DB2 for OS/390
Version 5

Data Sharing: Planning and Administration

SC26-8961-02

Note!

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

First Edition (June 1997)

This edition applies to Version 5 of IBM DATABASE 2 Server for OS/390 (DB2 for OS/390), 5655-DB2, and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

The technical changes for this edition are summarized under "Summary of Changes to this Book" in the Introduction. Specific changes are indicated by a vertical bar to the left of a change. A vertical bar to the left of a figure caption indicates that the figure has changed. Editorial changes that have no technical significance are not noted.

This softcopy version is based on the printed version of the book, and includes the changes indicated in the printed version by vertical bars. Additional changes made to this softcopy version of the manual since the hardcopy manual was published are indicated by the hash (#) symbol in the left-hand margin.

Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in
this document. The furnishing of this document does not give you any license to
these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Licensees of this program who wish to have information about it for the purpose of
enabling (1) the exchange of information between independently created programs
and other programs (including this one) and (2) the mutual use of the information
that has been exchanged, should contact:

IBM Corporation
IBM Corporation
J74/G4
555 Bailey Avenue
P.O. Box 49023
San Jose, CA 95161-9023

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

Programming Interface Information

This book is intended to help you plan for the use of DB2 data sharing.

This book also documents General-use Programming Interface and Associated Guidance Information and Product-sensitive Programming Interface and Associated Guidance Information provided by DB2 for OS/390 (DB2).

General-use programming interfaces allow the customer to write programs that obtain the services of DB2.

General-use Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

General-use programming interface

General-use Programming Interface and Associated Guidance Information ...

End of General-use programming interface

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of DB2. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, by the following marking:

Product-sensitive programming interface

Product-sensitive Programming Interface and Associated Guidance Information ...

End of Product-sensitive programming interface

Trademarks

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

C/370
CICS
CICS/ESA
CICS/MVS
CICSPlex SM
DATABASE 2
DB2
DFSMS
DFSMShsm
DFSMS/MVS
DRDA
Enterprise System/3090
Enterprise System/9000
ES/3090
IBM
IMS
IMS/ESA
MVS/ESA
MVS/SP
NetSP
OS/2
OS/390
Parallel Sysplex
QMF
PR/SM
RACF
RAMAC
Resource Measurement Facility
RMF
S/390
Sysplex Timer
System/390
VTAM

Throughout the library, the DB2 licensed program and a particular DB2 subsystem are each referred to as “DB2.” In each case, the context makes the meaning clear. The term *MVS* is used to represent MVS/Enterprise Systems Architecture (MVS/ESA). *CICS* is used to represent CICS/MVS and CICS/ESA; *IMS* is used to represent IMS/ESA; *C* and *C language* are used to represent the C/370 and C/C++ for MVS/ESA programming languages. *COBOL* is used to represent IBM COBOL for MVS & VM, OS/VS COBOL, VS COBOL II, and COBOL/370 programming languages.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Chapter 1. Introduction to this Book and the DB2 Library

This chapter contains specific information about this book, a general overview of the library for the licensed program DB2 for OS/390and a summary of changes to DB2.

Who Should Read This Book

This book is primarily intended for system and database administrators who are responsible for planning and implementing DB2 data sharing. Many of the task descriptions in this book assume the user is already familiar with administering DB2 without data sharing. See *Administration Guide* for any concepts not explained in this book.

How This Book Is Organized

This book contains several chapters on planning for and using DB2 data sharing:

"Chapter 2. Introduction to DB2 Data Sharing" on page 21 gives an overview of the advantages of data sharing, how it works, and the hardware and software requirements.

"Chapter 3. Planning for DB2 Data Sharing" on page 45 describes hardware and software configurations, availability planning, and a recommended naming convention.

| "Chapter 4. Installing and Enabling DB2 Data Sharing" on page 85 includes recommendations for subsystem parameters, procedures for creating a data sharing group, whether you are installing or migrating. Details on each step in the procedures are in Section 2 of *Installation Guide*. This chapter also includes information about disabling data sharing, falling back, or removing a member from the data sharing group.

"Chapter 5. Communicating with a Data Sharing Group" on page 123 describes how you can set up your data sharing group in a distributed network.

"Chapter 6. Operating with Data Sharing" on page 145 describes how you can control DB2 with commands, monitor activity in the group, and back up and recover data.

"Chapter 7. Performance Monitoring and Tuning" on page 193 describes how you can reduce locking contention and how to tune group buffer pools to | improve the performance of applications that run in the data sharing group. It also includes the majority of information about Sysplex query parallelism.

| This book also contains an appendix with reference information about naming.

How to Use this Book

This book is the main source of information about DB2 data sharing. It can be used to educate yourself about data sharing and to do many of the tasks associated with data sharing.

However, there are many tasks associated with data sharing, especially those of setting up the hardware and software environment for the parallel Sysplex, that require the use of other product libraries, such as MVS.

For installing DB2, use this book with *Installation Guide* to do initial planning and to develop your installation strategy. Detailed installation procedures are in *Installation Guide*. Exceptions and deviations from those procedures are noted in this book.

How to Use the DB2 Library

Titles of books in the library begin with DB2 for OS/390 Version 5. However, references from one book in the library to another are shortened and do not include the product name, version, and release. Instead, they point directly to the section that holds the information. For a complete list of books in the library, and the sections in each book, see the bibliography at the back of this book.

Throughout the library, the DB2 for OS/390 licensed program and a particular DB2 for MVS/ESA subsystem are each referred to as “DB2.” In each case, the context makes the meaning clear.

The most rewarding task associated with a database management system is asking questions of it and getting answers, the task called *end use*. Other tasks are also necessary—defining the parameters of the system, putting the data in place, and so on. The tasks associated with DB2 are grouped into the following major categories (but supplemental information relating to all of the below tasks for new releases of DB2 can be found in *Release Guide*):

Installation: If you are involved with DB2 only to install the system, *Installation Guide* might be all you need.

If you will be using data sharing then you also need this book, for data sharing installation considerations.

End use: End users issue SQL statements to retrieve data. They can also insert, update, or delete data, with SQL statements. They might need an introduction to SQL, detailed instructions for using SPUFI, and an alphabetized reference to the types of SQL statements. This information is found in *Application Programming and SQL Guide* and *SQL Reference*.

End users can also issue SQL statements through the Query Management Facility (QMF) or some other program, and the library for that program might provide all the instruction or reference material they need. For a list of some of the titles in the QMF library, see the bibliography at the end of this book.

Application Programming: Some users access DB2 without knowing it, using programs that contain SQL statements. DB2 application programmers write those programs. Because they write SQL statements, they need *Application Programming and SQL Guide*, *SQL Reference*, and *Call Level Interface Guide and Reference* just as end users do.

Application programmers also need instructions on many other topics:

- How to transfer data between DB2 and a host program—written in COBOL, C, or FORTRAN, for example
- How to prepare to compile a program that embeds SQL statements
- How to process data from two systems simultaneously, say DB2 and IMS or DB2 and CICS

- How to write distributed applications across platforms
- How to write applications that use DB2 Call Level Interface to access DB2 servers
- How to write applications that use Open Database Connectivity (ODBC) to access DB2 servers
- How to write applications in the Java programming language to access DB2 servers

The material needed for writing a host program containing SQL is in *Application Programming and SQL Guide* and *Application Programming Guide and Reference for Java*. The material needed for writing applications that use DB2 Call Level Interface or ODBC to access DB2 servers is in *Call Level Interface Guide and Reference*.

For handling errors, see *Messages and Codes*.

Information about writing applications across platforms can be found in *Distributed Relational Database Architecture: Application Programming Guide*.

System and Database Administration: *Administration* covers almost everything else. *Administration Guide* divides those tasks among the following sections:

- Section 2 (Volume 1) of *Administration Guide* discusses the decisions that must be made when designing a database and tells how to bring the design into being by creating DB2 objects, loading data, and adjusting to changes.
- Section 3 (Volume 1) of *Administration Guide* describes ways of controlling access to the DB2 system and to data within DB2, to audit aspects of DB2 usage, and to answer other security and auditing concerns.
- Section 4 (Volume 1) of *Administration Guide* describes the steps in normal day-to-day operation and discusses the steps one should take to prepare for recovery in the event of some failure.
- Section 5 (Volume 2) of *Administration Guide* explains how to monitor the performance of the DB2 system and its parts. It also lists things that can be done to make some parts run faster.

In addition, the appendixes in *Administration Guide* contain valuable information on DB2 sample tables, National Language Support (NLS), writing exit routines, interpreting DB2 trace output, and character conversion for distributed data.

If you are involved with DB2 only to design the database, or plan operational procedures, you need *Administration Guide*. If you also want to carry out your own plans by creating DB2 objects, granting privileges, running utility jobs, and so on, then you also need:

- *SQL Reference*, which describes the SQL statements you use to create, alter, and drop objects and grant and revoke privileges
- *Utility Guide and Reference*, which explains how to run utilities
- *Command Reference*, which explains how to run commands

If you will be using data sharing, then you need this book.

Additional information about system and database administration can be found in *Messages and Codes*, which lists messages and codes issued by DB2, with explanations and suggested responses.

Diagnosis: Diagnosticians detect and describe errors in the DB2 program. They might also recommend or apply a remedy. The documentation for this task is in *Diagnosis Guide and Reference* and *Messages and Codes*.

How to Obtain DB2 Information

DB2 on the Web

Stay current with the latest information about DB2. View the DB2 home page on the World Wide Web. News items keep you informed about the latest enhancements to the product. Product announcements, press releases, fact sheets, and technical articles help you plan your database management strategy. Technical professionals can access DB2 publications on the Web and follow links to other Web sites with more information about DB2 family and OS/390 solutions. Access DB2 on the Web with the following URL:

http://www.ibm.com/software/db2os390

DB2 Publications

The DB2 publications are available in both hardcopy and softcopy format. Using online books on CD-ROM, you can read, search across books, print portions of the text, and make notes in these BookManager books. With the appropriate BookManager READ product or IBM Library Readers, you can view these books on the MVS, VM, OS/2, DOS, AIX and Windows platforms.

When you order DB2 Version 5, you are entitled to one copy of the following CD-ROM, which contains the DB2 licensed book for no additional charge:

DB2 Server for OS/390 Version 5 Licensed Online Book, LK2T-9075.

You can order multiple copies for an additional charge by specifying feature code 8207.

When you order DB2 Version 5, you are entitled to one copy of the following CD-ROM, which contains the DB2 and DATABASE 2 Performance Monitor online books for no additional charge:

DB2 Server for OS/390 Version 5 Online Library, SK2T-9092

You can order multiple copies for an additional charge through IBM's publication ordering service.

Periodic updates will be provided on the following collection kit available to licensees of DB2 Version 5:

IBM Online Library Transaction Processing and Data Collection, SK2T-0730

SK2T-9092 will be superseded by SK2T-0730 when updates to the online library are available.

In some countries,including the United States and Canada, you receive one copy of the collection kit at no additional charge when you order DB2 Version 5. You will automatically receive one copy of the collection kit each time it is updated, for no additional charge. To order multiple copies of SK2T-0730 for an additional charge, see “How to Order the DB2 Library” on page 7. In other countries, updates will be available in displayable softcopy format in the IBM Online Book Library Offering (5636–PUB), SK2T-0730 IBM Online Library Transaction Processing and Data Collection at a later date.

See your IBM representative for assistance in ordering the collection.

DB2 Server for OS/390 books are also available for an additional charge on the following collection kits, which contain online books for many IBM products:

IBM Online Library MVS Collection, SK2T-0710, in English

Online Library Omnibus Edition OS/390 Collection, SK2T-6700, in English

IBM Online Library MVS Collection Kit, SK88-8002, in Japanese, for viewing on DOS and Windows platforms

How to Order the DB2 Library

You can order DB2 publications and CD-ROMs through your IBM representative or the IBM branch office serving your locality. If you are located within the United States or Canada, you can place your order by calling one of the toll-free numbers :

- In the U.S., call 1-800-879-2755.
- In Canada, call 1-800-565-1234.

To order additional copies of licensed publications, specify the SOFTWARE option. To order additional publications or CD-ROMs, specify the PUBLICATIONS & SLSS option. Be prepared to give your customer number, the product number, and the feature code(s) or order numbers you want.

DB2 Classes

IBM Education and Training offers a wide variety of classroom courses to help you quickly and efficiently gain DB2 expertise. Classes are scheduled in cities all over the world. For more information, including the current local schedule, please contact your IBM representative.

Classes can also be taught at your location, at a time that suits your needs. Courses can even be customized to meet your exact requirements. The diagrams below show the DB2 curriculum in the United States. *Enterprise Systems Training Solutions, GR28-5467* describes these courses. You can inquire about or enroll in them by calling 1-800-IBM-TEACh (1-800-426-8322).

