME=xxxxxxxx — identifies the system name. Each system must have a unique system name, which can be up to 8 characters. For simplicity, use the four-character SMF SID.
- GRS=TRYJOIN — indicates that a system is to become part of the global resource serialization complex by joining an already existing complex, or by starting the complex if this is the first system to IPL into the complex. The operator does not need to be aware of the status of the other systems in the complex and generally is not prompted for START or JOIN at IPL. See "IPL" on page 151 for restrictions.
- GRSRNL=xx — identifies the GRSRNLxx parmlib member that holds the RNLs the system is to use. GRSRNL=EXCLUDE, however, specifies that no resource requests are to be treated as global ENQs, other than those ENQ macros coded with SCOPE=SYSTEMS,RNL=NO.

Chapter 2, “Selecting the data,” on page 17 described how to use GRSRNLxx to define the RNLs. *z/OS MVS Initialization and Tuning Reference* contains complete information about how to code the GRSRNL system parameter.

- GRSCNF=xx identifies the GRSCNFxx parmlib member that contributes toward the definition of the complex. GRSCNF00 is the default member and contains ring-specific keywords, so another GRSCNFxx member should be defined by the installation for ring. The GRSDEF syntax for keywords specific to ring mode are shown in Figure 62 on page 121.

```
GRSDEF  [MATCHSYS{(name)}]
                 [{(*)}   ]
        [RESMIL(1-8 decimal digit)]
        [TOLINT(1-8 decimal digit)]
        [ACCELSYS(1-2 decimal digit)]
        [SYNCHRES     {(YES)}]
        [             {(NO)}]
        [ENQMAXA(1-8 decimal digit)]
        [ENQMAXU(1-8 decimal digit)]
```

Figure 62. Ring-relevant parameters for GRSCNFxx

Because you are initializing a multisystem sysplex that matches the complex, you can use GRSCNF00 with the default values to define all the systems in your complex. Whether you use GRSCNF00 or supply your own parmlib member, you can use the same member on all systems in the sysplex.

Planning Aids

Table 8 on page 125 provides a worksheet you can use for defining a complex that matches the sysplex.

Reference

Once you have completed your plan, see *z/OS MVS Initialization and Tuning Reference* for information about how to specify the global resource serialization parmlib member and system parameters.

Residency time value (RESMIL)

Use the RESMIL option in GRSCNFxx to specify the residency time value. The residency time value is the minimum length of time in milliseconds the RSA-message resides at each system in the complex in addition to the processing time. The actual amount of time an RSA-message resides at a system varies.

The system holds the RSA-message for a variable length of time — based on the minimum length of time specified by RESMIL and a maximum value calculated by global resource serialization. The amount of time is that which the system calculates to be a good balance between quick response time and effective processor utilization. The residency time calculated by the system will never be less than the specified RESMIL value.

You can also set the RESMIL option to OFF to specify that no residency time is required. In a ring made up of small systems, the amount of time used by I/O and other processing generally compensates for any additional residency time that RESMIL would have provided. By specifying OFF, you indicate that the minimum residency time is zero and that the system is not to do any tuning.

Choosing a RESMIL value

The following topics discuss considerations when choosing a RESMIL value:

- “Residency time value (RESMIL)” on page 121 discusses how the RESMIL value can affect the performance of your complex. The default value for RESMIL is 10 milliseconds. However, depending on your processor and workload configuration, this value might be too high for your installation.
- “Tuning the complex” on page 167 discusses the importance of achieving an acceptable response time and how the RESMIL value can affect that response time. Charts are provided that show examples of

what RESMIL value to specify to achieve the desired response time (Table 12 on page 170 and Table 13 on page 170).

Tolerance interval (TOLINT)

Use the TOLINT option in GRSCNFxx to specify the tolerance interval. The tolerance interval is the length of time that global resource serialization is to wait for an overdue RSA-message before it signals a disruption. Determining an acceptable time-out value warrants the following considerations in a global resource serialization complex:

- The excessive spin time and recovery actions for each system
- The number of systems
- The speed of the systems
- Inter-system signalling configuration, and activity
- Paging of the global resource serialization common area storage for each system
- The RESMIL time for each system

When the sysplex matches the complex, IBM guidelines suggest that TOLINT be set to the default value of 180 seconds. Depending on your installation, this value can be raised or lowered. Keeping it set to a relatively high time-out value (180 seconds) prevents premature global resource serialization disruptions in cases where there are unexpected, but recoverable, delays involving the ring.

If a system fails and is partitioned out of the sysplex, global resource serialization is notified of this condition and commences a ring rebuild operation without waiting for the TOLINT value to expire.

Typically, the RSA-message should proceed quickly around the ring. A system that fails or is stopped temporarily, or a link that fails or temporarily slows down communication, can cause a significant delay of the RSA-message, such as,

- An MVS image recovering from a spin loop
- An MVS image is taking an SVC dump
- Delays in inter-system communications
- Shortages in real storage
- Auxiliary storage page-in delays

During a ring disruption, all tasks that request or free global resources are suspended because the RSA-message is halted and there is no communication between systems in the ring. As the ring disruption continues, more and more tasks are suspended, slowing the throughput of each system in the ring. The ring is then automatically rebuilt without the failed system or link. Once the ring has been rebuilt, global resource serialization will resume processing resource requests.

The value you set for TOLINT affects how rapidly global resource serialization detects an overdue RSA-message, and setting the value properly requires a basic trade-off:

- To detect a system failure or a link failure, the best TOLINT value is one that recognizes the condition almost immediately.
- To deal with a temporary delay, the best TOLINT value is one that does not detect the condition. There are many reasons for a system entering a temporary stop, such as a spin loop or taking an SDUMP to capture the contents of common storage. For a temporarily stopped system or a temporary link delay, the best TOLINT value is one that is large enough to allow normal RSA-message processing to resume without causing a ring disruption.

Thus, the best TOLINT value is one that allows global resource serialization to detect a system or link failure promptly but does not cause it to continuously detect temporary delays. If you specify RESTART(YES) and REJOIN(YES), setting a low TOLINT value has minimal effect because ring recovery is automatic. If your installation chooses not to use automatic restart and rejoin, set a higher value to avoid unnecessary ring disruptions that require operator intervention. The default value for TOLINT is three minutes. Depending on your installation, this value can be lowered. In other complexes, or when MVS is

running in a PR/SM environment, a good value is between 40 and 60 seconds. If MVS is running as a guest under VM, set the set TOLINT value even higher.

Note: Systems in a multisystem sysplex will automatically rebuild a disrupted global resource serialization ring. A disabled system in a multisystem sysplex will automatically rejoin a disrupted global resource serialization ring once the disabled system has recovered.

Ring acceleration (ACCELSYS)

Use the ACCELSYS option in GRSCNFxx to specify whether or not the complex is to use ring acceleration and, if so, how many systems must see the RSA-message before a system sends the shoulder-tap acknowledgment.

Without ring acceleration, every system in the ring must see each request for a global resource before it can be granted. While the RSA-message makes a complete cycle around the ring, the task that requested the global resource is suspended to guarantee the integrity of resources; no global resource request is granted until all systems know about it. It does, however, mean that every task that requests a global resource must wait for at least one RSA-message cycle.

Ring acceleration offers an alternative technique, which protects the integrity of resources while potentially providing a significant reduction in global resource request response time (the time a task is suspended while waiting for ring processing).

Request processing

Using Figure 63 on page 123, assume that a task on SYS1 requested access to a resource and that the RSA-message is moving as shown. Without ring acceleration, the task on SYS1 would wait until the RSA-message made a complete cycle around the ring. Only when SYS2, SYS3, and SYS4 all know about the request can SYS1 grant the resource to the requester.

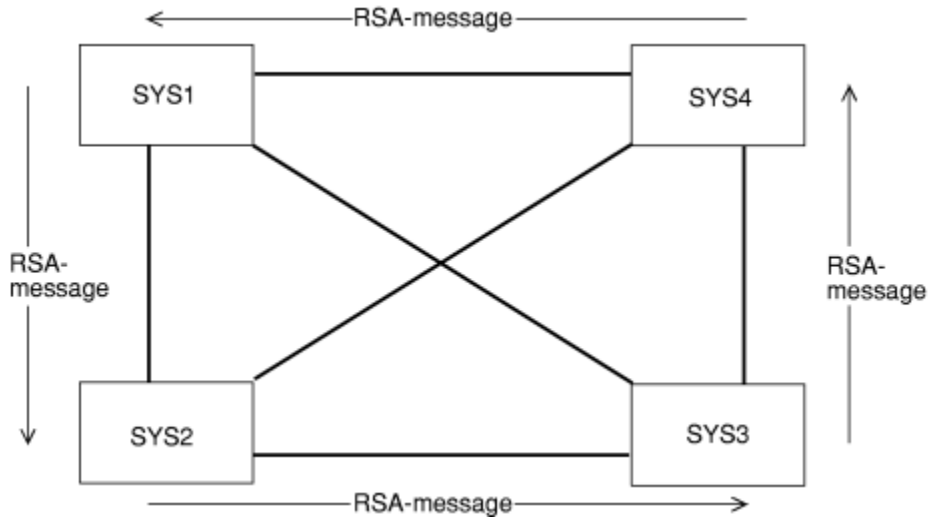


Figure 63. Ring Acceleration Configuration

In contrast, assume that GRSCNFxx contains ACCELSYS(2) to request ring acceleration. With ring acceleration, SYS1 still suspends the task that requested the resource, puts the request in the RSA-message, and sends the RSA-message on to the next system in the ring — SYS2 in this example.

SYS2, when it receives the RSA-message, uses the alternate link to send a shoulder-tap acknowledgment, the ring acceleration signal, to SYS1. SYS2 sends the signal because it is the second system to see the request, and ACCELSYS(2) means that two systems must see the request before it can be granted. After sending the shoulder-tap, SYS2 then processes the RSA-message. The RSA-message continues its cycle around the ring. All systems see and process the request, which preserves the integrity of the resource.

SYS1, as soon as it receives the ring acceleration signal, can grant the request. The task that requested the resource does not have to wait for the RSA-message to make a complete cycle around the ring. If the requested resource is available, the task can resume execution almost immediately.

Using ring acceleration can significantly reduce the amount of time that tasks must wait for access to global resources. On ACCELSYS, you specify the number of consecutive systems that must see the RSA-message before one of the systems sends the shoulder-tap to the originating system. If the complex shown in Figure 63 on page 123 used a value of ACCELSYS(3), a resource requested on SYS1 would be granted once SYS1 received a shoulder-tap from SYS3.

Recovery

Using ring acceleration can further improve ring performance in large complexes, but it might introduce recovery considerations. When the sysplex matches the complex and ACCELSYS is set to 2, if two consecutive systems in the ring should fail, XCF might prompt the operator to either start or remove one of the systems that is stopped. Otherwise, one of the failed systems might hold the RSA-message and might have sent the shoulder tap to the other, which could have then granted access to resources prior to the system failure. The other systems in the ring would then not be aware that access had been granted to those resources.

ACCELSYS recommendations

To minimize the chance of such a problem, avoid temporarily stopping more than one system at a time when you are using ring acceleration. See Chapter 9, "Operating the ring complex," on page 147 for additional information about operating a global resource serialization complex.

To use ring acceleration, specify ACCELSYS on all systems. ACCELSYS(2) provides the maximum performance benefits. Global resource serialization rejects an ACCELSYS value that is less than 2 or greater than 99. Specifying a value greater than the number of systems in the complex turns ring acceleration off. The default is ACCELSYS(99), which turns ring acceleration off.

Specifying a different ACCELSYS value for different systems in a complex is allowed, but the system uses the highest ACCELSYS value specified to determine when to send the shoulder-tap acknowledgment. If the highest value is greater than the number of systems in the ring, the shoulder-tap acknowledgment does not take place. To ensure that shoulder-tap processing occurs, specify ACCELSYS(n) on all systems in the ring, making sure that the value of n is less then the number of systems in the ring.

Specifying global resource serialization tracing options (CTRACE)

The CTRACE parameter in GRSCNFxx allows you to modify the default tracing options used by global resource serialization. Since the defaults provide adequate serviceability information, these should be changed only upon the recommendation of your IBM Service Representative.

Activating synchronous reserve processing (SYNCHRES)

The SYNCHRES parameter in GRSCNFxx indicates whether the system defined in the GRSCNFxx MATCHSYS parameter is to have synchronous reserve processing activated. The default is YES. See "Understanding the synchronous RESERVE feature" on page 9 for more information.

Specifying the maximum ENQ requests (ENQMAXA, ENQMAXU)

The ENQMAXA and ENQMAXU parameters in GRSCNFxx identify the system-wide maximum of concurrent ENQ requests for authorized (ENQMAXA) or unauthorized (ENQMAXU) requesters.

The ENQMAXA range is 250,000 to 99,999,999. The default MAXVALUE is 250,000.

The ENQMAXU range is 16,384 to 99,999,999. The default MAXVALUE is 16,384.

See "Limiting global resource serialization requests" on page 10 for more information.

Define the complex to MVS

If necessary, use the worksheet shown in Table 8 on page 125 to plan the content of GRSCNFxx.

Table 8. GRSCNF____ Definition (Sysplex Matches Complex)		
Statement	**Parameter**	**Comments**
GRSDEF	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	
GRSDEF	MATCHSYS(_____)	
	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	
GRSDEF	MATCHSYS(_____)	
	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	
GRSDEF	MATCHSYS(_____)	
	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	

Designing a mixed complex (sysplex does not match complex)

As stated earlier, a global resource serialization complex consists of all the systems that are able to share global resources. There are links between the systems that enable them to communicate, primarily by passing the RSA-message from one system to another. Each system in the complex indicates at IPL time that it is to be part of the complex.

You need to decide how many of these systems to combine into one complex. The design of global resource serialization allows up to 32 systems per complex. However, the practical limit is much lower. The actual number of systems your particular installation can reasonably configure into a complex depends on a number of factors. You should, for example, consider the operations and performance implications of a very large complex.

Note: Global resource serialization ring complexes can contain XCFLOCAL, MONOPLEX, or systems that are part of a multisystem sysplex. However, only one multisystem sysplex can exist in any global resource serialization complex.

Once you have selected the systems that are to be part of the complex, you must then define the communication links that connect the systems. Use the information in "Choosing the link configuration" on page 126 to help you with this process. Since systems outside of a sysplex cannot communicate using XCF, you must provide communication links that global resource serialization can use. Use the information in "Choosing the link configuration" on page 126 to help you with this process.

Choosing the link configuration is one major part of designing a mixed complex; the other is defining the processing options that your complex needs. For example, you must decide:

- The length of the residency time value, that is, the minimum length of time a system is to hold the RSA-message before sending it on to the next system in the ring.
- The length of the tolerance interval, that is, the maximum length of time that global resource serialization is to wait for the incoming RSA-message before it signals a disruption.
- Whether or not the system can automatically rebuild a disrupted global resource serialization ring.
- Whether or not the system can automatically rejoin the ring after it has temporarily stopped.

- Whether or not to use ring acceleration, and if so, the number of systems in the ring that need to know about global resource requests before they can be granted.

Use the information in "Processing options in a mixed complex" on page 130 to make these decisions.

Planning aids

"Defining the complex to MVS" on page 139 includes a blank configuration diagram and a worksheet for recording the design of your complex. The diagrams and worksheet can help you implement your plan.

Reference

Once you have completed your plan, see *z/OS MVS Initialization and Tuning Reference* for information about how to specify the global resource serialization parmlib member and system parameters.

Choosing the link configuration

Systems in the same XCF sysplex communicate with each other only through XCF links. However, global resource serialization links must be defined for communication between systems that cannot communicate with each other through XCF. These include links between all non-sysplex systems, as well as between sysplex systems and non-sysplex systems. See Figure 64 on page 126 for an illustration.

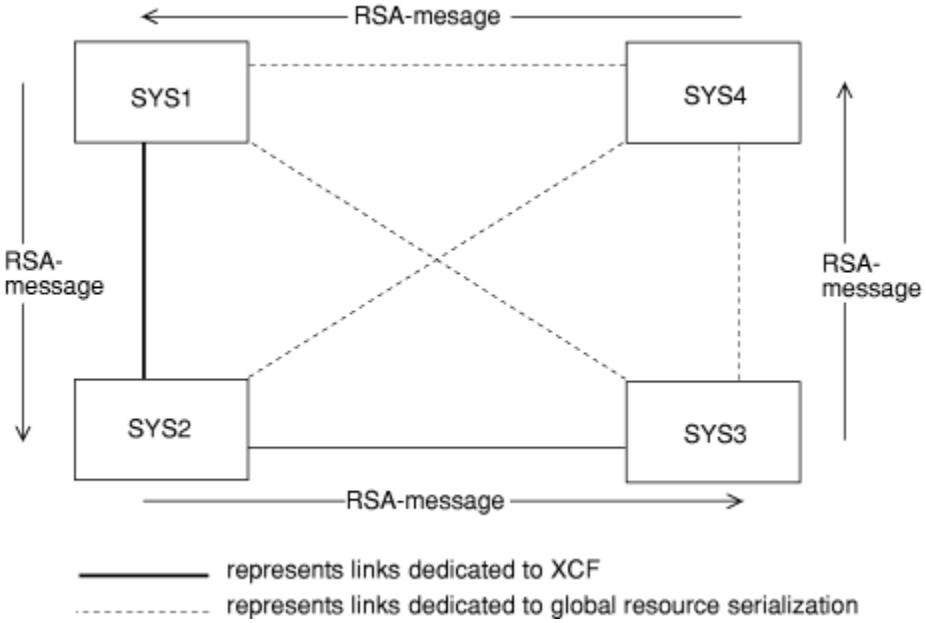


Figure 64. Links in a Mixed Complex

The GRS-managed links must be FICON (FCTC), basic mode (BCTC) ESCON CTCs or parallel CTC adapters dedicated to global resource serialization. Any other CTC type, such as ESCON SCTC, is not supported.

Link placement

Global resource serialization requires fast communication between systems. Without channel interference, this communication normally requires only a few milliseconds. To achieve the optimum communication speed, place the 3088 on a 3-megabyte or faster data streaming block multiplexor channel.

Other data links used by other MVS subsystems can share the channel, but do not connect data links on the same channel as any devices that keep the channel busy for a long period of time. For example, some DASD control units, such as the IBM 3880, might be compatible while tape and terminal control units are not. The 3088, however, might monopolize the channel, causing delays in I/O operations to DASD.

Note: If one or more of the systems in the complex are running as VM guest systems, provide real CTC links (not virtual CTCs) and dedicate the CTC links to global resource serialization.

Other configuration recommendations are:

- Design a fully-connected complex — one where every system has a communication link to every other system. See "Level of connectivity" on page 127 for more information.
- Provide at least one alternate (a second connection) for each communication link. Alternate links are required if you want to use ring acceleration to speed up the processing of global resource requests. See "Alternate links" on page 128 for more information.
- Provide a backup IBM 3088. See "Backup considerations" on page 130 for more information.

Following these recommendations ensures a complex that provides the best possible performance and availability. Configuration decisions have a significant effect on recovery planning.

Recovery

Any failure that disrupts ring processing requires recovery. Global resource serialization can both detect the failure and respond to it. Depending on options your installation selects, global resource serialization can automatically rebuild a disrupted ring. It can also automatically issue VARY commands to enable and disable communication links. These actions speed up recovery from a failure and reduce operator intervention in the recovery process. The primary causes of a failure that requires recovery are:

- A system fails because of a software problem, which can be either related to ring processing or independent of ring processing.
- A system fails because of a hardware problem.
- A CTC link fails. The failure can occur in the link itself or in any hardware or software component required in the communication path.
- Global resource serialization detects either a temporary problem on a system in the ring (such as when an operator stops a system) as a system failure or a temporary delay in communication as a communication link failure.

Whatever the cause of a break in ring processing, the result is a ring disruption; global resource serialization suspends the processing of requests for global resources until it recovers from the failure.

In designing your complex, plan a complex that can recover from a system or a CTC link failure, regardless of whether the failure is temporary or not.

After a system failure, there should be enough CTC links available to reconfigure a subset ring of **n-1** systems, where **n** is the number of systems in the original ring. That is, you want a complex where a system failure means that only the failed system must withdraw from the ring.

After a CTC link failure, there should be enough CTC links available to reconfigure the original ring. That is, you want a complex where the failure of a single link does not force any system to withdraw from the ring.

To design a complex that can recover effectively from failures, consider both the level of connectivity and the need for alternate CTC links. Alternate links are CTC links that global resource serialization can use for ring acceleration, which improves global resource request response time. Alternate links are available for use when there is a failure on a link that global resource serialization is using to pass the RSA-message around the ring, which improves recovery.

Level of connectivity

The number of CTC links available to the complex determines, to a great extent, the availability, ease of operation, and performance of the complex.

A **fully-connected complex** exists when every system in the complex has a communication link to every other system in the complex. Each system in a fully-connected n-system complex has **n-1** communication links, where **n** is the number of systems in the complex. For example, each system in a fully-connected four-system complex requires three link connections.

Note: Systems in a sysplex are fully connected with each other.

In contrast, a **partially-connected complex** is one where not every system has a communication link to every other system. That is, some systems in an **n**-system complex have fewer than **n-1** communication links. Systems operating outside of a multisystem sysplex or where the sysplex does not match the complex might be in a partially-connected complex. For example, some systems in a partially-connected four-system complex have less than three communication links. The level of connectivity of the complex affects the operation of the complex in two basic ways. One is the order in which one system starts the complex and other systems join the complex. A partially-connected complex of four or more systems requires coordination of IPLs so that global resource serialization can build a valid ring. The second effect is on recovery from a system or CTC link failure that cannot be repaired immediately. A partially-connected complex limits the reconfiguration options available to recover from the failure.

Figure 65 on page 128 illustrates the problem of a system failure on a partially-connected complex. It shows an active four-system ring with five communication link connections.

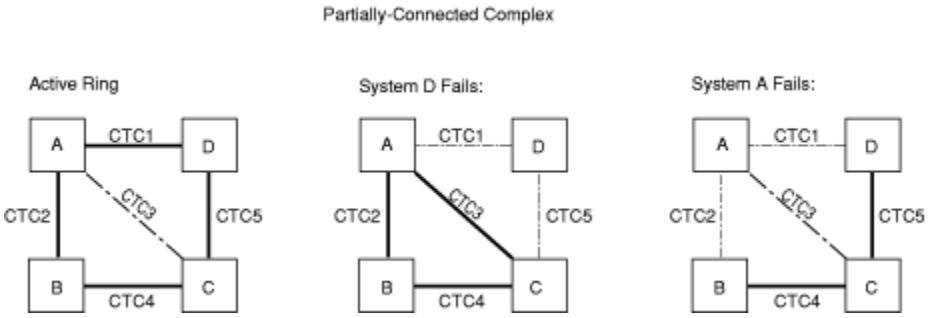


Figure 65. Recovery Problems with a Partially-Connected Complex

If system D fails, global resource serialization can automatically rebuild systems A, B, and C into a three-system ring using CTC2, CTC3, and CTC4, as shown in the figure. The three-system ring can resume processing requests for global resources. This four-system ring can recover just as quickly from a failure on system B.

If system A fails, however, the problem is entirely different because the complex is not fully connected. Global resource serialization cannot rebuild a three-system ring. It can automatically rebuild a two-system ring consisting of systems D and C (using CTC5) or a two-system ring consisting of systems B and C (using CTC4), and there is no way to predict which ring it would build. The same problem occurs if system C fails.

If the complex shown in the figure were fully connected, it would include a CTC link between system B and system D. With this fully-connected complex, global resource serialization can respond to a failure on any system by rebuilding a three-system ring that can quickly resume processing requests for global resources.

The distinction between a fully-connected complex and a partially-connected complex does not exist for a two-system complex or a three-system complex. For these complexes, the number of communication link connections required for full connectivity and the minimum number of communication link connections required are the same.

IBM recommends that you design a fully-connected global resource serialization complex, especially if you are operating a mixed complex that includes a multisystem sysplex. Even a fully-connected complex, however, might not meet the level of reliability or performance your installation requires. Designing a complex that includes alternate links offers significant advantages for both reliability and performance.

Alternate links

Global resource serialization uses one link, the primary link, to send the RSA-message from one system in the ring to another. Alternate links provide additional connections. At IPL time, global resource serialization uses one link as the primary connection and marks any other links as alternates. Alternate links provide two very important benefits.

If alternate links exist, you can use ring acceleration during normal ring processing. Ring acceleration speeds up the process of granting requesters access to global resources. See “Ring acceleration (ACCELSYS)” on page 134 for more information.

Alternate links also provide improved recovery. If the primary link fails during ring processing, a ring disruption occurs. If an alternate link is available, however, global resource serialization automatically selects an alternate link to replace the primary link, and ring processing can resume almost immediately. If global resource serialization was using the alternate link to send the ring acceleration signal, and no other link is available, the ring acceleration signal can no longer be sent between the two systems. This loss might affect performance, but the ring continues processing.

Figure 66 on page 129 shows the problem of a link failure in a three-system complex. If CTC1 fails, system A and system B cannot communicate. Global resource serialization can rebuild a two-system ring, but the two-system ring omits either system A or system B. One of the two systems cannot continue to serialize access to global resources.

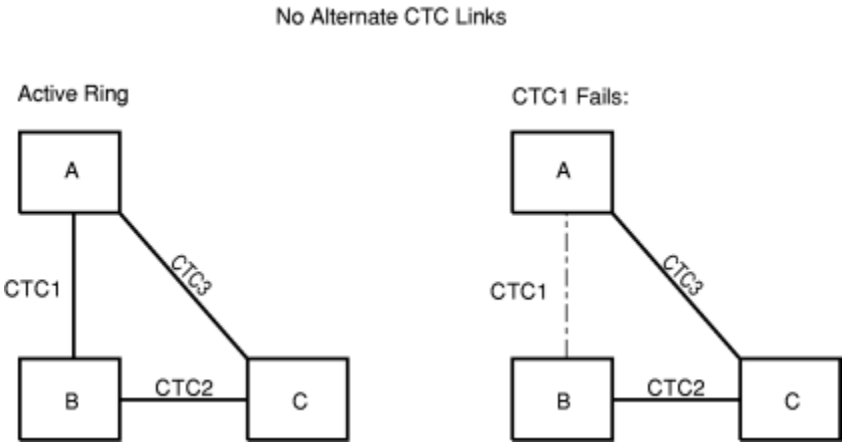


Figure 66. CTC Link Failure — No Alternate Available

Figure 67 on page 129 shows how alternate links can solve this problem. If there is an alternate link between system A and system B, global resource serialization can use the alternate to rebuild the original three-system ring. The alternate link means that the two systems can continue to serialize access to global resources even when the primary link fails; global resource serialization uses the alternate in place of the primary, and ring processing continues.

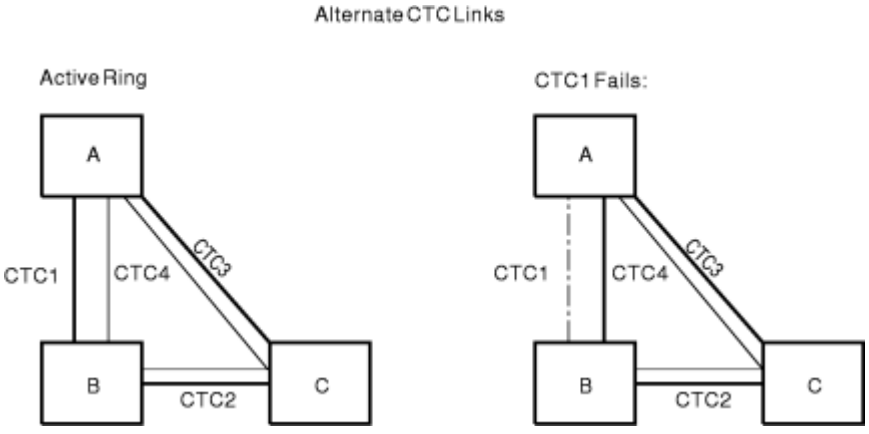


Figure 67. CTC Link Failure — Alternate Available

Because of the recovery and performance benefits, IBM recommends that you provide alternate links for the global resource serialization complex.

For additional information, see “Alternate link failure ” on page 207.

Backup considerations

A fully-connected complex with alternate links provides the option of using ring acceleration as well as a very high level of reliability. It does not, however, remove the IBM 3088 as a potential single point of failure. Thus, IBM recommends that you provide an alternate 3088 to act as a backup. (Integrated CTC adapters can also act as backup connections between systems.)

In such a configuration, dedicate two data links from the primary 3088 and two data links from the alternate 3088 for each connection in your fully-connected complex. If a system runs on a processor that, like the 3090 Model 400E, can be partitioned, connect links from both 3088 units to both sides of the processor complex.

Processing options in a mixed complex

Processing options provide the information that global resource serialization needs about the systems in the complex. Some of this information comes from the system parameters specified at IPL time. These parameters are:

- GRS=START, TRYJOIN, or JOIN — indicates whether the system is to start or join the complex.
- SYSNAME=name — identifies the name the system has in the global resource serialization complex.
- GRSCNF=xx — identifies the GRSCNFxx parmlib member that defines the complex.
- GRSRNL=xx — identifies the GRSRNLxx parmlib member that holds the RNLs the system is to use.

Chapter 2, “Selecting the data,” on page 17 described how to use GRSRNLxx to define the RNLs. This section describes how to use GRSCNFxx to define the processing options for the complex. You must create GRSCNF. Because there is no way of predicting the configuration of a particular installation's links, IBM does not supply a default GRSCNFxx member for a mixed complex.

The information that you can specify in GRSCNFxx for each system is:

- The name of the system. See “System name (MATCHSYS)” on page 131.
- The specific device numbers of all CTC links attached to the system and available to global resource serialization. See “CTC link device numbers (CTC)” on page 131.
- The minimum length of the RSA-message residency time. See “Residency time value (RESMIL)” on page 131.
- The length of the tolerance interval — the length of time that global resource serialization is to wait for an overdue RSA-message before it signals a disruption. See “Tolerance interval (TOLINT)” on page 133.
- Whether or not the complex is to use ring acceleration and, if so, how many systems must see the RSA-message before a system sends the shoulder-tap acknowledgment. See “Ring acceleration (ACCELSYS)” on page 134.
- Whether or not the system can automatically rebuild a disrupted ring. See “Automatically rebuilding a disrupted ring (RESTART)” on page 136.
- Whether or not the system can automatically rejoin a ring after the system has been temporarily stopped. See “Automatically rejoining the ring (REJOIN)” on page 137.
- The tracing options you want available for CTRACE processing. See “Specifying global resource serialization tracing options (CTRACE)” on page 138.
- The system-wide maximum of concurrent ENQ requests for authorized requesters (ENQMAXA). See “Specifying the maximum ENQ requests (ENQMAXA, ENQMAXU)” on page 100.
- The system-wide maximum of concurrent ENQ requests for unauthorized requesters (ENQMAXU). See “Specifying the maximum ENQ requests (ENQMAXA, ENQMAXU)” on page 100.
- The synchronous RESERVE feature (SYNCHRES). See “Understanding the synchronous RESERVE feature” on page 9.

z/OS MVS Initialization and Tuning Reference contains complete syntactical information about how to create GRSCNFxx and how to specify the system parameters. Before you can create GRSCNFxx, however, you need to understand the options and what effect each option has on the complex.

System name (MATCHSYS)

Global resource serialization matches the name specified on the SYSNAME system parameter at IPL time to a name specified on MATCHSYS to locate the information in GRSCNFxx for a particular system. Each system name must be unique within the complex. It is a good practice to use the same four-character name for MATCHSYS that you use to identify the system to SMF, that is, the same value you specify for the SMF SID parameter. Consistent use of the same system name makes identifying the system in various records and messages easier and provides a consistent identifier for the operators.

The syntax of the MATCHSYS parameter in GRSCNFxx allows you to specify MATCHSYS(*). Global resource serialization considers MATCHSYS(*) to be a match to any SYSNAME value. If an operator enters an erroneous value for SYSNAME, global resource serialization sees MATCHSYS(*) as a match and might build the complex incorrectly. To avoid this problem, identify each system explicitly by name in GRSCNFxx.

CTC link device numbers (CTC)

In a mixed complex, use the CTC option in GRSCNFxx to specify the device numbers of all CTC links attached to the system and available to global resource serialization. Specifying CTC links that you might have included at system generation but not yet installed causes global resource serialization to issue an error message (ISG046E) during system initialization.