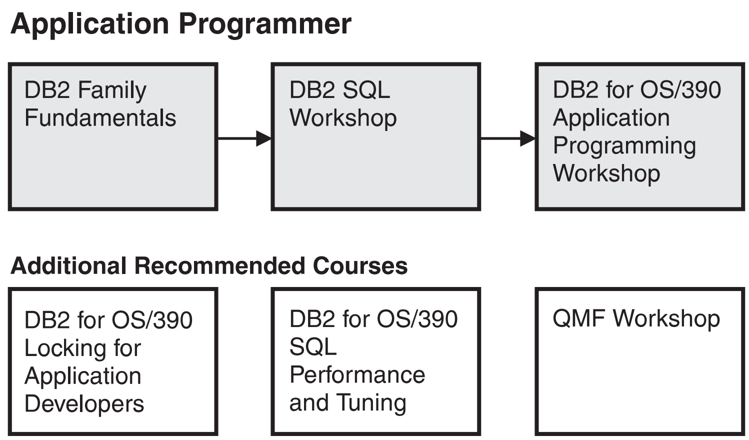

Figure 1. Application Programmer Curriculum

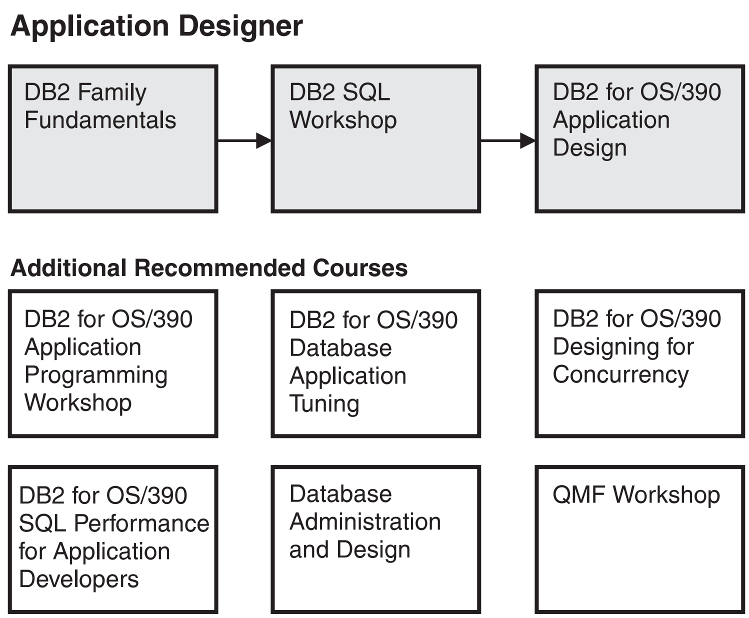

Figure 2. Application Designer Curriculum

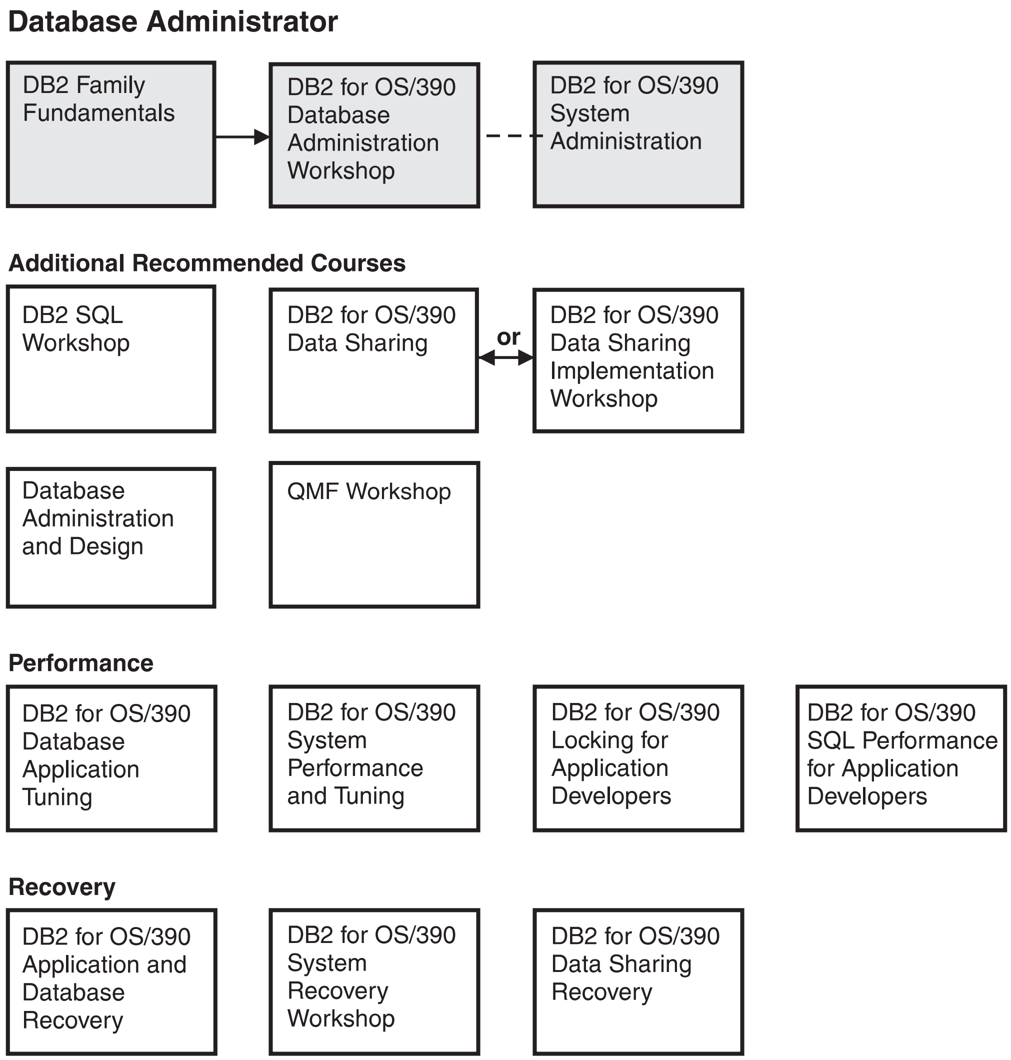

Figure 3. Database Administrator Curriculum

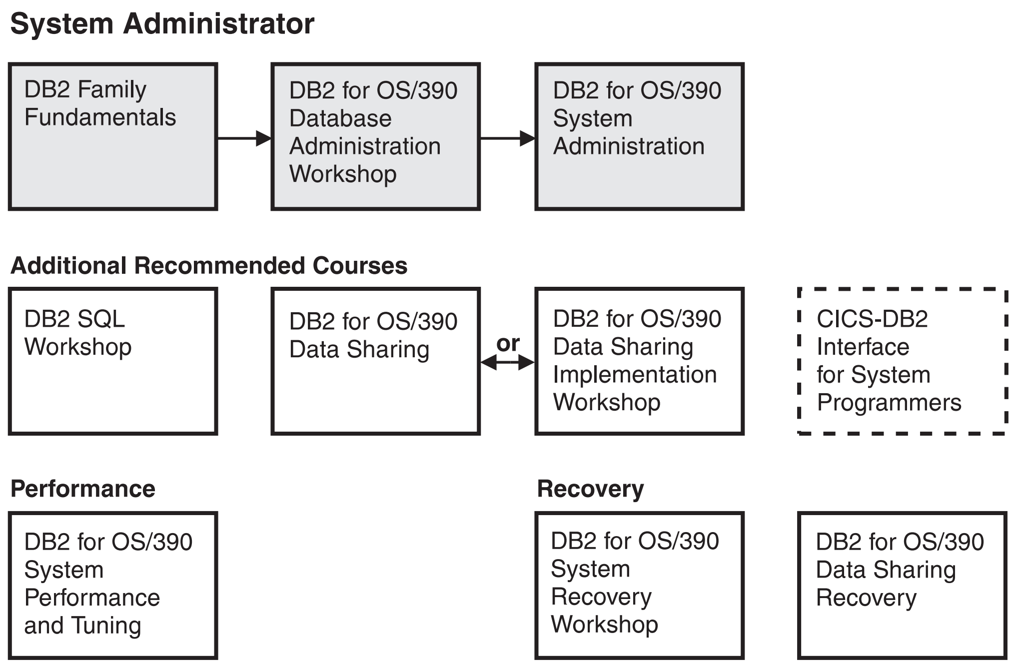

Figure 4. System Administrator Curriculum

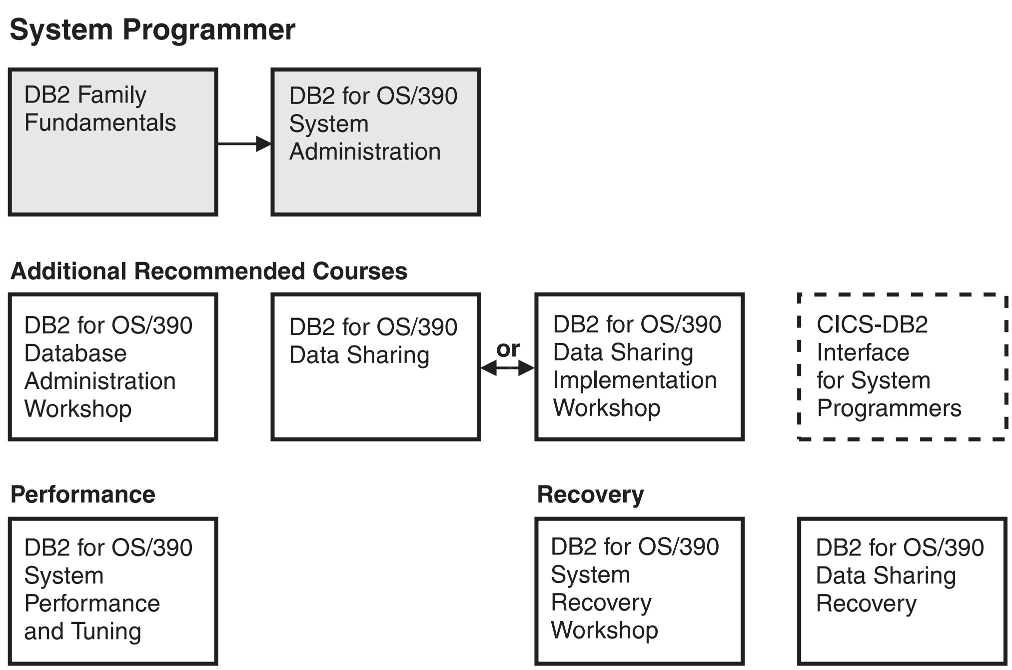

Figure 5. System Programmer Curriculum

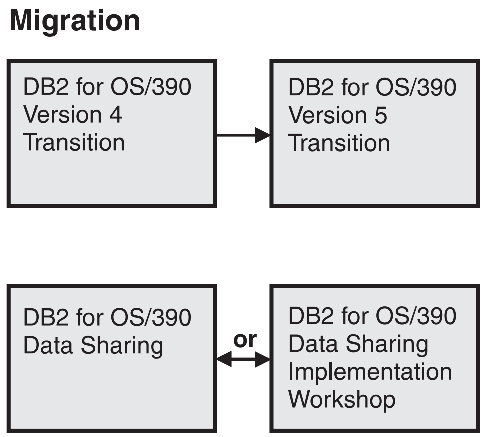

Figure 6. Migration Curriculum

Summary of Changes to DB2 for OS/390 Version 5

DB2 for OS/390 Version 5 delivers a database server solution for OS/390. Version 5 supports all functions available in DB2 for MVS/ESA Version 4 plus enhancements in the areas of performance, capacity, and availability, client/server and open systems, and user productivity.

If you are currently using DB2, **you can migrate only from a DB2 for MVS/ESA Version 4 subsystem**. This summary gives you an overview of the differences to be found between these versions.

Server Solution

OS/390 retains the classic strengths of the traditional MVS/ESA operating system, while offering a network-ready, integrated operational environment.

The following features work directly with DB2 for OS/390 applications to help you use the full potential of your DB2 subsystem:

- Net.Data for OS/390
- DB2 Installer
- DB2 Estimator for Windows
- DB2 Visual Explain
- Workstation-based Performance Analysis and Tuning
- DATABASE 2 Performance Monitor

Net.Data for OS/390

Net.Data provides support for Internet access to DB2 data through a Web server. Applications built with Net.Data make data stored in any DB2 server more accessible and useful. Net.Data Web applications provide continuous application availability, scalability, security, and high performance.

This no charge feature can be ordered with DB2 Version 5 or downloaded from Internet. The Net.Data URL is:

http://www.ibm.com/software/data/net.data/downloads.html

DB2 Installer

DB2 Installer offers the option to install DB2 on an OS/2 workstation. Now, you can use a friendly graphical interface to complete installation tasks easily with DB2 Installer.

This function is delivered on CD-ROM with DB2 Visual Explain.

DB2 Estimator for Windows

DB2 Estimator provides an easy-to-use capacity planning tool. You can estimate the sizes of tables and indexes, and the performance of SQL statements, groups of SQL statements (transactions), utility runs, and groups of transactions (capacity runs). From a simple table sizing to a detailed performance analysis of an entire DB2 application, DB2 Estimator saves time and lowers costs. You can investigate the impact of new or modified applications on your production system, *before* you implement them.

This no charge feature can be ordered with DB2 Version 5 or downloaded from the Internet. From the internet, use the IBM Software URL:

http://www.ibm.com/software/

From here, you can access information about DB2 Estimator using the download function.

DB2 Visual Explain

DB2 Visual Explain lets you tune DB2 SQL statements on an OS/2 workstation. You can see DB2 EXPLAIN output in a friendly graphical interface and easily access, modify, and analyze applications with DB2 Visual Explain.

Workstation-based Performance Analysis and Tuning

The new workstation-based Performance Analysis and Tuning function simplifies system administration. You can access statistical data to help you analyze and improve system performance. This function works with the optional DB2 PM feature to provide full analysis and tuning functionality.

DATABASE 2 Performance Monitor (DB2 PM)

DB2 PM lets you monitor, analyze, and optimize the performance of DB2 Version 5 and its applications. An online monitor, for both host and workstation environments, provides an immediate "snap-shot" view of DB2 activities and allows for exception processing while the system is operational. The workstation-based online monitor can connect directly to the Visual Explain function of the DB2 base product.

DB2 PM also offers a history facility, a wide variety of customizable reports for in-depth performance analysis, and an EXPLAIN function to analyze and optimize SQL statements. For more information, see *DB2 PM for OS/390 General Information* .

This feature can be ordered with DB2 Version 5.

Performance

Sysplex Query Parallelism

The increased power of Sysplex query parallelism in DB2 for OS/390 Version 5 allows DB2 to go far beyond DB2 for MVS/ESA Version 4 capabilities; from the ability to split and process a single query within a DB2 subsystem to processing that same query across many different DB2 subsystems in a data sharing group.

The advances this release offers in scalable query processing let you process queries quickly while accommodating the potential growth of data sharing groups and the increasing complexity of queries.

Prepared Statement Caching

DB2 reduces the cost of duplicate prepares for the same dynamic SQL statement by saving them in a cache. Now, different application processes can share prepared statements and they are preserved past the commit point. This performance improvement offers the most benefit for:

- Client/server applications that frequently use dynamic SQL for repeated execution of SQL statements
- Relatively short dynamic SQL statements for which PREPARE cost accounts for most of the CPU expended

Reoptimization

When host variables, parameter markers, or special registers were used in previous releases, DB2 could not always determine the best access path because the values for these variables were unknown. Now, you can tell DB2 to reevaluate the access path at run time, after these values are known. As a result, queries can be processed more efficiently, and response time is improved.

Faster Transactions and Batch

- Caching of package authorization improves performance at run time for remote packages and applications that use pattern-matching characters in a package list.
- You can define a table space to use ***selective partition locking***, which can reduce locking costs for applications that do partition-at-a-time processing. It also can reduce locking costs for certain data sharing applications that rely on an affinity between members and data partitions.
- A new standalone utility lets you preformat active logs.
- With LOAD and REORG, you can preformat data sets up to the high allocated RBA, which can make processing for sequential inserts more predictable.

Faster Utilities

- LOAD and REORG jobs run faster and more efficiently with enhanced index key sorting that reduces CPU and elapsed time, and an inline copy feature that lets you make an image copy without a separate copy step.
- New REORG options let you select rows to discard during a REORG and, optionally, write the discarded records to a file.
- When you run the REBUILD, RECOVER, REORG, or LOAD utility on DB2-managed indexes or table spaces, a new option lets you logically reset and reuse the DB2-managed objects.

- RECOVER INDEX and LOAD run faster on large numbers of rows per page.
- Sampling support for RUNSTATS reduces the processing required to collect nonindexed column statistics.
- BSAM striping improves the I/O capability of DB2 utilities.

Other Performance Enhancements

- There are several significant performance enhancements to data sharing, including selective partition locking, the MAXROWS option, and several optimizations to reduce data sharing overhead.
- DB2 installations that run in the OS/390 Version 2 Release 6 environment can now have as many as (approximately) 25 000 open DB2 data sets at one time. The maximum number of open data sets in earlier releases of OS/390 is 10000.
- You can easily alter the length of variable-length character columns using the new ALTER COLUMN clause of the ALTER TABLE statement.
- SQL CASE expressions let you eliminate queries with multiple UNIONs and improve performance by using only one table scan.
- You can collect a new statistic on concatenated index keys to improve the performance of queries with correlated columns. The statistic lets DB2 estimate the number of rows that qualify for the query more accurately, and select access paths more efficiently.
- DB2 scans partitions more efficiently and allows scans during parallel processing.
- Query enhancements include the ability to:
  - Use indexes for joins on string columns that have different lengths
  - Use an index to access predicates with noncorrelated IN subqueries
- Noncolumn expressions in simple predicates are evaluated at stage 1 and can be indexable.

Increased Capacity

DB2 for OS/390 Version 5 introduces the concept of a *large* partitioned table space. Defining your table space as large allows a substantial capacity increase: to approximately one terabyte of data and up to 254 partitions. In addition to accommodating growth potential, large partitioned table spaces make database design more flexible, and can improve availability.

Improved Availability

Online REORG

DB2 for OS/390 Version 5 adds a major improvement to availability with *Online REORG*. Now, you can avoid the severe availability problems that occurred while offline reorganization of table spaces restricted access to read only during the unload phase and no access during reload phase of the REORG utility. Online REORG gives you full read and write access to your data through most phases of the process with only very brief periods of read only or no access.

Data Sharing Enhancements

- Version 5 provides continuous availability with group buffer pool duplexing. Prior releases of DB2 rely on DASD and the merged recovery logs to recover group buffer pool (GBP) data that is lost if a coupling facility fails. With group buffer pool duplexing, DB2 writes changed pages to both a *primary GBP* and a *secondary GBP.* Overlapped writes to the GBPs provide good performance and eliminate the writes to DASD.
- Group buffer pool rebuild makes coupling facility maintenance easier and improves access to the group buffer pool during connectivity losses.
- Automatic group buffer pool recovery accelerates GBP recovery time, eliminates operator intervention, and makes data available faster when GBPs are lost because of coupling facility failures.
- Improved restart performance for members of a data sharing group reduces the impact of retained locks by making data available faster when a group member fails.
- Changes to traces and DISPLAY GROUPBUFFERPOOL output improve monitoring.

Tracker site for disaster recovery

You can set up a tracker site that shadows the activity of a primary site, and eliminate the need to constantly ship image copies.

Client/Server and Open Systems

Native TCP/IP Network Support

DB2's support of TCP/IP networks allows DRDA clients to connect directly to DDF and eliminate the gateway machine. In addition, customers can now use asynchronous transfer mode (ATM) as the underlying communication protocol for both SNA and TCP/IP connections to DB2.

Stored Procedures

- Return multiple SQL result sets to local and remote clients in a single network operation.
- Receive calls from applications that use standard interfaces, such as Open Database Connectivity** (ODBC) and X/Open** Call Level Interface, to access data in DB2 for OS/390.
- Run in an enhanced environment. DB2 supports multiple stored procedures address spaces managed by the MVS Workload Manager (WLM). The WLM environment offers efficient program management and allows WLM-managed stored procedures to run as subprograms and use RACF security.
- Use individual MVS dispatching priorities to improve stored procedure scheduling.
- Access data sources outside DB2 with two-phase commit coordination.
- Use an automatic COMMIT feature on return to the caller that reduces network traffic and the length of time locks are held.
- Have the ability to invoke utilities, which means you can now invoke utilities from an application that uses the SQL CALL statement.

- Support IMS Open Database Access (ODBA). Now a DB2 stored procedure can directly connect to IMS DBCTL and access IMS data.

Dynamic Query and Network Performance

Improvements for DRDA Applications

- Reduced processing costs for block fetch operations
- DRDA support for OPTIMIZE FOR n ROWS on SELECT
- Faster dynamic SQL queries and reduced processing costs for VTAM network operations
- Reduced message traffic for dynamic SQL SELECT statements

Improved Application Portability

- DB2 for OS/390 Version 5 introduces the DB2 Call Level Interface (CLI) to MVS/ESA. Unlike applications that use embedded SQL to access DB2 data, applications that choose CLI are not tied to a precompiler, packages, or a plan.

  Workstation and desktop applications use standard interfaces, such as Open Database Connectivity (ODBC), to access relational data. Standard interfaces need one version of an application to access many data sources. Now, you can port UNIX workstation and PC desktop applications to DB2 for OS/390 and exploit the CLI (ODBC) capabilities without modification. In addition, applications can issue ODBC or CLI calls from within a stored procedure.
- You can now access DB2 for OS/390 databases in your Java applications. You can use DB2 Connect Java Database Connectivity (JDBC) for your dynamic SQL applications, or SQLJ for your static SQL applications.
- DB2 adds DRDA support for the DESCRIBE INPUT statement to improve performance for many ODBC applications.
- Now, you can write multithreaded DB2 CLI applications, and restrictions on connection switching no longer exist.
- DB2 now provides ASCII table support for clients and servers across platforms. This support reduces the cost of translation between EBCDIC and ASCII encoding schemes. ASCII table support also offers an alternative to writing field procedures that provide the ASCII sort sequence, which improves performance.

Improved Security

- DB2 for OS/390 supports Distributed Computing Environment (DCE) for authenticating remote DRDA clients. DCE offers the following benefits:
  - Network security: By providing an encrypted DCE ticket for authentication, remote clients do not need to send an MVS password in readable text.
  - Simplified security administration: End users do not need to maintain a valid password on MVS to access DB2; instead, they maintain their DCE password only.
- New descriptive error codes help you determine the cause of network security errors.
- You can change end user MVS passwords from DRDA clients.

User Productivity

Improved SQL Compatibility

DB2 conforms to the ANSI/ISO SQL entry level standard of 1992. Application programmers can take advantage of a more complete set of standard SQL to use across the DB2 family to write portable applications. New SQL function includes:

- More check options for view definitions.
- Foreign keys that reference UNIQUE keys as well as PRIMARY keys.
- An extension to GRANT that lets the REFERENCES privilege apply to a list of columns.
- A new delete rule, NO ACTION, that you can use to define referential constraints for self-referencing tables.
- SQL CASE expressions provide the capability to create conditional logic wherever an expression is allowed.
- SQL temporary tables allow application programs to easily create and use temporary tables that store results of SQL transactions without logging or recovery.

New Access Choice

A new attachment facility, the Recoverable Resource Manager Services attachment facility, improves access in a client/server environment. It coordinates two-phase commit processing between DB2 and other participating resource managers in any MVS application environment. Other key features include the ability for multiple users to run in a single address space, thread reuse, and moving threads between MVS tasks.

Image Copy Enhancements

The COPY, LOAD, and REORG utilities provide:

- Features of the COPY utility that help you quickly determine what type of image copy to take, when to take it, and let DB2 automatically take it for you.
- Inline copy in LOAD and REORG that lets you create an image copy while improving data availability.

Improved Integration of C++ and IBM COBOL for MVS & VM Support

It is easier for application programmers to use object-oriented programming techniques in their DB2 applications. DB2 for OS/390 Version 5 adds COBOL and C++ languages as options on installation panels, DB2I panels, the DSNH command, and DCLGEN.

Other Usability Enhancements

- To prevent long running units of work and to help avoid unnecessary work during the recovery phase of restart, DB2 issues new warning messages at an interval of your choice.
- A new special register for decimal precision provides better granularity, so that applications that need different values for decimal precision can run in the same DB2 subsystem.

- Trace records for IFCID 0022 now include most information in the PLAN_TABLE.
- An increase from 127 to 255 rows on a page improves table space processing and eliminates the need for compression.
- Install SYSOPR can recover objects using the START DATABASE command.
- A filtering capability for DISPLAY BUFFERPOOL limits statistics information to a specified set of page sets.
- You can enter comments within the SYSIN input stream for DB2 utilities.

Summary of Changes to this Book

The information that was formerly in Chapter 3 has been split into two chapters:

- "Chapter 3. Planning for DB2 Data Sharing" on page 45

  See this chapter for enhancements made to group buffer pool recovery, including DB2's ability to dynamically rebuild or recover group buffer pools. See also the latest recommendations for IRLM storage.