When you have more than one CTC link between systems, global resource serialization uses the last device number specified as the primary link. Thus, the order in which you specify the device numbers affects how global resource serialization chooses the primary and the alternate links. If two links are on-line and available when a system joins the ring, global resource serialization chooses the second CTC link specified as the primary link and marks the first as an alternate. For example, if GRSCNFxx specifies CTC(860), followed by CTC(780), global resource serialization selects 780 as the primary and 860 as the alternate.

Residency time value (RESMIL)

Use the RESMIL option in GRSCNFxx to specify the residency time value. The residency time value is the minimum length of time in milliseconds the RSA-message spends in each system. The default value is 10 milliseconds.

If processing the RSA-message takes longer than the RESMIL value, the system holds the RSA-message until processing is complete. Systems adjust the actual residency time to optimize processor use and enqueue response time. The actual residency time will never be less than the RESMIL value.

The RESMIL value you specify can further affect the performance of your complex. In general, a low value, which means that the RSA-message passes around the ring more frequently, improves ring performance. That is, a low RESMIL value:

- Increases ring capacity — the number of global resource requests the ring can process
- Decreases response time — the amount of time the ring requires to process a specific request for a global resource.

Thus, setting the RESMIL value is important to tuning your complex for efficient operation. Setting a low RESMIL value increases ring capacity and decreases response time, though it can slightly increase processor utilization. The RESMIL value, however is not the only factor that affects ring performance; the configuration and workload are also important.

Configuration considerations

To set an optimal value for RESMIL, you must also consider the configuration:

- The transfer rate of the communication device
- The number of systems in the ring

To see how these factors relate, assume that you reduce by half the value of RESMIL on each system in a four-system ring. This action cuts the RSA-message cycle time roughly in half, thus approximately doubling ring capacity and cutting response time in half.

You can obtain a similar result by replacing a CTC adapter connection with an IBM 3088 link connection. The transmission rate of a 3088 data link is about three times as fast as the transmission rate of a CTC adapter. The faster communication speed reduces the RSA-message cycle time, thus increasing ring capacity and decreasing response time.

Adding a system to the ring, however, has the opposite effect; it decreases ring capacity and increases response time. The additional system adds time to the RSA-message cycle. If you are adding a system to the ring, decreasing the RESMIL value or switching to a faster communication device can maintain the existing ring capacity and response time.

Workload considerations

The rate of global resource requests is the number of requests a system generates in a given time period. The rate depends on the workload. Thus, the workload on the systems in the ring also affects ring performance. TSO/E or a batch workload normally creates more global resource requests than a workload that is mostly CICS, or IMS.

The workload can also determine how important ring performance is. Ring performance is especially important to installations that:

- Run time-sensitive jobs that use global resources or jobs (like HSM back-out) that must complete in a predetermined amount of time
- Are experiencing resource contention problems
- Are planning to use the ring to serialize catalog access

Such installations need the lowest possible RESMIL value. See “Tuning the complex” on page 167 for more information on performance.

Setting the RESMIL value differently according to the processor's power does not affect the time (or processor utilization) a system spends processing global resource requests; a system, regardless of the RESMIL value, processes all requests in the incoming RSA-message before it sends the outgoing RSA-message. The RESMIL value affects only the time (or processor utilization) a system spends on ring processing. There is no benefit in trying to use the RESMIL value to adjust for differences in processor power; only the sum of RESMIL values is significant.

RESMIL recommendations

You do not have to specify the same RESMIL value on all systems, but there is no reason to set different values. The only effect of specifying a large value for one system is to increase the RSA-message cycle time, which decreases ring capacity and increases response time. It is simpler to use the same RESMIL value for all systems.

For a complex that consists only of small systems (such as systems running on an IBM 4381 or only of large systems (such as systems running on an IBM 3090 Model 400E), set RESMIL to the lowest value that works for your complex.

For a complex that includes systems running on both large and small processors, there is a basic problem: the total percent of a small processor's power spent on global resource serialization might be very high because it must process global resource requests generated on all systems.

Setting the RESMIL value differently according to the processor's power does not affect the time (or processor utilization) a system spends processing global resource requests; a system, regardless of the RESMIL value, processes all requests in the incoming RSA-message before it sends the outgoing RSA-message. The RESMIL value affects only the time (or processor utilization) a system spends on ring processing. There is no benefit in trying to use the RESMIL value to adjust for differences in processor power; only the average value is significant.

Thus, use the workload on the large processor to select the RESMIL value and set the same value on all systems. If the workload on the large processor generates many requests for global resources, set RESMIL to a lower value on all systems. Otherwise, make no adjustment for processor power.

The same recommendation applies to a ring where each system runs a distinct workload. Use the workload on the system that generates the largest number of global resource requests as a base for setting RESMIL on all systems.

See "Tuning the complex" on page 167 for a description of some techniques you can use to determine the best value for your complex.

Tolerance interval (TOLINT)

Use the TOLINT option in GRSCNFxx to specify the tolerance interval. The tolerance interval is the length of time that global resource serialization is to wait for an overdue RSA-message before it signals a disruption. Determining an acceptable time-out value warrants the following considerations in a global resource serialization complex:

- The excessive spin time and recovery actions for each system
- The number of systems
- The speed of the systems
- Inter-system signalling configuration, and activity
- Paging of the global resource serialization common area storage for each system
- The RESMIL time for each system

Typically, the RSA-message should proceed quickly around the ring. A system that fails or is stopped temporarily, or a link that fails or temporarily slows down communication, can cause a significant delay of the RSA-message. Such as,

- An MVS image recovering from a spin loop
- An MVS image is taking an SVC dump
- Delays in inter-system communications
- Shortages in real storage
- Auxiliary storage page-in delays

During a ring disruption, all tasks that request or free global resources are suspended because the RSA-message is halted and there is no communication between systems in the ring. As the ring disruption continues, more and more tasks are suspended, slowing the throughput of each system in the ring.

A ring disruption requires recovery. Global resource serialization can recover automatically from most ring disruptions when you specify automatic restart and automatic rejoin. Specifying RESTART(YES) and REJOIN(YES) allows recovery without operator intervention; global resource serialization issues messages but does not usually require operator action. See "Automatically rebuilding a disrupted ring (RESTART)" on page 136 and "Automatically rejoining the ring (REJOIN)" on page 137.

The value you set for TOLINT affects how rapidly global resource serialization detects an overdue RSA-message, and setting the value properly requires a basic trade-off:

- To detect a system failure or a link failure, the best TOLINT value is one that recognizes the condition almost immediately.
- To deal with a temporary delay, the best TOLINT value is one that does not detect the condition. There are many reasons for a system entering a temporary stop, such as a spin loop or taking an SDUMP to capture the contents of common storage. For a temporarily stopped system or a temporary link delay, the best TOLINT value is one that is large enough to allow normal RSA-message processing to resume without causing a ring disruption.

Thus, the best TOLINT value is one that allows global resource serialization to detect a system or link failure promptly but does not cause it to continuously detect temporary delays. If you specify RESTART(YES) and REJOIN(YES), setting a low TOLINT value has minimal effect because ring recovery is

automatic. If your installation chooses not to use automatic restart and rejoin, set a higher value to avoid unnecessary ring disruptions that require operator intervention. The default value for TOLINT is three minutes. Depending on your installation, this value can be lowered. In other complexes, or when MVS is running in a PR/SM environment, a good value is between 40 and 60 seconds. If MVS is running as a guest under VM, set the TOLINT value even higher.

If your installation chooses not to use automatic restart and automatic rejoin, set a higher value. The higher value avoids unnecessary ring disruptions that require operator intervention.

The TOLINT value does not have to be the same for all systems, but it is a good idea to specify it consistently. If you set it to different values, the system with the smallest value is the first system to detect the disruption and initiate recovery.

Ring acceleration (ACCELSYS)

Use the ACCELSYS option in GRSCNFxx to specify ring acceleration and the number of systems that must see a resource request before it is granted.

Without ring acceleration, every system in the ring must see each request for a global resource. While the RSA-message makes a complete cycle around the ring, the task that requested the global resource is suspended. This processing guarantees the integrity of resources; no global resource request is granted until all systems know about it. It does, however, mean that every task that requests a global resource must wait for at least one RSA-message cycle.

Ring acceleration offers an alternative technique, which protects the integrity of resources while potentially providing a significant reduction in global resource request response time (the time a task is suspended while waiting for ring processing).

Ring acceleration requires alternate links, used to send the ring acceleration signal from one system to another. In addition, IBM suggests that the complex be a fully-connected complex. Using ring acceleration can significantly improve ring performance in large complexes; in two-system complexes, it provides minimal benefits.

Figure 68 on page 134 shows an example of a complex that could use ring acceleration. It is a fully-connected four-system complex with primary links, shown as heavy lines, and alternate links, shown as light lines.

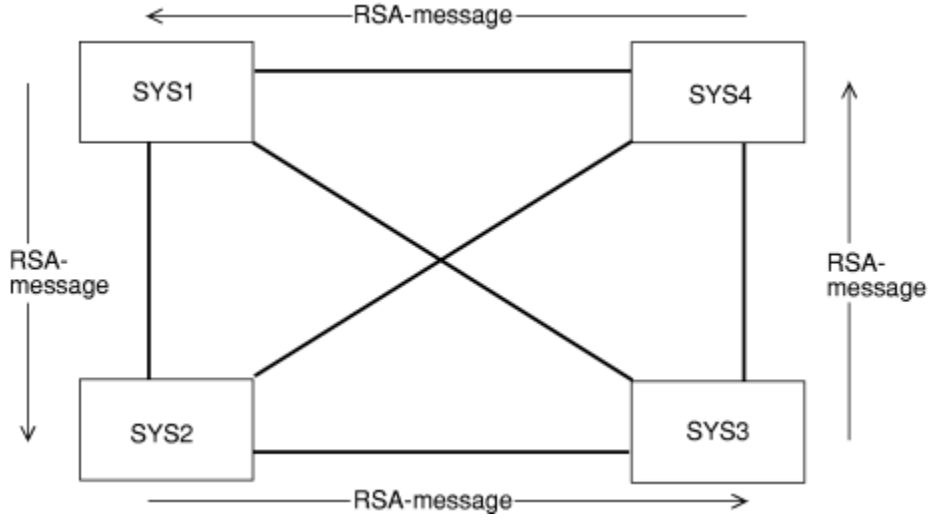


Figure 68. Ring Acceleration Configuration

Request processing

Using Figure 68 on page 134, assume that a task on SYS1 requested access to a resource and that the RSA-message is moving as shown. Without ring acceleration, the task on SYS1 would wait until the RSA-message made a complete cycle around the ring. Only when SYS2, SYS3, and SYS4 all know about the request can SYS1 grant the resource to the requester.

In contrast, assume that GRSCNFxx contains ACCELSYS(2) to request ring acceleration. With ring acceleration, SYS1 still suspends the task that requested the resource, puts the request in the RSA-message, and sends the RSA-message on to the next system in the ring — SYS2 in this example.

SYS2, when it receives the RSA-message, uses the alternate link to send a shoulder-tap acknowledgment, the ring acceleration signal, to SYS1. SYS2 sends the signal because it is the second system to see the request, and ACCELSYS(2) means that two systems must see the request before it can be granted. After sending the shoulder-tap, SYS2 then processes the RSA-message. The RSA-message continues its cycle around the ring. All systems see and process the request, which preserves the integrity of the resource.

SYS1, as soon as it receives the ring acceleration signal, can grant the request. The task that requested the resource does not have to wait for the RSA-message to make a complete cycle around the ring. If the requested resource is available, the task can resume execution almost immediately.

Using ring acceleration can significantly reduce the amount of time that tasks must wait for access to global resources. On ACCELSYS, you specify the number of consecutive systems that must see the RSA-message before one of the systems sends the shoulder-tap to the originating system. If the complex shown earlier in Figure 68 on page 134 used a value of ACCELSYS(3), a resource requested on SYS1 would be granted once SYS1 received a shoulder-tap from SYS3.

Recovery

Ring acceleration provides significant performance improvement, but it does introduce recovery considerations. The ACCELSYS value, as stated earlier, specifies the number of systems that must see the RSA-message before the originating system can grant a request. It also, in effect, specifies the number of consecutive systems that can fail before ring acceleration introduces a possible data integrity exposure.

For example, look again at the complex shown in Figure 68 on page 134 and assume that ACCELSYS(2) is in effect. If any single system fails or is stopped temporarily, global resource serialization can safely rebuild the ring because at least one of the remaining systems has current resource information. If two non-consecutive systems, like SYS1 and SYS3, fail or are stopped temporarily, rebuilding the ring safely is still possible because either SYS2 or SYS4 has current resource information.

If, however, two consecutive systems, like SYS1 and SYS2, fail, and one of the failed systems held the RSA-message, then rebuilding the ring safely is not possible. There is a potential data integrity exposure because the failed systems are the systems that have current resource information. For example, assume that SYS2 held the RSA-message and sent the shoulder-tap to SYS1. SYS1, after receiving the shoulder-tap, granted access to one or more resources. If both systems then fail or are stopped temporarily, the active systems (SYS3 and SYS4) do not know about the resources granted on SYS1 and, if the ring were rebuilt, might grant other tasks access to these same resources.

In this situation, global resource serialization does not automatically rebuild the ring. Instead, it issues a unique operator message **ISG080E** . The operator must follow normal installation procedures to resolve any data integrity exposure and then issue a restart command to rebuild the ring.

ACCELSYS recommendations

To minimize the chance of such a problem, avoid temporarily stopping more than one system at a time. If your operators regularly issue VARY GRS(sysname),QUIESCE for a system before stopping it, you can avoid any recovery problems related to temporarily stopped systems. See Chapter 9, “Operating the ring complex,” on page 147 for additional information about operating a global resource serialization complex.

To use ring acceleration, specify ACCELSYS on all systems. ACCELSYS(2) provides the maximum performance benefits. Global resource serialization rejects an ACCELSYS value that is less than 2 or greater than 99. The operator must re-IPL with a valid value, or specify GRS=NONE to continue with the IPL. Specifying a value greater than the number of systems in the complex turns ring acceleration off. The default is ACCELSYS(99), which turns ring acceleration off. You can change the ACCELSYS value to tune enqueue response time. See “ACCELSYS value” on page 170 for more information.

Automatically rebuilding a disrupted ring (RESTART)

Use the RESTART option in GRSCNFxx to indicate whether or not the system can automatically rebuild the ring when it detects an error that disrupts global resource serialization processing.

Specifying RESTART(YES) indicates that the system can automatically rebuild the ring when either one of the following conditions is true:

1. The system can communicate with more than half of all systems that were in the ring (including itself).
2. The system can communicate with exactly one half of all systems that were in the ring; all the systems with which it cannot communicate have the RESTART(NO) option specified.

Having a system rather than the operator initiate recovery has several advantages:

- It can speed up the recovery process significantly.
- It reduces operator intervention in recovery.
- It avoids the possibility of split rings, which can occur when more than one operator tries to restart the ring at the same time, causing the ring to split into multiple independent rings. See "Split rings" on page 159.

Thus, specify RESTART(YES) whenever possible:

- In two-system mixed complexes, specify RESTART(YES) on one system and RESTART(NO) on the other. Global resource serialization can automatically rebuild a one-system ring if the link fails or if the RESTART(NO) system fails. If you specify RESTART(YES) on both systems, global resource serialization does not restart either system if the link fails.

  Specifying RESTART in this way means that an operator must rebuild a disrupted ring only when:

  – The primary link fails and there is no alternate available
  – The RESTART(YES) system itself fails.

  If one system is more critical than the other, specify RESTART(YES) on the critical system and RESTART(NO) on the on the other; otherwise, arbitrarily choose one system as the RESTART(YES) system.
- In complexes of three or more systems, specify RESTART(YES) for all systems.

Note:

1. An installation that specifies both RESTART(YES) and REJOIN(YES) can reduce the need for operator intervention in recovery from a ring disruption.
2. RESTART is forced to YES on systems in a sysplex.

Example of automatic restart

For this example, the operator actions assume a fully-connected four-system complex. Assume that SYS1 failed and that SYS2 detected the failure. Assume also that GRSCNFxx specified RESTART(YES) and REJOIN(YES) for all four systems.

On all other systems, the following message appears:

```
ISG023E GLOBAL RESOURCE SERIALIZATION HAS BEEN DISRUPTED - GLOBAL
        RESOURCE REQUESTORS WILL BE SUSPENDED
```

This message indicates that an error has disrupted ring processing, though it does not pinpoint the source of the error. All active systems become inactive; any task that tries to obtain or free a global resource is suspended. All systems then attempt to restart the complex, but only one coordinates the process. In this example, assume that SYS2 is the system that will recover the complex.

On SYS2, the system coordinating the complex recovery, the following message indicates that automatic restart has begun:

```
ISG024I SYSTEM SYS2 INITIATED AUTO RESTART PROCESSING
```

On SYS3 and SYS4, the following message appears:

```
ISG025E SYSTEM sysx UNABLE TO INITIATE AUTO RESTART
        PROCESSING - PERMISSION GRANTED TO SYS2
```

This message, where *sysx* is SYS3 or SYS4, indicates that this system has given permission to SYS2 to rebuild the complex. It is a message that the operators should expect to see during automatic restart processing. It normally requires no operator action.

As SYS2 rebuilds the complex, assuming that SYS3 and SYS4 rejoin the complex in that order, the following messages appear as each system rejoins the complex.

On SYS2:

```
ISG011I SYSTEM SYS2 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS2 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

On SYS3:

```
ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

On SYS4:

```
ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

Global resource serialization has recovered the complex automatically without operator intervention. The recovered complex consists of SYS2, SYS3, and SYS4.

Automatically rejoining the ring (REJOIN)

Use the REJOIN option in GRSCNFxx to indicate whether or not a system that has been temporarily stopped can automatically rejoin the ring when the system resumes processing. A system that stops temporarily causes a ring disruption; the ring is rebuilt without the stopped system. REJOIN(YES) allows the system to automatically rejoin the ring when it resumes processing; no operator intervention is required. REJOIN(NO) means that the operator must bring the system back into the ring when the system resumes processing.

IBM recommends that you always specify REJOIN(YES) to avoid the problems of operator intervention and operator delay during recovery. REJOIN(YES) does not cause any potential ring processing problems or data integrity exposures. When the automatic rejoin does not succeed, global resource serialization issues messages, and the operator can then intervene to bring the system back into the ring. In general, the automatic rejoin succeeds when there is at least one active system in the ring and an available link between the active system and the rejoining system.

Specify REJOIN(NO) only when your installation needs absolute control over the process of rebuilding a disrupted ring.

Note:

1. An installation that specifies both RESTART(YES) and REJOIN(YES) can reduce the need for operator intervention in recovery from a ring disruption.
2. REJOIN is forced to YES on systems in a sysplex.
3. On systems running MVS Version 2, REJOIN(YES) cannot be specified with RESTART(NO).

Recovery actions for a system failure with automatic restart

From an active system, the operator must determine if the failed system, SYS1, stopped temporarily or if the failure requires reIPL.

- If SYS1 stopped temporarily, do nothing.

  When SYS1 resumes processing, it automatically rejoins the complex because REJOIN(YES) was specified. The following messages appear on all active systems:

  ```
  ISG011I SYSTEM SYS1 - RESTARTING GLOBAL RESOURCE SERIALIZATION
  ISG013I SYSTEM SYS1 - RESTARTED GLOBAL RESOURCE SERIALIZATION
  ```

- If SYS1 requires reIPL, do the following:

  1. On an active system, enter VARY GRS(SYS1),PURGE to remove SYS1 from the complex. The following message appears:

     ```
     ISG017D CONFIRM PURGE REQUEST FOR SYSTEM SYS1 - REPLY YES OR NO
     ```

     Reply YES. The following message appears:

     ```
     ISG011I SYSTEM SYS1 - BEING PURGED FROM GRS COMPLEX
     ```

  2. If the following appears before ISG013I, the message indicates a possible data integrity exposure.

     ```
     ISG016I SYSTEM SYS1 OWNS OR IS WAITING FOR GLOBAL RESOURCES
     ```

     Notify the system programmer. Give the system log to the system programmer. The system log normally contains messages about the problem. The following message describes any resources that might have been damaged by purging the system:

     ```
     ISG018I REQUESTORS FROM SYSTEM SYS1 HAVE BEEN PURGED FROM
             RESOURCES NAMED xxxx,yyyy
     ```

  3. On SYS1, reIPL the system with GRS=JOIN.
  4. On the active systems, the following messages appear when SYS1 rejoins the complex:

     ```
     ISG011I SYSTEM SYS1 - JOINING GRS COMPLEX
     ISG004I GRS COMPLEX JOINED BY SYSTEM SYS1
     ```

Specifying global resource serialization tracing options (CTRACE)

Use the CTRACE parameter in GRSCNFxx to specify the CTRACE options you want for your installation. For more information about global resource serialization component trace, see *z/OS MVS Diagnosis: Tools and Service Aids*. For information about how to specify the options you want, see *z/OS MVS Initialization and Tuning Reference*.

The CTRACE parameter in GRSCNFxx allows you to modify the default tracing options used by global resource serialization. Because the defaults provide adequate serviceability information, you should change them only upon the recommendation of your IBM service representative.

The CTRACE parameter in GRSCNFxx is available only in systems running MVS/SP Version 4 or later.

Minimum options (MINOPS)

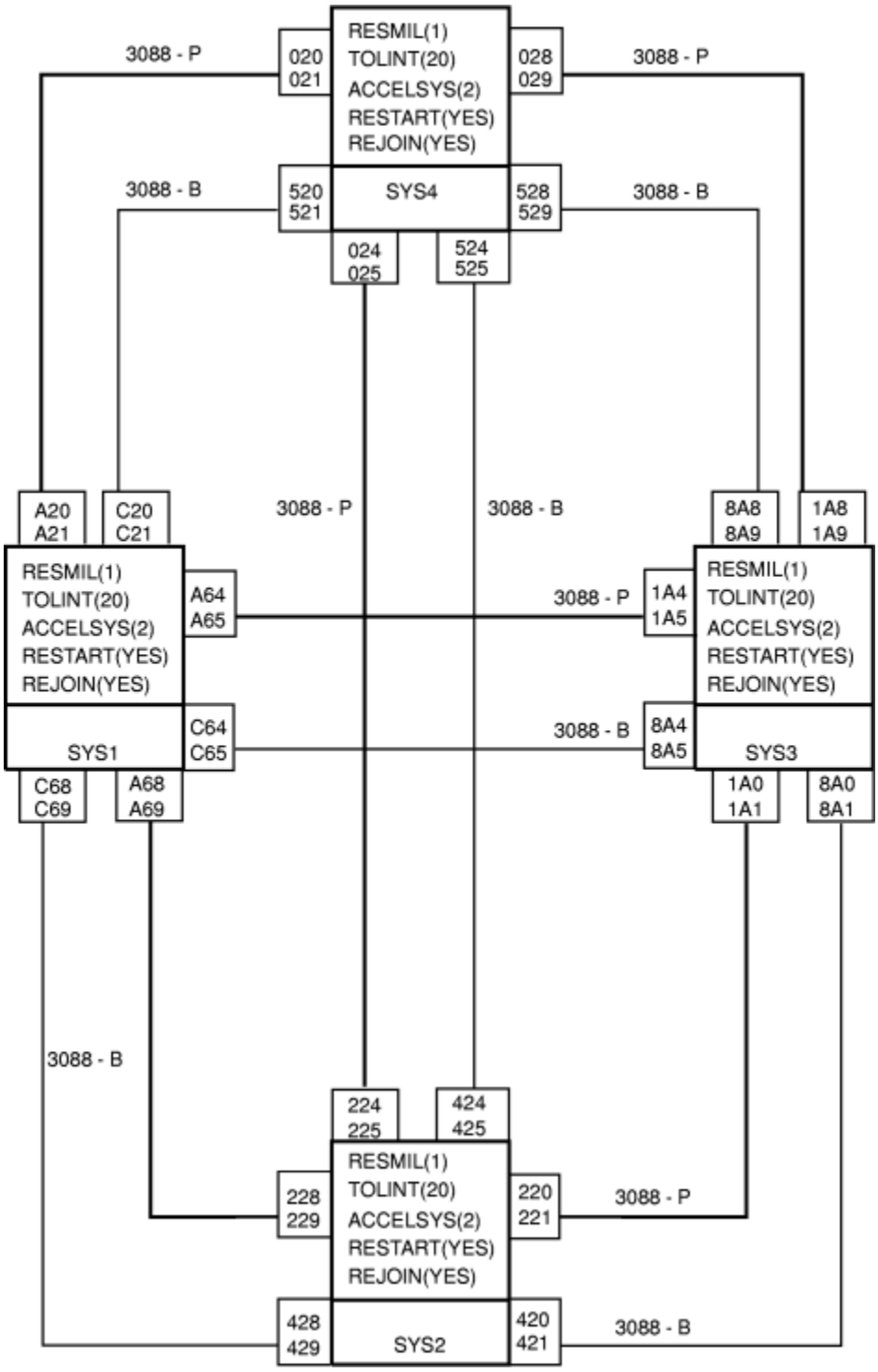
Minimum options (MINOPS) are defined for GRS=RING and GRS=NONE so that exceptional events will be traced at all times, regardless of the options specified. CTRACE will permit the trace state to be ON (with both the specified and minimum options in effect) or MIN (only the minimum options in effect). The trace will never be OFF. When viewing the trace data using the IPCS CTRACE subcommand, filtering options will be provided to limit the scope of the data displayed.

Defining the complex to MVS

A diagram consisting of system blocks and CTC link connection lines is a useful way to represent the configuration of a global resource serialization complex. To make the block diagram also useful as input to the process of defining your complex to MVS, expand the basic diagram to include the information that global resource serialization needs about the systems in the complex.

Figure 69 on page 140 shows one way to expand the basic block diagram to combine the information MVS needs about your complex with the block diagram design of your complex. The complex shown in the figure consists of 4 systems and 24 CTC links. It is a fully-connected complex with alternate links. The configuration includes a primary IBM 3088 MCCU (3088-P) and a backup 3088 (3088-B). Figure 70 on page 141 is a blank configuration diagram for your use.

To define your complex to MVS, you use the GRSCNFxx parmlib member. You can create a single member that defines all systems in the complex and copy it to the parmlibs of all systems, or you can create a parmlib member for each system that defines only that system. For more efficient testing and control, it is better to define the entire complex in a single member, then copy that member to the parmlib of each system.



P

Primary

B

Backup

Figure 69. Sample Complex Design and Definition Diagram

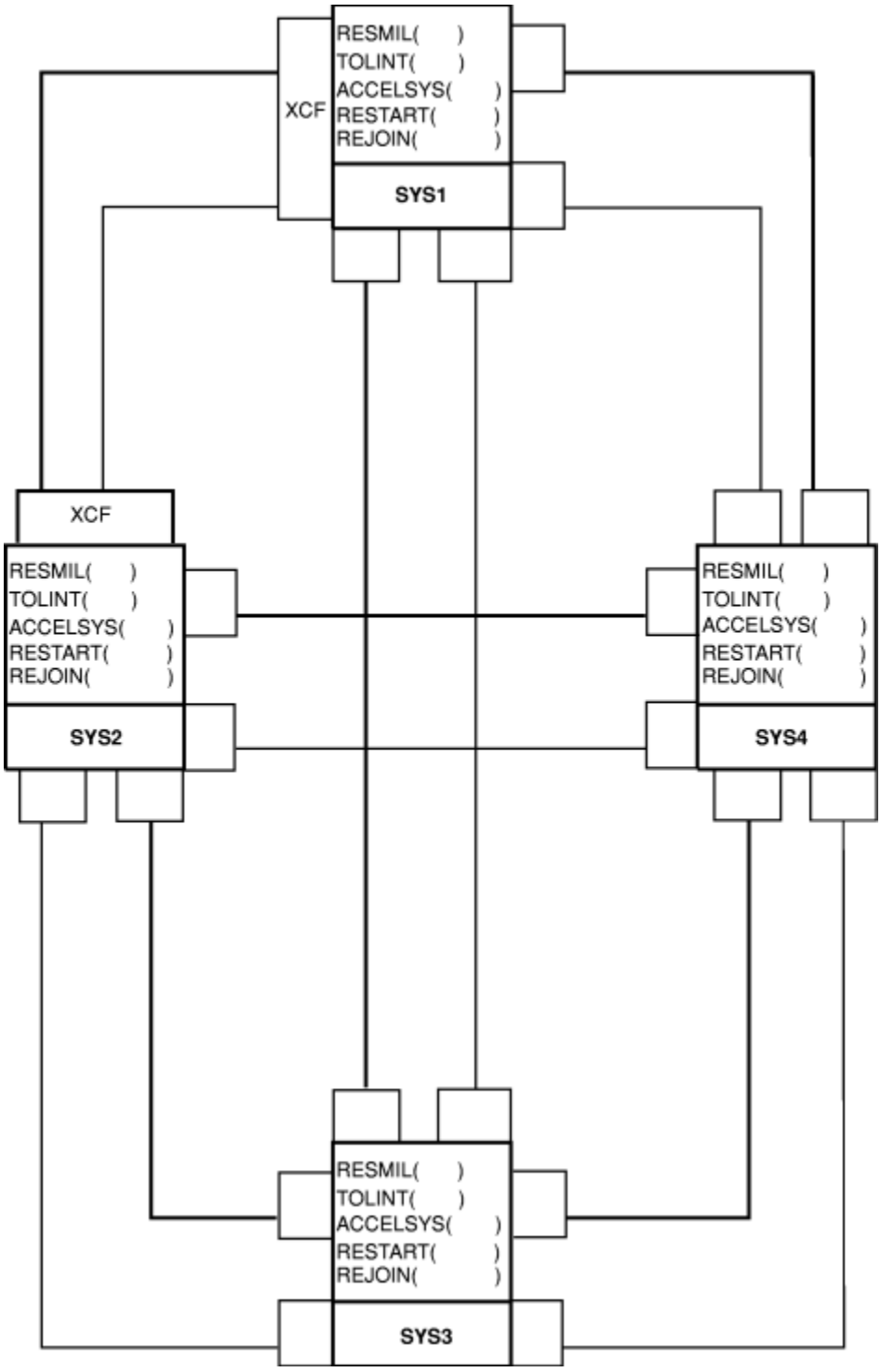


Figure 70. Sample Design and Definition Diagram for a Mixed Complex

GRSCNFxx worksheet

One way to describe the design of your complex in a format you can use to actually create GRSCNFxx is to use the GRSCNFxx worksheet. Table 9 on page 142 shows a completed worksheet for the complex shown in Figure 69 on page 140. Table 10 on page 143 is a blank worksheet for your use as you plan for systems that are not part of a sysplex. Table 8 on page 125 is a blank worksheet for your use as you plan a complex that matches the sysplex.

Table 9. Sample Complex Definition Plan

Statement	Parameter	Comments
GRSDEF	MATCHSYS(SYS1)	Matches SYSNAME=SYS1 system parameter
	RESMIL(1)	RSA-message residency = 1 millisecond
	TOLINT(20)	Tolerance interval = 20 seconds
	ACCELSYS(2)	Ring acceleration definition
	CTC(C68)	3088-B link to SYS2
	CTC(C69)	3088-B link to SYS2
	CTC(C64)	3088-B link to SYS3
	CTC(C65)	3088-B link to SYS3
	CTC(C20)	3088-B link to SYS4
	CTC(C21)	3088-B link to SYS4
	CTC(A68)	3088-P link to SYS2
	CTC(A69)	3088-P link to SYS2
	CTC(A64)	3088-P link to SYS3
	CTC(A65)	3088-P link to SYS3
	CTC(A20)	3088-P link to SYS4
	CTC(A21)	3088-P link to SYS4
	RESTART(YES)	Can rebuild disrupted ring automatically
	REJOIN(YES)	Can rejoin ring after temporary stop
GRSDEF	MATCHSYS(SYS2)	Matches SYSNAME=SYS2 system parameter
	RESMIL(1)	RSA-message residency = 1 millisecond
	TOLINT(20)	Tolerance interval = 20 seconds
	ACCELSYS(2)	Ring acceleration definition
	CTC(428)	3088-B link to SYS1
	CTC(429)	3088-B link to SYS1
	CTC(420)	3088-B link to SYS3
	CTC(421)	3088-B link to SYS3
	CTC(424)	3088-B link to SYS4
	CTC(425)	3088-B link to SYS4
	CTC(228)	3088-P link to SYS1
	CTC(229)	3088-P link to SYS1
	CTC(220)	3088-P link to SYS3
	CTC(221)	3088-P link to SYS3
	CTC(224)	3088-P link to SYS4
	CTC(225)	3088-P link to SYS4
	RESTART(YES)	Can rebuild disrupted ring automatically
	REJOIN(YES)	Can rejoin ring after temporary stop

Table 9. Sample Complex Definition Plan (continued)

Statement	Parameter	Comments
GRSDEF	MATCHSYS(SYS3)	Matches SYSNAME=SYS3 system parameter
	RESMIL(1)	RSA-message residency = 1 millisecond
	TOLINT(20)	Tolerance interval = 20 seconds
	ACCELSYS(2)	Ring acceleration definition
	CTC(8A4)	3088-B link to SYS1
	CTC(8A5)	3088-B link to SYS1
	CTC(8A0)	3088-B link to SYS2
	CTC(8A1)	3088-B link to SYS2
	CTC(8A8)	3088-B link to SYS4
	CTC(8A9)	3088-B link to SYS4
	CTC(1A4)	3088-P link to SYS1
	CTC(1A5)	3088-P link to SYS1
	CTC(1A0)	3088-P link to SYS2
	CTC(1A1)	3088-P link to SYS2
	CTC(1A8)	3088-P link to SYS4
	CTC(1A9)	3088-P link to SYS4
	RESTART(YES)	Can rebuild disrupted ring automatically
	REJOIN(YES)	Can rejoin ring after temporary stop
GRSDEF	MATCHSYS(SYS4)	Matches SYSNAME=SYS4 system parameter
	RESMIL(1)	RSA-message residency = 1 millisecond
	TOLINT(20)	Tolerance interval = 20 seconds
	ACCELSYS(2)	Ring acceleration definition
	CTC(520)	3088-B link to SYS1
	CTC(521)	3088-B link to SYS1
	CTC(524)	3088-B link to SYS2
	CTC(525)	3088-B link to SYS2
	CTC(528)	3088-B link to SYS3
	CTC(529)	3088-B link to SYS3
	CTC(020)	3088-P link to SYS1
	CTC(021)	3088-P link to SYS1
	CTC(024)	3088-P link to SYS2
	CTC(025)	3088-P link to SYS2
	CTC(028)	3088-P link to SYS3
	CTC(029)	3088-P link to SYS3
	RESTART(YES)	Can rebuild disrupted ring automatically
	REJOIN(YES)	Can rejoin ring after temporary stop

Table 10. GRSCNF____ Definition (Mixed Complex)

Statement	Parameter	Comments
GRSDEF	MATCHSYS(_____)	

Table 10. GRSCNF____ Definition (Mixed Complex) (continued)

Statement	Parameter	Comments
	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	RESTART(YES|NO)	
	REJOIN(YES|NO)	
GRSDEF	MATCHSYS(_____)	
	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	RESTART(YES|NO)	
	REJOIN(YES|NO)	

Table 10. GRSCNF____ Definition (Mixed Complex) (continued)

Statement	Parameter	Comments
GRSDEF	MATCHSYS(_____)	
	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	RESTART(YES|NO)	
	REJOIN(YES|NO)	
GRSDEF	MATCHSYS(_____)	
	RESMIL(_____)	
	TOLINT(_____)	
	ACCELSYS(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	CTC(_____)	
	RESTART(YES|NO)	
	REJOIN(YES|NO)	

Chapter 9. Operating the ring complex

As shown in Chapter 8, “Designing a ring complex,” on page 119, the design of the global resource serialization complex is a critical factor in the successful operation of the complex. However, the procedures you provide for the operators of the systems in the complex are also important.

Educate the operators and prepare operator procedures for operating the global resource serialization complex. These procedures, often called a run book or an operations workbook, describe how your operators build and operate the complex. When the sysplex matches the complex, there is little need for any operator action. When you are operating a mixed complex, however, you will need to provide explicit instructions tailored to your installation's needs.

As you read this topic and as you develop your operator procedures, consult:

- *z/OS MVS System Messages, Vol 9 (IGF-IWM)* for the exact text and responses to global resource serialization messages.
- *z/OS MVS System Commands* for a complete description of how to use the VARY and DISPLAY commands with global resource serialization.
- Chapter 11, “Diagnosing global resource serialization,” on page 175 for recovery and diagnosis procedures.
- *z/OS MVS Setting Up a Sysplex* for sysplex-related concerns.

Your operating environment determines how you plan for operation. If all of the systems in your global resource serialization complex belong to the same sysplex, see “Operating a complex that matches a sysplex” on page 147. If you are operating a mixed complex (at least one system in your global resource serialization complex is not a part of the sysplex), then you must make precise, detailed plans for operations — how your operators build the complex, keep an eye on its normal operation, and respond to the problems that automatic recovery does not handle. See “Operating a mixed complex (complex does not match sysplex)” on page 150 for those considerations.

Operating a complex that matches a sysplex

In general, your operational planning focuses on three areas: building the complex, normal operations, and recovery operations. The design of your complex affects all three areas. See Appendix B, “Recovery actions for global resource serialization,” on page 203 for information about planning your recovery operations.

In a sysplex, global resource serialization uses XCF signalling paths as communication links.

Building the complex

The process of building a global resource serialization complex can have two phases: a configuration check and the IPL of the systems.

Configuration check

Before an operator actually IPLs a system that is to start or join a global resource serialization complex, ensure that the operator verifies that the shared resource (such as DASD) connections are correct.

If the shared resource connections are incorrect, a serious data integrity exposure could occur. This exposure occurs when systems in the complex are serializing access to a global resource by means of an ENQ macro with a scope of SYSTEMS. For example, if the RESERVE conversion RNL for the complex contains an entry for a resource, this entry causes global resource serialization on each system in the complex to suppress the reserve for that resource. If a system outside the complex can use a reserve to access the same resource at the same time, the resource could be damaged.

IPL

There are three system parameters (GRS, GRSCNF, and GRSRNL) that indicate to MVS at IPL time that a system is to be part of a global resource serialization complex. The GRS and GRSCNF parameters remain in effect for the duration of the IPL, or until they are changed using the SETGRS command. You can change the RNLs without having to reinitialize the entire complex. Use the SET GRSRNL command to accomplish this change. See “Changing the RNLs for a ring” on page 149 for more information.

As with all system parameters, there are several ways you can specify the global resource serialization parameters. To minimize operator intervention during IPL, it is generally best to specify GRS, GRSRNL, and GRSCNF in IEASYSxx or take the default values.

You can specify GRS=TRYJOIN in IEASYSxx on each system in the sysplex. If a system is IPLed and there is no complex, that system will start one. If a complex exists, the system joins the complex. The system that starts the complex builds a one-system ring and issues a message stating that a complex is active. As each additional system IPLs, an active system processes the request to join the complex. Global resource serialization writes messages to the system log on the joining system that identify the system that is assisting the joining system.

Global resource serialization also writes messages to the system logs on all systems in the complex to indicate that a new system is joining the ring. These messages indicate that the IPLs are proceeding normally and that the global resource serialization ring can process requests for global resources.

Normal operations in a sysplex

Once the complex is built, it requires little, if any, operator intervention. If a problem occurs, either global resource serialization or some other system component will detect the problem and issue messages that describe it before the operator could notice it.

For example, some of the error messages that global resource serialization issues indicate damage to resources or to the resource control blocks. These messages are ISG031E, ISG032E, ISG033E, ISG034E and ISG035E. The problem that causes any of these messages can also cause the job requesting the resource to terminate abnormally. If the damage is extensive, the problem can cause multiple jobs to terminate abnormally, requiring the system to IPL again to restore the control blocks. This problem is, of course, only one example of a problem that can force a system in the ring to IPL again.

During normal processing, operators can use system commands to monitor and control global resource serialization. The system commands related to global resource serialization are:

- DISPLAY GRS, which displays the status of each system in the complex
- VARY XCF,sysname,OFFLINE, which removes a system from the sysplex. (Any action that removes a system from the sysplex also removes it from the global resource serialization complex.) See *z/OS MVS Setting Up a Sysplex* for more information.
- SET GRSRNL, which changes the RNLs dynamically.
- SETGRS, which can change such parameters as TOLINT, RESMIL, ENQMAXA, and ENQMAXU. See *z/OS MVS System Commands* for more information.

Displaying ring status

The DISPLAY GRS (D GRS) command shows the state of each system in the complex. Note that D GRS shows system status only as it relates to the global resource serialization ring. D GRS does not reflect how well a system is running.

You can also use D GRS to display the local and global resources requested by the systems in the ring, contention information, the contents of the RNLs, and jobs that are delaying or suspended by a SET GRSRNL command. These uses are described in *z/OS MVS System Commands*.

You can issue D GRS from any system in the ring and at any time after the ring has been started. The D GRS display shows the status of the ring from that system's point of view; thus, the displays issued from different systems might show different results. Figure 71 on page 149 shows an example of the information D GRS produces and explains the values that can appear in each field.