- "Chapter 4. Installing and Enabling DB2 Data Sharing" on page 85

  This chapter contains information about how to migrate a data sharing group to a new release, and how to handle the *coexistence* issues that occur until all members have been migrated.

"Chapter 5. Communicating with a Data Sharing Group" on page 123 contains information about setting up a data sharing group to use Transmission Control Protocol/Internet Protocol (TCP/IP) for its DRDA processing.

"Chapter 6. Operating with Data Sharing" on page 145 contains changes to the recovery scenarios caused by DB2's ability to dynamically rebuild group buffer pools for which there was a 100 percent loss of connectivity. Scenarios are also changed because of DB2's ability to automatically recover from a group buffer pool structure failure.

"Chapter 7. Performance Monitoring and Tuning" on page 193 contains information about using Sysplex query parallelism. See this chapter also for the latest recommendations for performance.

The information that was previously in the following appendixes has been removed from this book:

- Appendix B: Commands

  See *Command Reference* for complete information about commands.

- Appendix C: Utilities

  See *Utility Guide and Reference* for complete information about utilities.

- Appendix D: Stand-alone Log Services

  See Appendix C (Volume 2) of *Administration Guide* for complete information about stand-alone log services.

- Appendix E: SQL Statements

  See *SQL Reference* for complete information about SQL statements.

- Appendix F: Sysplex Transaction Program

  See *Reference for Remote DRDA Requesters and Servers* for complete information about the Sysplex transaction program.

- Appendix G: New and Changed IFCIDs for Data Sharing

  See *Release Guide* for information about new and changed IFCIDs in Version 5. See DSN510.SDSNMACS for descriptions of all IFCIDs.

Chapter 2. Introduction to DB2 Data Sharing

With the data sharing function of the licensed program DB2 for OS/390, applications running on more than one DB2 subsystem can read from and write to the same set of data concurrently.

DB2 subsystems that share data must belong to a DB2 data sharing group. A data sharing *group* is a collection of one or more DB2 subsystems accessing shared DB2 data. Each DB2 subsystem belonging to a particular data sharing group is a *member* of that group. All members of the group use the same shared DB2 catalog and directory. Currently, the maximum number of members in a group is 32.

Some capabilities described in this book can be used whether or not you are sharing data. We use the term *data sharing environment* to mean a situation in which a group has been defined with at least one member. In a non-data-sharing environment, no group is defined.

This chapter describes the following topics of DB2 data sharing:

- “Advantages of DB2 Data Sharing”
- “How Data Sharing Works” on page 31
- “Using DB2 Data Sharing” on page 37
- “Software and Hardware Requirements” on page 42

Advantages of DB2 Data Sharing

DB2 data sharing improves price for performance, improves the availability of DB2, extends the processing capacity of your system, and provides more flexible ways to configure your environment. There is no need to change SQL in your applications to use data sharing, although some tuning might be needed for optimal performance.

Improves Price for Performance

DB2 data sharing gives you a database solution that is powerful enough to handle complex business requirements, but which can be run on one or more smaller, less expensive System/390 (S/390) microprocessors.

Improves Availability

More DB2 users demand access to DB2 data every hour of the day, every day of the year. DB2 data sharing helps you meet this service objective by improving availability during both planned and unplanned outages.

As Figure 7 illustrates, if one subsystem comes down, users can access their DB2 data from another subsystem. Transaction managers are informed that DB2 is down and can switch new user work to another DB2 subsystem in the group.

For unplanned outages, restart and recovery can be automated, if you are running on MVS/ESA Version 5 Release 2 or later.

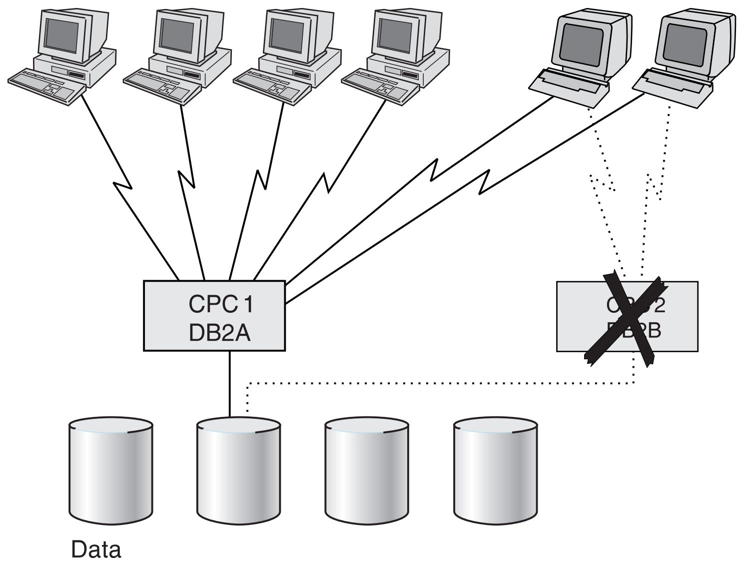

Figure 7. Data Sharing Improves Availability During Outages. If a DB2 or the entire central processor complex (CPC) comes down, transactions can be routed to another system.

DB2's increased availability does have some performance cost, but the overhead for inter-processor communication and caching of changed data is minimized by DB2's efficient locking and caching mechanisms, along with its use of the *coupling facility* hardware. High availability for the DB2 structures in the coupling facility is provided by using automatic structure rebuild and duplexing of the structures used for caching data.

Expands Capacity

As you move more data processing onto DB2, your processing needs can exceed the capacity of a single system to handle it. Before DB2 data sharing, you had these options for relief:

- Move some applications and copies of the data onto another DB2.

  This approach requires that you maintain separate copies of the data. There is no communication among DB2s and no shared DB2 catalog and directory.

- Install another DB2 and rewrite applications on it to access the original data as distributed data.

  This might relieve the workload on the original DB2, but it does require changes to your applications and has performance overhead of its own. Nevertheless, for DB2s that are separated by great distance, or if DB2 needs to share data with a system outside the data sharing group, the distributed data facility is still your only option.

- Use shared read-only data.

Shared read-only data lets you have only one DB2, the owning DB2, that can both read and write data. Other DB2s can only read data. There are complex operational issues for dealing with shared read-only data. **Shared read-only data is not likely to be supported for much longer.**

- Install a larger processor and move data and applications to that machine.

  This can be an expensive option and is often the choice of last resort. Not only is it costly, but it means your system must come down while you move to the new machine.

Incremental Growth: The Parallel Sysplex can grow incrementally. You can add a new DB2 onto another central processor complex and access the same data through the new DB2. There is no need to manage copies or distribute data. All DB2s in the data sharing group have concurrent read-write access, and there is a single DB2 catalog and directory used by all DB2s.

Workload Balancing: DB2 data sharing provides flexibility for growth and workload balancing. Unlike the partitioned data approach to parallelism (sometimes called the *shared-nothing* architecture), in which there is a one-to-one relationship between a particular DBMS and a segment of data, data in a DB2 data sharing environment does not have to be redistributed when a new subsystem is added or if the workload becomes unbalanced. The new DB2 member has the same direct access to the data as all other existing members of the data sharing group.

Capacity when You Need It: Consider using a data sharing configuration to handle your peak loads. You can start members to handle peak loads (such as end of quarter processing), and then stop them when the peak passes.

Higher Transaction Rates

As Figure 8 on page 26 illustrates, you can run the same application on more than one DB2 subsystem to achieve higher transaction rates than possible on a single subsystem.

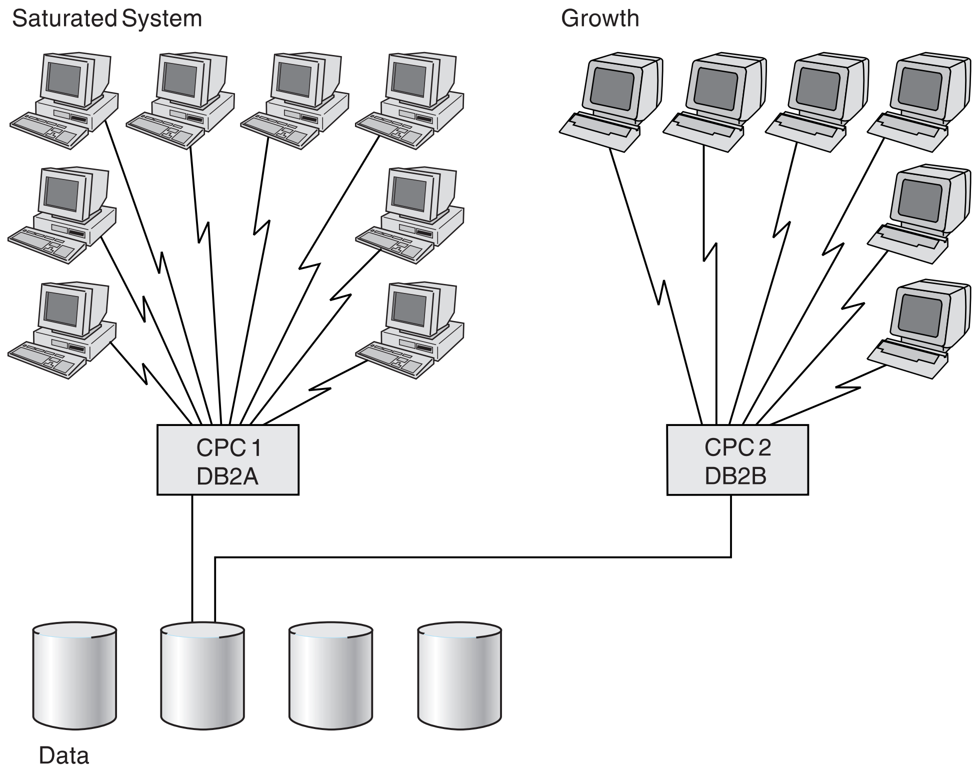

Figure 8. Data Sharing Enables Growth. Move some of your existing DB2 workload onto another central processor complex (CPC).

More Capacity to Process Complex Queries

With *Sysplex query parallelism*, DB2 can use all the processing power of the data sharing group to process a single query. For those of you who do complex data analysis or decision support, Sysplex query parallelism is a scalable solution; because the data sharing group can grow, you can put more power behind those queries even as those queries become increasingly complex and run on larger and larger sets of data.

Figure 9 on page 27 shows all members of a data sharing group participating in processing a single query.

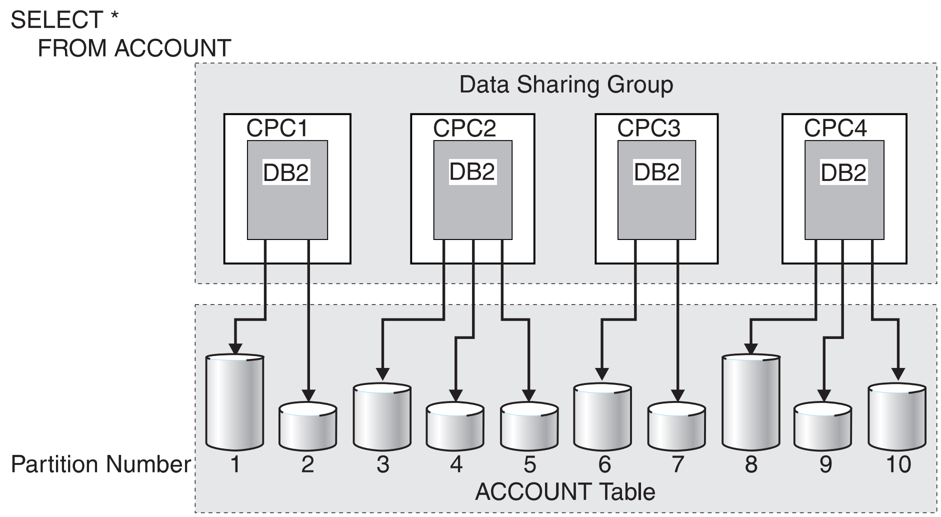

Figure 9. Query Processed in Parallel by Members of a Data Sharing Group. Different DB2 members are processing different partitions of the data.

This is a simplification of the concept— several DB2s can access the same physical partition. To take full advantage of parallelism, use partitioned table spaces.

Allows for Flexible Configurations

With DB2 data sharing, you can configure your system environment much more flexibly.

As shown in Figure 10 on page 28, it is possible to have more than one DB2 data sharing group on the same MVS Sysplex. You might, for example, want one group for testing and another for production data. There is also a single, non-data-sharing DB2 in this example.

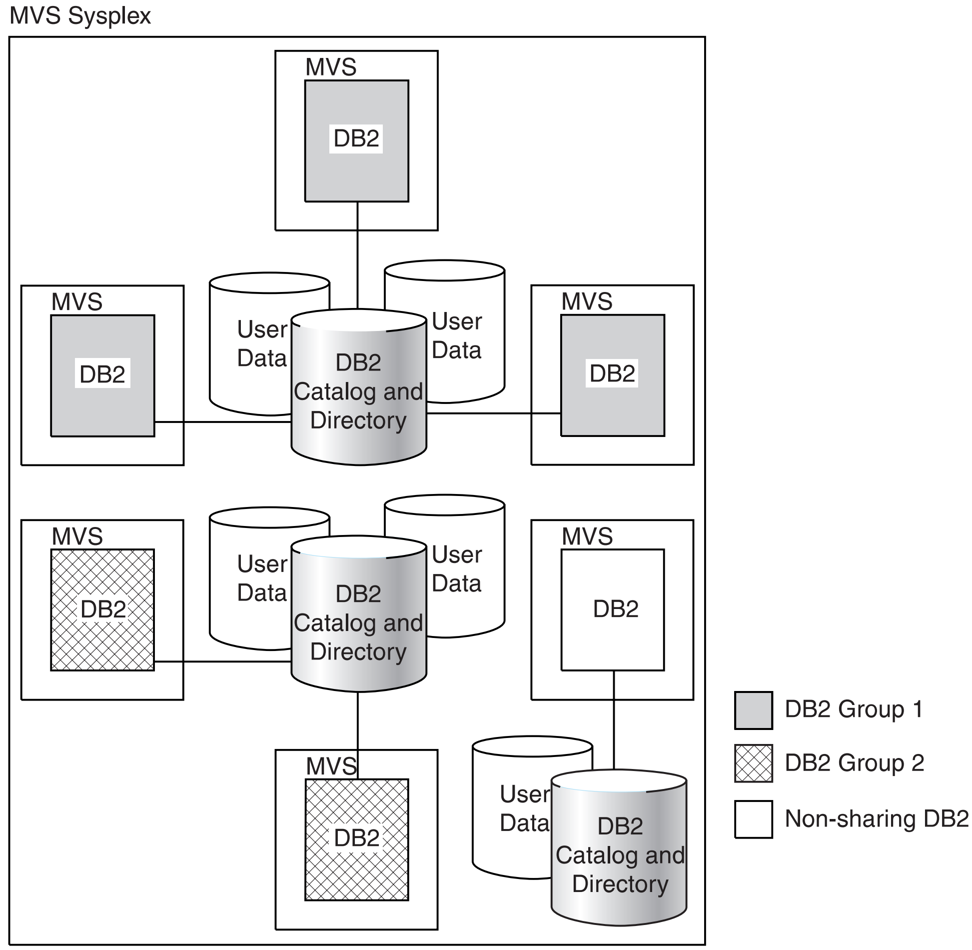

Figure 10. A Possible Configuration of DB2 Data Sharing Groups. Although this example shows one DB2 per MVS, it is possible to have more.

Flexible Operational Systems

Figure 11 on page 29 shows how, with data sharing, you can have query user groups and online transaction user groups on separate MVS images. This configuration lets you tailor each system specifically for that user set, control storage contention, and provide predictable levels of service for that set of users. Previously, you might have had to manage separate copies of data to meet the needs of different user groups.

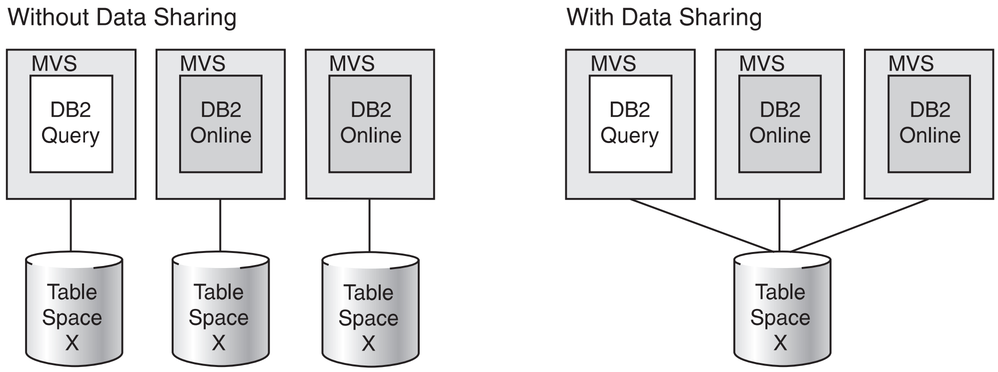

Figure 11. Flexible Configurations with DB2 Data Sharing. Data sharing lets each set of users access the same data, which means it is no longer necessary to manage copies.

Flexible Decision Support Systems

Figure 12 on page 30 shows two different decision support configurations. A *typical* configuration separates the operational data from the decision support data. Use this typical configuration when the operational system has environmental requirements that are different from those of the decision support system. The decision support system might be in a different geographical area, or there might be different security requirements for the two systems.

DB2 offers another option—a *combination* configuration. A combination configuration combines your operational and decision support systems into a single data sharing group, which has these advantages:

- You can occasionally join decision support data and operational data with SQL.
- You can reconfigure the system dynamically to handle fluctuating workloads. (You can dedicate CPCs to decision support processing or operational processing at different times of the day or year.)
- You can reduce the cost of computing:
  - The infrastructure used for data management is already in place.
  - You can create a prototype of a decision support system in your existing system, and then add processing capacity as the system grows.

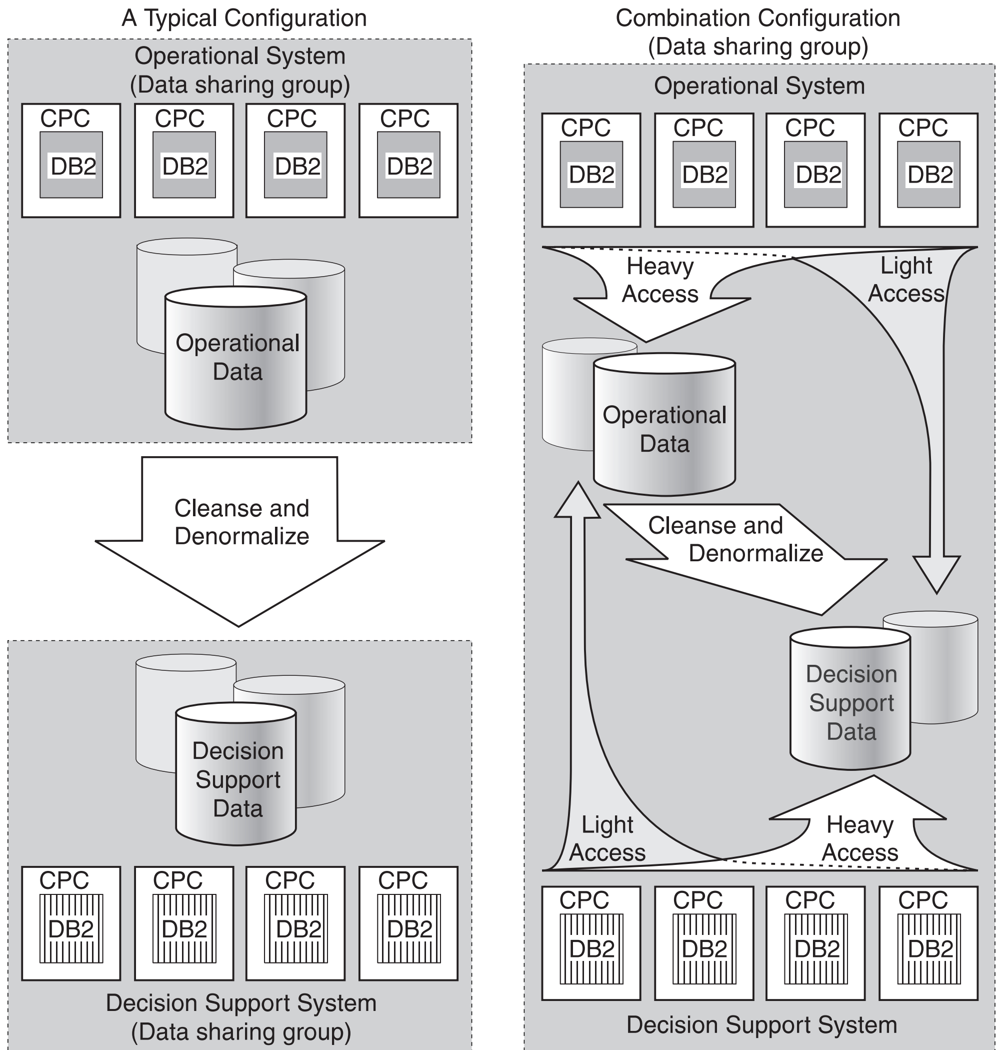

Figure 12. Flexible Configurations for Decision Support. DB2 data sharing lets you configure your systems in the way that works best with your environment.

If you want to configure a combination system configuration, you must separate decision support data from operational data as much as possible. Buffer pools, DASD, and control units that are used in your decision support system should be separate from those used in your operational system. This separation greatly minimizes any negative performance impact on the operational system.

If you are unable to maintain that level of separation, or if you have separated your operational data for other reasons, such as security, then using a separate decision support system is your best option.

Application Interface Is Unchanged

Your investment in people and skills is protected because their current knowledge extends to this environment. Existing SQL interfaces and attachments remain intact when sharing data.

It is possible to bind a package or plan on one DB2 subsystem and run that package or plan on any other DB2 subsystem in the data sharing group.

How Data Sharing Works

This section provides background information about how shared data is updated and how DB2 protects the consistency of that data. It also introduces operational and database design considerations for the shared environment.

For data sharing, you must have an *MVS Sysplex* that is enhanced with the coupling facility technology. Such an MVS Sysplex is a group of central processor complexes (CPCs) running MVS and cooperating as a common computing entity. Hardware and software requirements for DB2 data sharing are listed in “Software and Hardware Requirements” on page 42.

You define the data sharing group and its members by using the installation and migration process. See “Creating the DB2 Data Sharing Group” on page 92 for an overview of this process.

How Data Consistency is Protected

Data can be accessed by any DB2 in the group. There can be potentially many subsystems reading and writing the same data. DB2 uses special data sharing locking and caching mechanisms to ensure data consistency.

| When multiple members of a data sharing group have opened the same table space, index space, or partition, and at least one of them has opened it for writing, then the data is said to be “of inter-DB2 R/W interest to” the members. (Sometimes we shorten this to “inter-DB2 interest.”) To control access to data that is of inter-DB2 interest, whenever the data is changed DB2 caches it in a storage area, called a *group buffer pool*.

When there is inter-DB2 R/W interest in a particular table space, index, or partition, it is *dependent* on the group buffer pool, or *GBP-dependent* (group buffer pool dependent).

You define group buffer pools using coupling facility resource management (CFRM) policies. For more information about these policies, see *MVS/ESA Setting Up a Sysplex*.

As shown in Figure 13 on page 32, there is a mapping between a group buffer pool and the buffer pools of the group members. For example, each DB2 has a buffer pool named BP0. For data sharing, you must define a group buffer pool (GBP0) in the coupling facility that maps to buffer pool BP0. GBP0 is used for caching the DB2 catalog and directory table spaces and indexes, along with any other table spaces, indexes, or partitions that use buffer pool 0.

Although a single group buffer pool cannot reside in more than one coupling facility (unless it is duplexed), you can put group buffer pools in more than one coupling facility.

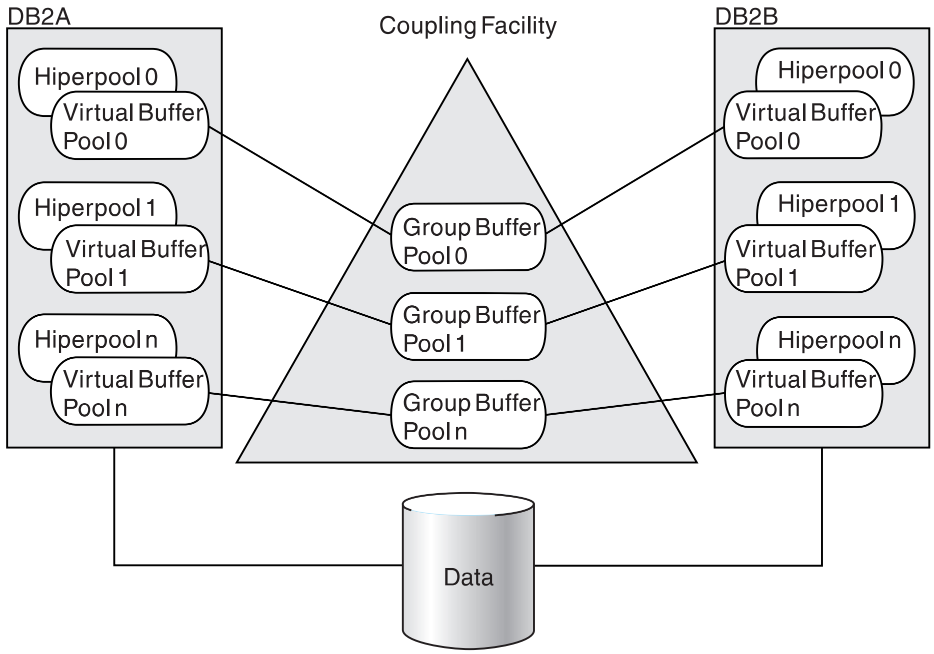

Figure 13. Relationship of Virtual Buffer Pools and Hiperpools to Group Buffer Pools. There is one group buffer pool for all buffer pools of the same name.