```
18.40.07  ISG343I  18:40:06  GRS STATUS 340
SYSTEM    STATE                SYSTEM    STATE
 SYS2      ACTIVE               SYS1      ACTIVE
 SYS3      QUIESCED             SYS4      QUIESCED
```

SYSTEM
The name of the system.

STATE
The state of the system at the time when the command was issued.

ACTIVE
The system is part of the ring and is actively participating in global resource serialization. ACTIVE is the normal condition.

QUIESCED
The system is temporarily suspended from the ring, in response to a ring disruption. The system does not have current information about global resources and is not currently processing global resource requests. Users of global resources retain ownership, but any users who try to obtain or free a global resource are suspended. The system will restart and become active as soon as it is able.

INACTIVE
The system is not part of the ring. INACTIVE appears when a ring disruption has occurred. The system has current information about global resources but is not currently processing global resource requests. Users of global resources retain ownership, but any users who try to obtain or release a global resource are suspended. Multiple systems can be INACTIVE, and an inactive system can restart the ring. A system will restart the ring as soon as possible.

JOINING
The system is joining the ring as part of its IPL process.

RESTARTING
The system is re-entering the ring after a ring disruption.

ACTIVE+VARY
The system is executing an internal command.

ACTIVE+WAIT
The system is waiting to process an internal command.

MIGRATING
The system is in the process of migrating from a ring to a star, as a result of issuing a SETGRS command. The system suspends all tasks from obtaining global resources.

Note: GQSCAN does not work during migration from a ring complex to a star complex.

Figure 71. D GRS Explanation (Complex Matches Sysplex)

As a part of your planning, decide what to do if a system remains quiesced or inactive for an extended period of time. See Appendix B, “Recovery actions for global resource serialization,” on page 203 to determine what recovery operations you should perform.

Changing the RNLs for a ring

You can dynamically change the RNLs that global resource serialization uses, as long as the sysplex matches the complex. If your complex has any systems that are not in the sysplex, the change will not take effect. In addition, any single system complex can dynamically change its RNLs.

To change the RNLs currently being used by global resource serialization, set up the GRSRNLxx parmlib members with the new RNLs. Next, issue the SET GRSRNL command on a system that has access to those members. The new RNLs are then communicated to all systems in the complex. Keep in mind that you can not use SET GRSRNL=EXCLUDE or issue SET GRSRNL=xx in a complex already using GRSRNL=EXCLUDE. See Chapter 2, “Selecting the data,” on page 17 for details.

Note: Even though only one system needs the updated parmlib members to start the change, be sure to copy the updated GRSRNLxx parmlib members to each system's parmlib. Any system that needs to can then IPL again into the same complex. Otherwise, the change will be in effect only for the duration of the IPL.

Global resource serialization ensures that the integrity of all resources is maintained throughout the RNL change. In particular, before an RNL change can complete, special processing might be performed if any jobs are using the resources that are different in the old and new RNLs. These resources are known as *affected* resources. Jobs issuing new requests for these resources are suspended until the RNL change

is complete. These are known as *suspended jobs*. The following message is issued on each system in the complex that has suspended one or more jobs:

```
ISG210E RNL CHANGE WAS INITIATED BY SYSTEM sysname
        SOME JOBS ARE BEING SUSPENDED UNTIL RNL CHANGE COMPLETES.
```

If any job currently holds one or more of the affected resources, the change is delayed until all of the affected resources are freed. Jobs holding an affected resource (and thereby delaying the RNL change) are *delaying jobs*. When jobs are holding affected resources and delaying the change, the following messages are issued on whichever console originated the RNL change:

```
ISG219E RNL CHANGE WAITING FOR RESOURCES TO BE FREED.
        TO LIST DELAYING JOBS, USE ROUTE SYSNAME,DISPLAY GRS,DELAY.
        TO LIST SUSPENDED JOBS, USE ROUTE SYSNAME,DISPLAY GRS,SUSPEND.
ISG220D REPLY C TO CANCEL RNL CHANGE COMMAND, OR S FOR SUMMARY OF RNL
        CHANGE PROGRESS.
```

The DISPLAY GRS,DELAY (D GRS,DELAY) operator command lists the jobs that hold affected resources and are causing the change to be delayed. The jobs listed might release the affected resources normally, or they can be cancelled at the discretion of the installation. Once these jobs release the affected resources, the RNL change completes.

The DISPLAY GRS,SUSPEND (D GRS,SUSPEND) operator command lists the jobs that are being suspended due to the RNL change. The jobs listed will remain suspended until the RNL change completes, or until the RNL change is cancelled.

Replying to message ISG220D with an S produces a summary of the RNL change progress. This summary indicates the number of jobs on each system that are delaying or are suspended by the RNL change. Replying to message ISG220D with a C causes the RNL change to be cancelled.

If the operator chooses not to respond to message ISG220D with a C, the change will take place when all delaying jobs release the affected resources.

There may be instances where the operator must either cancel the RNL change command or cancel jobs that hold the affected resources:

1. A job that is not cancellable is holding affected resources for a long time.
2. A job holding an affected resource cannot DEQ that resource because it is suspended by global resource serialization pending a new ENQ for another affected resource, or else the job is waiting for some other work in the system that has issued an ENQ for an affected resource and has become suspended.
3. A job that is suspended by the RNL change is considered more important than the RNL change.

See *z/OS MVS System Commands* for more information about the DISPLAY GRS and SET GRSRNL commands.

Operating a mixed complex (complex does not match sysplex)

In general, the messages, and thus your operational planning, focus on three areas:

- building the complex
- normal operations
- recovery operations

The design of your complex affects all three areas. See Appendix B, “Recovery actions for global resource serialization,” on page 203 for information about planning your recovery operations. Chapter 8, “Designing a ring complex,” on page 119 describes these design considerations in detail, but the following list summarizes the most important recommendations:

- Use an ESCON channel operating in basic mode, rather than CTC adapters, to connect the systems.
- Design a fully-connected complex, one where each system has at least one link to every other system.

Note: You can, of course, operate a complex that is not fully connected. However, a minimal configuration can create availability, operations, and performance problems.

- Provide alternate links for each connection. Alternate links make recovery from a ring disruption easier. In addition, MVS/ESA SP Versions 3 and above can use the alternate link to send the ring acceleration signal.
- Provide a backup IBM 3088 to increase availability and eliminate the 3088 as a single point of failure.
- In GRSCNFxx, specify RESTART(YES) and REJOIN(YES) to minimize operator intervention during recovery.
- Tune the TOLINT value to meet your installation's needs. The TOLINT value determines the maximum length of time required to detect a ring disruption.
- On any pre-MVS/ESA SP Version 4 system, tune the RESMIL value to meet your installation's needs. The RESMIL value determines the minimum length of time the RSA-message spends in each system. MVS/ESA SP Version 4 and later systems tune the RSA-message residency time automatically.

Figure 72 on page 151 shows a sample configuration diagram of a four-system mixed complex. The system is fully-connected, and there are alternate links for all connections. Examples throughout this chapter use this configuration.

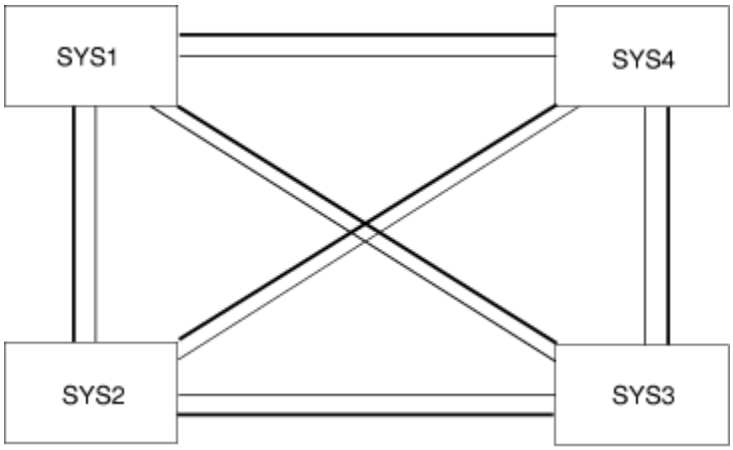


Figure 72. Sample mixed complex configuration

Building the complex

The process of building a global resource serialization complex can have two phases: a configuration check and the IPL of the systems.

Configuration check

Before an operator actually IPLs a system that is to start or join a global resource serialization complex, ensure that the operator verifies that the CTC link connections and the shared resource (such as DASD) connections are correct.

If the shared resource connections are incorrect, a serious data integrity exposure could occur. This exposure occurs when systems in the complex are serializing access to a global resource by means of an ENQ macro with a scope of SYSTEMS. For example, if the RESERVE conversion RNL for the complex contains an entry for a resource, this entry causes global resource serialization on each system in the complex to suppress the reserve for that resource. If a system outside the complex can use a reserve to access the same resource at the same time, the resource could be damaged.

IPL

There are three system parameters (GRS, GRSCNF, and GRSRNL) that indicate to MVS at IPL time that a system is to be part of a global resource serialization complex. In a mixed complex, these parameters remain in effect for the duration of the IPL; the only way to change a value is to IPL the system again.

See “Transitions to and from a mixed complex” on page 161 for information about the effects of changing from a mixed complex to a complex matches sysplex environment.

As with all system parameters, there are several ways you can specify the global resource serialization parameters. To minimize operator intervention during IPL, it is generally best to specify GRS, GRSRNL, and GRSCNF in IEASYSxx. However, the best way to specify the GRS= system parameter is a less clear-cut choice:

1. If the same system always starts the complex, you can place GRS=START in IEASYSxx for that system and place GRS=JOIN explicitly or by default in IEASYSxx for the other systems.
2. You can tell the operators to enter GRS=START or GRS=JOIN at the console during IPL.
3. You can place GRS=JOIN either explicitly or by default in IEASYSxx for all of the systems. When you make this choice, the system that actually is to start the complex IPLs with GRS=JOIN. Global resource serialization issues a message stating that there is no active complex, followed by a prompting message. The operator can then respond START to the prompting message, but the system-operator interaction requires extra time.
4. You can specify GRS=TRYJOIN in IEASYSxx for all of the systems that are operating MVS/ESA SP Version 4 or later. If a global resource serialization complex exists, the system joins the complex. If the complex does exists, the system joins the existing complex. If the system does not find any active global resource serialization complex systems, the system will start the complex.

It is both neater and more efficient to have the operator of the system that is to start the mixed complex explicitly override the default by entering GRS=START in response to the SPECIFY SYSTEM PARAMETERS message. This procedure, which IBM recommends, is especially useful when different systems might start the complex at different times. One operator explicitly specifies GRS=START; the others use the default GRS=JOIN in IEASYSxx.

Whatever way you choose to specify GRS=START, the effect is the same. The system that starts the complex builds a one-system ring and issues a message stating that a complex is active. As each additional system IPLs with GRS=JOIN, an active system processes the request to join the complex. When a system IPLs with GRS=JOIN, global resource serialization issues messages on the joining system that identify the system that is assisting the joining system. Global resource serialization also issues messages on all systems in the complex to indicate that a new system is joining the ring. These messages indicate that the IPLs are proceeding normally and that the global resource serialization ring can process requests for global resources.

One major reason for having the systems IPL explicitly with GRS=START for the starting system and GRS=JOIN for any joining system is to make sure that two potentially critical messages appear only in abnormal situations. These messages are:

```
ISG005I GRS START OPTION INVALID - SYSTEM sysname EXISTS IN A GRS COMPLEX
```

This message, followed by a prompting message, can occur when the system IPLs with GRS=START after another system has already started the complex. It thus might be a “normal” or expected message, and the operator can respond JOIN to the prompting message. However, it can also occur when there is a serious problem, such as a combination of errors having created multiple independent complexes using incorrectly-connected CTC links or duplications of the same system name.

```
ISG006I GRS JOIN OPTION INVALID - NO ACTIVE GRS SYSTEM
```

This message, also followed by a prompting message, can occur when the system IPLs with GRS=JOIN before another system has started the complex. It thus might be a “normal” or expected message, and the operator can respond START to the prompting message. However, this message can also occur when there are some very serious problems, such as a combination of errors having created either multiple independent complexes using incorrectly-connected CTC links.

If systems IPL explicitly with GRS=START or GRS=JOIN, the operator can treat each occurrence of either of these messages as a potentially serious error rather than something that is usually normal. There is no chance the operator can think the serious error is a normal condition.

Normal operations in a mixed complex

Once the complex is built, it requires little, if any, operator intervention unless a problem disrupts ring processing or the ring requires reconfiguration for some other reason. If a problem occurs that is not related to ring processing, either global resource serialization or some other system component will detect the problem and issue messages that describe it before the operator could notice it.

For example, some of the error messages that global resource serialization issues indicate damage to resources or to the resource control blocks, rather than a problem with ring processing. These messages are ISG031E, ISG032E, ISG033E, ISG034E and ISG035E. The problem that causes any of these messages can also cause the job requesting the resource to terminate abnormally. If the damage is extensive, the problem can cause multiple jobs to terminate abnormally, requiring the system to IPL again to restore the control blocks. This problem is, of course, only one example of a problem not directly related to ring processing that can force a system in the ring to IPL again.

During normal processing, as well as during recovery from a ring disruption, operators use system commands to monitor and control global resource serialization. The system commands related to ring processing are:

- DISPLAY GRS, which displays the status of each system in the complex. D GRS is the operator's primary way of checking ring processing and determining the source of problems. See “Displaying ring status” on page 153.
- VARY GRS with the QUIESCE operand, which allows the operator to remove a system from the ring. See “Quiescing a system” on page 156.
- VARY GRS with the PURGE operand, which allows the operator to remove a system from the complex. See “Purging a system” on page 157.
- VARY GRS with the RESTART operand, which allows the operator to restart a quiesced system or an inactive system. See “Restarting a system” on page 159.
- VARY *devnum*, which allows the operator to bring a link online or take a link offline. See “Controlling CTC links” on page 161.
- VARY CTC,OFFLINE,FORCE, which permanently removes a CTC from use by global resource serialization and releases it for other uses. See “Migrating an existing complex into a sysplex” on page 163 for an example of how and when to use this command.

It is a good practice, whenever possible, to issue all VARY commands for the ring from the same system; this practice simplifies operations procedures, especially during recovery.

If your mixed complex contains systems in an XCF multisystem sysplex, additional system commands for those systems related to ring processing are:

- VARY XCF,sysname,OFFLINE, which removes a system from the sysplex. (Any action that removes a system from the sysplex also removes it from the global resource serialization complex.) See *z/OS MVS Setting Up a Sysplex* for more information.

Displaying ring status

The DISPLAY GRS (D GRS) command shows the state of each system in the complex and the status of the links that connect the systems. Note that D GRS shows system status only as it relates to the global resource serialization ring. D GRS does not reflect how well a system is running generally; for example, MVS on a system shown as QUIESCED or INACTIVE in the global resource serialization complex might run successfully for quite a while.

You can also use D GRS to display the local and global resources requested by the systems in the ring, contention information, or the contents of the RNLs. These uses are described in *z/OS MVS System Commands*.

You can issue D GRS from any system in the ring and at any time after the ring has been started. The D GRS display shows the status of the ring from that system's point of view; thus, the displays issued from different systems might show different results. Figure 73 on page 154 shows an example of the information D GRS produces and explains the values that can appear in each field.

D GRS is most useful, however, when a ring failure has occurred. The information displayed can help the operator to make informed decisions about the cause of an error and the correct response to the problem. Note that D GRS does not diagnose a problem; it simply reports status. Figure 74 on page 155 shows how an operator can use D GRS to determine the cause of a problem with ring processing.

```
18.40.07  ISG343I  18:40:06  GRS STATUS 340
SYSTEM    STATE       COMM    SYSTEM    STATE      COMM
SYS2      ACTIVE              SYS1      ACTIVE     YES
SYS3      QUIESCED    YES     SYS4      QUIESCED   NO

LINK      STATUS      TARGET  LINK      STATUS     TARGET
 220       ALTERNATE   SYS3    420       ALTERNATE   SYS3
 221       ALTERNATE   SYS3    421       ALTERNATE   SYS3
 224       QUIET       SYS4    424       QUIET       SYS4
 225       QUIET       SYS4    425       QUIET       SYS4
 228       ALTERNATE   SYS1    428       ALTERNATE   SYS1
 229       IN-USE      SYS1    429       ALTERNATE   SYS1
```

Figure 73. D GRS Explanation (Mixed Complex)

SYSTEM
The name of the system.

STATE
The state of the system at the time when the command was issued. There are seven possible states:

ACTIVE
The system is part of the ring and is actively participating in global resource serialization. ACTIVE is the normal condition. The system accepts all commands related to ring processing.

QUIESCED
The system is temporarily suspended from the ring, in response to either a ring disruption or operator command. The system does not have current information about global resources and is not currently processing global resource requests. Users of global resources retain ownership, but any users who try to obtain or free a global resource are suspended. The system remains quiesced, and the users remain suspended, until the system is restarted. **Note:** Access to local resources is not affected, but an attempt to cancel a job might not succeed if a global resource is involved.

INACTIVE
The system is not part of the ring. INACTIVE appears when a ring disruption has occurred. The system has current information about global resources but is not currently processing global resource requests. Users of global resources retain ownership, but any users who try to obtain or release a global resource are suspended. Multiple systems can be INACTIVE, and an inactive system can restart the ring. An inactive system remains inactive until any system in the complex is restarted. **Note:** Access to local resources is not affected, but an attempt to cancel a job might not succeed if a global resource is involved.

JOINING
The system is joining the ring as part of its IPL process.

RESTARTING
The system is re-entering the ring as a result of a RESTART command.

ACTIVE+VARY
The system is executing a VARY GRS command.

ACTIVE+WAIT
A VARY GRS command was issued, but it is waiting because another VARY GRS command is now executing. When ACTIVE+WAIT appears, another system normally shows ACTIVE+VARY.

COMM
An indication of whether or not the system has responded to a request for status. YES indicates that the system shown can communicate with the system issuing D GRS. NO indicates that there is no communication link, the system is temporarily stopped, or the system has failed. If NO appears, the state shown for the system might not be accurate. The field is blank for the system that issued D GRS.

LINK
The address of each CTC data link defined for global resource serialization on the system.

STATUS
The status of the link. There are four possible states:

IN-USE
The link is a primary link now being used to send the RSA-message from one system to another. IN-USE appears only for a link that connects active systems.

ALTERNATE
The link is an alternate. If a primary link fails, an alternate link can automatically replace it. An alternate link might be used to send the ring acceleration signal, which D GRS does not report; if an alternate link used for ring acceleration replaces a failed primary link, it can no longer send the ring acceleration signal.

DISABLED
The link is not physically connected or was taken offline because of an error.

QUIET
The link does not have any apparent problems, but the system it connects to did not respond to the request for status.

TARGET
The name of the system that last responded from the other side of the link. The field is blank when the link has been disabled since the IPL of the system or when the system did not respond to the request for status.