When a particular page of data is changed by one DB2, DB2 caches that page in the group buffer pool. The coupling facilityinvalidates any image of the page in the separate buffer pools of all the members. Then, when a request for that same data is subsequently made by another DB2, it looks for the data in the group buffer pool.

Update Example

Let's follow a page of data as it goes through the update process. The most recent version of the data page is shaded in the illustrations. This scenario also assumes that the group buffer pool is used for caching just changed data and that it is *duplexed*, for high availability.

Figure 14 shows an application issuing an UPDATE statement from DB2A. In this instance, the data does not reside in the member's buffer pool, or in the group buffer pool, so DB2A retrieves the data from DASD and gets the appropriate locks to prevent another DB2 from updating the same record at the same time.

Because no other DB2 subsystem is sharing the table at this time, DB2 does not need to use data sharing processing for DB2A's update.

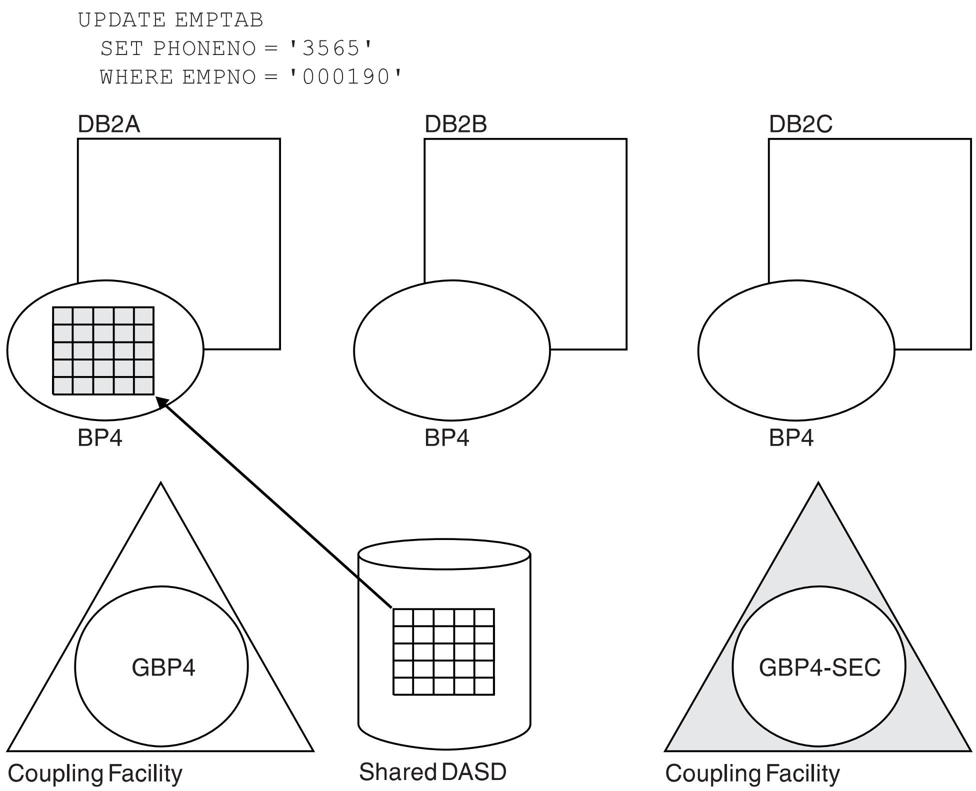

Figure 14. Data Is Read from DASD and Updated by an Application Running on DB2A

Next, suppose another application, running on DB2B, needs to update that same data page. Figure 15 illustrates the situation. DB2 knows that inter-DB2 interest exists; DB2A writes the changed data page to the primary group buffer pool. The write to the backup group buffer pool, the *secondary* group buffer pool, is overlapped with the write to the primary group buffer pool. DB2B then retrieves the data page from the primary group buffer pool.

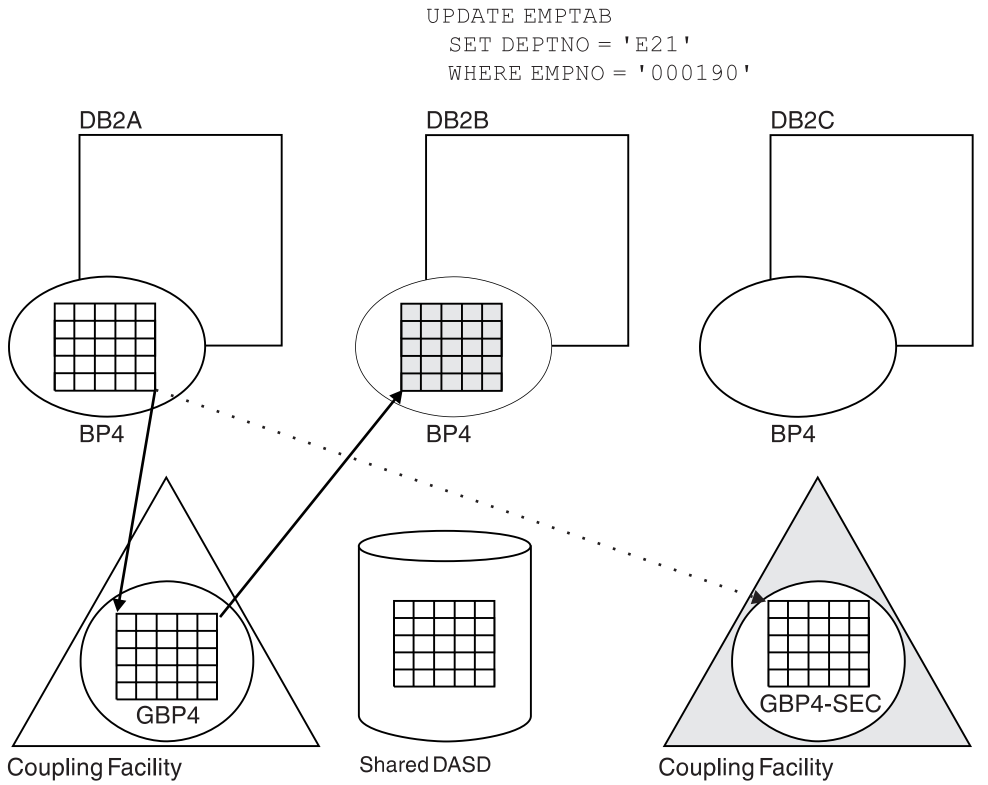

Figure 15. DB2B Updates the Same Data Page. When DB2B references the page, it gets the most current version of the data from the primary group buffer pool.

Figure 16 shows that after DB2B updates the data, a copy of the data page is
moved into the group buffer pool (both primary and secondary) , and the data page
is invalidated in DB2A's buffer pool.

COMMIT

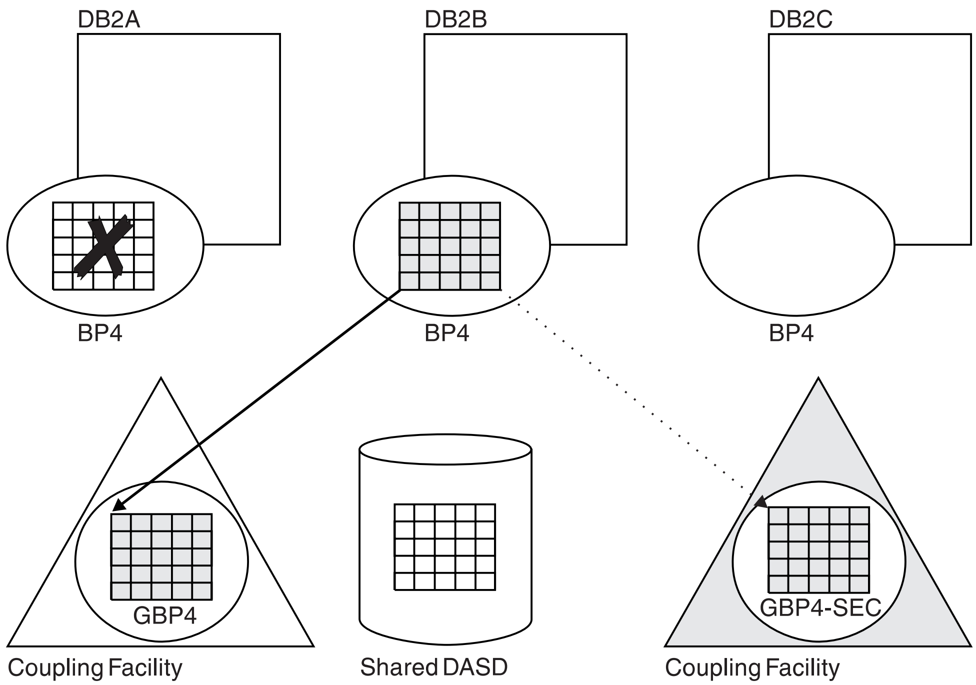

Figure 16. The Updated Page Is Written to the Group Buffer Pool. The data page is invalidated in DB2A's buffer pool.

Now, as shown in Figure 17, when DB2A needs to read the data, the data page in its own buffer pool is invalid. Therefore, it reads the latest copy from the primary group buffer pool.

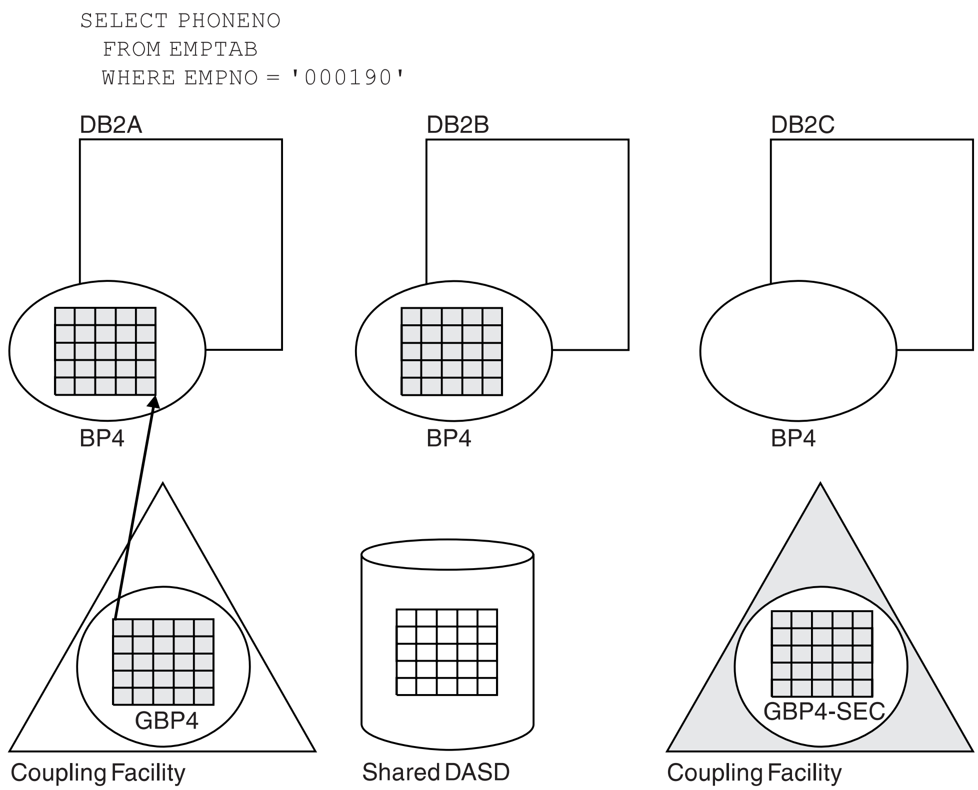

Figure 17. Data Is Being Read from the Group Buffer Pool by DB2A

Writing Changed Data to DASD

DB2 uses a *castout* process to write changed data to DASD from a group buffer pool. Figure 18 shows that when data is cast out from a group buffer pool to DASD, that data must first pass through a DB2's address space because there is no direct connection from a coupling facility to DASD. This data passes through a private buffer, not DB2's virtual buffer pools.