```
18.40.07  ISG020I  18:40:06  GRS STATUS 340
SYSTEM    STATE       COMM    SYSTEM    STATE      COMM
 SYS2      ACTIVE              SYS1      ACTIVE     YES
 SYS3      QUIESCED    YES     SYS4      QUIESCED   NO

LINK      STATUS      TARGET  LINK      STATUS     TARGET
 220       ALTERNATE   SYS3    420       ALTERNATE   SYS3
 221       ALTERNATE   SYS3    421       ALTERNATE   SYS3
 224       QUIET       SYS4    424       QUIET       SYS4
 225       QUIET       SYS4    425       QUIET       SYS4
 228       ALTERNATE   SYS1    428       ALTERNATE   SYS1
 229       IN-USE      SYS1    429       ALTERNATE   SYS1
```

Figure 74. Using D GRS to Analyze a Problem

The COMM field for system SYS2 is blank; the D GRS command was issued on system SYS2. The display shows the following:

1. System SYS2 and system SYS1 are active; they are processing global resource requests.
2. System SYS2 and system SYS1 are using link 229 to send the RSA-message. Link 229 is a primary link; its status is IN-USE.
3. All other links between system SYS2 and system SYS1 (228, 428, 429) are shown as ALTERNATE. One of these links might be sending the ring acceleration shoulder-tap.
4. The status of system SYS3 is QUIESCED. It is not part of the ring and is not processing global resource requests. YES appears in the COMM field for system SYS3, indicating that system SYS3 responded to the request for status. MVS is still active on system SYS3.
5. Because system SYS3 is quiesced, all of its links to system SYS2 (220, 221, 420, 421) are marked as ALTERNATE.
6. The status of system SYS4 is QUIESCED. Like system SYS3, it is not part of the ring and is not processing global resource requests. NO appears in the COMM field for system SYS4, indicating that system SYS4 (unlike system SYS3) did not respond to the request for status. Also, all links between system SYS4 and system SYS2 (224, 225, 424, 425) are marked as QUIET. System SYS4 is the source of the problem.

   D GRS can report a problem but it cannot diagnose the reason for the problem. Possible reasons for the problem shown in this example are:

   a. System SYS4 is temporarily stopped. Perhaps the system has stopped to take a dump, or MVS might be in a spin loop.
   b. System SYS4 has failed.

c. All links have failed on the system SYS4 side. This possibility is unlikely; an I/O error on a link is normally detected by both systems (SYS2 and SYS4 in this case), and the status of a failed link normally appears as DISABLED.

Quiescing a system

In the context of ring processing, quiescing a system means removing it from the ring. The system can continue to run, but it cannot access or free global resources. Quiescing a system is normally done as the first step in removing a system from the complex.

To quiesce a system, the operator on the system to be quiesced (or on any active system that can communicate with the system to be quiesced) can issue the VARY GRS(sysname),QUIESCE command. A quiesced system is no longer part of the ring; it is, however, still known to the other systems in the ring.

Quiescing a system can slow down performance because no global resources are released:

- On the quiesced system, any task that controls any global resources retains control of those resources, and any task that is waiting for a global resource continues to wait.

  Programs on the quiesced system do continue to process, but only until they need to access or free a global resource. For example, a program that had exclusive control of a global resource can finish with the resource; however, the quiesced system cannot completely process the DEQ for the resource or tell the active systems that the resource is now available.
- On the active systems, any task that needs a global resource held by a task on the quiesced system continues to wait. This condition continues until the quiesced system either rejoins the ring or is purged from the ring.

Thus, quiescing a system is an action that you should take very seldom and for as short a time as possible. It might, for example, be part of the process of physical reconfiguration. When it is necessary to quiesce a system, the operator must first bring work on the system to an orderly shutdown. This procedure minimizes ring performance problems and data integrity exposures if the system is to be purged from the ring.

Global resource serialization, in response to the VARY GRS(sysname),QUIESCE command, places the target system in a quiesced state and forms a new ring without the quiesced system. The operators can use the global resource serialization messages (ISG011I and ISG013I) and, if necessary, D GRS, to verify that the system is now quiesced.

Global resource serialization can build a new ring without the quiesced system only when the links needed for the new ring are available. When the complex is fully-connected and alternate links are available, rebuilding the ring without the quiesced system is not a problem. If a link that it needs to build a new ring without the quiesced system is missing, global resource serialization rejects the VARY GRS(sysname),QUIESCE command. After enabling the required links, the operator can enter the command again. Once the system is quiesced, it can either rejoin the ring, which does not require a reIPL, or be purged from the ring.

Quiesce Messages

The global resource serialization messages related to quiescing a system include ISG011I, ISG012I, ISG013I, ISG014I, and ISG015I.

Example — Quiescing a System

Using the four-system complex shown earlier in Figure 72 on page 151, assume that it is necessary to stop SYS4 temporarily.

On SYS4, the operator would take the following steps:

1. Bring the work to an orderly shutdown by taking such actions as stopping the subsystems and terminating jobs.

2. Issue VARY GRS(*),QUIESCE or VARY GRS(SYS4),QUIESCE. (An operator on any active system could also issue the second command.)

   On SYS4, the following message appears:

   ```
   ISG012I QUIESCE REQUEST PASSED TO SYSTEM SYS2
   ```

   This message indicates that global resource serialization has accepted the command and that SYS2 will assist in the quiesce process. SYS2 is the assisting system, and SYS4 is the target system. (This message appears only when the operator on the target system issues the VARY command.)

   On SYS2 and SYS4, the following message indicates that global resource serialization has started to quiesce the target system (SYS4):

   ```
   ISG011I SYSTEM SYS4 - QUIESCING GLOBAL RESOURCE SERIALIZATION
   ```

   On all systems, the following message appears after the system has been successfully quiesced.

   ```
   ISG013I SYSTEM SYS4 - QUIESCED GLOBAL RESOURCE SERIALIZATION
   ```

3. Once the quiesce is successful, the operator on SYS4 can stop the system.

Issuing D GRS would show the state of SYS4 as QUIESCED. The COMM field would show YES, indicating that communication still existed, and all links to SYS4 would appear as ALTERNATE.

Purging a system

To purge a system, the operator on any active system issues the VARY GRS(sysname),PURGE command. Purging a system is normally needed when:

- The operator must remove the system from the ring for a long period of time (perhaps for preventive maintenance).
- The system is no longer needed in the ring (perhaps because of a configuration change).
- The system has failed and must reIPL.

In response to the purge command, each active system in the ring deletes all information related to the target system, including its requests for global resources, its control of global resources, and any appearance of its system name. In short, global resource serialization removes all indications that the purged system was ever a part of the complex.

If users on the purged system held global resources, these resources are freed. Message ISG018I, issued to SYSLOG, describes these resources. Your installation must plan in advance to investigate the state of resources as part of the process of removing a system from the complex.

The purged system must reIPL with GRS=JOIN to rejoin the ring. Also, a system that has been part of an active ring cannot reIPL with GRS=JOIN unless it has first been purged. Until the system is purged, global resource serialization knows about it and rejects its attempt to rejoin the ring because it detects a duplicate system name.

Because purging the system does not stop MVS, stop the system before purging it so that the system cannot continue to access shared resources. This procedure prevents a potential data integrity exposure. For example, assume that a job on a purged system, SYS1, was updating a resource and did not complete before SYS1 was purged. Purging SYS1 frees the resource and makes it available to other requesters, but, unless SYS1 is stopped, the job on SYS1 can continue to update the resource.

The target system — the system to be purged — can be an active system or a quiesced system. When the target system is an active system, global resource serialization issues a message to remind the operator that the system is active; that is, the operator must bring the work to an orderly shutdown before proceeding. See “Example — Purging an Active System” on page 158 for an example of purging an active system. See “Example — Purging a Quiesced System” on page 158 for an example of purging a quiesced system.

Example — Purging an Active System

Using the four-system complex shown earlier in Figure 72 on page 151, assume that it is necessary to purge SYS1 from the ring. SYS1 is an active system. The operator on SYS2 issues the purge command; thus, SYS2 is the assisting system. The required steps are:

1. On SYS1, the operator, to avoid potential data integrity exposures, must bring the workload to an orderly shutdown by taking such actions as stopping the subsystems and terminating jobs.
2. On SYS2, the operator then issues VARY GRS(SYS1),PURGE. The following messages appear:

   ```
   ISG100E SYSTEM SYS1 IS STILL AN ACTIVE GRS SYSTEM
   ISG101D CONFIRM PURGE FOR ACTIVE SYSTEM SYS1 - REPLY NO OR YES
   ```

   In this example, the operator can safely reply YES. These messages remind the operator that SYS1 is still active; purging it from the ring might create a data integrity exposure.
3. On SYS2, the operator replies YES to message ISG101D.
4. On all active systems, the following messages appear:

   ```
   ISG011I SYSTEM SYS1 - QUIESCING GLOBAL RESOURCE SERIALIZATION
   ISG013I SYSTEM SYS1 - QUIESCED GLOBAL RESOURCE SERIALIZATION
   ISG011I SYSTEM SYS1 - BEING PURGED FROM GRS COMPLEX
   ISG013I SYSTEM SYS1 - PURGED FROM GRS COMPLEX
   ```

5. On SYS1, the operator must stop the system. At this point, SYS1 is no longer known to global resource serialization. The ring consists of SYS2, SYS3, and SYS4. To rejoin the ring, SYS1 must reIPL with GRS=JOIN.

   While purging SYS1, global resource serialization might detect a potential data integrity exposure. After purging an active system, the operator should follow the installation's procedures for resolving a data integrity exposure, such as contacting the system programmer responsible for investigating the state of resources. More information about the problem normally appears in SYSLOG, where the following message appears to describe any resources that might have been damaged by purging the system:

   ```
   ISG018I REQUESTORS FROM SYSTEM SYS1 HAVE BEEN PURGED FROM RESOURCES
             NAMED xxxx,yyyy
   ```

Example — Purging a Quiesced System

Using the four-system complex shown earlier in Figure 72 on page 151, assume that it is necessary to purge SYS1 from the ring. SYS1 is a quiesced system. The operator on SYS2 issues the purge command; thus, SYS2 is the assisting system. The required steps are:

1. On SYS2, the operator issues VARY GRS(SYS1), PURGE. The following messages appear:

   ```
   ISG016I SYSTEM SYS1 OWNS OR IS WAITING FOR GLOBAL RESOURCES
   ISG017D CONFIRM PURGE REQUEST FOR SYSTEM SYS1 - REPLY NO OR YES
   ```

   These messages indicate that users on SYS1 still own, or are waiting for, global resources. The operator on SYS2 should reply NO unless the operator knows the work on SYS1 has been shut down. The operator on SYS1 must shut the work down. Replying YES to message ISG017D might create a data integrity exposure. When the work on SYS1 has been shut down, the operator on SYS2 can reissue the VARY GRS(SYS1), PURGE command.
2. On SYS2, the following message appears:

   ```
   ISG011I SYSTEM SYS1 - BEING PURGED FROM GRS COMPLEX
   ```

3. On all active systems, the following message appears:

   ```
   ISG013I SYSTEM SYS1 - PURGED FROM GRS COMPLEX
   ```

4. On SYS1, the operator must stop the system. At this point, SYS1 is no longer known to global resource serialization. The ring consists of SYS2, SYS3, and SYS4. To rejoin the ring, SYS1 must reIPL with GRS=JOIN.

   While purging SYS1, global resource serialization detected a potential data integrity exposure, indicated by message ISG016I. When this message appears, the operator should follow the installation's procedures for resolving the problem, such as contacting the system programmer responsible for investigating the state of resources. More information about the problem normally appears in SYSLOG, where the following message appears to describe any resources that might have been damaged by purging the system:

   ```
   ISG018I REQUESTORS FROM SYSTEM SYS1 HAVE BEEN PURGED FROM RESOURCES
            NAMED xxxx,yyyy
   ```

Restarting a system

Specifying the automatic recovery options, RESTART and REJOIN, means that operators seldom need to intervene to restart a system. When necessary, the operator on any system in the complex can issue the VARY GRS(sysname),RESTART command to bring the target system back into the ring. The target system can be either an inactive system or a quiesced system.

Restarting a system is the opposite of quiescing a system. VARY GRS(sysname),QUIESCE suspends global resource serialization and removes the target system from the ring. VARY GRS(sysname),RESTART resumes global resource serialization and brings the system back into the ring. The restarted system, because it is now part of the ring, can release any resources already freed by users and resume processing requests for global resources.

The VARY GRS(sysname),RESTART command can bring back into the ring a system that the operator had quiesced or a system that had become inactive or quiesced as a result of a ring disruption. After issuing the command, the operator can use D GRS to verify that the target system is now part of the ring.

There are three ways to issue the command. Issuing the first or second form of the command when all systems are inactive makes the specified system active, while all others become quiesced. The active system can then bring the other systems back into the ring.

1. VARY GRS(sysname),RESTART — issued from any active system to bring the named system back into the ring.
2. VARY GRS(*),RESTART — issued on the system to be restarted to bring that system back into the ring.
3. VARY GRS(ALL),RESTART — issued on any inactive system to restart the ring. Ensure that operators:
   - Allow automatic recovery processing to complete before issuing this command.
   - Issue this command only when there is no active system in the ring and at least one inactive system.
   - Issue this command only once during the process of recovery from a ring disruption.

   The command changes the state of all inactive systems from inactive to active. The command will not bring back systems that were quiesced before the ring disruption, such as those that the operator quiesced specifically.

Issuing any form of the command when all systems are quiesced invokes the reactivate function, which is designed for very unusual recovery situations.

To avoid the possibility of split rings, ensure that the operators issue VARY GRS(ALL),RESTART with extreme care.

Split rings

Split rings can occur when more than one operator tries to restart the ring at the same time, causing the ring to split into multiple independent rings, each able to grant access to global resources at the same time. Split rings create a severe data integrity exposure. Actions you can take to avoid split rings include:

- Ensure that an operator issues VARY GRS(ALL),RESTART only from an inactive system.

- Ensure that only one operator issues VARY GRS(ALL), RESTART
- Provide alternate links.
- Specify RESTART(YES) whenever possible, which allows automatic restart and reduces operator intervention in recovery from a ring disruption.

An operator trying to restart the ring should always issue VARY GRS(ALL),RESTART to restart all of the systems rather than VARY GRS(sysname),RESTART or VARY GRS(*),RESTART to restart a specific system. When the operator issues VARY GRS(ALL),RESTART, global resource serialization issues the following messages:

```
ISG026I SYSTEM SYS2 MAY CREATE A SPLIT RING IF ANY OTHER GRS SYSTEM
          IS ACTIVE. VERIFY THAT NO GRS SYSTEM IS ACTIVE BEFORE
          CONFIRMING RESTART
  ISG027D CONFIRM RESTART RING FOR SYSTEM SYS2 - REPLY NO OR YES
```

Before replying to message ISG027D, the operator must issue D GRS and/or check with the other operators to verify that there are no active systems. If all other systems are inactive, the operator can safely reply YES to continue the restart. If any system is active, the operator must reply NO to avoid split rings.

For example, consider the two-system complex shown in Figure 75 on page 160. SYS1 is active, SYS2 is quiesced, and the communication link has failed. If the operator issues a restart command on SYS2, message ISG026I appears to warn the operator that split rings might occur, followed by a prompting message. If the operator replies YES to the prompt, SYS2 will create a ring of one system. Because SYS1 is also active and there is no communication, there are two one-system rings. Both rings can grant access to the same global resources, and neither system can rejoin the ring created by the other without an IPL. Note that global resource serialization does not force the reIPL; it is, however, required to resolve the data integrity exposure.

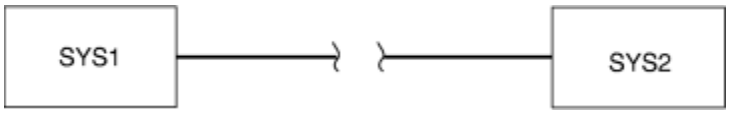


Figure 75. Two-System Ring with Link Failure

There are several ways to avoid split rings in this situation:

1. If the configuration includes an alternate link, and the alternate link has not failed, the problem does not occur; global resource serialization could use the alternate link and resume processing almost immediately.
2. If the operator issues the restart command on the active system, split rings do not occur. Instead, the following message appears:

   ```
   ISG014I VARY GRS RESTART REQUEST FOR SYSTEM SYS1 REJECTED -
             SYSTEM NOT RESPONDING
   ```

3. If the operator replies NO to the prompt following message ISG027D, split rings do not occur.

In the last two cases, message ISG026I or message ISG014I alert the operator to the actual problem; SYS1 and SYS2 cannot communicate. The operator could then respond correctly — fix the link problem, then reissue the VARY GRS(SYS2),RESTART command for the quiesced system.

Example — Restarting a System

Using the four-system complex shown in Figure 72 on page 151, assume that SYS4 has been quiesced but is now ready to rejoin the ring. An operator on any active system could restart SYS4, but assume that the operator on SYS2 is to handle the restart.

On SYS2, the operator issues VARY GRS(SYS4),RESTART.

On SYS2 and SYS4, the following message appears to indicate that the process of restarting SYS4 has begun:

```
ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
```

On all active systems, the following message appears to indicate that SYS4 is now part of the active global resource serialization ring:

```
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

If the operator on SYS4 issued the restart command, global resource serialization would pass the command to an active system for processing. If the system selected was SYS2, the following message would appear on SYS4:

```
ISG012I RESTART REQUEST PASSED TO SYSTEM SYS2
```

Controlling CTC links

To bring a CTC link online or take a CTC link offline, use the VARY command. All links that you want global resource serialization to use must be defined in the GRSCNFxx member.

To bring a defined link online, issue VARY devnum,ONLINE. In response to this command, the link comes online, and global resource serialization changes its status from DISABLED to ALTERNATE. It is then available to send the ring acceleration signal or to act as a backup for a primary link.

To take a defined link offline, issue VARY devnum,OFFLINE. In response to this command, the link goes offline, and global resource serialization changes its status from ALTERNATE to DISABLED. The specified link must be an alternate link; if an operator tries to take the primary link offline, global resource serialization rejects the command.

Transitions to and from a mixed complex

During normal operations, the actual operating environment may alternate between a mixed complex and a complex that matches a sysplex as systems join and are purged from the complex. For example, in a three system mixed complex where two systems are in the same sysplex, removing the non-sysplex system (by issuing VARY GRS PURGE) creates the complex equal sysplex environment.

Even in this transitional complex equal sysplex environment, many of the operational benefits described in “Operating a complex that matches a sysplex” on page 147 will exist. For example, all systems have automatic rebuild and rejoin characteristics, so the VARY GRS commands are not supported in this environment. In addition, you can issue the SET GRSRNL command to change the RNLs of the systems that are part of a sysplex, and then IPL the other systems with the updated RNLs.

When the non-sysplex system is re-IPLed into the complex, the operating environment becomes mixed again.

Chapter 10. Installing and tuning the complex

There are many possible ways to install a global resource serialization complex.

Installing the complex

You may decide to bring all of your systems down and reIPL them into a multisystem sysplex that matches the complex. You may choose to start a complex with a single system enabled for a sysplex and continue to use RESERVE/DEQ or job scheduling to serialize shared resources until you bring your remaining systems into the complex and sysplex. If you are planning your installation's first global resource serialization complex, the best approach is to plan to create a sysplex that matches the complex. See “Installing the complex” on page 163 for a list of considerations. *z/OS MVS Setting Up a Sysplex* provides detailed information on installing a sysplex.

However, if your installation is already using a global resource serialization complex, you will want to plan on moving one system at a time into a sysplex. See “Migrating an existing complex into a sysplex” on page 163 for a detailed migration plan.

You will also want to evaluate the performance of the complex as you progress and take actions that will minimize the performance cost of the availability benefits that a global resource serialization complex provides; see “Tuning the complex” on page 167.

Installing a new complex

How you actually install the global resource serialization complex depends — like every other decision related to global resource serialization — on the needs of your installation. Here are some installation considerations:

- Identify the resources that you must include in the RNLs to get benefits from the complex as quickly as possible.
- Define the long-term resource processing goals for your installation.
- Evaluate the data set naming conventions at your installation and begin making any changes required to make better use of global resource serialization.
- Evaluate the possibility of modifying existing applications to use global resource serialization more effectively.
- Determine how global resource serialization will affect the design of future applications.
- Prepare the initial educational and procedural material for the system operators.
- Leave the RESERVE conversion RNL empty if you plan to continue using RESERVE/DEQ for data set serialization until the complex matches the sysplex.
- Complete any shared DASD changes required to ensure that the complex does not share DASD volumes with systems outside the complex.

Migrating an existing complex into a sysplex

You can migrate an existing global resource serialization complex into a sysplex one system at a time. There are two ways you can accomplish the migration. The easier way is to add communication links for XCF to use in addition to the links already dedicated to global resource serialization. Providing the extra links allows you to keep your complex operational in the event that you encounter an unexpected situation and need to revert to the previous environment. Once your operating environment is stable after the migration, however, you can release the links that were dedicated to global resource serialization, freeing them for other uses.

Another way to migrate the systems in a complex into a sysplex requires releasing a link that is dedicated to global resource serialization and using that link for XCF, as well as providing every pair of systems in the complex with one additional link, also for XCF. The additional link between each pair of systems leaves one link dedicated to global resource serialization, enabling you to return to the previous environment with at least minimal link connections still dedicated to global resource serialization if an unexpected situation should make it necessary.

Note: You must use the VARY ctc,OFFLINE,FORCE command to release a CTC dedicated to global resource serialization before XCF can use the CTC.

The following example illustrates this migration technique. Figure 76 on page 164 represents a fully-connected global resource serialization complex before migration described in the example begins.

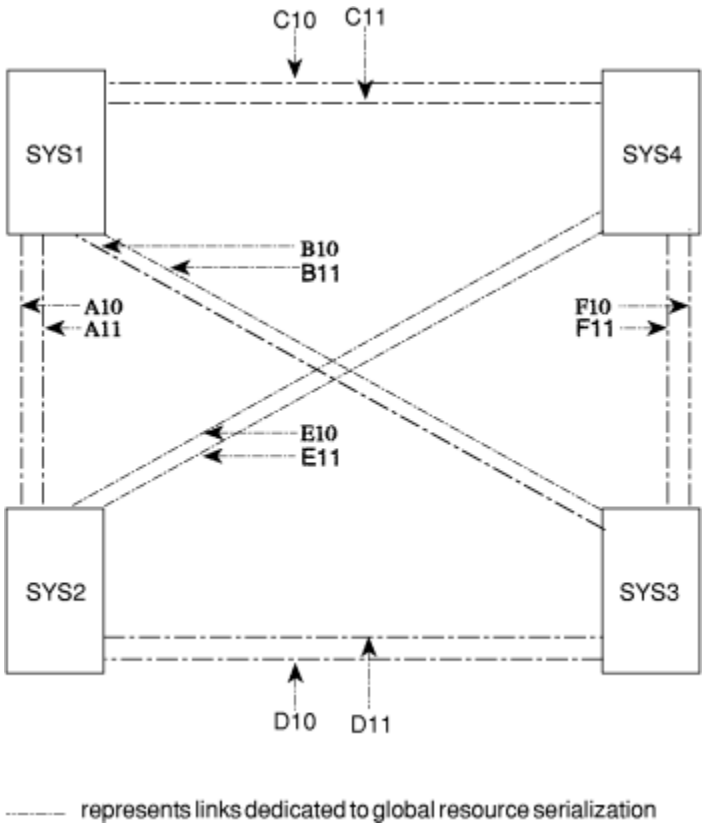


Figure 76. Global Resource Serialization Complex before Migration

The systems are migrating to the next release sequentially, beginning with SYS1.

1. **Migrating SYS1 into the sysplex**
   - All links remain dedicated to global resource serialization.
   - ReIPL SYS1 into the sysplex.
2. **Migrating SYS2 into the sysplex**
   - Remove one link (A11 in this example) from use by global resource serialization and make it available to XCF.
     - On SYS1, issue the VARY A11,OFFLINE,FORCE command so global resource serialization can no longer use the link. (Respond *free* to message ISG186D.)
     - On SYS1, issue the VARY A11,ONLINE command to make the link available to XCF.
     - On SYS1, issue the SETXCF START,PATHIN,DEVICE=(A11) command to let XCF begin using the link.
   - Make another link (A12 in this example) available to XCF.

- On SYS1, issue the SETXCF START,PATHOUT,DEVICE=(A12) command to let XCF begin using this new link.

- ReIPL SYS2 into the sysplex, specifying that XCF on SYS2 use links A11 and A12.

3. **Migrating SYS3 into the sysplex**
   - Remove one link (B11) between SYS1 and SYS3 from use by global resource serialization and make it available to XCF.
     - On SYS1, issue the VARY B11,OFFLINE,FORCE command so global resource serialization can no longer use the link. (Respond *free* to message ISG186D.)
     - On SYS1, issue the VARY B11,ONLINE command to make the link available to XCF.
     - On SYS1, issue the SETXCF START,PATHIN,DEVICE=(B11) command to let XCF begin using the link.
   - Make another link (B12 in this example) between SYS1 and SYS3 available to XCF.
     - On SYS1, issue the SETXCF START,PATHOUT,DEVICE=(B12) command to let XCF begin using this new link.
   - Remove one link (D11) between SYS2 and SYS3 from use by global resource serialization and make it available to XCF.
     - On SYS2, issue the VARY D11,OFFLINE,FORCE command so global resource serialization can no longer use the link. (Respond *free* to message ISG186D.)
     - On SYS2, issue the VARY D11,ONLINE command to make the link available to XCF.
     - On SYS2, issue the SETXCF START,PATHIN,DEVICE=(D11) command to let XCF begin using the link.
   - Make another link (D12) between SYS1 and SYS3 available to XCF.
     - On SYS2, issue the SETXCF START,PATHOUT,DEVICE=(D12) command to let XCF begin using this new link.
   - ReIPL SYS3 into the sysplex, specifying that XCF on SYS3 add and use links B11, B12, D11, and D12.
4. **Migrating SYS4 into the sysplex**
   - Remove one link (C11) between SYS1 and SYS4 from use by global resource serialization and make it available to XCF.
     - On SYS1, issue the VARY C11,OFFLINE,FORCE command so global resource serialization can no longer use the link. (Respond *free* to message ISG186D.)
     - On SYS1, issue the VARY C11,ONLINE command to make the link available to XCF.
     - On SYS1, issue the SETXCF START,PATHIN,DEVICE=(C11) command to let XCF begin using the link.
   - Make another link (C12) between SYS1 and SYS4 available to XCF.
     - On SYS1, issue the SETXCF START,PATHOUT,DEVICE=(C12) command to let XCF begin using this new link.
   - Remove one link (E11) between SYS2 and SYS4 from use by global resource serialization and make it available to XCF.
     - On SYS2, issue the VARY E11,OFFLINE,FORCE command so global resource serialization can no longer use the link. (Respond *free* to message ISG186D.)
     - On SYS2, issue the VARY E11,ONLINE command to make the link available to XCF.
     - On SYS2, issue the SETXCF START,PATHIN,DEVICE=(E11) command to let XCF begin using the link.
   - Make another link (E12) between SYS2 and SYS4 available to XCF.
     - On SYS2, issue the SETXCF START,PATHOUT,DEVICE=(E12) command to let XCF begin using this new link.

- Remove one link (F11) between SYS3 and SYS4 from use by global resource serialization and make it available to XCF.
  - On SYS3, issue the VARY F11,OFFLINE,FORCE command so global resource serialization can no longer use the link. (Respond *free* to message ISG186D.)
  - On SYS3, issue the VARY F11,ONLINE command to make the link available to XCF.
  - On SYS3, issue the SETXCF START,PATHIN,DEVICE=(F11) command to let XCF begin using the link.
- Make another link (F12) between SYS3 and SYS4 available to XCF.
  - On SYS3, issue the SETXCF START,PATHOUT,DEVICE=(F12) command to let XCF begin using this new link.
- ReIPL SYS4 into the sysplex, specifying that XCF on SYS4 add and use links C11, C12, E11, E12, F11, and F12.

Figure 77 on page 166 shows the same complex after migration, but before the links dedicated to global resource serialization are released.

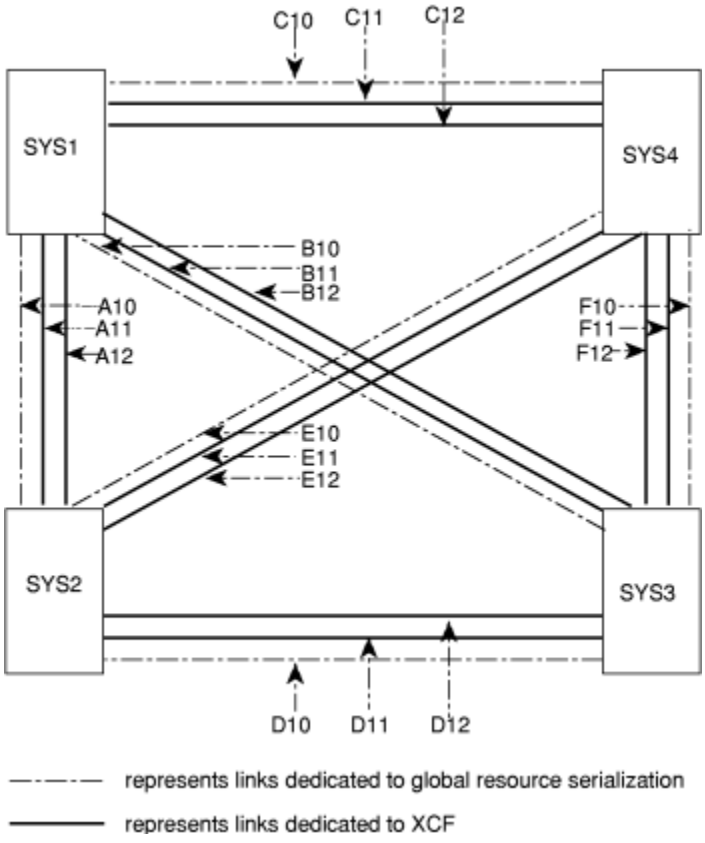


Figure 77. Global Resource Serialization Complex after Migration

The final step, once you are satisfied that your sysplex is performing normally, is to release the links that remain dedicated to global resource serialization. Use the VARY device,OFFLINE,FORCE command and respond *free* to message ISG186D for each link that you wish to free on each system. Using the sysplex shown in Figure 77 on page 166, you would issue the VARY device,OFFLINE,FORCE command:

- On SYS1 to release links A10, B10, and C10
- On SYS2 to release links A10, D10, and E10
- On SYS3 to release links B10, D10, and F10
- On SYS4 to release links C10, E10, and F10.

Tuning the complex

A global resource serialization complex increases the availability of systems and resources, and might improve system performance, because converting reserves can:

- Reduce the number of system interlocks that occur because of reserves
- Decrease contention for resources on volumes now serialized by use of a reserve
- Remove the need to use job scheduling to serialize access to resources on shared DASD volumes
- Avoid the data integrity exposure that occurs when a system reset prematurely ends a reserve
- Avoid the situation when one processor can monopolize a shared volume

A global resource serialization complex does, however, increase the system's use of the processor, channels and devices, and storage. This additional overhead occurs for two main reasons:

1. Every system in the ring processes every request for a global resource and maintains information about the status of every global resource.
2. The RSA-message used to pass information about global resource requests from one system to another requires processing overhead each time it makes a cycle around the ring. This overhead occurs even when the RSA-message is empty.

In addition, every task that requests a global resource is suspended while the ring processes the request. The time that a task is suspended while it is waiting for access to a global resource is the global resource request response time.

Thus, tuning a global resource serialization complex involves balancing your installation's need for an acceptable response time while minimizing the effect that ring processing has on overall system performance. Except for very small processors or systems that are already running close to capacity, the effect of ring processing is normally not significant. Thus, focus your tuning work on response time — the delay that individual global resource requesters encounter. Your installation must determine what response time is acceptable.

If the response time is acceptable, there is no need to tune the complex. If the response time is not acceptable, then you need to investigate the problem and take action to resolve, or at least minimize, the difference between the actual response time and the acceptable response time.

The design of your complex, described in detail in Chapter 8, “Designing a ring complex,” on page 119, is very closely connected to ring performance. Thus, many of the factors that affect ring performance are relatively fixed; for example, if all other factors are equal, a three-system ring takes more time to process a global resource than a two-system ring. If you need three systems in the ring, however, you cannot resolve a performance problem by removing one system.

To evaluate the impact of these factors, you first need to understand how to calculate average response time. To illustrate the calculations, assume the following configuration:

- There are four systems in the ring.
- The RESMIL value for all systems is 1 millisecond.
- The RSA-message size is 4K.

Average response time

Average response time is the average amount of time a requester must wait before a request for a global resource can be granted. It depends on cycle time — the amount of time that the RSA-message requires to make a complete cycle around the ring. In tuning ring performance, the primary goal is to ensure that the average response time is as close as possible to the acceptable response time at your installation.

One of the factors used to calculate average response time is the transmission delay for a global resource serialization signal. Table 11 on page 168 shows the transmission delays for the global resource serialization signalling paths.

Table 11. Transmission Delays by Signalling Path and Device

Signalling Path	Device	Transmission Delay (in milliseconds)
Global resource serialization	ESCON channel in basic mode	0.5
XCF	ESCON channel in CTC or basic mode	0.5

You can use the following formulas to calculate average response time and predict the effect of changes you might make. The formulas you need depend on whether your ring is using ring acceleration.

If the ring is using ring acceleration, then **n** is the number of systems, and **a** is the ACCELSYS value. The calculations for cycle time and average response time are:

```
cycle time = (RESMIL * n) + (transmission delay * n)
response time = cycle time/2 + (RESMIL * (a - 1)) + (transmission delay * a)
```

Because the configuration includes a 3088, the transmission delay is about 1 millisecond for the RSA-message and 1 millisecond for the ring acceleration signal. Assuming ACCELSYS(2), the average response time is about 7 milliseconds, as follows:

```
cycle time = (1 millisecond * 4) + (1 millisecond * 4)
response time = 8 milliseconds/2 + 1 millisecond + 2 milliseconds
```

If the ring is not using ring acceleration, **n** is the number of systems, and the calculations for cycle time and average response time are:

```
cycle time = (RESMIL * n) + (transmission delay * n)
response time = cycle time * 1.5
```

In this case, the average response time is about 12 milliseconds, as follows:

```
cycle time = (1 millisecond * 4) + (1 millisecond * 4)
response time = 8 milliseconds * 1.5
```

Experience has shown that the cycle time is a stable value; it does not normally vary significantly during any given time period. Resource contention, however, can add to the response time. That is, average response time measures only the time a global resource requester waits for ring processing to complete. It does not reflect the time spent waiting until a requested resource becomes available (this delay is not directly related to ring processing).

Tuning factors

Achieving acceptable response time for global resource requests involves many factors:

- Number of systems in the ring
- Transfer rate on the communication links
- Level (version and release) of MVS
- RSA-message size
- Global resource request rate
- RESMIL value
- Ring acceleration (ACCELSYS)

The following sections describe the factors that can affect how successfully your complex meets your goals. Where applicable, the descriptions include specific actions for particular problems.

Number of systems

It is obvious that a two-system complex can respond more quickly to requests for global resources than a four-system complex. The fewer the systems, the shorter the cycle. Adding a system to the ring, however, need not cause a corresponding increase in response time. Lowering the residency time (RESMIL value) on all systems can make the addition of another system transparent to global resource requesters.

Transmission rate

The transmission rate reflects how quickly the systems in the ring can communicate. It is like the number of systems in the ring, relatively fixed. Using an ESCON channel operating in basic mode to provide communication links yields significant improvement over using integrated CTC adapters. Following the channel placement recommendations in "Link placement" on page 126 can avoid channel delay. The level of MVS installed can also affect the transmission rate.

RSA-message size

The maximum size of the RSA-message, as well as the cycle time and the average global resource request size, determine ring capacity — the number of global resource requests the ring can process in a given period of time. Whether ring capacity is a performance problem or not depends on the global resource request rate.

Global resource request rate

The global resource request rate is the number of global resource requests that the systems in the ring actually generate, most conveniently expressed as the number of global resource requests per second.

If the systems generate more requests than the ring can process, the RSA-message cannot hold all of the requests. Some requests must wait for at least one additional cycle before getting into the RSA-message, thus increasing response time. Using the RNLs to reduce the number of unnecessary global resource requests is a possible solution. Lowering the residency time (RESMIL value) is often a better way to deal with a high global resource request rate. The lower RESMIL value increases ring capacity; the RSA-message makes more cycles around the ring in any given time period, and thus the ring can process more requests in the same amount of time.

If the systems generate very few requests, global resource serialization might raise the actual RESMIL to avoid excessive CPU utilization. In rare cases, this can noticeably increase the average response time. If the increased response time is unacceptable, specify a RESMIL slightly lower than the value specified in Table 12 on page 170 or Table 13 on page 170.

RESMIL value

The residency time (also called the RESMIL value) is the amount of time that the RSA-message spends in each system in the ring in addition to the time needed to processes the RSA-message.

"Residency time value (RESMIL)" on page 131 and "Residency time value (RESMIL)" on page 121 describe how to set an initial value when you set up your complex. Use this information in tuning your complex. Setting the initial RESMIL value correctly is your best technique for tuning the complex.

Establishing a response time objective is one way to select a value for RESMIL. (See Chapter 12, "Measuring response time," on page 195 for one way to measure actual response time.) To establish a response time objective, determine what response time is acceptable or desirable, then set the RESMIL value to meet that objective.

Choose a low response time objective (10 milliseconds or lower) if all of the systems in the complex are 3090 processors or if any one of the following conditions is true:

- Your installation has chosen to convert catalog reserve SYSIGGV2.
- Your installation has batch jobs that issue many global resource requests and that must complete in a fixed time.

Once you have set a response time objective, you next translate that objective into a RESMIL value, depending on the number of systems in the ring. Table 12 on page 170 and Table 13 on page 170 can help you with this process. The RESMIL values shown are based on the formulas shown earlier in "Average response time" on page 167, with the following assumptions:

- The RSA-message contains only one request.
- The transmission time is one millisecond.
- There is no contention time for the requested resource.

The RESMIL values shown have been rounded down to the nearest integer.

As described earlier, there are many factors (such as data transmission rate, RSA-message size, and global resource request rate) that affect the actual response time. Thus, setting a particular RESMIL value does not guarantee meeting the corresponding response time objective. You can, however, use the RESMIL value in the table as a starting point. Use the values in Table 12 on page 170 if you use ring acceleration; use the values in Table 13 on page 170 if you do not.

Table 12. RESMIL Values with Ring Acceleration — ACCELSYS(2)

Response Time Objective (Milliseconds)	2-System Ring	3-System Ring	4-System Ring	5-System Ring
10	3	2	2	1
20	8	6	5	4
30	13	10	8	7
40	18	14	12	10
50	23	18	15	13

Table 13. RESMIL Values without Ring Acceleration

Response Time Objective (Milliseconds)	2-System Ring	3-System Ring	4-System Ring	5-System Ring
10	2	1	0	0
20	5	3	2	1
30	9	5	4	3
40	12	7	5	4
50	15	10	7	5

ACCELSYS value

The value you specify for ACCELSYS can be between 2 and **n**, where **n** is the number of systems in the global resource complex. Specify ACCELSYS(2) to obtain the maximum performance benefits. The default value for ACCELSYS is 99, which turns ACCELSYS off. See "Recovery" on page 124 or "Recovery" on page 135 for information about recovery considerations.

Tuning process

Tuning a global resource serialization complex, like any system tuning process, requires a disciplined approach of measuring the system's performance, setting specific goals, taking actions to reach the goals, then measuring and evaluating the results of the actions.

You will probably want to take a base set of measurements of your system performance before you begin the process of installing your complex and repeat the measurements at several points along the way. You might want to measure your system:

1. Before you begin to install the complex
2. While each system is running as a one-system complex
3. After the initial complex is running

4. After each change to the design of the initial complex — such as adding a system
5. After each tuning change that you make

Obviously, each set of measurements should be taken under the same set of conditions. That is, come as close as you can to ensuring that the system is processing the same workload during each measurement period.

Measurements

Gathering the information you need to measure and evaluate ring performance is a very important part of the tuning process. RMF reports provide much useful information about system performance and about resource use. You can use GTF to gather information about the size of the RSA-message, and you can measure response time.

Using RMF

Ring performance is only one aspect of total system performance. Focus your monitoring efforts on total system performance and look more closely at ring performance only when it appears to be affecting how well your system meets your overall performance objectives.

RMF Monitor I session reports are particularly useful in determining how ring performance affects overall system performance. One key report is **Workload Activity**. Use this report to determine the resources (processor, I/O, and storage) that global resource serialization is using. To obtain the clearest picture of resource consumption, place global resource serialization in its own performance group or report performance group.

RMF Monitor II session reports can also provide information about how global resource serialization affects the system. For example, you can use the **Address Space State Data** report to determine how much processor time global resource serialization consumes and the number of page faults the GRS address space experiences. To provide the optimum response time, the page-in rate for the GRS address space should be close to zero.

RMF Monitor III (workload delay monitor) reports can also help you deal with problems related to delays of jobs or users. These reports can show address spaces delayed and whether resource contention is contributing to the delay, as well as the jobs affected. They can point out jobs that are being slowed by ENQ delays. If the reports indicate ENQ delays, possible reasons might be:

- A job is delayed because the system is waiting for the RSA-message to complete its cycle. Lowering the RESMIL value might resolve the problem.
- A job is delayed because the RSA-message is encountering a communication delay. Ensure that the links used to send the RSA-message are not installed on the same channels as devices that monopolize the channel for long periods of time.
- A job is delayed because there is contention for the resource; another job is using it. If the delay is caused by a reserve, investigate the possibility of converting the reserve to reduce contention for the resource.

The Monitor I **Enqueue Contention Activity** report can also help you to identify the resources that are causing the most significant contention delays.

Using GTF

For systems that are not part of a multisystem sysplex, using the generalized trace facility (GTF) can provide additional information. You can use GTF to monitor:

- How frequently the RSA-message passes around the ring
- The actual size of the RSA-message
- The duration of the cycle time

To gather this information, trace the global resource serialization CTC channel program for a short period of time during peak processing. Place the following GTF trace options in a parmlib member that GTF reads when the trace is started:

```
TRACE=SSCHP,IOP,CCWP
IO=SSCH=(C44,C4C,C54)
CCW=(SI,DATA=8)
END
```

These sample statements trace the CTC links identified by device numbers C44, C4C, and C54. Replace these device numbers to trace the links that global resource serialization is using in your installation at the time of the trace. The data count (DATA=8) is small because the actual contents of the RSA-message are not of interest. The trace statements do record the number of bytes in the RSA-message; a length of 39 bytes indicates an empty RSA-message (an RSA-message that contains no resource requests).

If the RSA-message is almost always empty, you can probably set a higher RESMIL value without affecting response time. If the RSA-message is frequently close to full (more than 25K when the maximum size is 32K), set a lower RESMIL value to send the RSA-message around the ring more frequently. If the RSA-message is full even occasionally (more than five percent of the time), it is probably having an adverse effect on response time. Set a lower RESMIL value to send the RSA-message around the ring more frequently.

For more information about using GTF, see *z/OS MVS Diagnosis: Tools and Service Aids* and *z/OS MVS Initialization and Tuning Reference*.

When you examine the data, using either the GTFTRACE function of IPCS (interactive problem control system) or your own post-processing program, be aware that the direction (clockwise or counter clockwise) of the RSA-message is determined dynamically when the systems build the ring. Thus, an address used to send the RSA-message at one time might receive it at another time. Once determined, the direction of the message cannot change unless a system enters or leaves the ring.

Part 4. Global Resource Serialization Diagnosis

Chapter 11. Diagnosing global resource serialization

Occasionally, situations will arise during the operation of a global resource serialization complex where workload slows down or comes to a complete halt. There are times when these situations are due to problems with the allocation of global resource serialization managed (ENQ) resources. This topic describes strategies to aid in the diagnosis and correction of these problems in a sysplex environment for either a global resource serialization ring complex or a global resource serialization star complex.

This covers the following topics:

- "Discriminating between system and application problems" on page 175
- "Check if the complex is operating normally" on page 176
- "Tuning the global resource serialization ring" on page 176
- "Ring disruption recovery" on page 177
  - "ISG177E and ISG178E recovery" on page 177
- "Global resource serialization ring rebuild" on page 179
- "Checking XCF/XES connectivity and performance" on page 179
- "ISGLOCK structure request processing" on page 179
- "Checking for ENQ contention problems" on page 180
- "Using SMF 87 records" on page 185

Discriminating between system and application problems

The first step in determining what actions to take is to discriminate between a problem with the global resource serialization complex and the applications it serves. There are several kinds of problems that can occur which will affect global resource serialization processing:

- Tuning:

  A poorly tuned system can elongate global resource serialization requests (ENQ and DEQ). An example of a poorly tuned system is a ring complex where some of the systems have too high a RESMIL value. Another example is a star complex where the lock structure is too small causing excessive false contention in the lock structure. Tuning the complex will alleviate these problems.

- Intersystem communication breakdown:

  Global resource serialization relies on intersystem communication, through XCF communication facilities, which can be either CTCs or coupling facility signalling structures. Communication failures or delays might cause global resource serialization to take recovery actions which can delay and/or elongate ENQ and DEQ request processing.

- Coupling facility availability:

  The loss of a coupling facility might cause problems with a global resource serialization ring. In star mode, if the ISGLOCK structure fails or the containing coupling facility is lost, the systems in the sysplex cooperate to rebuild the structure.

- Software:

  Global resource serialization occasionally runs into situations that are not understood by the software or that are not automatically corrected. If a problem is detected that could cause a resource allocation integrity error (for example, more than one exclusive owner of a resource), global resource serialization will take appropriate actions to ensure that such an error does not occur. These actions include fencing a set of resources from being allocated or partitioning a system from the complex.

- Resource allocation:

Even if your global resource serialization complex is well tuned, a combination of applications, system utilities, and online users can impede workload progress due to the use of resources. For example, a long running job or utility can hold data sets exclusively, effectively blocking other jobs and users from proceeding. In more extreme cases, it is possible that a set of requests can cause a deadlock for resource requests by causing a situation where a set of users requires resources held by other users. This situation can only be remedied by breaking the deadlock, usually by cancelling one or more of the jobs in the deadlock.

Check if the complex is operating normally

First, the installation should check to ensure that the complex is operating normally. This is best accomplished by issuing the DISPLAY GRS,SYSTEM command. The following examples show normal output from these commands for both ring and star mode.

```
ISG343I 17.43.42 GRS STATUS          FRAME  1     F      E   SYS=PROD2
SYSTEM    STATE               SYSTEM    STATE
PROD1     ACTIVE              PROD2     ACTIVE
TEST1     ACTIVE

Ring complex, normal response for DISPLAY GRS,SYSTEM
command
```

```
ISG343I 17.43.42 GRS STATUS          FRAME  1     F      E   SYS=PROD2
SYSTEM    STATE               SYSTEM    STATE
PROD1     CONNECTED           PROD2     CONNECTED
TEST1     CONNECTED

Star complex, normal response for DISPLAY GRS,SYSTEM
command
```

Tuning the global resource serialization ring

If a global resource serialization ring complex is running without disruption, slowdowns are usually caused by an improperly tuned ring. Delay of the RSA message can become pronounced on larger complexes, delaying the initiation of new workload. To "speed up the ring", the installation can take one, two, or all of the following actions.

- **Speed up the RSA.**

  The speed of the RSA message is dependent on the RESMIL values specified in GRSCNFxx or the SETGRS RESMIL= command. Determine the RESMIL values used by each of the systems in the complex. To improve ENQ/DEQ response time, decrease the RESMIL value used on all the systems in the complex. You can use the ROUTE *ALL command to effect the change on all systems at one time. Try using a RESMIL of 1 or 2. If this does not meet your performance goals, try a RESMIL of 0.

  There is also a RESMIL setting of OFF. Use this setting carefully, as all of the other RESMIL values are tuned automatically by the complex. If the ring is lightly loaded, global resource serialization will tune the RESMIL value up one millisecond each time an empty RSA makes a trip around the sysplex until RESMIL reaches the specified value plus 5 (RESMIL=1 will tune between 1 and 6 milliseconds). When the ring becomes loaded, RESMIL returns to the specified value. When an installation specifies RESMIL=OFF, the RSA will be sent immediately after receipt and processing by each system, without tuning. This might adversely impact processor performance.

- **Use Ring Acceleration.**

  Ring acceleration improves performance by reducing the number of systems that must see an ENQ before the issuing system might grant ownership of the resource. This reduces the time between a system receiving the RSA and granting a resource request (but does not decrease the system's wait time for the RSA). There are possible integrity concerns introduced by the use of ring acceleration, but the opportunity for such failures is small. By specifying ACCELSYS(2), an installation can reduce the overall response time for an ENQ/DEQ request. See “Ring acceleration (ACCELSYS)” on page 134 for more information.

- **Use Star Mode.**

  By far, a global resource serialization star complex outperforms any ring complex, usually by orders of magnitude, with fewer recovery and operational concerns. IBM guidelines suggest that all installations migrate to a star complex whenever possible.

Ring disruption recovery

In a global resource serialization ring, communication and connectivity errors are often associated with the following messages:

- ISG177E
- ISG178E

For information about ISG177E and ISG178E disruption messages, see "ISG177E and ISG178E recovery" on page 177.

During a ring disruption and the ensuing recovery, the status of the systems should to change from:

- ACTIVE (before the disruption)
- INACTIVE (during the disruption)
- QUIESCED
- ACTIVE

As the ring is rebuilt, the systems will be returned to ACTIVE status one at a time. One of the systems in the complex is charged with bringing a QUIESCED system back into the ring. This system will be indicated by a status of ACTIVE+VARY.

Even in the best tuned complexes, ring disruptions periodically occur, especially if there are systems with small LPAR percentages or single processor machines that might become unresponsive due to other error conditions (a large system dump). After the error condition is corrected, the complex should automatically return to normal operation.

If these messages appear frequently, it is likely that there is a configuration error, either:

- The specified TOLINT value is too small

  TOLINT (Toleration Interval) is the maximum amount of time that the system will wait for the RSA to arrive before indicating an error condition and disrupting the ring. The default is TOLINT(180), which is 180 seconds or 3 minutes. You need to ensure that the TOLINT value is not too low, causing false error indications or too high, causing an excessive delay in recognizing a ring disruption situation. Use the SETGRS TOLINT= command to increase the value. You can use the ROUTE *ALL command to effect the change on all systems in the sysplex at the same time.
- XCF communication is not responsive

  Check your XCF signalling configuration to ensure that messages for the SYSGRS group are being serviced without excessive delay. Use data from RMF (or another performance monitor) to ensure that XCF signalling is optimized for the SYSGRS group. For more information on how to tune XCF signalling, see Tuning a sysplex in *z/OS MVS Setting Up a Sysplex*.

ISG177E and ISG178E recovery

If GRS seems hung after an ISG177E or ISG178E disruption message, use the process that follows to recover the sysplex.

1. Issue `D GRS` and `D XCF,S,ALL` on all the systems to obtain the status of each system.
2. GRS auto-restart processing might be in progress if any system has a GRS status of ACTIVE. Give GRS enough time, usually 4 to 6 minutes, to restart without manual intervention.
3. If all the systems in the GRS display show either INACTIVE or QUIESCE, issue `XCF PATHIN` and `PATHOUT` commands to ensure the entire sysplex has good connectivity. If XCF on any system

is unable to deliver signals, one of the systems might not have proper status keeping GRS from restarting. Recover paths as necessary to ensure that all systems have good connectivity.

4. If the sysplex has good connectivity, yet all the systems in the GRS display still show either INACTIVE or QUIESCE, you can restart the ring by manually driving the GRS group notification exits. To do this, temporarily stop a system using either the hardware console or QUIESCE command.

   Attention: Do not IPL.

5. Stop the system for one GRS TOLINT interval. Restart the system after the GRS TOLINT interval has expired. Restarting the system should re-drive the GRS group notification exits. If stopping and restarting a system does not restart GRS, then GRS on that system might not be the problem. Pick a different system and try stopping and restarting that system.

   Note:

   a. If you are running with an SFM policy that will take a stopped system out of the sysplex, stop the policy before stopping the system by using:

   ```
   SETXCF STOP,POLICY,TYPE=SFM
   ```

   b. If you are able to restart the ring, start the SFM policy using:

   ```
   SETXCF START,POLICY,TYPE=SFM,POLNAME=XXXXX
   ```

   where XXXXX is the SFM policy.

If you have completed the steps above on all the systems and the D GRS output still displays INACTIVE, you can restart the sysplex using the process that follows.

1. Use the hardware console or the QUIESCE command to temporarily stop the systems until only one is remaining.

   Attention: Do not IPL.

2. Use D XCF,S,ALL to check systems status. The XCF display output should show only one system as ACTIVE and the other systems as MONITOR-DETECTED STOP.
3. When only one system is ACTIVE, wait a TOLINT interval until the remaining system restarts as a one system ring.
4. Issuing a D GRS after waiting a TOLINT interval will show one system as ACTIVE and the other systems as QUIESCED.
5. When the D GRS command displays an ACTIVE system, start the other systems to have it join the ring.

If GRS is not able to restart, obtain the following data before calling IBM service:

1. SYSLOG from all systems.
2. LOGREC from all systems.
3. SADUMP from any system that requires an IPL

Use the following JCL to obtain a dump of primary and alternate CDS:

```
DUMP COMM=(your dump title)
R x,ASID=(1,6,7,A),REMOTE=(SYSLIST=*(1,6,7,A),DSPNAME,SDATA),CONT
R y,DSPNAME=('XCFAS'.*,'GRS'.*),CONT
R z,SDATA=(COUPLE,XESDATA,GRSQ,RGN,ALLNUC,CSA,PSA,SQA,SUM,TRT),END
```

where x, y, and z are reply numbers.

Note: If a SADUMP of a critical system is not possible, take a console dump.

Global resource serialization ring rebuild

If a ring rebuild seems to hang, quiesce one of the systems until the XCF failure detection interval is reached. The system issues message IXC402D to indicate that communications with that system has been lost. Restart the system and see if the ring successfully rebuilds. If the ring does not completely rebuild after one or two attempts, the system(s) that have not rejoined the complex will not do so. Partition the system(s) from the sysplex (using the VARY XCF,system,OFF command) and re-IPL. If possible, obtain a standalone dump of the failing system(s) and contact IBM Service.

Checking XCF/XES connectivity and performance

In both ring and star complexes, XCF and XES connectivity are critical to global resource serialization function. If either are not well tuned, the performance of the allocation of global resource requests will suffer. If your installation is running a global resource serialization ring complex, see System/390 Parallel Sysplex Performance (www.redbooks.ibm.com/redbooks/SG244356.html) to ensure optimal tuning of the sysplex.

In a global resource serialization star complex, you should apply appropriate resource to the coupling facility where you have allocated the ISGLOCK structure. Make sure that the structure is large enough to minimize false contention and that the coupling facility containing the ISGLOCK structure is fast enough to satisfy lock requests from the z/OS images. If you do not, you could see the following problems with your global resource serialization complex:

Symptom	Likely causes
Excessive processor utilization in the global resource serialization address space	1. The CF with the ISGLOCK structure is using shared CPs. 2. The CF with the ISGLOCK structure is of an older technology than the z/OS system. 3. High contention for global resource serialization managed resources.
Long ENQ response times	1. The CF does not have enough paths to handle the incoming requests. 2. The ISGLOCK structure is too small, causing high false contention rates. 3. The CF with the ISGLOCK structure is using shared CPs. 4. The CF with the ISGLOCK structure is of an older technology than the z/OS system.
ENQ requests suspended	1. The ISGLOCK structure is being rebuilt. 2. There are hung requests to the ISGLOCK structure. 3. A system in the sysplex is not responding to XES requests.

The Coupling Facility Structure Sizer Tool (CFSIZER) is a web based application that will return the structure sizes based on the latest CFLEVEL for the IBM products that exploit the Coupling Facility. See z/OS Internet Library (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

ISGLOCK structure request processing

Infrequently, there are situations that can cause requests to the ISGLOCK structure to be left in an incomplete state. When this occurs, resource requests for those resources that have been hung up will never complete, holding up the requesters indefinitely. The DISPLAY GRS,CONTENTION command provides information regarding resources and requesters that might be in this state. To determine if there is a problem with request processing, you should issue the command on all systems in the sysplex, because the data required to determine the condition is located on each system. When the system is running correctly, the response to the DISPLAY GRS,CONTENTION command, message ISG343I, will look as follows:

```
ISG343I 13.46.47 GRS STATUS 379
[resource contention information]
NO REQUESTS PENDING FOR ISGLOCK STRUCTURE
[latch contention information]
```

When there is a potential problem, the command will indicate the resource and requesters associated with the hung resources. Potentially hung resources are noted by the indication of a delay of more than two seconds in response time. If this text does not appear, the request might still be active, and the appearance in the output is not indicative of a problem. However, if the situation persists, the number of hung requesters will increase. To correct the situation, rebuild the ISGLOCK structure with the SETXCF START,REBUILD,STRNAME=ISGLOCK. An example of the output from the DISPLAY GRS,CONTENTION command when there might be a problem is presented below:

```
ISG343I 13.46.47 GRS STATUS 380
[resource contention information]
GLOBAL REQUESTS PENDING FOR ISGLOCK STRUCTURE:
SYSDSN   SYS1.PROD1.LINKLIB
PRODTOOL 001E 007E7B68 ENQ-EXCL LOCK REQUEST AT 10/25/1999 13:59:21
THIS REQUEST IS DELAYED MORE THAN 2 SECONDS
[latch contention information]
```

Checking for ENQ contention problems

When workload slows down and global resource serialization appears to be operating normally, the problem is often due to some part of the workload dominating ENQ resources in the sysplex. Because many workloads require exclusive access to resources (for example, to update a file), resource contention occurs between different parts of the workload when incompatible requests are made for resources. By itself, resource contention is not a sign of a problem. However, contention held for a long period of time among the same resources and requesters might be an indication of a problem. In many cases, the root cause of a resource holder not releasing a resource is due to something else blocking it which GRS knows nothing about. As such, IBM recommends using Runtime Diagnostics and Predictive Failure Analysis (PFA). Runtime Diagnostics uses GRS ENQ and Latch contention information, long ENQ holder information, deadlock detection, and information for various other resource providers, on the system in which it was initiated as well as the systems on which the holder resides to assist with narrowing down the root cause of a problem. Predictive Failure Analysis (PFA) includes the PFA_ENQUEUE_REQUEST_RATE check which can detect damage to an address space or system by using the number of enqueue requests per amount of CPU used as the tracked metric. For more information, see *z/OS Problem Management*. Global resource serialization and Runtime Diagnostics provides diagnostic commands to help determine the source of contention.

Global resource serialization provides diagnostic commands to help determine the source of contention.

Command	Use
DISPLAY GRS,ANALYZE,BLOCKER	Provides a list of the requesters that have been blocking ENQ resources for the longest time. Each blocker is reported with: 1. Resource name and scope. 2. The count of waiters and blockers of the resource. The block time, the system the blocker ENQed from, jobname, and the type of access requested are also reported.
DISPLAY GRS,ANALYZE,DEPENDENCY	Provides resource allocation dependency analysis: 1. Starting with each of the longest ENQ waiters, an analysis is performed, iteratively chaining from waiter to top blocker until either a request that is not waiting is found, or a resource allocation deadlock is detected. 2. Starting with the top blockers of a specified resource, an analysis is performed, iteratively chaining from waiter to top blocker until either a request that is not waiting is found, or a resource allocation deadlock is detected.

Command	Use
DISPLAY GRS,ANALYZE,WAITER	Provides a list of requesters that have been waiting the longest for ENQ resources. Each waiter is reported with: 1. The resource name and scope. 2. The count of waiters and blockers of the resource. 3. The top blocker of the resource. Each waiter is reported with its wait time, system resource that it was ENQed on, and the type of access (shared or exclusive) requested. The counts of waiters and blockers are explicitly returned only when the count is greater than one.
DISPLAY GRS,CONTENTION	Provides an alphabetized list of all visible[1] ENQ resources that are in contention. Each resource is reported with the owners and waiters of the resource. For SCOPE=SYSTEM resources, D GRS,C only reports on contention that occurred on the system where the command was issued. It does not report contention for SCOPE=SYSTEM resources on other systems in the complex.
MODIFY HZR,ANALYZE	This Runtime Diagnostic command is probably the best place to start if you suspect some type of problem. It will analyze both ENQ and Latch information along with other information to assist with narrowing down the root cause of potential system problems. It can follow the holding ASIDs even to other systems in the sysplex to determine problems that may be the root cause of the contention. It can also detect deadlocks between ENQs and Latches on the system on which the command was initiated as well as the systems where the blockers reside. For more information, see *z/OS Problem Management*.

To illustrate how to use contention analysis, an example is presented. In this example, the three-system sysplex is made up of systems PROD1, PROD2 and TEST.

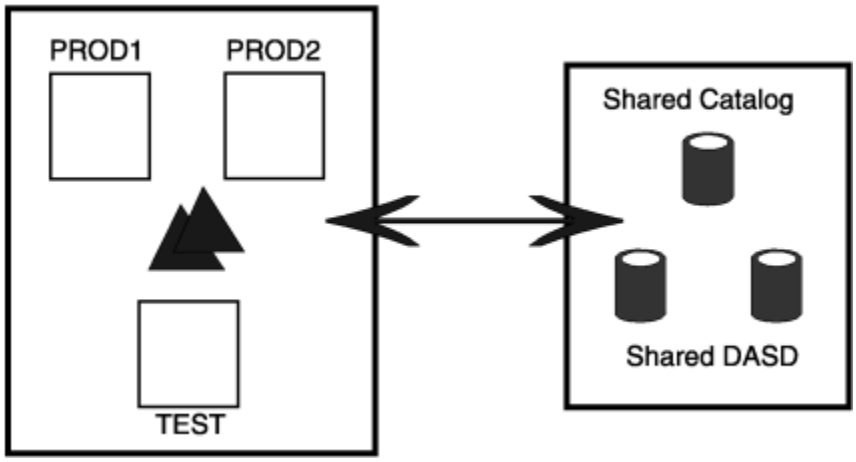


Figure 78. Example of Three System Sysplex

In the scenario, four different work units will be affected:

1. The master scheduler address space (*MASTER*) on PROD1
2. A production job (PRODJOB), running on PROD1
3. A database clean up job (CLEANUP), running on PROD2
4. The system programmer's TSO session (SYSPROG), running on TEST

The production job on PROD1 is a multistep process that submits the cleanup job, which is to run after the completion of the production run. The cleanup job is kept from running by the exclusive data set ENQ, [SYSDSN, PROD.DB] owned by the production job.

The scenario begins with the production job running. It reaches the step where it submits the cleanup job. The cleanup job initiates but is blocked in allocation on the global ENQ for [SYSDSN, PROD.DB]. However, as part of allocation, it takes exclusive ownership of [SYSDSN, PROD.PROCS]. The current view of contention is displayed in the following figure. In the figure, units of work are represented by rectangles

and resources are represented by ovals. The arrow and text from the unit of work to the resource represents the dependency.

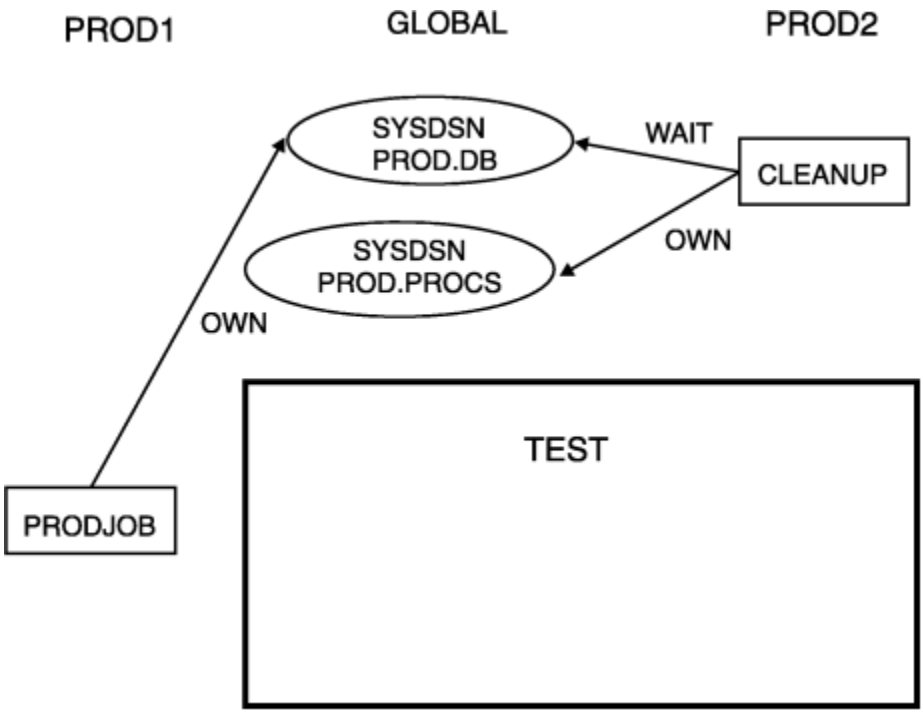


Figure 79. Current View of Contention

While the production job is executing, a task in the master scheduler address space (*MASTER*) fails, but does not end, while holding the system command resource, [SYSIEFSD, Q10]. This resource is required by tasks that need to issue MVS system commands. Following this failure, the production job invokes the MGCRE macro to issue a system command. Because the command resource is permanently hung up by the *MASTER* task, ownership cannot be granted to PRODJOB. Contention for ENQ resources now looks like:

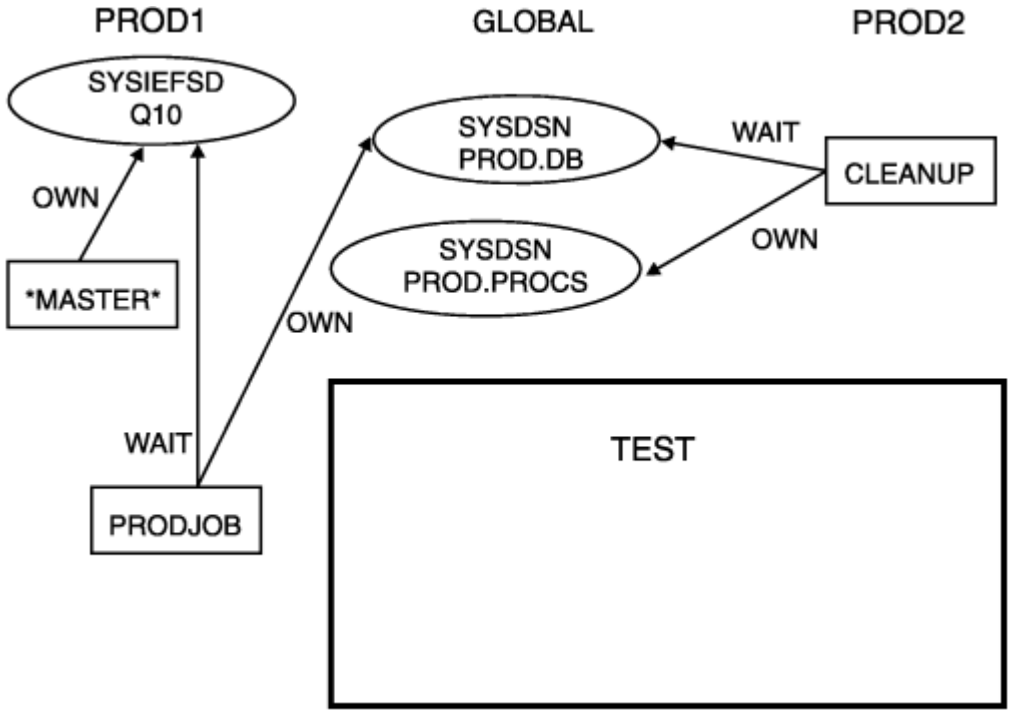


Figure 80. Contention for ENQ Resources

You discover that there is some sort of problem with the production database; the production job and the cleanup job seem to be hung up. Interactive requests for the database fail with an indication that the database is unavailable. You run an exec from the TSO session, which attempts to allocate both the production database, [SYSDSN, PROD.DB] and the production procedures library, [SYSDSN, PROD.PROCS]. This, of course, hangs the TSO session in an ENQ wait. The final state of contention is as follows:

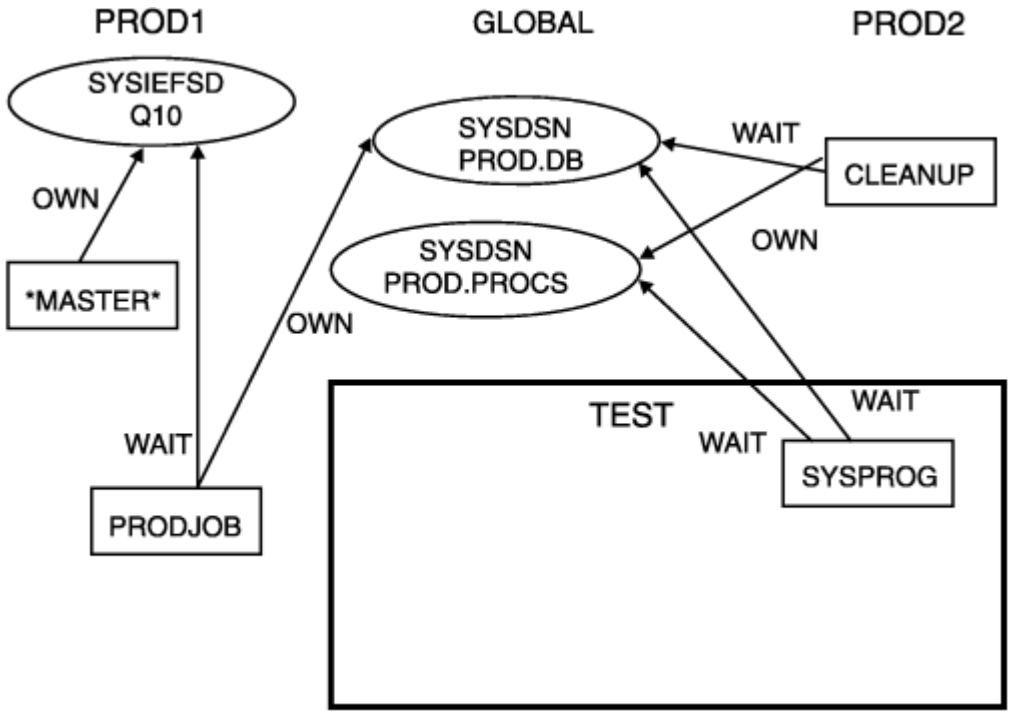


Figure 81. Final State of Contention

Note that on a normal system, there will always be some level of "background" contention that is going on all the time. The above example ignores that level of contention and only displays the contention that is applicable.