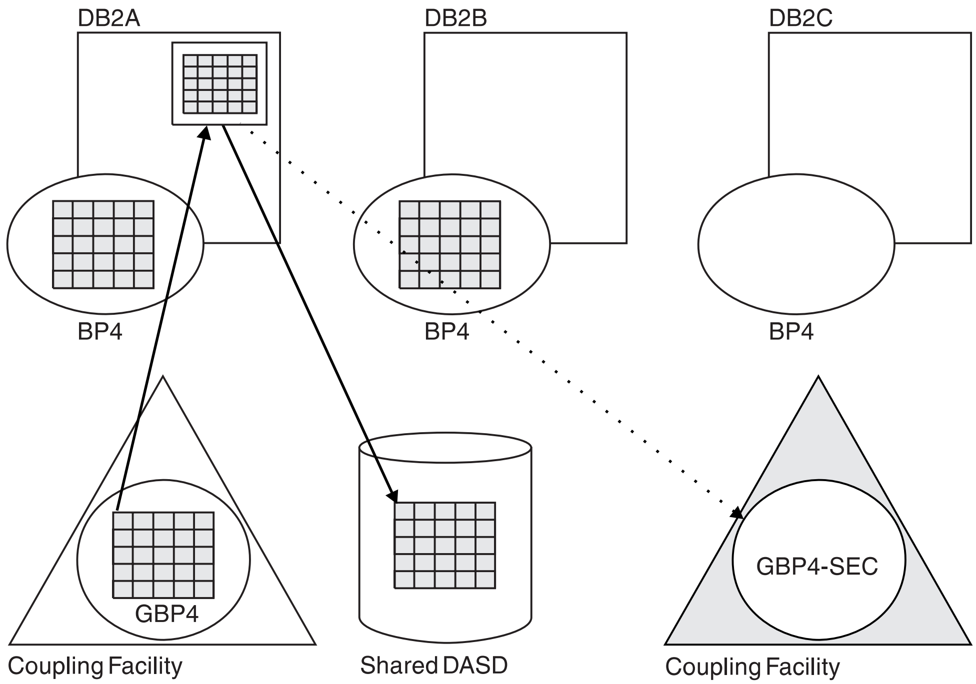

Figure 18. Writing Data to DASD. There is no direct connection from the coupling facility to DASD. The data must pass through DB2's address space before being written to DASD.

When a group buffer pool is duplexed, data is not cast out from the secondary
group buffer pool to DASD. When a set of pages has been written to DASD from
the primary group buffer pool, DB2 deletes those pages from the secondary group
buffer pool.

Using DB2 Data Sharing

Because the application interface is unchanged for data sharing, there are no new tasks for application programmers and end users. However, system programmers, operators, and database administrators have additional tasks in a data sharing environment.

Enabling Data Sharing

Plan a naming convention before enabling data sharing on the first DB2 in the group (the *originating* member). Because many names in the Sysplex and in the group must be unique, you must have a naming convention before you create the group. Not only must shared data objects have unique names, but unique names must also be given for every group resource (such as a name for the group). See "Planning a Naming Convention" on page 52 for more information about naming.

| The originating member is an existing Version 5 subsystem that was either
| migrated from Version 4 or is a new installation of Version 5. The originating

member's DB2 catalog is the catalog for the entire group. Additional members of the group use the originating member's catalog and are added as new installations.

If you have data from existing DB2s to move into the group, you must move data from those subsystems and merge their catalog definitions into the catalog used by the data sharing group. DB2 does not provide a way to merge DB2 catalogs automatically.

Using the Data Sharing Group for Network Computing

Applications can communicate with a data sharing group by using either systems network architecture (SNA) or, if you are running on OS/390 Release 3 or subsequent releases, Transmission Control Protocol/Internet Protocol (TCP/IP) network protocols. Applications attach to a data sharing group by using a single location name, and a single-system image is provided to all requesting applications. For more information about setting up a data sharing group for communicating in a network, see “Chapter 5. Communicating with a Data Sharing Group” on page 123.

Administering a Database

One of the main tasks in database design and administration is to make sure that every data object has a unique name, accounting for the fact that existing data and applications from other systems might be merged into the data sharing group. You can use SQL to define data objects from any DB2 in the data sharing group. Make sure that data to be shared resides on shared DASD.

DB2 data sharing implements referential integrity within a data sharing group in the same way as DB2 without data sharing. Referential integrity is controlled by the DB2 catalog, and all DB2s share the same catalog.

Improving Performance for Read-Intensive Applications: You can improve performance for certain read-intensive applications that run on more than one DB2 subsystem by caching all data pages in the group buffer pool to avoid DASD I/O. The GBPCACHE keyword on CREATE TABLESPACE, CREATE INDEX, ALTER TABLESPACE and ALTER INDEX allows you to specify this. See “Read Operations” on page 245 for more information.

However, if the read-intensive applications are mainly complex queries, it is best not to cache all data in the group buffer pool. See “Improving the Response Time for Read-Only Queries” on page 199 for more information.

Planning for Exit Routines: If you use exit routines, such as a field or validation procedure, or the access control authorization routine, you must ensure that all members of the group are using the same routines. We recommend that all such routines be placed in a program library shared by all members of the group.

Authorizing Users

Use the same authorization mechanisms that are in place for non-data sharing DB2s to control access to shared DB2 data and the member DB2 subsystem itself. Because all DB2s in the group access the same catalog, any authorization ID has the same granted privileges and authorities in every member of the group.

As suggested for non-data sharing DB2s, use a security system outside of DB2 (such as RACF or its equivalent) to control which user IDs can access any particular DB2 subsystem. RACF does not recognize an entire data sharing group as a single resource. Therefore, you must separately define DB2 resources to RACF for each member of the group. Then connect all user IDs to a RACF group that permits access to all those resources. Or, permit separate groups of user IDs to have access to different sets of resources. In the latter case, however, you cannot move work freely among all members of the data sharing group.

Each member of the data sharing group uses the same names for the connection and sign-on exit routines. We recommend that these routines be shared by all members of the group to avoid authorization anomalies, such as having primary authorization IDs treated differently by different members of the group, or associated with different sets of secondary IDs by different members.

Loading and Reorganizing Data

You can run utilities to load or reorganize data from any DB2 in the data sharing group.

Operating a Data Sharing Group

This section describes some of the operational considerations for data sharing:

Entering Commands

Sysplex technology allows you to manage the DB2 data sharing group from consoles attached to a single MVS system, or from separate systems. Figure 19 shows an example of routing commands from a single MVS system.

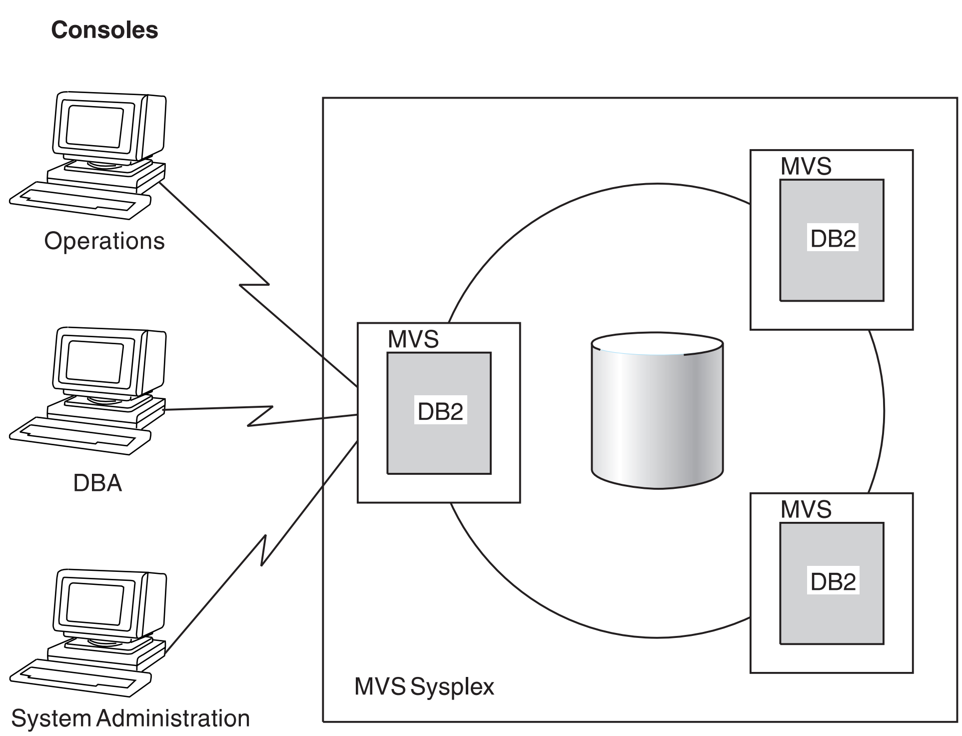

Figure 19. Issuing Commands

Using Commands: Some commands manage group resources while others manage resources on a member-wide basis. See *Command Reference* for more information about specific commands.

Recovering Data

DB2 recovers data from information contained in the members' logs and bootstrap data sets (BSDSs). However, because updates to any particular table can be logged on many different DB2 subsystems, the group uses the *shared communications area* (SCA) in a coupling facility to coordinate recovery. The SCA contains the member names and BSDS data set names, as well as status information about objects and members in the group. It is also used to coordinate startup.

The RECOVER Utility: You can run the RECOVER utility from any DB2 subsystem in the group. The process for data recovery is basically the same for DB2 data sharing as it is for non-sharing DB2s. However, a single table space can be updated by many different DB2 subsystems. Therefore, to recover that object DB2 must merge log records from those DB2s, using a *log record sequence number* (LRSN), which is a value derived from the store clock timestamp and synchronized across the members by the Sysplex Timer.

You can specify an LRSN value on the RECOVER TABLESPACE utility for recovery to a prior point in time. See “How Recovery Works in a Data Sharing Group” on page 161 for more information.

Because DB2 must merge logs from many different DB2 subsystems, we recommend avoiding a situation where you must recover from archive logs on tape. Archives on DASD provide the best recovery performance, but if you do archive to tape, it is best to have large active logs and frequent recovery points (such as image copies and commits in long-running units of recovery) so you can avoid going to those tape archives.

If it is necessary to recover from tape, the DB2 subsystem doing the recovery must have enough tape drives allocated to merge the log, or you can copy the tape data sets to DASD and recatalog them before running the recovery job.

Do not archive logs from more than one system on the same tape.

Coupling Facility Availability and Recovery: In addition to data objects, coupling facilities contain vital resources needed for data sharing. We recommend that you have more than one coupling facility, to allow for group buffer pool duplexing, and for automatic recovery in the event a coupling facility fails. See “Coupling Facility Availability” on page 61 for more detailed suggestions.

Stopping and Starting DB2

You can stop and start individual members in a data sharing group while the other members continue to run. The startup process for each DB2 member is largely the same as non-data-sharing DB2s.

The process called *group restart* is needed only in the rare event that critical resources in a coupling facility are lost and cannot be rebuilt. When this happens, all members of the group terminate abnormally. Group restart is required to rebuild this lost information from individual member logs. However, unlike with data recovery, this information can be applied in any order. Because there is no need to merge log records, many of the restart phases for individual members can be done in parallel. An automated procedure can be used to start all members of the group. If a particular DB2 is not started, then one of the started DB2s performs group restart on behalf of that stopped DB2.

Maintaining a Data Sharing Group

To apply maintenance, most changes can be made on one DB2 at a time, as shown in Table 1. If a DB2 or MVS must come down for the change to take place, and the outage is unacceptable to users, you can move those users onto another DB2.

The recommended way of testing maintenance is to apply that maintenance to a test data sharing group before moving it onto the production group.

Table 1. Planned Maintenance Changes

Type of Change	Action Required
Early code	Bring down one MVS at a time and re-IPL.
DB2 code	Bring down and restart each DB2 member independently.
IRLM code	Bring down and restart each IRLM member independently.
Attachment code	Apply the change and restart the transaction manager or application.
Subsystem parameters	For those that cannot be changed dynamically, update using DB2's update process. Stop and restart the DB2 to activate the updated parameter.

Applying Maintenance to IRLM: Each DB2 has its own IRLM. The set of IRLMs have their own data sharing group. Similar to DB2, you have to stop and restart each IRLM in the group to roll the change throughout the group.

IRLM has a concept of a *function level* to control changes that affect the entire group. The function level for a particular IRLM member indicates the latest function that IRLM is capable of running. The *group* function level is the lowest function level that exists among the group. The group function level is the function level at which all IRLMs in the group must run, even if individual members are capable of running at a later function level.