To debug this problem, use the contention analysis features provided by global resource serialization. The first thing that you discover is that commands do not seem to work on PROD1, so any systems analysis would have to occur either on TEST or PROD2. You must determine if any resources are in contention. If DISPLAY GRS,C were issued on PROD2 or TEST, the result would be as follows:

```
ISG343I 15.05.24 GRS STATUS 539
S=SYSTEMS SYSDSN   PROD.DB
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
PROD1     PRODJOB            001A       007E7B68 EXCLUSIVE    OWN
PROD2     CLEANUP            0029       007E7B68   SHARE      WAIT
TEST      SYSPROG            0027       007E7B68   SHARE      WAIT
S=SYSTEMS SYSDSN   PROD.PROCS
SYSNAME        JOBNAME         ASID     TCBADDR   EXC/SHR    STATUS
PROD2     CLEANUP            0029       007E7B68 EXCLUSIVE    OWN
```

Looking at this output, it would appear that the problem is with PRODJOB; it is blocking both CLEANUP and SYSPROG from continuing. However, the DISPLAY GRS,C command does not return information about local resources on PROD1, as it was issued on PROD2 or TEST. If system commands were working on PROD1, DISPLAY GRS,C from PROD1 would provide a more complete picture:

```
IISG343I 15.02.58 GRS STATUS 981
S=SYSTEMS SYSDSN   PROD.DB
SYSNAME   JOBNAME     ASID     TCBADDR     EXC/SHR   STATUS
PROD1     PRODJOB     001A     007E7B68   EXCLUSIVE   OWN
PROD2     CLEANUP     0029     007E7B68     SHARE     WAIT
TEST      SYSPROG     0027     007E7B68     SHARE     WAIT
S=SYSTEMS SYSDSN   PROD.PROCS
SYSNAME   JOBNAME     ASID     TCBADDR     EXC/SHR   STATUS
PROD2     CLEANUP     0029     007E7B68   EXCLUSIVE   OWN
TEST      SYSPROG     0027     007E7B68   EXCLUSIVE   WAIT
S=SYSTEM  SYSIEFSD Q10
SYSNAME   JOBNAME     ASID     TCBADDR     EXC/SHR   STATUS
PROD1     *MASTER*    0001     007E7B68   EXCLUSIVE   OWN
PROD1     PRODJOB     001A     007E7B68   EXCLUSIVE   WAIT
```

You can see that the local resource [SYSIEFSD, Q10], held by *MASTER* is really holding up all of the workload.

In a fully loaded system, where there is a considerable amount of workload being processed concurrently, the opportunities for contention and the number of units of work involved in that contention can become much higher. It might be impossible to quickly analyze what resources and units of work are part of a ENQ lockout and which ones are not. When this occurs, using the DISPLAY GRS,ANALYZE command is much more useful. Additionally, the GRS analysis command options are truly sysplex-wide in scope. The analysis will include local resources on all systems in the sysplex.

The analysis provided by this command is based on the fact that most of the "benign" contention in the sysplex is short term. That is, if you issue the same command over a period of time, the contention that is most affecting the sysplex remains in the output of the command. The output from the command is ordered by the length of time that the contention has been in effect. In a serious resource lockout, where one requester dominates ownership of a resource for a long period of time, or a resource deadlock, where a set of requesters requires resources held by the others in the set such that no request can be granted, the contention will quickly rise to the top of the output.

Using the previous lockout scenario, the DISPLAY GRS,ANALYZE,BLOCKER command would return:

```
ISG349I 15.03.09 GRS ANALYSIS 984
LONG BLOCKER ANALYSIS:  ENTIRE SYSPLEX
BLOCKTIME SYSTEM  JOBNAME E/S SCOPE QNAME    RNAME
00:01:33 PROD1    PRODJOB *E*  SYSS SYSDSN   PROD.DB
                              OTHER BLOCKERS: 0  WAITERS: 2
00:00:57 PROD1    *MASTER**E*  SYS  SYSIEFSD Q10
                              OTHER BLOCKERS: 0  WAITERS: 1
00:00:44 PROD2    CLEANUP *E*  SYSS SYSDSN   PROD.PROCS
                              OTHER BLOCKERS: 0  WAITERS: 1
```

It is clear from this output that PRODJOB has been blocking other requesters for the longest time. The display does not tell the complete story. The view obtained from DISPLAY GRS,ANALYZE,WAITER command shows that PRODJOB is also a waiter:

```
ISG349I 15.03.31 GRS ANALYSIS 987
LONG WAITER ANALYSIS:  ENTIRE SYSPLEX
WAITTIME  SYSTEM  JOBNAME E/S SCOPE QNAME    RNAME
00:01:33 PROD2    CLEANUP *S*  SYSS SYSDSN   PROD.DB
BLOCKER  PROD1    PRODJOB  E  OTHER BLOCKERS: 0  WAITERS: 1
00:00:57 PROD1    PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER  PROD1    *MASTER* E
00:00:44 PROD2    SYSPROG *S*  SYSS SYSDSN   PROD.DB
BLOCKER  PROD1    PRODJOB  E  OTHER BLOCKERS: 0  WAITERS: 1
00:00:44 PROD2    SYSPROG *E*  SYSS SYSDSN   PROD.PROCS
```

Again, because this is a simple case, it is easy to see that, although PRODJOB has been blocking the longest, PRODJOB is itself blocked by *MASTER* for [SYSIEFSD, Q10].

What if the scenario is far more complicated? Then the DISPLAY GRS,ANALYZE,DEPENDENCY command is very useful in determining if a single or a small set of jobs is causing the lockout. The command can also detect a resource allocation deadlock. For this scenario, the output from the command would be:

```
ISG349I 15.03.54 GRS ANALYSIS 990
DEPENDENCY ANALYSIS:  ENTIRE SYSPLEX
----- LONG WAITER #1
WAITTIME  SYSTEM  JOBNAME E/S SCOPE QNAME    RNAME
00:01:33 PROD2    CLEANUP *S*  SYSS SYSDSN   PROD.DB
BLOCKER  PROD1    PRODJOB  E
00:00:57 PROD1    PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER  PROD1    *MASTER* E
--:--:-- PROD1    *MASTER*
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
----- LONG WAITER #2
WAITTIME  SYSTEM  JOBNAME E/S SCOPE QNAME    RNAME
00:00:57 PROD1    PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER  PROD1    *MASTER* E
--:--:-- PROD1    *MASTER*
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
----- LONG WAITER #3
WAITTIME  SYSTEM  JOBNAME E/S SCOPE QNAME    RNAME
00:00:44 PROD2    SYSPROG *S*  SYSS SYSDSN   PROD.DB
BLOCKER  PROD1    PRODJOB  E
00:01:33 PROD1    PRODJOB *E*  SYS  SYSIEFSD Q10
BLOCKER  PROD1    *MASTER* E
--:--:-- PROD1    *MASTER*
ANALYSIS ENDED: THIS UNIT OF WORK IS NOT WAITING
----- LONG WAITER #4
WAITTIME  SYSTEM  JOBNAME E/S SCOPE QNAME    RNAME
00:00:44 PROD2    SYSPROG *E*  SYSS SYSDSN   PROD.PROCS
BLOCKER  PROD2    CLEANUP  E
```

From this analysis, it is obvious that the problem is with *MASTER* and not PRODJOB. In fact, all of the dependency analyses end with *MASTER*. Although you cannot restart *MASTER*, the only way to clear up this lockout is to reIPL PROD1. In the case where the unit of work that all of the other requests depend on is a cancelable job or a subsystem that can be recycled, the operator can take appropriate action against that address space to resume normal system operations.

For complete information on the DISPLAY GRS,ANALYZE command, see *z/OS MVS System Commands*.

Checking for latch contention problems

In many cases, the root cause of a resource holder not releasing a resource is due to something else blocking it which GRS knows nothing about. As such, IBM recommends using Runtime Diagnostics and Predictive Failure Analysis (PFA). Runtime Diagnostics uses GRS ENQ and Latch contention information, long ENQ holder information, deadlock detection, and information for various other resource providers. It uses this information on the system in which it was initiated, as well as the systems on which the holder resides, to assist with narrowing down the root cause of a problem. Predictive Failure Analysis (PFA) includes the PFA_ENQUEUE_REQUEST_RATE check which can detect damage to an address space or system by using the number of enqueue requests per amount of CPU used as the tracked metric. See z/OS Problem Management for more information.

For more information on GRS EQ and Latch contention, use the DISPLAY GRS,CONTENTION and DISPLAY GRS,ANALYZE commands to display information for the current GRS complex. For additional information, see “Checking for ENQ contention problems” on page 180 and *z/OS MVS System Commands*.

Using SMF 87 records

SMF 87 record subtype 1 helps you identify global generic queue scan issuers. SMF 87 record subtype 2 helps you diagnose application problems related to ENQ and DEQ and monitor the usage of ENQ/DEQ/ISGENQ/RESERVE requests.

SMF 87 records are mapped by macro SYS1.SAMPLIB(ISGYSMFR). In order to access a record programmatically, use macro IFASMFR.

Record environment

Macro: SMFEWTM BRANCH=YES (record exit: IEFU84)

Mode: Subtype Mode
1 and 2 Task (Non-Cross Memory)

Storage Residency: 31-bit

SUBSYS: SYSSTC

Using SMF 87 subtype 1 records to identify global generic queue scan issuers

In APAR OA42221, GRS provides monitoring of global generic queue scans using SMF type 87 records. GQSCAN and ISGQUERY REQINFO=QSCAN services can cause spikes in GRS CPU usage and GRS private storage when invoked many times for global, generic queue scans. For GQSCAN, a scan is global and generic if SCOPE=SYSTEMS or ALL, XSYS=YES or is not specified, and either the QNAME or RNAME (or both) are not specific. For ISGQUERY, a scan is global and generic if SCOPE=SYSTEMS or ANY, GATHERFROM=SYSPLEX, and either the QNAME or RNAME (or both) are not specific.

In GRS Star mode, the potential impact is greater than in other modes. SMF 87 records can help identify programs that issue global generic queue scans.

The records contain some information about the caller. For example, the TCB, ASID, PSW, and some details about the queue scan service invocation.

To activate GRS monitoring, specify MONITOR(YES) in your GRSCNFxx PARMLIB member or issue the SETGRS MONITOR=YES command. Ensure that either no GRSMON is active (D GRS shows GRSMON: NONE) or GRSMONXX contains FILTER INCLUDE QSCAN. You can dynamically turn the monitor on and off.

Issue DISPLAY GRS to view the MONITOR status (YES or NO) on your system.

GRS might not write the records in some cases that include, for example, if the caller is holding a local lock or if there is an unexpected error with the queue scan processing.

Record structure

An SMF 87 subtype 1 (QSCAN) record contains one SMFRCD87 DSECT (standard header and self-defining sections), followed by one SMF87REQ DSECT (requester section), then one SMF87QSCAN DSECT (QSCAN data section). One record is issued for each matching request.

In z/OS V2R1 and earlier, the requester section is version 1. In z/OS V2R2, it is version 2.

```
SMF87RCD
        Header (SMF87RHS)
        Self Defining Section (SMF87DEF)

Requester Section (SMF87REQ)

QSCAN Data Section (SMF87QSCAN)
```

Figure 82. Record structure for SMF 87 subtype 1

Record format

Table 14. SMFRCD87 DSECT

Offset	Name	Length	Format	Description
0(X'0')	SMFRCD87	44(X'2C')		SMF standard header and self defining section
0(X'0')	SMF87RHS	24(X'18')		SMF standard header
0(X'0')	SMF87LEN	2(X'2')	Binary	Record length

Table 14. SMFRCD87 DSECT (continued)

Offset	Name	Length	Format	Description
2(X'2')	SMF87SEG	2(X'2')	Binary	Segment descriptor
4(X'4')	SMF87FLG	1(X'1')	Binary	Flag byte
5(X'5')	SMF87RTY	1(X'1')	Binary	Record type 87(X'57')
6(X'6')	SMF87TME	4(X'4')	Binary	Time at which the record was written
10(X'A')	SMF87DTE	4(X'4')	Packed	Date on which the record was written
14(X'E')	SMF87SID	4(X'4')	EBCDIC	System identification
18(X'12')	SMF87SSI	4(X'4')	EBCDIC	Subsystem identification
22(X'16')	SMF87STP	2(X'2')	Binary	Record subtype. Possible values include: • X'1' SMF87STP_QSCAN – QSCAN subtype • X'2' SMF87STP_ENQ – ENQ subtype See macro ISGYSMFR for more information about the constants.
24(X'18')	SMF87DEF	20(X'14')		Self defining section
24(X'18')	SMF87DEF_LEN	4(X'4')	Binary	Length of self defining section
28(X'1C')	SMF87DEF_REQ_OFF	4(X'4')	Binary	Requester section offset
32(X'20')	SMF87DEF_REQ_LEN	2(X'2')	Binary	Requester section length
34(X'22')	SMF87DEF_REQ_NUM	2(X'2')	Binary	Number of requester sections
36(X'24')	SMF87DEF_DATA_OFF / SMF87DEF_QSCAN_OFF	4(X'4')	Binary	Qscan data section offset
40(X'28')	SMF87DEF_DATA_LEN / SMF87DEF_QSCAN_LEN	2(X'2')	Binary	Qscan data section length
42(X'2A')	SMF87DEF_DATA_NUM / SMF87DEF_QSCAN_NUM	2(X'2')	Binary	Number of Qscan data sections

Table 15. SMF87REQ DSECT – Maps the Requester section

Offset	Name	Length	Format	Description
0(X'0')	SMF87REQ	92(X'5C') at version 1, 108(X'6C') at version 2		Requester section
0(X'0')	SMF87REQ_COMP	8(X'8')	EBCDIC	Requester component ID. Possible values include: • 'SCSDS' SMF87REQ_COMP_GRS_0TO3 • SMF87REQ_COMP_GRS_4TO7 See macro ISGYSMFR for more information about the constants.
8(X'8')	SMF87REQ_VERSION	4(X'4')	Binary	Requester section version. Possible values include: • X'1' SMF87REQ_VERSION_1 – version 1 will be used on z/OS V2R1 and below • X'2' SMF87REQ_VERSION_2 – version 2 will be used on z/OS V2R2 See macro ISGYSMFR for more information about the constants.
12(X'C')	SMF87REQ_SYS	8(X'8')	EBCDIC	Requester system name
20(X'14')	SMF87REQ_STOK	8(X'8')	Binary	Requester STOKEN
28(X'1C')	SMF87REQ_ASID	2(X'2')	Binary	Requester ASID
32(X'20')	SMF87REQ_TCB	4(X'4')	Binary	Requester TCB address

Table 15. SMF87REQ DSECT – Maps the Requester section (continued)				
Offset	**Name**	**Length**	**Format**	**Description**
36(X'24')	SMF87REQ_JOB	8(X'8')	EBCDIC	Requester jobname
44(X'2C')	SMF87REQ_STIME	16(X'10')	Binary	Requester start timestamp (ETOD)
60(X'3C')	SMF87REQ_CTIME	16(X'10')	Binary	Requester completion timestamp (ETOD)
76(X'4C')	SMF87REQ_RC	4(X'4')	Binary	Requester return code
80(X'50')	SMF87REQ_RSN	4(X'4')	Binary	Requester reason code
84(X'54')	SMF87REQ_PSW	8(X'8')	Binary	Requester PSW

Table 16. SMF87QSCAN DSECT – Maps the QSCAN data section				
Offset	**Name**	**Length**	**Format**	**Description**
0(X'0')	SMF87QSCAN	284(X'11C')		QSCAN data section
0(X'0')	SMF87QSCAN_EYE	8(X'8')	EBCDIC	Service eye-catcher. Possible values include: • 'GQSCAN' SMF87QSCAN_EYE_GQSCAN_0TO3 and SMF87QSCAN_EYE_GQSCAN_4TO7 – GQSCAN Service • 'ISGQUERY' SMF87QSCAN_EYE_ISGQUERY_0TO3 and SMF87QSCAN_EYE_GQSCAN_4TO7 – ISGQUERY Service See macro ISGYSMFR for more information about the constants.
8(X'8')	SMF87QSCAN_VERSION	4(X'4')	Binary	QSCAN data section version. Possible values include X'1' SMF87QSCAN_VERSION_1 – version 1. See macro ISGYSMFR for more information about the constants.
12(X'C')	SMF87QSCAN_SCANACTION	1(X'1')	Binary	Requested scan action. Possible values include: • X'1' SMF87QSCAN_SCANACTION_START – Start • X'2' SMF87QSCAN_SCANACTION_RESUME – Resume See macro ISGYSMFR for more information about the constants.
13(X'D')	SMF87QSCAN_FLAGS	1(X'1')	Binary	Flags. Possible values include: • X'80' SMF87QSCAN_QNAME_SPECIFIC – When on, the request specified a specific QNAME • X'40' SMF87QSCAN_RNAME_SPECIFIC – When on, the request specified a specific RNAME
14(X'E')	SMF87QSCAN_QNAMELEN	1(X'1')	Binary	Requested QNAME length
15(X'F')	SMF87QSCAN_RNAMELEN	1(X'1')	Binary	Requested RNAME length
16(X'10')	SMF87QSCAN_MINWAITERS	4(X'4')	Binary	Requested minimum waiters
20(X'14')	SMF87QSCAN_QNAME	8(X'8')	EBCDIC	Requested QNAME or, if nonspecific, QNAME pattern. If not specified on the request, then zeroes.
28(X'1C')	SMF87QSCAN_RNAME	256(X'100')	EBCDIC	Requested RNAME or, if nonspecific, RNAME pattern. If not specified on the request, then zeroes. For the requested RNAME length, see SMF87QSCAN_RNAMELEN.

Using SMF 87 subtype 2 records to diagnose application problems related to ENQ and DEQ and monitor the usage of ENQ/DEQ/ISGENQ/RESERVE requests

SMF 87 subtype 2 records are unique among SMF record types because, since they are related directly to ENQ and DEQ activity, you can issue them much more frequently. It is also possible but less likely to issue SMF 87 subtype 1 as frequently. In order to manage data volume on the SMF side, and avoid data loss of other SMF types, target SMF 87 subtype 2 to a SMF logstream that is different from other types. Share DASD resources as infrequently as possible. Use a residency period MAXDORM to define how long to keep the data if there is a maximum buffer condition to avoid loss of data. See "Preserving SMF Data " in *z/OS MVS System Management Facilities (SMF)* for more information about avoiding SMF data loss.

Notes:

- When issuing SMF 87 subtype 2 records frequently, use GRSMONxx filtering to control the output data volume. For more information, see "GRSMONxx (global resource serialization monitor)" in *z/OS MVS Initialization and Tuning Reference* and the instructional sample GRSMONxx filters provided in SYS1.SAMPLIB(ISGMON00).
- The ISGAUDIT ISPF application SYS1.SBLSCLI0(ISGACLS0) does not support SMF 87 records.

Each record potentially contains multiple requests. Each request has its own REQUESTER section and ENQ DATA section. However, the DATA section does not immediately follow the REQUESTER section. To address the corresponding REQUESTER and DATA sections for a request, start at the self defining section. From there, go to the first requester section and the data section. These first sections correspond to the same request. The second request section and the second data section also correspond, and so on. So, requester section *n* corresponds to data section *n*, where 1 <= *n* <= *(number of requester sections)*. The number of requester sections is always the same as the number of data sections.

The requester section is mapped by DSECT SMF87REQ.

The ENQ data section is mapped by DSECT SMF87ENQ.

SMF 87 subtype 2 use cases:

- To assist in planning an RNL change, use the RNLMATCH filter to see which requests matched current RNLs.
- To assist in eliminating RESERVEs or understanding dependencies on RESERVE within your environment, use the RESERVE filter to determine unconverted hardware reserves.
- Use the WAITER filter to gather information about resource contention over time. This functionality is similar to that of other contention reports such as RMF, but is uses different media and is under your control.
- Use the JOBNAME filter or SCOPE filter to focus on the ENQs/DEQs for a certain application or exclude noise from the SMF data.
- Use the AUTHQLVL filter to plan a migration to AUTHQLVL(2) and identify any impacted programs before restarting the system with the new value.

Warning: Avoid issuing FILTER ENQ INCLUDE QNAME(*) RNAME(*) with no other filters. It will match every ENQ/DEQ in the system, which produces an exorbitant amount of data and consumes resources unnecessarily.

Note: To increase ENQ/DEQ overhead as minimally as possible when monitoring and filtering, use the fewest needed number of FILTER statements in GRSMONxx. You can also utilize wildcard placement in pattern strings to help avoid additional wildcard processing overhead. For example, use "?" or "*" in the middle of the string. For more information, see "GRSMONxx (global resource serialization monitor)" in *z/OS MVS Initialization and Tuning Reference* .

Record structure

```
SMF87RCD
        Header (SMF87RHS)
        Self Defining Section (SMF87DEF)

Requester Section 1 (SMF87REQ)

Requester Section 2

...

Requester Section SMF87DEF_REQ_NUM

ENQ Data Section 1 (SMF87ENQ)

ENQ Data Section 2

...

ENQ Data Section SMF87DEF_DATA_NUM
```

Figure 83. Record structure for SMF 87 subtype 2

Record format

Table 17. SMFRCD87 DSECT

Offset	Name	Length	Format	Description
0(X'0')	SMFRCD87	44(X'2C')		SMF standard header and self defining section
0(X'0')	SMF87RHS	24(X'18')		SMF standard header
0(X'0')	SMF87LEN	2(X'2')	Binary	Record length
2(X'2')	SMF87SEG	2(X'2')	Binary	Segment descriptor
4(X'4')	SMF87FLG	1(X'1')	Binary	Flag byte
5(X'5')	SMF87RTY	1(X'1')	Binary	Record type 87(X'57')
6(X'6')	SMF87TME	4(X'4')	Binary	Time at which the record was written
10(X'A')	SMF87DTE	4(X'4')	Packed	Date on which the record was written
14(X'E')	SMF87SID	4(X'4')	EBCDIC	System identification
18(X'12')	SMF87SSI	4(X'4')	EBCDIC	Subsystem identification
22(X'16')	SMF87STP	2(X'2')	Binary	Record subtype. Possible values include: • X'1' SMF87STP_QSCAN – QSCAN subtype • X'2' SMF87STP_ENQ – ENQ subtype See macro ISGYSMFR for more information about the constants.
24(X'18')	SMF87DEF	20(X'14')		Self defining section
24(X'18')	SMF87DEF_LEN	4(X'4')	Binary	Length of self defining section
28(X'1C')	SMF87DEF_REQ_OFF	4(X'4')	Binary	Requester section offset

Table 17. SMFRCD87 DSECT (continued)				
Offset	**Name**	**Length**	**Format**	**Description**
32(X'20')	SMF87DEF_REQ_LEN	2(X'2')	Binary	Requester section length
34(X'22')	SMF87DEF_REQ_NUM	2(X'2')	Binary	Number of requester sections
36(X'24')	SMF87DEF_DATA_OFF	4(X'4')	Binary	Data section offset
40(X'28')	SMF87DEF_DATA_LEN	2(X'2')	Binary	Data section length
42(X'2A')	SMF87DEF_DATA_NUM	2(X'2')	Binary	Number of data sections

Table 18. SMF87REQ DSECT – Maps the Requester section				
Offset	**Name**	**Length**	**Format**	**Description**
0(X'0')	SMF87REQ	108(X'6C') at version 2		Requester section
0(X'0')	SMF87REQ_COMP	8(X'8')	EBCDIC	Requester component ID. Possible values include: • 'SCSDS' SMF87REQ_COMP_GRS_0TO3 • SMF87REQ_COMP_GRS_4TO7 See macro ISGYSMFR for more information about the constants.
8(X'8')	SMF87REQ_VERSION	4(X'4')	Binary	Requester section version. Possible values include: zzz • X'1' SMF87REQ_VERSION_1 – version 1 is used on z/OS V2R1 and earlier • X'2' SMF87REQ_VERSION_2 – version 2 is used on z/OS V2R2 See macro ISGYSMFR for more information about the constants.
12(X'C')	SMF87REQ_SYS	8(X'8')	EBCDIC	Requester system name
20(X'14')	SMF87REQ_STOK	8(X'8')	Binary	Requester STOKEN
28(X'1C')	SMF87REQ_ASID	2(X'2')	Binary	Requester ASID
32(X'20')	SMF87REQ_TCB	4(X'4')	Binary	Requester TCB address
36(X'24')	SMF87REQ_JOB	8(X'8')	EBCDIC	Requester jobname
44(X'2C')	SMF87REQ_STIME	16(X'10')	Binary	Requester start timestamp (ETOD)
60(X'3C')	SMF87REQ_CTIME	16(X'10')	Binary	Requester completion timestamp (ETOD)
76(X'4C')	SMF87REQ_RC	4(X'4')	Binary	Requester return code
80(X'50')	SMF87REQ_RSN	4(X'4')	Binary	Requester reason code
84(X'54')	SMF87REQ_PSW	8(X'8')	Binary	Requester PSW
92(X'5C')	SMF87REQ_FLAGS	1(X'1')	Binary	Flags. Possible values include: • X'80' SMF87REQ_JOBINIT – Initiated job • X'40' SMF87REQ_STCTSU – Started task, TSO user, or other system process See macro ISGYSMFR for more information about the constants.
92(X'5D')	SMF87REQ_AUTH1	1(X'1')	Binary	Copy of CDATTR2, mapped by macro IHACDE. Value might be zero if unavailable.
96(X'60')	SMF87REQ_SMFID	4(X'4')	Binary	SMF system ID
100(X'64')	SMF87REQ_PROGRAMNAME	8(X'8')	EBCDIC	Program name. Field might be blank if unavailable

Table 19. SMF87ENQ DSECT – Maps the ENQ data section				
Offset	**Name**	**Length**	**Format**	**Description**
0(X'0')	SMF87ENQ	292(X'124') for version 1		ENQ data section
0(X'0')	SMF87ENQ_EYE	8(X'8')	EBCDIC	Service eye-catcher. Padded to eight characters with blanks. Possible values include: • ''ENQ' SMF87ENQ_EYE_ENQ_0TO3 and SMF87ENQ_EYE_ENQ_4TO7 – ENQ request • 'DEQ' SMF87ENQ_EYE_DEQ_0TO3 and SMF87ENQ_EYE_DEQ_4TO7 – DEQ request • 'ISGENQ' SMF87ENQ_EYE_ISGENQ_0TO3 and SMF87ENQ_EYE_ISGENQ_4TO7 – ISGENQ request For more information, see constants beginning with SMF87ENQ_EYE in the ISGYSMFR macro.
8(X'8')	SMF87ENQ_VERSION	4(X'4')	Binary	ENQ section version. Possible values include X'1' SMF87ENQ_VERSION_1 – version 1
12(X'C')	SMF87ENQ_REQUESTFIELDS	8(X'8')	Binary	Flags
12(X'C')	SMF87ENQ_FLAGS1	1(X'1')	Binary	Flags. Possible values include: • X'01' SMF87ENQ_LIST – When on, this request was part of a list request • X'02' SMF87ENQ_RNLNO – When on, this request was made requesting no changes allowed by RNLs • X'04' SMF87ENQ_PCREQUEST – When on, this request was a result of a PC-ENQ, PC-DEQ, or ISGENQ • X'08' SMF87ENQ_CONVERTED – When on, this request was converted from a RESERVE to a global ENQ by RNLs • X'10' SMF87ENQ_EXCLUDED – When on, this request was excluded by RNLs • X'20' SMF87ENQ_INCLUDED – When on, this request was included by RNLs • X'40' SMF87ENQ_EXITCHANGED – When on, this request was potentially changed by an exit **Note:** Exits may not change the scope or UCB status when RNL=NO is specified on the request. • X'80' SMF87ENQ_REQWAITED – When on, this request was queued to wait for the resource See macro ISGYSMFR for more information about the constants.
13(X'D')	SMF87ENQ_FLAGS2	1(X'1')	Binary	Flags. Possible values include: • X'08' SMF87ENQ_OBTAIN – When on, this request was to obtain or change an ENQ. When off, this request was to release an ENQ. • X'10' SMF87ENQ_RESERVE – When on, this request was for a RESERVE. See SMF87ENQ_CONVERTED to see if the RESERVE was converted. • X'20' SMF87ENQ_SHARE – When on, this request was for shared access • X'40' SMF87ENQ_STEPMUSTCOMPLETE – When on, this request asked for "Step must complete". • X'80' SMF87ENQ_DIRECTEDTCB – When on, this request was directed to another request See macro ISGYSMFR for more information about the constants.

Table 19. SMF87ENQ DSECT – Maps the ENQ data section (continued)

Offset	Name	Length	Format	Description
14(X'E')	SMF87ENQ_SCOPE	1(X'1')	Binary	Final scope of the request. Possible values include: • X'1' SMF87ENQ_SCOPE_STEP – Step • X'2' SMF87ENQ_SCOPE_SYSTEM – System • X'3' SMF87ENQ_SCOPE_SYSTEMS - Systems Refer to the constants starting with SMF87ENQ_SCOPE in macro ISGYSMFR
15(X'F')	SMF87ENQ_RNAMELEN	1(X'1')	Binary	RNAME length
16(X'10')	SMF87ENQ_REQUESTTYPE	1(X'1')	Binary	Type of request. Possible values include: • X'0' SMF87ENQ_REQUESTTYPE_NONE – RET=NONE • X'1' SMF87ENQ_REQUESTTYPE_HAVE – RET=HAVE • X'2' SMF87ENQ_REQUESTTYPE_CHANGE – RET=CHNG • X'3' SMF87ENQ_REQUESTTYPE_USE – RET=USE • X'4' SMF87ENQ_REQUESTTYPE_ECB – ECB= specified • X'7' SMF87ENQ_REQUESTTYPE_TEST – RET=TEST For more information, see constants starting with SMF87ENQ_REQUESTTYPE in macro ISGYSMFR. Also refer to the ENQ RET to ISGENQ Mapping Table for mapping ENQ/DEQ to ISGENQ request types.
20(X'14')	SMF87ENQ_DEVICEINFO	8(X'8')	Binary	DeviceInfo subfields are valid for only RESERVEs. The subfield values are zeros if unavailable.
20(X'14')	SMF87ENQ_DEVN	2(X'2')	Binary	Device number if available, or zeros if not
22(X'16')	SMF87ENQ_UCBVOLI	6(X'6')	EBCDIC	VOLSER if available, or zeros if not
28(X'1C')	SMF87ENQ_QNAME	8(X'8')	EBCDIC	QNAME
36(X'24')	SMF87ENQ_RNAME	256(X'100')	EBCDIC	RNAME. For more information, see SMF87ENQ_RNAMELEN for length of the request RNAME.

Table 20. ENQ RET to ISGENQ mapping

SMF87ENQ_REQUESTTYPE	ENQ/DEQ	ISGENQ
SMF87ENQ_REQUESTTYPE_NONE	RET=NONE (when others do not apply)	ISGENQ REQUEST=OBTAIN or RELEASE with COND=NO (when others do not apply)
SMF87ENQ_REQUESTTYPE_HAVE	RET=HAVE (when others do not apply)	ISGENQ REQUEST=OBTAIN or RELEASE with COND=YES (when others do not apply)
SMF87ENQ_REQUESTTYPE_CHANGE	RET=CHNG	ISGENQ REQUEST=CHANGE
SMF87ENQ_REQUESTTYPE_USE	RET=USE	ISGENQ REQUEST=OBTAIN CONTENTIONACT=FAIL
SMF87ENQ_REQUESTTYPE_ECB	ECB= specified	ISGENQ REQUEST=OBTAIN with WAITTYPE=ECB
SMF87ENQ_REQUESTTYPE_TEST	RET=TEST	ISGENQ REQUEST=OBTAIN with TEST=YES

Chapter 12. Measuring response time

This contains the following topics:

- "Viewing system performance" on page 195
- "Changing RESMIL values" on page 196

Viewing system performance

The cause of a response time problem that shows up either as an increase in interactive user response time or an increase in the time it takes to complete a batch job, is sometimes difficult to pinpoint. One way to obtain a very useful end user view of performance is to code a program that issues repetitive global resource requests (ENQ macros with a scope of SYSTEMS) and measures the elapsed time required for a response.

Figure 85 on page 196 shows an example of how to invoke a program that measures request response time from the perspective of the problem program issuing the request. You can take repetitive samples, allowing you to analyze response time over a longer period. The program, GENQRESP, is provided in SYS1.SAMPLIB(ISGNQRSP). You will need to ensure that G.SAMPLES DD is appropriate for your installation.

The program uses STIMER to suspend itself for a user-specified interval. When the interval has expired, the program issues an ENQ with a scope of SYSTEMS for a nonexistent resource. The program notes the value of the system TOD clock before issuing the ENQ request and again when control returns. The difference between the two values is the global resource request response time. The program then frees the resource and repeats the request.

The GENQRESP program encounters the same contention for the processor and storage as does any other problem program in your system. Thus, even after the system grants the global resource request, the program might not receive control immediately because of other factors, such as waiting for the processor, paging, or swapping. To ensure that the response time GENQRESP measures is as close as possible to the actual response time:

1. It is a good idea to run GENQRESP on a lightly-loaded system. Run the program on the system in the global resource serialization complex that has the lightest workload to minimize the effect of dispatching, paging and swapping delays on the results. Also, try to run the program each time under the same set of conditions. That is, come as close as you can to ensuring that the system is processing the same workload each time you run the program.
2. Run GENQESP at a dispatching priority that matches the component or subsystem you are analyzing (CATALOG, CICS, BATCH, ON-LINE, and others).

Use the RNL=NO keyword to prevent the alteration of the requested ENQ's SCOPE and to insure that global resource serialization processes a GLOBAL ENQ. When specifying the default of RNL=YES, the resulting SCOPE can be altered by the RNL, installation exit, or an alternate method of serializing global resources. You can have different versions of GENQRESP to use RNL=NO or RNL=YES. Use these to measure the differences between having the ENQ altered and managed by another serialization method and Global Resource Serialization. You can also use the ISGENQ macro TEST=YES option to determine if the ENQ issued is being altered, managed, or both by alternate serialization methods.

Figure 84 on page 196 shows an example of the output of GENQRESP.

```
    TOD CLOCK               ENQ TIME (IN U-SEC)
0A2CC4DDE984EAB1             00000000011336
0A2CC4DDE9D9F421             00000000008485
0A2CC4DDEA239931             00000000014236
0A2CC4DDEA856D41             00000000014689
0A2CC4DDEAE7DA41             00000000006145
0A2CC4DDEB28D4A1             00000000007376
0A2CC4DDEB6E86D1             00000000012141
0A2CC4DDEBC6B451             00000000008454
0A2CC4DDEC182481             00000000006383
0A2CC4DDEC59CCD1             00000000010125
```

Figure 84. Sample Output of GENQRESP

```
//GENQRESP  JOB accounting information...
//*
//* THE NUMBER OF REQUESTS TO ISSUE BEFORE TERMINATING AND THE INTERVAL
//* BETWEEN REQUESTS ARE SPECIFIED AS CONSTANTS IN THE CODE.
//*
//* THE OUTPUT DATASET, DD 'SAMPLES', HAS THE FOLLOWING FORMAT:
//*
//* START   END
//* COL     COL   FORMAT            DESCRIPTION
//*-----   -----  ------   ------------------------------------------
//*  1      16   TODCLOCK  TOD CLOCK VALUE WHEN ENQ WAS ISSUED
//* 31      44   DECIMAL   RESPONSE TIME FOR GLOBAL ENQ REQUEST
//*                        IN MICROSECONDS (1 SECOND = 10**6 MICROSECOND
//*
//* BECAUSE THE PROGRAM SAVES ONLY 32 BITS OF THE TOD CLOCK, RESPONSE
//* TIMES LONGER THAN ABOUT ONE HOUR ARE NOT RECORDED CORRECTLY.
//*
//* FOR THE MOST CONSISTENT RESULTS, ASSIGN THIS JOB TO A DOMAIN
//* WITH A MINIMAL MPL HIGH ENOUGH TO KEEP IT FROM BEING SWAPPED.
//*
//STEP 1 EXEC ASMHCLG,PARM.L='MAP,LIST,NET,NCAL',
// REGION.L=400K
//C.SYSPUNCH DD DUMMY
//C.SYSIN   DD DSN=SYS.SAMPLIB(ISGNQRSP),DISP=SHR
//C.SYSIN   DD *
        ENTRY GENQRESP
/*
//G.SAMPLES DD DSN=yourhlq.ENQRESP.DATA,
// UNIT=SYSDA,
// DISP=(NEW,CATLG),
// DCB=(LRECL=80,BLKSIZE=3200, RECFM=FB),
// SPACE=(CYL,(1,1))
```

Figure 85. Measurement Program Example

Changing RESMIL values

If any of these measurements reveal problems, the most important and useful action is to change the RESMIL value. Divide the change equally among all systems. There is no need to have a different RESMIL value for each system. “Residency time value (RESMIL)” on page 131 and “RESMIL value” on page 169 contain more detailed information.

If your major area of concern is the response time for global resource requests (that is, your concerns are resource availability or delays for work because of ring processing), you can reduce the cycle time by reducing the RESMIL value for each system. Reducing the cycle time reduces the time global resource serialization requires to process each request for a global resource and makes the complex

more responsive to individual resource requests. Reducing the cycle time, however, increases the number of times the system must process the RSA-message per second; it thus tends to increase the global resource serialization overhead — the demands it makes on the processor, channels, and devices.

If your major area of concern is the system overhead that global resource serialization causes (that is, your concern is system throughput), you can increase the cycle time by increasing the RESMIL value for each system. Increasing the cycle time reduces the system overhead. It decreases the number of times the system must process the RSA-message per second and thus tends to decrease the demands global resource serialization makes on the processor, channels, and devices. A longer cycle time, however, tends to increase the time global resource serialization requires to process each request for a global resource and makes the complex less responsive to individual requests for global resources.

There are two other basic tuning actions. One is to reduce the number of requests for global resources that the complex must process. That is, you want to ensure that all resources processed as global resources actually require global serialization. To reduce the number of requests for global resources, place entries in the SYSTEMS exclusion RNL to exclude from global serialization those resources, such as temporary data sets, that require only local serialization.

Another basic action is to convert as many reserves as possible. This action does not directly affect the performance of the complex, but it does increase the availability benefits your installation gets from the complex. Therefore, consider developing a long-term plan to work toward the goal of increasing the number of reserves converted to requests for global resources. For example, if you cannot convert reserves only because a volume can be accessed both by systems in the complex and systems outside the complex, change your use of the volume so that only systems in the complex can access it. You can then convert the reserves.

Comparing response time

Steps for comparing global ENQ response time to RESERVE

About this task

Follow these steps to compare global ENQ response time to RESERVE:

Procedure

1. Copy out SYS1.SAMPLIB(ISGNQRSP).
2. Modify the following section of the program by appending ,RNL=NO as shown:

```
LOOP     EQU   *
*
         STCK  TSTART                  SAVE TOD AT START OF REQUEST
         ENQ   (QNAME,RNAME,S,8,SYSTEMS),RNL=NO
         STCK  TSTOP                   SAVE TOD AT END OF REQUEST
         DEQ   (QNAME,RNAME,8,SYSTEMS),RNL=NO
```

3. To test RESERVE, modify the following section:

```
LOOP     EQU   *
*
         STCK  TSTART                  SAVE TOD AT START OF REQUEST
         RESERVE (QNAME,RNAME,E,8,SYSTEMS),UCB=UCBADDR
         STCK  TSTOP                   SAVE TOD AT END OF REQUEST
         DEQ   (QNAME,RNAME,8,SYSTEMS)
```

4. Change the following assignments at the bottom of the program:

```
QNAME    DC    CL8'QNAMEABC'           FICTITIOUS ENQ RESOURCE NAME
RNAME    DC    CL8'ENQTIMER'
*
```

5. Change QNAMEABC to a resource that is in your exclusion list (EXCL) and not in your conversion list (CONV).
6. After the RNAME line, add:

```
UCBADDR  DC    XL4'nnnnnnnn'          Address of UCB to RESERVE
```

7. Locate a UCB for a device that can be RESERVED by this system without causing an impact to other systems. Ensure that SYNCHRES=YES while the job is running, so that the RESERVE actually occurs under the ENQ process. The following system commands can help you determine the reserve status:

Table 21. System commands for reserve status

Command	**Function**
D, U,DASD,ONLINE	Displays the unit device numbers of online DASD with VOLSER and status
DS QP,(DEVICE NUMBER),UCB	Displays the UCB information, given the device number
D GRS	Displays global resource serialization information for a given system — including SYNCHRES indication.
SETGRS SYNCHRES=YES	Sets the SYNCHRES indication for a given system to YES

Appendix A. Data set ENQ contention monitor

The data set ENQ contention monitor is a sample application that determines which data sets are experiencing contention from multiple jobs. The data set ENQ contention monitor, which runs as a started task, periodically examines the data returned by the GQSCAN macro to locate data set contention. If a TSO/E user is the sole owner of a data set that other jobs are waiting to use, the TSO/E user receives a message asking the user to free the data set.

The data set ENQ contention monitor works only for TSO/E users that hold resources on the system where it is started. Therefore, you might want to start the data set ENQ contention monitor on each system in a global resource serialization complex. The data set ENQ contention monitor can detect contention for both local and global resources.

Installing the data set ENQ contention monitor

The data set ENQ contention monitor is written in IBM Assembler-H, and is a single program called ISGECMON, which is located in SYS1.SAMPLIB. To install the monitor:

1. Assemble the source code for ISGECMON
2. Link-edit the object code into a system library.
3. Create a new member in SYS1.PROCLIB that will invoke the program.

The next sections describe the listed steps in detail.

Assembling the data set ENQ contention monitor source module

In the JCL below, underscores indicate values that you can replace with information specific to your installation.

The numbers are not part of the JCL statements but correspond to notes that follow the example. Do not code these as part of the JCL.

First, assemble the module. Example JCL is shown below.

```
1.  //ASSEMBLE JOB
    //* INVOKING ASSEMBLER
2.  //ASMSTEP EXEC PGM=ASMBLR,REGION=1024K,
3.  //         PARM='RENT,LOAD,NODECK'
4.  //SYSLIB DD DSN=SYS1.MODGEN,DISP=(SHR,PASS)
    //       DD DSN=SYS1.MACLIB,DISP=(SHR,PASS)
    //SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE)
    //SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE)
    //SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(NEW,DELETE)
5.  //SYSLIN DD DSN=USERID.MY.OBJ(ISGECMON),DISP=OLD
6.  //SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBM,LRECL=121,BLKSIZE=3509),
    //          COPIES=1
    //SYSPUNCH DD SYSOUT=A,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),COPIES=1
7.  //SYSIN DD DSN=SYS1.SAMPLIB(ISGECMON),DISP=OLD,UNIT=SYSDA
    //SYSUDUMP DD SYSOUT=A
    //SYSABEND DD SYSOUT=A
    //
```

Notes on the JCL to assemble the monitor module

Each numbered JCL statement in the previous example is described as follows:

1. The JOB statement specifies the beginning of a job and assigns a job name.
2. The EXEC statement specifies the beginning of a job step and identifies the assembler.
3. The PARM parameter specifies options for the assembler.
4. The SYSLIB DD statement specifies the macro library concatenation.

5. The SYSLIN DD statement specifies the data set that will contain the object module (ISGECMON). The partitioned data set USERID.MY.OBJ must contain the assembled member ISGECMON when this job ends.
6. The DCB (data set control block) specifies the DCB options.
7. The SYSIN DD statement specifies the data set that contains the source code.

Link-editing the data set ENQ contention monitor source module

Once you have assembled the module, you can use JCL modeled on the example below to install the program into a system library in the LNKLST concatenation.

In the following JCL, underscores indicate values that you can replace with information specific to your installation.

The numbers are not part of the JCL statement but correspond to notes that follow the example. Do not code these as part of the JCL.

```
1.  //LINKEDIT JOB
2.  //LINK EXEC PGM=IEWL,
3.  //    PARM='RENT,XREF,REUS,LET,LIST,NCAL,SIZE=(750K,200K)'
    //SYSPRINT DD SYSOUT=A
4.  //OBJLIB DD DSN=USERID.MY.OBJ,DISP=SHR
    //SYSUT1 DD SPACE=(1024,(200,20)),UNIT=SYSDA
5.  //SYSLMOD DD DSN=LINKLIST.LIB,DISP=OLD
6.  //SYSLIN DD  *
        INCLUDE OBJLIB(ISGECMON)
        ENTRY ISGECMON
        NAME ISGECMON(R)
    /*
    //
```

Notes on JCL to Link-edit the ENQ contention monitor

Each numbered JCL statement in the previous example is described as follows:

1. The JOB statement specifies the beginning of a job and assigns a name to the job.
2. The EXEC statement specifies the beginning of a job step and identifies the linkage editor.
3. The PARM parameter specifies linkage editor options.
4. The OBJLIB statement identifies a data set containing the object module ISGECMON. The partitioned data set USERID.MY.OBJ must contain the assembled member ISGECMON.
5. The SYSLMOD DD statement specifies the data set that receives the object module.
6. SYSLIN DD * contains the linkage editor or control statements for the data set ENQ contention monitor.

Changes to SYS1.PROCLIB

Install the following JCL in SYS1.PROCLIB to allow the data set ENQ contention monitor to be started from the operator console:

```
//ISGECMON PROC INTERVAL=60
//ISGECMON EXEC PGM=ISGECMON,PARM='&INTERVAL'
//*
//*  Invoke from console with
//*       START ISGECMON
//*  to have the program scan for contention
//*  every 60 seconds.
//*
//*  Optionally, invoke from console with
//*       START ISGECMON,INTERVAL=ssss
//*  where ssss is the number of seconds (in decimal)
//*  to pause between scans.
//*
```

Using the data set ENQ contention monitor

Enter the START command from an operator console to start the data set ENQ contention monitor; use the CANCEL command to stop it.

Starting the monitor

Enter the following command to start the monitor from the operator console:

```
START ISGECMON,INTERVAL=ssss
```

The INTERVAL parameter specifies, in decimal, the number of seconds that the monitor is to pause between each contention check. You can use the INTERVAL parameter to control how sensitive the program is. A larger value means that data set contention is not detected immediately. A smaller value causes contention to be detected more promptly, but a smaller value also causes the monitor to use more CPU cycles.

Stopping the monitor

If you need to stop the data set ENQ contention monitor for any reason, issue the following command from the operator console:

```
CANCEL ISGECMON
```

Output

When the monitor detects that a TSO/E user owns a data set that another job or user is waiting for, it sends the following message to the TSO/E user:

```
===>> Please free 'Data Set.Name'. Other jobs are waiting to use it. (ISGECMON)
```

Appendix B. Recovery actions for global resource serialization

Loss of connectivity to the coupling facility

In the event of a loss of connectivity to the coupling facility, coupling facility structure failure, or operator initiated rebuild command, the following message is issued on each system notified by global resource serialization of a rebuild in progress.

```
ISG323A GLOBAL RESOURCE SERIALIZATION STOPPED ON sysname.
   LOCK STRUCTURE (ISGLOCK) REBUILD IS DUE TO reason.
```

As global resource serialization on each system is notified, global resource serialization requesters will be suspended due to a rebuild in progress. Each system will then proceed to perform its part of the rebuild process.

While the global resource serialization lock structure is being rebuilt, no systems are allowed to join the star complex. If a system attempts to connect the complex during the rebuild, the following message is issued:

```
      ISG318I GRS INITIALIZATION IS SUSPENDED UNTIL LOCK
              STRUCTURE (ISGLOCK) REBUILD IS COMPLETE.
```

In the case of a structure or connectivity error, the first system in the complex to be notified will respond by automatically initiating a rebuild of the lock structure. XES will determine if a rebuild is warranted under the current policy. If so, global resource serialization will continue with the rebuild process. If not, the rebuild will be stopped and global resource serialization will attempt to return to and use the original lock structure. Any systems without connectivity to the original lock structure will be terminated with a X'0A3' wait state with reason code X'C8' or X'D0'. For either of the reason codes, the following message is issued:

```
      ISG309W GRS PROCESSING TERMINATED. UNRECOVERABLE FAILURE
              DURING LOCK STRUCTURE, REBUILD PROCESSING.
```

When a lock structure rebuild successfully completes, the following message is issued:

```
      ISG325I GRS LOCK STRUCTURE (ISGLOCK) REBUILD HAS
              COMPLETED ON sysname.
```

The examples in this chapter assume that global resource serialization uses channel-to-channel (CTC) links.

Recovery when some ring complex systems are in sysplex

In a global resource serialization complex that has more than one system in the same sysplex, perform the sysplex recovery first; see *z/OS MVS Setting Up a Sysplex*. Performing the sysplex recovery actions may remove the need for performing actions for global resource serialization recovery. For example, when a system in the sysplex fails, another system in the sysplex will automatically do the needed global resource serialization purge when the system is removed from the sysplex.

Recovery when sysplex and ring complex contain same systems

When all the systems in a global resource serialization ring complex become part of the same sysplex, recovery becomes automatic. For global resource serialization recovery in a sysplex, see *z/OS MVS Setting Up a Sysplex*.

See the following:

- *z/OS MVS Planning: Global Resource Serialization* for information about global resource serialization
- *z/OS MVS Setting Up a Sysplex* for information about sysplexes

Recovery actions for ring complex with some systems not in a sysplex

For problems in a global resource serialization ring complex with some systems not in a sysplex, prepare recovery actions for the following:

- Purge a system from the complex before reIPL. See the next topic.
- Ring disruption in progress during IPL. See "Ring disruption in progress during IPL" on page 205.
- Primary link failure. See "Primary link failure" on page 205.
- Alternate link failure. See "Alternate link failure " on page 207.
- System failures. See "System failure" on page 207.
- Reactivating a quiesced system. See "Reactivating a quiesced system " on page 212.

An installation can control the amount of operator intervention needed for complex recovery. As a general rule, minimize operator intervention as much as possible; specify RESTART(YES) and REJOIN(YES) for as many systems as possible in the GRSCNFxx parmlib member.

See *z/OS MVS Initialization and Tuning Reference* for the GRSCNFxx member.

Purge a system from the ring complex before reIPL

Before reinitializing a failed system, the operator must purge the system from an active global resource serialization ring complex.

If the operator does not first purge the system, the following messages appear during reIPL of the failed system (SYS4):

```
ISG006I GRS JOIN OPTION INVALID-SYSTEM SYS4 ALREADY A GRS SYSTEM
```

- This message says that the global resource serialization ring complex already knows the system name: SYS4. It means that, because the operator did not purge a failed system from the complex before starting reIPL, global resource serialization finds two systems in the complex with the same name.

  ISG006I also appears if the operator types the wrong system name during IPL or specifies the wrong IEASYSxx parmlib member.

```
ISG007I fc-rc ERROR PROCESSING GRS JOIN OPTION.
ISG009D RELOAD THE SYSTEM OR REPLY JOIN OR NONE
```

  For an obvious typing error, the operator should restart the IPL and specify the system name correctly.

Actions to purge a system before reIPL

1. On an active system, enter DISPLAY GRS to check the state of the systems in the complex.
2. On an active system, enter a VARY command to remove SYS4 from the complex:

- VARY GRS(SYS4),PURGE

3. On SYS4, after the purge command completes successfully, continue to reIPL the system. Reply JOIN to message ISG009D.

The IPL continues. Message ISG011I appears on all systems to indicate that SYS4 has successfully joined the complex.

Ring disruption in progress during IPL

If a system failed and the operator purged it from the complex, the operator should wait until the disruption clears up and one system has restarted. Then the operator should reIPL the purged system for it to resume processing and rejoin the complex.

Another type of failure can occur during that reIPL. During reIPL, when the system tries to rejoin the complex, the complex could be experiencing a ring disruption. In this case, the following messages appear on the failed system (SYS4):

```
ISG006I GRS JOIN OPTION INVALID - GRS DISRUPTION MAY BE IN PROGRESS
ISG009D RELOAD THE SYSTEM OR REPLY START OR JOIN
```

On other systems, message ISG023E indicates that a ring disruption occurred.

Recovery actions for a ring disruption during IPL

Assuming that GRSCNFxx specifies RESTART(YES) for all systems, after the failure is corrected so that at least one system has restarted, then do the following:

1. On SYS4, continue with the IPL. Reply JOIN to message ISG009D.

Primary link failure

The primary link is the link global resource serialization uses to send the RSA-message. Use a DISPLAY GRS,LINK operator command to identify the links. The response for DISPLAY GRS,LINK shows IN USE for the primary link and ALTERNATE, QUIET, or DISABLED for all other links.

A primary link failure occurs when a channel-to-channel (CTC) link fails while global resource serialization is using the link to send the RSA-message. A system in the complex detects a primary link failure when the link it is using to send the RSA-message receives unanticipated interrupts, cannot perform an I/O operation, or receives unanticipated messages. The cause of the failure is not necessarily the link itself; the cause might also be a channel or software error.

Global resource serialization detects a primary link failure almost immediately. It does not wait for the tolerance interval to expire. On the system that detected the link failure, global resource serialization reports the problem to the operator as a link failure and a ring disruption. Other systems in the complex report only the ring disruption; they detect a problem but not its cause.

If a primary link fails, global resource serialization automatically selects an alternate link, if one is available. If there is no alternate available, global resource serialization tries to rebuild a valid ring using the connections that are available. A fully-connected complex with alternate links can always recover from the loss of a single link.

To permit this recovery without operator intervention, specify RESTART(YES) for all systems in the GRSCNFxx parmlib member.

To recover the disrupted ring, global resource serialization varies the failed link offline and selects an alternate link to replace it. If the complex has only one alternate and it is being used to send the ring acceleration signal, global resource serialization stops sending the ring acceleration signal between the two affected systems. If the complex has two alternates, global resource serialization uses one as the primary link and the other to send the acceleration signal.

For the following sequence of messages, assume a four-system complex and failure of the primary link between SYS1 and SYS2. SYS1 detects the failure. The device number of the failed link is A01. The installation specified RESTART(YES) for all systems.

On SYS1, the following messages appear:

```
ISG046E CTC A01 DISABLED DUE TO HARDWARE ERROR FAILURE CODE=05
ISG047I CTC A01 DISABLED
ISG022E SYSTEM SYS1 DISRUPTED GLOBAL RESOURCE SERIALIZATION DUE
        TO COMMUNICATION FAILURE - GLOBAL RESOURCE REQUESTORS WILL
        BE SUSPENDED
```

These messages indicate that SYS1 was using link A01 as the primary link, but it failed because of a hardware error. The operator should contact hardware support to fix the problem. Global resource serialization attempts to disable the link.

On other systems, the following message appears:

```
ISG023E GLOBAL RESOURCE SERIALIZATION HAS BEEN DISRUPTED - GLOBAL
        RESOURCE REQUESTORS WILL BE SUSPENDED
```

- Whether this message or ISG022E appears on the other systems depends on how each system detected the error. This message appears if the system detected the error because the tolerance interval expired or because SYS1 began recovery from the ring disruption. Message ISG022E appears if the system detected the communication failure. In most cases, message ISG023E appears.

  Make sure that the operators understand that these messages simply reflect the different points at which the systems detected the error.

On SYS1, the system coordinating the ring recovery, the following message indicates that automatic restart has begun:

```
ISG024I SYSTEM SYS1 INITIATED AUTO RESTART PROCESSING
```

After a ring disruption, all systems defined with RESTART(YES) attempt to restart the ring. Only one actually coordinates the process; this message identifies that system, which will bring all of the others back into the ring.

On other systems (SYS2, SYS3, and SYS4 in this example), the following message appears:

```
ISG025E SYSTEM sysx UNABLE TO INITIATE AUTO RESTART
        PROCESSING - PERMISSION GRANTED TO SYS1
```

This message, where *sysx* is SYS2, SYS3, or SYS4, indicates that this system has given permission to SYS1 to rebuild the ring. It is a message that the operators should expect to see during automatic restart processing. It normally requires no operator action.

As SYS1 rebuilds the ring, assuming that SYS2, SYS3, and SYS4 rejoin the ring in that order, the following messages appear:

On SYS1:

```
ISG011I SYSTEM SYS1 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS1 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM SYS2 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS2 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

On SYS2:

```
ISG011I SYSTEM SYS2 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS2 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

On SYS3:

```
ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

On SYS4:

```
ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

Recovery actions for a primary link failure

1. After ring processing resumes, enter DISPLAY GRS,LINK to see the links that are now IN USE.
2. DISPLAY GRS,LINK does not identify a link used for ring acceleration. For this information, use the RMF Monitor I report on communication equipment activity.
3. Contact hardware support for any failed links.
4. Vary links online for the system to be able to use them again.

Alternate link failure

An alternate link problem occurs when a CTC link fails while global resource serialization is using the link to send the ring acceleration signal. An alternate link failure does not always disrupt ring processing.

If global resource serialization detects an alternate link failure, it issues messages ISG046E and ISG047I to describe the error.

If global resource serialization was using the alternate link to send the ring acceleration shoulder-tap signal, it tries to locate another alternate link to use for ring acceleration. If one is available, global resource serialization uses it to send the signal. If one is not available, global resource serialization stops sending the signal between the two systems and issues the following message:

```
ISG083E GRS CANNOT SEND ACK-TAP SIGNALS TO SYSTEM sysname
```

Ring processing continues normally, but throughput might slow down on one of the systems because all requests for global resources must wait for ring processing. If an alternate link becomes available, global resource serialization begins using it to send the ring acceleration signal and deletes message ISG083E.

Recovery actions for an alternate link failure

Do the following:

1. Contact hardware support to repair the problem.
2. When the link is repaired, enter a VARY devnum,ONLINE command to make the link available again.

System failure

Automatic restart example

For this example, the operator actions assume a fully-connected four-system complex. Assume that SYS1 failed and that SYS2 detected the failure. Assume also that GRSCNFxx specified RESTART(YES) and REJOIN(YES) for all systems.

On all other systems, the following message appears:

```
ISG023E GLOBAL RESOURCE SERIALIZATION HAS BEEN DISRUPTED - GLOBAL
        RESOURCE REQUESTORS WILL BE SUSPENDED
```

This message indicates that an error has disrupted ring processing, though it does not pinpoint the source of the error. All active systems become inactive; any task that tries to obtain or free a global resource is suspended. All systems then attempt to restart the complex, but only one coordinates the process. In this example, assume that SYS2 is the system that will recover the complex.

On SYS2, the system coordinating the complex recovery, the following message indicates that automatic restart has begun:

```
ISG024I SYSTEM SYS2 INITIATED AUTO RESTART PROCESSING
```

On SYS3 and SYS4, the following message appears:

```
ISG025E SYSTEM sysx UNABLE TO INITIATE AUTO RESTART
        PROCESSING - PERMISSION GRANTED TO SYS2
```

This message, where *sysx* is SYS3 or SYS4, indicates that this system has given permission to SYS2 to rebuild the complex. It is a message that the operators should expect to see during automatic restart processing. It normally requires no operator action.

As SYS2 rebuilds the complex, assuming that SYS3 and SYS4 rejoin the complex in that order, the following messages appear as each system rejoins the complex.

On SYS2:

```
ISG011I SYSTEM SYS2 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS2 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

On SYS3:

```
ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

On SYS4:

```
ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

Global resource serialization has recovered the complex automatically without operator intervention. The recovered complex consists of SYS2, SYS3, and SYS4.

Recovery Actions for a System Failure with Automatic Restart

From an active system, the operator must determine if the failed system, SYS1, stopped temporarily or if the failure requires reIPL.

- If SYS1 stopped temporarily, do nothing.

  When SYS1 resumes processing, it automatically rejoins the complex because REJOIN(YES) was specified. The following messages appear on all active systems:

```
ISG011I SYSTEM SYS1 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS1 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

- If SYS1 requires reIPL, do the following:

  1. On an active system, enter VARY GRS(SYS1),PURGE to remove SYS1 from the complex. The following message appears:

```
ISG017D CONFIRM PURGE REQUEST FOR SYSTEM SYS1 - REPLY YES OR NO
```

     Reply YES. The following message appears:

```
ISG011I SYSTEM SYS1 - BEING PURGED FROM GRS COMPLEX
```

  2. If the following appears before ISG013I, the message indicates a possible data integrity exposure.

```
ISG016I SYSTEM SYS1 OWNS OR IS WAITING FOR GLOBAL RESOURCES
```

     Notify the system programmer. Give the system log to the system programmer. The system log normally contains messages about the problem. The following message describes any resources that might have been damaged by purging the system:

```
ISG018I REQUESTORS FROM SYSTEM SYS1 HAVE BEEN PURGED FROM
        RESOURCES NAMED xxxx,yyyy
```

  3. On SYS1, reIPL the system with GRS=JOIN.
  4. On the active systems, the following messages appear when SYS1 rejoins the complex:

```
ISG011I SYSTEM SYS1 - JOINING GRS COMPLEX
ISG004I GRS COMPLEX JOINED BY SYSTEM SYS1
```

Manual restart

The operator actions assume a fully-connected four-system complex. Assume that SYS1 failed and that SYS2 detected the failure. Assume also that GRSCNFxx specified RESTART(NO) and REJOIN(NO) for all systems.

On SYS2, the following messages appear:

```
ISG023E GLOBAL RESOURCE SERIALIZATION HAS BEEN DISRUPTED - GLOBAL
        RESOURCE REQUESTORS WILL BE SUSPENDED
ISG025E SYSTEM SYS2 UNABLE TO INITIATE AUTORESTART PROCESSING - THIS
        SYSTEM IS NOT AUTHORIZED
```

- These messages indicate that an error has disrupted ring processing, and that the system is not authorized to restart the complex automatically. All active systems become inactive; any task that tries to obtain or free a global resource is suspended. Message ISG023E and message ISG025E appear on all of the other systems in the complex.

  Tasks suspended while waiting for resources will quickly affect throughput on all systems; it is therefore imperative that an operator start recovery immediately.

Recovery Actions for a System Failure with Manual Restart

The recovery actions assume the operator on SYS2 is to initiate complex recovery.

1. On SYS2, enter VARY GRS(ALL),RESTART. Enter this command rather than trying to restart specific systems. Enter the VARY command only once. Otherwise, a split ring may be created.

   As SYS2 rebuilds the complex, assuming that SYS3 and SYS4 rejoin the complex in that order, the following messages appear as each system rejoins the complex.

   On SYS2:

   ```
   ISG011I SYSTEM SYS2 - RESTARTING GLOBAL RESOURCE SERIALIZATION
   ISG013I SYSTEM SYS2 - RESTARTED GLOBAL RESOURCE SERIALIZATION
   ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
   ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
   ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
   ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
   ```

   On SYS3:

   ```
   ISG011I SYSTEM SYS3 - RESTARTING GLOBAL RESOURCE SERIALIZATION
   ISG013I SYSTEM SYS3 - RESTARTED GLOBAL RESOURCE SERIALIZATION
   ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
   ```

   On SYS4:

   ```
   ISG011I SYSTEM SYS4 - RESTARTING GLOBAL RESOURCE SERIALIZATION
   ISG013I SYSTEM SYS4 - RESTARTED GLOBAL RESOURCE SERIALIZATION
   ```

2. On SYS2, after the restart is complete, enter DISPLAY GRS to verify that the systems are active. Determine if the failure on SYS1 requires reIPL or if the operator action stopped SYS1 temporarily.
3. If the failure is temporary, do the following to recover the system:
   a. On SYS1, resume system processing.
   b. On SYS2, after determining that SYS1 has resumed processing, enter VARY GRS(SYS1),RESTART.
   c. On the active systems, the following messages appear when SYS1 rejoins the complex:

      ```
      ISG011I SYSTEM SYS1 - RESTARTING GLOBAL RESOURCE SERIALIZATION
      ISG013I SYSTEM SYS1 - RESTARTED GLOBAL RESOURCE SERIALIZATION
      ```

1. If the failure requires reIPL of SYS1, do the following:
   a. On an active system, enter VARY GRS(SYS1),PURGE to remove SYS1 from the complex. The following message appears:

      ```
      ISG017D CONFIRM PURGE REQUEST FOR SYSTEM SYS1 - REPLY YES OR NO
      ```

   b. If the following appears before ISG013I, the message indicates a possible data integrity exposure.

```
ISG016I SYSTEM SYS1 OWNS OR IS WAITING FOR GLOBAL RESOURCES
```

Notify the system programmer. Give the system log to the system programmer. The system log normally contains messages about the problem. The following message describes any resources that might have been damaged by purging the system:

```
ISG018I REQUESTORS FROM SYSTEM SYS1 HAVE BEEN PURGED FROM
        RESOURCES NAMED xxxx,yyyy
```

c. On SYS1, reIPL the system with GRS=JOIN.

d. On the active systems, the following messages appear when SYS1 rejoins the complex:

```
ISG011I SYSTEM SYS1 - JOINING GRS COMPLEX
ISG004I GRS COMPLEX JOINED BY SYSTEM SYS1
```

Restart of two-system complex

For a two-system complex, specify in the GRSCNFxx parmlib member RESTART(YES) and REJOIN(YES) for SYS1 and RESTART(NO) and REJOIN(YES) for SYS2.

If SYS2 fails, SYS1 automatically restarts the complex. If SYS2 was temporarily stopped, it can automatically rejoin the complex when it resumes processing. In this case, the complex recovers without operator intervention.

If SYS1 fails, however, recovery requires operator intervention.

Recovery actions for a two-system complex if SYS1 fails

1. On SYS2, the following messages appear:

```
ISG023E GLOBAL RESOURCE SERIALIZATION HAS BEEN DISRUPTED - GLOBAL
        RESOURCE REQUESTORS WILL BE SUSPENDED
ISG025E SYSTEM SYS2 UNABLE TO INITIATE AUTO RESTART
        PROCESSING - THIS SYSTEM IS NOT AUTHORIZED
```

   These messages indicate that an error disrupted ring processing, and that SYS2 is not authorized to restart the complex automatically. SYS2 becomes inactive; any task that tries to obtain or free a global resource is suspended. Suspended tasks will quickly degrade performance on both systems.

2. On SYS2, enter VARY GRS(ALL),RESTART.

3. On SYS2, the following messages appear:

```
ISG026I SYSTEM SYS2 MAY CREATE A SPLIT RING IF ANY OTHER GRS
        SYSTEM IS ACTIVE. VERIFY THAT NO GRS SYSTEM IS ACTIVE
        BEFORE CONFIRMING RESTART
ISG027D CONFIRM RESTART RING FOR SYSTEM SYS2 - REPLY NO OR YES
```

4. Before replying to message ISG027D, enter DISPLAY GRS or check with the other operator to verify that SYS1 is not active.

   - If SYS1 is inactive, reply YES to continue the restart.
   - If SYS1 is active, reply NO to avoid a split ring.

   On SYS2, reply YES to message ISG027D (assuming that SYS1 is inactive). The following messages appear:

```
ISG011I SYSTEM SYS2 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS2 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

5. On SYS2, determine if the failure on SYS1 requires reIPL of SYS1 or if operator action stopped SYS1 temporarily.
6. If the failure is temporary, no further action is needed. When SYS1 resumes processing, it automatically rejoins the complex because REJOIN(YES) was specified on both systems. The following message appears:

```
ISG012I RESTART REQUEST PASSED TO SYS2
```

The following messages appear on both systems when SYS1 rejoins the complex:

```
ISG011I SYSTEM SYS1 - RESTARTING GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM SYS1 - RESTARTED GLOBAL RESOURCE SERIALIZATION
```

1. If the failure requires reIPL of SYS1, do the following:
   a. On SYS2, enter VARY GRS(SYS1),PURGE to remove SYS1 from the complex. The following messages might appear:

```
ISG017D CONFIRM PURGE REQUEST FOR SYSTEM SYS1 - REPLY
        YES OR NO
```

Reply YES. The following message appears:

```
ISG011I SYSTEM SYS1 - BEING PURGED FROM GRS COMPLEX
```

   b. If the following appears before ISG013I, the message indicates a possible data integrity exposure.

```
ISG016I SYSTEM SYS1 OWNS OR IS WAITING FOR GLOBAL RESOURCES
```

Notify the system programmer. Give the system log to the system programmer. The system log normally contains messages about the problem. The following message describes any resources that might have been damaged by purging the system:

```
ISG018I REQUESTORS FROM SYSTEM SYS1 HAVE BEEN PURGED FROM
        RESOURCES NAMED xxxx,yyyy
```

   c. On SYS1, reIPL the system with GRS=JOIN.
   d. On both systems, the following messages appear when SYS1 rejoins the complex:

```
ISG011I SYSTEM SYS1 - JOINING GRS COMPLEX
ISG004I GRS COMPLEX JOINED BY SYSTEM SYS1
```

Reactivating a quiesced system

Restarting a quiesced system requires help from an active system — a system that is currently actively participating in the global resource serialization complex. The active system provides current information about global resource requests. A quiesced system does not have this current information. For this reason, a quiesced system cannot make itself an active system without introducing a data integrity exposure.

In contrast, an inactive system has current information about the requests and can make itself an active system without introducing a data integrity exposure.

Under normal conditions, including most ring disruptions, at least one system is active and can restart a quiesced system, or an inactive system can make itself active and can restart a quiesced system.

Under unusual conditions, all systems in the complex might be either quiesced or failed. This condition can occur when the operators quiesce all but one system and then the only active system fails. When all systems are quiesced or have failed, an installation can recover the complex in two ways:

1. Perform a complex-wide IPL. This process is cumbersome and time-consuming, particularly when MVS is still running on the quiesced systems.
2. Allow global resource serialization to reactivate a system. **Reactivating a system** means turning a quiesced system into an active system. It avoids the complex-wide IPL but can introduce data integrity exposures or other problems related to global resources.

Reactivating a system is less disruptive than a complex-wide IPL. Careful planning for the process can reduce the impact of the data integrity exposure.

The recovery procedure shows how to reactivate a four-system complex. The procedure assumes a four-system complex, SYS1, SYS2, SYS3, and SYS4. The following events occurred:

1. At 10:00, the operator entered VARY GRS(*),QUIESCE on SYS4.
2. At 10:01, the operator entered VARY GRS(*),QUIESCE on SYS3.
3. At 10:02, the operator entered VARY GRS(*),QUIESCE on SYS2.
4. At 10:03, the only remaining active system SYS1, failed.

Recovery actions for reactivating quiesced systems

The operator on SYS4 begins the reactivation.

1. On SYS4, enter DISPLAY GRS. The response does not show SYS1, but does show that all other systems are QUIESCED, and that the links between the quiesced systems are ALTERNATE.
2. On SYS4, enter any one of the following:
   - VARY GRS(ALL),RESTART
   - VARY GRS(*),RESTART
   - VARY GRS(SYS4),RESTART

   On SYS4, the following message appears:

   ```
   ISG121I VARY REQUEST REJECTED - ENTER COMMAND ON SYSTEM SYS2
   ```

   This message indicates that global resource serialization has determined that another system, SYS2 in this example, has more current resource information. The message tells the operator to enter the restart command on SYS2, which minimizes but does not entirely eliminate the data integrity exposure in reactivating a quiesced system.
3. On SYS2, enter VARY GRS(SYS2),RESTART. The following messages appear:

   ```
   ISG110E NO ACTIVE SYSTEMS FOUND IN THE GRS COMPLEX. REACTIVATE
           SHOULD BE ATTEMPTED IF THERE ARE NO ACTIVE GRS SYSTEMS
           IN OPERATION. REACTIVATE MUST NOT BE ATTEMPTED IF THERE
           ARE ACTIVE SYSTEMS.
   ISG111D CONFIRM REACTIVATE SHOULD BE ATTEMPTED - REPLY NO OR YES
   ```

   These messages indicate that SYS2 has the most current resource information of all the quiesced systems and has thus accepted the command.
4. Before replying to ISG111D, verify the following:

a. The complex contains no ACTIVE or INACTIVE system.

b. All required links are operational.

c. No system has been temporarily stopped since SYS2 was quiesced.

Also, contact the system programmer, describe the problem, and ask if a complex-wide IPL should be done. If the complex-wide IPL is preferable, reply NO to message ISG111D.

1. *The following steps assume that reactivation is to continue.*

   On SYS2, reply YES to message ISG111D. Some of the following messages appear:

```
ISG112I SYSTEM SYS3 RESPONDED WITH GLOBAL RESOURCE STATUS -- IT
        WILL BE PERMITTED TO RESTART AFTER REACTIVATE COMPLETES
```

   This message indicates that the system is responding and does not have the most current information about global resources.

```
ISG114I RESOURCE STATUS NOT RECEIVED FROM SYSTEM SYS4 -- IT MUST
        BE STOPPED BEFORE CONFIRMING REACTIVATE COMPLETION
```

   This message indicates that the system is responding but did not supply information about global resources. The system might have more current information, but it cannot reactivate the complex. (Only systems that include MVS/SP Version 2.2 with the fix for APAR OY13087 can reactivate the complex.) The operator must make sure the system is stopped before proceeding.

```
ISG113I NO RESPONSE RECEIVED FROM SYSTEM SYS1 - IT MUST BE
        STOPPED BEFORE CONFIRMING REACTIVATE COMPLETION
```

   This message indicates that the named system did not respond. In this case, this message confirms that SYS1 has failed. In other cases, the operator must investigate the state of the named system and make sure the system is stopped before proceeding.

```
ISG115I IF ANY SYSTEMS WERE NOT LISTED IN THE PREVIOUS LIST,
        THEY ARE UNKNOWN TO THIS SYSTEM - THEY MUST BE STOPPED
        BEFORE CONFIRMING REACTIVATE COMPLETION
ISG116E STOP INDICATED SYSTEMS BEFORE CONFIRMING REACTIVE
        COMPLETION. REPLY NO TO CANCEL REACTIVATE
ISG117D CONFIRM REACTIVATE SHOULD BE COMPLETED - REPLY NO OR YES
```

   - These messages warn the operator to make a final check before allowing the reactivate process to complete. At this point, you could verify, for example, that no system had joined the complex after SYS2 was quiesced but before SYS1 failed.

     Message ISG112I, ISG113I, or ISG114I must appear for every system in the complex. If there is no message for a system, investigate the reason before proceeding. If you find any problems, reply NO to message ISG117D to stop reactivation.

1. *The following steps assume that reactivation is to complete.*
2. On SYS2, reply YES to message ISG117D. The following message indicates that SYS2 is now an active system:

```
ISG118I REACTIVATE FUNCTION IS COMPLETE. SYSTEM SYS2 HAS
        RESTARTED AS A RING OF ONE SYSTEM.
```

   On SYS2, enter the following to complete the recovery:

   a. VARY GRS(SYS1),PURGE to purge SYS1 from the complex

   b. VARY GRS(SYS4),PURGE to purge SYS4 from the complex

c. VARY GRS(SYS3),RESTART to bring SYS3 back into the complex

3. On SYS1, determine the problem, fix it, then reIPL the system with GRS=JOIN.
4. On SYS4, reIPL the system with GRS=JOIN.

At this point, the recovery process is complete.

Appendix C. Accessibility

Accessible publications for this product are offered through IBM Documentation (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the Contact the z/OS team web page (www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

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

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

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
Site Counsel
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 document 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

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

COPYRIGHT LICENSE:

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

Index

Special Characters

(resource name list)
    overview 17

A

abends
    monitor tool 62
ACCELSYS
    recommendations 124
ACCELSYS option
    effect on RESMIL 170
    effect on response time 168
    specifying 134
ACCELSYS value
    use in tuning 170
accessibility
    contact IBM 217
address space state data RMF report 171
affected resource 33
alternate link
    global resource serialization 207
    recovery
        global resource serialization 207
alternate links
    in complex design 128
    selection during initialization 131
    with ring acceleration 134
API 5
assistive technologies 217
asynchronous I/O
        deadlock
    example 8
automatically rebuilding a disrupted ring
    specifying 136
automatically rejoining the ring
    specifying 137
automatically switchable tape devices 39
average response time 167

B

backup 130
block diagram
    use in defining a complex to MVS 139
building the complex in a sysplex 105, 147

C

catalogs 41
CFRM 97
changing the RNLs 107, 108, 150
checkpoint data set
    JES2 37
    JES3 37
CICS (Customer Information Control System) 35
communication rate 169
complex
    building 147
    decision making 120
    defining 124
    designing 119
    diagnosis 176
    installation 163
    matching sysplex 120
    mixed 125
    multiple 119
    operation 147
    overview 147
    ring 12
    star 12
configuration
    check before IPL 105, 151
    considerations in complex 131
    guidelines 95, 119, 130
    recommendation summary
        in a sysplex 105
        systems not in a sysplex 150
configuration check 147
configuration diagram 139
connectivity
    recovery considerations 128
contact
    z/OS 217
contention for data set ENQs
    monitoring 199
controlling CTC links 161
converting reserves
    effect on performance 197
    guidelines 7
    recommendations 43
couple data set 96
coupling facility
    loss of connectivity 103
critical system
    RESTART value in two-system complex 136
cross system processing 93
CTC 91, 102
CTC (channel-to-channel)
    checking connections before IPL 151
    device numbers 131
    guidelines for complex design 126, 129
    link 205
    link failure 129
    option 131
    recovery
        link 205
CTC link 14
CTIGRS00 99
CTRACE
    parameter 99
CTRACE parameter

CTRACE parameter *(continued)*
specifying 138
cycle time
computing 168
effect on tuning 196

D

DAE (dump analysis elimination)
single-system 35
sysplex 35
data integrity exposure
shared DASD connections 105, 147, 151
data set
CFRM 97
couple 96
naming conventions 18
restrictions 18
sysplex couple 97
temporary 68
data set ENQ contention monitor
JCL to assemble source module 199
JCL to link-edit source module 200
data transfer rate 169
DB2 (DATABASE 2) 36
deadlock
example 8
defining the complex to MVS 139, 143–145
defining the RNLs
in SYS1.PARMLIB 45
defining the sysplex couple data set 96
delaying jobs
displaying 108, 150
DFSMShsm 36
diagnosis
aids 175
avoiding errors 102
complex operation 176
connectivity in coupling facility 103
contention states 182
coupling facility availability 175
coupling facility structure 103
disruption 177
GTF 171
guidelines 175
intersystem communication 175
monitor tool 49, 80
possible error scenarios 101
procedures 175
resource allocation 175
ring acceleration 176
RSA 176
software 175
TOLINT value 177
tuning 175
tuning the ring 176
XCF 177
DISPLAY GRS command
contention 34
delaying jobs 108, 150
explanation of fields 107, 149, 154
suspended jobs 108, 150
to display ring status 153
to display ring status in a complex 106
DISPLAY GRS command *(continued)*
to display ring status in a sysplex 148
use in problem analysis 155
disrupted ring
automatically rebuilding 136
disruption
definition 127
double serialization deadlock
example 8
DSECT 85
dynamic exit
additional information 20
RNL processing 20
dynamic RNL 17
dynamic RNL processing 97

E

EDSORTED 86
ENQ and DEQ processing summary 30
ENQ delay 79
ENQ macro
use 10
ENQ monitor
planning RNL 34
ENQ resources allocation
problems 180
ENQ/RESERVE resources report 84
enqueue contention activity RMF report 171
entry type indicator 45
error messages 177
ESCON (Enterprise Systems Connection)
link configurations 126
operating modes 14
transmission delay 167
example
AUDIT MODIFY 58
complex configuration diagram 140
complex definition plan 143–145
contention states 182
ENQ/DEQ monitor main menu 52
filter 63
filter coding 65
GFLG 63
levels of serialization 5
minor name list 54
mixed complex diagram 141
monitor jobname list 54
monitor option 1 53
monitor option 2 55
monitor option 3 55
monitor option 4 57
monitor volume active RESERVE list 56
naming conventions 4
resource name 4
resource name list table 55
resource usage 68
ring concept 13
scope 4
star complex 14
star parmlib definition 100
three system sysplex 181
volume list 55
volume reserve report 68

examples
purging a quiesced system 158
purging an active system 158
quiescing a system 156
restarting a system 160
excluding requests from RNL
description 24
exit search routine 21

F

feedback xvii
filter facility 62
filter list 57
fully-connected complex
definition 115
recovery considerations 127
required in a multisystem sysplex 127

G

generic resource name entry
definition 21
specifying in RNL entry 45
GENQRESP
example 195
measurement example 196
sample output 196
using 195
GFLG 63
global ENQ
star complex 31
global ENQ/DEQ 92
global resource
effect on tuning the complex 197
processing 116, 118
request rate 169
request response time 167
global resource request processing 116
global resource serialization
advantages 3
automatic restart
example 208
complex
purging system 204
quiesced system 212
reactivating quiesced system 212
definition 3
diagnostic commands
ANALYZE 181
ANALYZE,BLOCKER 180
ANALYZE,DEPENDENCY 180
ANALYZE,WAITER 181
CONTENTION 181
how it works 3
introduction 3
limiting requests 10
manual restart
example 209
quiesced system
reactivating 212
request 10
restart
global resource serialization *(continued)*
restart *(continued)*
two-system complex 211
ring rebuild 179
global resource serialization complex
defining to MVS 139
designing 119, 143–145
installing 163
migrating into a sysplex 163, 166
mixed complex 151, 152
operating 147
processing overview 4
selecting the resources 17, 46, 47
tuning 167, 197
global resource serialization problems
diagnosing 175
global resource serialization ring
definition 115
processing 115
global resource serialization star complex
designing 95
operating 105
GQSCAN
overview 92
GQSCAN macro
use 10
GRS Initialization Parameter 99
GRS LOCK STRUCTURE 92
GRS ring
tuning 176
GRS system parameter
purpose 120, 130
specifying 152
GRS=STAR 101
GRS=System Parameter 99
GRSCNF system parameter 106, 130, 148, 152
GRSCNFxx 99, 100, 109
GRSCNFxx member
Parmlib
example 209, 211
REJOIN parameter 208, 209, 211
RESTART parameter 208, 209, 211
restarting two-system complex 211
GRSCNFxx parmlib 99
GRSCNFxx parmlib member
example 121
overview of use 121, 130
parameters 130, 139
sample definition 143–145
worksheet 141, 143–145
worksheet for sysplex matches complex 125
GRSDEF Statement 99, 100
GRSQ
parameter 100
GRSRNL 17
GRSRNL system parameter 17, 106, 120, 130, 148
GRSRNLxx worksheet 46, 47
GTF (generalized trace facility)
for diagnosis 171
RSA 171
sample statements 171
trace CTC 171

I

IEA101A 99
IEASYSxx 109
IEASYSxx Member of parmlib 99
IEASYSxx member of SYS1.PARMLIB 106, 148, 152
IEASYSxx parmlib 101
IMS (Information Management System) 36
in-stream procedure variables 68
initialization actions 102
input search argument 21
installation
  new complex 163
  overview 163
installation procedure for complex 163
IPL (initial program load)
  configuration check before 105, 147, 151
  global resource serialization
    failure 205
    recovery 205
  indicators 148
IPL errors 101
ISG020I message 155, 156
ISG177E
  recovery procedure 177
ISG1787E
  recovery procedure 177
ISG236I 103
ISG300I 110
ISG309W 203
ISG318I 203
ISG322A 98
ISG323A 102, 203
ISG325I 103, 203
ISG331I 110
ISG3371I 98
ISG337I 110
ISG338W 98
ISG343I 149
ISG343I message 107
ISGAMCTM 87
ISGENQ 5
ISGENQ macro
  abilities 6
  services 7
ISGLOCK
  diagnosis 103
  rebuild processing 102
  star 101
ISGLOCK structure request
  processing 179
ISGNQXIT 20
ISGNQXITFAST 20
ISGQUERY 22
ISGQUERY macro 10
ISPF
  application 52
  filter list 57
  global resource ring menu 52
  major name list 53
  minor name list 54
  monitor tool 52
  volume entry list 57
  volume list 57
ISPF (Interactive System Productivity Facility)
  resource requests 36
IXLLOCK Macro 92

J

JCL
  monitor tool 51
  resource usage report 76
  to assemble the data set ENQ contention monitor source module 199
  to link-edit the data set ENQ contention monitor source module 200
  trace report 70
  volume reserve time report 72
JES2 37
JES3 37
job scheduling 9

K

keyboard
  navigation 217
  PF keys 217
  shortcut keys 217

L

lack of granularity
  example 7
latch contention 185
level of connectivity 127, 128
levels of serialization 5
link
  alternate 128
  in a sysplex 14
  outside of a sysplex 14
  placement 126
  primary 128
link configuration
  choosing 130
link connection
  effect on quiescing a system 156
local and global resource 5
local resource 4, 116
Lock Structure 98
log 67
logrec 38
loss of data
  monitor 80

M

major name 4
major name list 53
MATCHSYS
  option 131
  parameter 99
measurements used in tuning 171
messages
  monitor tool 60
  quiescing a system 156
migrating an existing complex into a sysplex 166

migration
existing complex 163
release links 166
SYS1 164
SYS2 164
SYS3 165
SYS4 165
Migration 109
Minimum options (MINOPS) 104, 139
minor name 4
minor name list 54
mixed complex
building 151, 152
processing options 130
monitor tool
abends 62
analysis aid 82
buffers requirements 80
command examples 58
control 57
diagnostic information 80
DSECT 85
ENQ/RESERVE/DEQ restrictions 79
filter 62
filter combinations for AUTHQ2 65
filter combinations for NCRESERVE 65
flag byte 63
guidelines 80
installation 49
JCL sample 51
jobname list 54
log 67
messages 60
minor name list 54
objectives 49
option 1 53
option 2 55
option 3 55
option 4 57
physical output 80
real storage requirement 80
reinitialize dataspace 60
resource name list table 55
resources used 80
security 50
set up 50
starting 52
termination 58
trace 82
using 49
utilization hints 81
volume list 55
monitoring data set ENQ contention 199
multiple complex 95, 119
MULTISYSTEM option of PLEXCFG 100

N

name of system in complex 131
name of system in sysplex 131
NAME Parameter 97
naming conventions 18
navigation
keyboard 217
new complex 163
normal operations
in a mixed complex 153, 161
in a sysplex 108, 148, 150
number
of communication links to provide 127
of systems in complex 125

O

operating the complex
building the complex in a sysplex 148
building/IPL
systems in a sysplex 105, 147, 148
systems not in a sysplex 151, 152
normal operations
systems in a sysplex 108, 148, 150
systems not in a sysplex 153
overview 147
Operating the complex
overview 105
operating the star complex
building/IPL 106
normal operations
systems in a star sysplex 106
operation examples
purging a quiesced system 158
purging an active system 158
quiescing a system 156
restarting a system 160
options in GRSCNFxx parmlib member 130, 139

P

parameters
in GRSCNFxx parmlib member 130, 139
in IEASYSxx parmlib member 106, 148, 151
systems in a complex 120
parm field 68
Parmlib GRSCNFxx member
REJOIN parameter
automatic restart 208
example 208, 209, 211
manual restart 209
restarting two-system complex 211
RESTART parameter
automatic restart 208
example 208, 209, 211
manual restart 209
restarting two-system complex 211
partially-connected complex
recovery considerations 128
path 14
pattern resource name 21
performance
checking 179
converting reserves 167
number of systems 169
planning aid
complex configuration diagram 140
GRSCNFxx worksheet 141
mixed complex diagram 141
planning aids 46, 47

PR/SM environment 119
primary link
  global resource serialization complex
    recovery 205
  selection during initialization 131
problems
  coupling facility availability 175
  intersystem communication breakdown 175
  resource allocation 175
  software 175
  tuning 175
  types of 175
procedure
  for installing a new complex 163
  for migrating a complex into a sysplex 163, 166
  installing the complex 163
processing a request for a resource 29, 116
processing options 139
purging a system 157, 159

Q

qname
  format in RNL entry 45
  use in resource request processing 21
quiesced system
  reactivating
    global resource serialization 212
quiescing a system 156, 157

R

RACF (Resource Access Control Facility) 38, 43
rebuilding a disrupted ring automatically 136
records layout 85
recovery
  definition 127
  global resource serialization
    manual restart 209
  reasons for 127
  ring disruption 177
recovery procedures 177
reduction programs
  output record fields 86
REJOIN option
  specifying 137
  using 159
REJOIN parameter
  Parmlib GRSCNFxx member
    automatic restart 208
    example 208, 209, 211
    manual restart 209
    restarting two-system complex 211
release links 166
reports
  ENQ/RESERVE resources report 84
  monitor trace report for multiple systems 83
  monitor trace report for single system 82
  PARM field 68
  region requirement 69
  resource report 67
  resource usage 68
  resource usage report 76
reports *(continued)*
  restriction 68
  sequential trace report 66
  time of max reserve time 85
  trace 70
  volume reserve time report 67, 68
  volume-reserve-time 69
  volumes reserve time report 83
request for a resource 21
request processing 134
request rate 169
reserve
  status 198
  system commands 198
RESERVE
  activity 67
  conversion restrictions 7
  conversion to global ENQ 31
  problems 7
  processing summary 30
  RNL processing 30
reserve conversion
  effect on tuning 197
  guidelines 44, 45
RESERVE conversion RNL
  definition 5
  description 23
  entry format 45
  exit interaction 23
  recommended entries 43
RESERVE inclusion RNL 17
RESERVE macro
  guidelines 7
  protecting resources 8
  use 10
residency time
  definition 116
  specifying 121, 131
  use in tuning 196
residency time (RESMIL) 169
RESMIL
  change value 196
  complex recommedations 132
  ENQ delay 57
  use in tuning 196
  values with ring acceleration 170
  values without ring acceleration 170
RESMIL option
  effect of processor power 132
  specifying 121, 131
  use in tuning 169
  value tables 170
  workload considerations 132
resource name
  generic 21, 45
  pattern 21, 45
  qname 21
  rname 21
  specific 21, 45
  symbolic 4
resource name list table 55
resource request processing 21, 29, 116
resource usage 68
response time

response time *(continued)*
calculate 168
calculating the average 167
contention 168
definition 167
global ENQ 197
measuring 195
number of systems 169
objective 169
restart
global resource serialization
automatic 208
manual 209
two-system complex 211
RESTART option
specifying 136
using 159
RESTART parameter
Parmlib GRSCNFxx member
automatic restart 208
example 208, 209, 211
manual restart 209
restarting two-system complex 211
restarting a system 159, 161
return codes
monitor tool 60
ring
global resource serialization
disrupting IPL 205
ring acceleration
description 134
effect on RESMIL 170
effect on response time 168
overview 117
specifying 134
ring disruption
recovery 177
ring formation 148, 152
ring processing
overview 115, 116
role of RSA-message 116
use of RNLs 21, 29
ring rebuild 179
ring status
displaying 148, 153
RMF (Resource Management Facility)
address space data 171
monitor I 171
monitor III 171
use in data selection 34
use in tuning 171
workload activity 171
rname
format in RNL entry 45
use in resource request processing 21
RNL (resource name list)
altering 20
candidates 42, 44, 45
catalogs 41
changing
dynamically 107
considerations 33
default 33
defaults 34, 45
RNL (resource name list) *(continued)*
defining
in parmlib 46, 47
definition 5
dynamic change 32
dynamic exit 20
dynamic processing 97
entry format 45
example 22
excluding requests 24
guideline 6
IBM supplied 45
indentifier 45
input search argument 21
ISGQUERY 22
matching 21
planning 45–47, 49
planning aid 49
processing 21, 29
processing sequence 29
recommended entries 42, 44, 45
RESERVE conversion 23
RESERVE request 30
restrictions 17
rule 17
rules 24
scanning 21, 22
selecting the resources 17, 46, 47
SMF records 33
SYS1.BRODCAST 39
SYS1.UADS 39
SYSTEM inclusion
guidelines 42
SYSTEMS exclusion 23
tape volume 39
TSO/E 40
tuning 49
types 5, 22
VIO journaling data set 40
VSAM 41
worksheet 46, 47
RSA-message
effect on performance 172
in ring processing 116

S

scope
changing 29, 32
request processing 22, 29
resource serialization 5
selecting 5
security considerations 50
sending to IBM
reader comments xvii
sequential trace report 70
serialization
choosing a method 15
methods 12
ring 12
star 12
serialization of resources
alternate method 17
SET GRSRNL command 17, 32, 107, 108, 149, 150

SETGRS Command 110
shared DASD connections
checking before IPL 105, 147, 151
shortcut keys 217
SMF records of RNL changes 33
SMS 36
specific resource name entry
specifying in RNL entry 45
split ring 159
SPZAPLIB RNL entry 43
star
advantages 15
star complex 95
star formation 106
STAR initialization 101
STAR initialization option 102
STAR Initialization Option 99, 102
star method 14
star status
displaying 106
starvation
example 7
structure size 98
summary of changes
z/OS MVS Planning: Global Resource Serialization xix
suspended jobs
displaying 108, 150
SYNCHRES
parameter 100
SYNCHRES option 9
Synchronous RESERVE feature 9
syntax
IXCL1DSU 96
syntax checker 45
SYS1.BRODCAST 39
SYS1.BRODCAST data set 42
SYS1.LOGREC data set 38
SYS1.UADS 39
SYS1.UADS data set 42
SYSDSN QNAME
effect on resource serialization 33
SYSIEWLP RNL entry 43
SYSIGGV2 RNL entry 43
SYSIKJBC 40
SYSIKJBC RNL entry 42
SYSIKJUA 40
SYSIKJUA RNL entry 42
SYSNAME system parameter 120, 130, 152
sysplex
migrating a complex into 166
not matching complex 125
number of systems in a 12
parameters 120
processing options 120
sysplex matches complex 12
system
purging
global resource serialization complex 204
reinitializing 204
SYSTEM inclusion
recommended RNL entries 42
SYSTEM inclusion RNL
definition 5
SYSTEM inclusion RNL *(continued)*
description 22
entry format 45
system logger 38
system name 120, 131
system parameters
relation to building the complex 120, 130
specifying 106, 148, 152
SYSTEMS exclusion RNL
definition 5
description 23
entry format 45
SYSTEMS inclusion RNL 17
SYSZRACF RNL entry 43

T

tape volume 39
tasks
comparing response time
steps 197
temporary data set
local resources 39
serialization 39
tolerance interval
specifying 133
TOLINT 122
TOLINT option
specifying 133
trace options 103
trace report 82
trademarks 222
transmission rate 169
TSO/E 40
tuning
ACCELSYS value 170
factors 168
process 170, 197
the complex 167, 197
two-system complex
restart 211
types of problems
discriminating between 175

U

user interface
ISPF 217
TSO/E 217

V

VIO journaling data set 40
volume list 55
volume RESERVE list 56
volume reserve time report 68, 72
VSAM (virtual storage access method)
data sets 41

W

wait state code
0A3 203

workload
considerations 132
delay monitor in RMF 171
RMF report 171
worksheet
GRSCNFxx 141
GRSRNLxx parmlib member 46, 47
planning GRSCNFxx 124

X

XCF 91, 99, 102
XCF (cross-system coupling facility)
checking connections before IPL 105, 147
XCF/XES connectivity
checking 179
XSYS Option 93

Z

z/OS MVS Planning: Global Resource Serialization
summary of changes xix

Product Number: 5650-ZOS

SA23-1389-50