It is a good idea to keep all members of the group at the same function level. This ensures that all IRLMs are running with the same maintenance that affects the entire group. (Maintenance that does not affect the group does not increment the function level.)

To see the IRLM function levels, use the MODIFY *irlmproc*,STATUS,ALLI command of MVS. See “Determining the Function Level of the IRLM Group” on page 109 for more information.

Software and Hardware Requirements

This section describes at a high level the software and hardware required to run data sharing. For more detailed information about setting up the Parallel Sysplex, see *System/390 MVS Sysplex Hardware and Software Migration*.

Software

DB2 data sharing requires MVS/ESA Version 5 Release 1 (or subsequent releases). MVS Version 5 Release 2 or Operating System/390 (OS/390) offers these enhancements:

- The ability to automatically restart after a failure, as described in "Using MVS's Automatic Restart Capability" on page 60
- MVS workload management support for dynamic routing of DDF transactions, as described in "Using A Single Location Name with a List of LU Names (Member Routing)" on page 126
- The ability to dynamically change structure sizes, as described in "Changing Structure Sizes" on page 72. Structures must be allocated in a coupling facility at CFLEVEL=1 or higher.

Coupling facility performance enhancements as described in "Prefetch Processing" on page 245 and "Locking Optimizations" on page 217 require MVS Version 5 Release 2 or OS/390 *and* CFLEVEL=2 or higher.

The checkpoint performance enhancement described in "Group Buffer Pool Checkpoint" on page 249 requires OS/390 Release 3 (or later) with APAR OW28460. (This performance enhancement is included in the OS/390 Release 6 base product.) The group buffer pool must be allocated in a coupling facility at CFLEVEL=5 or higher.

Group buffer pool duplexing requires OS/390 Release 3 or subsequent releases and APAR OW28460. It also requires CFLEVEL=5 or higher. See "Duplexing group buffer pools" on page 65 for more information about group buffer pool duplexing.

RACF Version 2 Release 1 or later, or an equivalent external security facility that supports a Sysplex-wide security scope, is recommended.

Hardware

DB2 data sharing requires a S/390 Parallel Sysplex:

- Central processor complexes (CPCs) that can attach to the coupling facility
- At least one coupling facility and the appropriate channels and channel cards
- At least one Sysplex Timer
- Connection to shared DASD

If archiving the DB2 log to tape, you might need a number of tape units greater than or equal to the number of DB2s in the group. These tape units must be accessible and shareable by the DB2 running a RECOVER utility.

Storage Estimates

Installers must estimate the sizes of the various structures in the coupling facility. See “Storage for Coupling Facility Structures” on page 67 for more information.

Chapter 3. Planning for DB2 Data Sharing

To plan for the data sharing function of the licensed program DB2 for OS/390, you must coordinate your efforts with system hardware and software groups. You must complete these tasks before creating a DB2 data sharing group:

- "Planning for DB2 in the Parallel Sysplex"
- "Planning a Naming Convention" on page 52
- "Planning for Availability" on page 59
- "Estimating Storage" on page 67
- "Planning to Enable Data Sharing" on page 75

The process of enabling data sharing is described in "Chapter 4. Installing and Enabling DB2 Data Sharing" on page 85.

| If you already have a Version 4 data sharing group, read this chapter for new
| information, and see "Migrating an Existing Data Sharing Group to the New
| Release" on page 107.

Planning for DB2 in the Parallel Sysplex

This section has information about planning for DB2 in the MVS Parallel Sysplex and about the special connectivity needs for DB2 data sharing. For more information about specific hardware and software requirements for the Parallel Sysplex, see *System/390 MVS Sysplex Hardware and Software Migration*.

Considerations for the Parallel Sysplex

This section has information about the Parallel Sysplex and its relationship to DB2 data sharing. DB2 data sharing is dependent on the hardware and software components in the Parallel Sysplex.

Cross-System Coupling Facility (XCF) Component of MVS

The cross-system coupling facility (XCF) component of MVS must be set up to enable the Sysplex. During startup, DB2 members join one XCF group, and the associated internal resource lock managers (IRLMs) join another XCF group. To join a particular group, the IRLMs and DB2s use the names you specify during DB2 installation.

| DB2 uses the XCF for certain intersystem communications. It is recommended that
| you use both the coupling facility and channel-to-channel connections for XCF
| signalling. See *MVS/ESA Setting Up a Sysplex* for more information about
configuring the XCF.

Sysplex Timer

At least one Sysplex Timer must be installed in the Sysplex. The Sysplex Timer keeps the timestamps of the S/390 processors synchronized for all DB2s in the data sharing group. DB2 data sharing uses a value derived from the timestamp to recover data using the log.

Coupling Facility

At least one coupling facility must be installed and defined to MVS before you can run DB2 with data sharing capability.

DB2 relies on areas of storage in the coupling facility called *structures*. There are three types of structure: lock, list, and cache. Each type of structure has a unique function that DB2 uses. Figure 20 on page 48 shows the coupling facility structures used by DB2. Before starting DB2 for data sharing, you must have defined one lock structure, one list stucture, and at least one cache structure (group buffer pool 0).

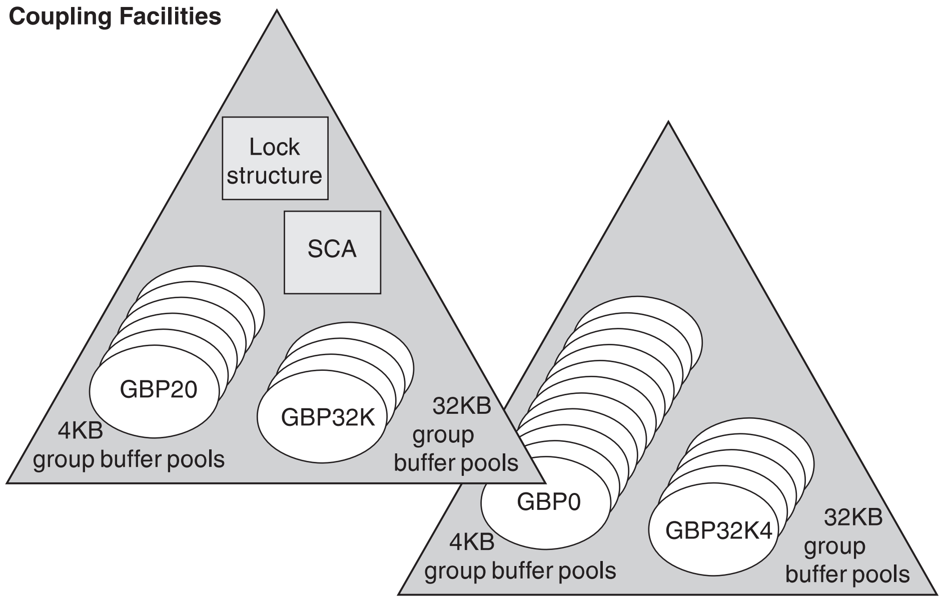

Figure 20. Coupling Facility Structures Used by DB2. This is a sample configuration. The lock structure and SCA do not have to be in the same coupling facility. Individual structures cannot span coupling facilities.

- Lock structure

  There is one lock structure per data sharing group used by IRLM to control locking.

- List structure (DB2's SCA)

  There is one list structure per data sharing group used as the shared communications area (SCA) for the members of the group. The SCA contains information about databases in an exception condition and other information.

- Cache structures (DB2 group buffer pools)

  Cache structures are used as group buffer pools for the DB2 data sharing group. DB2 uses a group buffer pool to cache data that is of interest to more than one DB2 in the data sharing group. Group buffer pools are also used to maintain the consistency of data across the buffer pools of members of the group by using a cross-invalidating mechanism. Cross-invalidation is used when a particular member's buffer pool does not contain the latest version of the data.

  One group buffer pool is used for all buffer pools of the same name in the DB2 group. For example, a buffer pool 0 (BP0) must exist on each member, to contain the catalog and directory table spaces. Thus, there must be a group buffer pool 0 (GBP0) in a coupling facility.

  Similarly, if any member creates table space X and associates it with buffer pool 1, then X is associated with BP1 on every member, because there is only one definition of X in the catalog, which is used by the entire group. If the data in X is to be shared, you must define a group buffer pool 1 (GBP1) cache structure. If you don't define the group buffer pool, a single DB2 can update X, or more than one DB2 can read X. But there can be no inter-DB2 R/W activity for X.

Recommendation: For data that is private to each member, such as work files or user data that only one member reads, choose a buffer pool for those non-shared page sets. Assume you choose BP6. If you wanted just DB2A to have access to non-shared table space Y, define Y (and any indexes) to buffer pool 6. Every other member must define its virtual buffer pool 6 with a size of 0 and then there is no need to define the coupling facility structure for group buffer pool 6.

By moving this data to buffer pools that are separate from those used by shared data, it is easier to monitor the performance and provide more predictable performance for that private data.

Defining Coupling Facility Structures: Before you start DB2 data sharing, you must define coupling facility structures. Use MVS's coupling facility resource management (CFRM) policies to define these structures to the MVS Sysplex. A CFRM policy determines how and where the structure resources are allocated.

We strongly recommend that you plan for and define the availability characteristics of the SCA and lock structure for loss of connectivity failures, which also includes total failure of the coupling facility. This requires a Sysplex failure management (SFM) policy, as described in "Rebuilding Structures when Connectivity is Lost" on page 63.

See *MVS/ESA Setting Up a Sysplex* for information about how to create CFRM and SFM policies.

A sample CFRM policy as it might appear in a Sysplex running MVS Version 5 Release 2 is shown in Figure 21 on page 50.

```
//POLICYX  JOB MSGCLASS=Z,REGION=2000K,CLASS=A,
//             MSGLEVEL=(1,1)
//STEP1  EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN  DD *
 DATA TYPE(CFRM) REPORT(YES)
 DEFINE POLICY NAME(POLICYX) REPLACE(YES)
   STRUCTURE NAME(DSNDB0G_LOCK1)
             INITSIZE(32000)
             SIZE(64000)
             REBUILDPERCENT(5)
             PREFLIST(CF01,CF02)
   STRUCTURE NAME(DSNDB0G_GBP0)
             INITSIZE(50000)
             SIZE(100000)
             REBUILDPERCENT(5)
             DUPLEX(ALLOWED)
             PREFLIST(CF02,CF01)
   STRUCTURE NAME(DSNDB0G_SCA)
             INITSIZE(10000)
             SIZE(20000)
             REBUILDPERCENT(5)
             PREFLIST(CF01,CF02)

   CF NAME(CF01) TYPE(009674)
                 MFG(IBM)
                 PLANT(00)
                 SEQUENCE(000000040016)
                 PARTITION(1)
                 CPCID(00)
                 DUMPSPACE(1200)

   CF NAME(CF02) TYPE(009674)
                 MFG(IBM)
                 PLANT(00)
                 SEQUENCE(000000040029)
                 PARTITION(1)
                 CPCID(00)
                 DUMPSPACE(1200)
//
```

Figure 21. Sample CFRM Policy

What You Need to Know: For DB2, know the following before creating the policy definitions:

- Initial size and maximum size (see “Storage for Coupling Facility Structures” on page 67)

  With MVS Version 5 Release 1, there is only one size possible (the SIZE parameter). With MVS Version 5 Release 2 and subsequent releases, and with the coupling facility level 1 (CFLEVEL=1), the structures can be dynamically resized from INITSIZE up to the value in SIZE. See “Changing Structure Sizes” on page 72 for more information.

- Structure names (see “Coupling Facility Structure Names” on page 56)

- Availability characteristics

  You need to know the preference list (PREFLIST) for rebuilding or reallocating a structure should the coupling facility fail. See “Coupling Facility Availability” on page 61 for more information.

  See “Rebuilding Structures when Connectivity is Lost” on page 63 for information about specifying the value for REBUILDPERCENT.

  See “Duplexing group buffer pools” on page 65 for information about specifying the value for DUPLEX.

Authorize DB2 to Access the Structures: Optionally, you can set up a facility class profile to limit access to the structures in the coupling facility. If you do this, ensure that DB2 does have access by ensuring that the IDs associated with the DB2 address spaces have update access authority to the coupling facility structures through RESOURCE(IXLSTR.*structure_name*) in SAF class CLASS(FACILITY).

If you do not create a facility class profile, the default allows any authorized user or program (supervisor state and program key mask allowing key 0-7) to issue coupling facility macros for the structure.

Make sure that the user IDs associated with the DB2 address spaces have update access authority to the coupling facility structures through RESOURCE(IXLSTR.*structure_name*) in SAF class CLASS(FACILITY).

Common MVS Libraries

As you will read in "Naming Recommendations" on page 56, DB2 supports a configuration with a SYS1.PARMLIB and SYS1.PROCLIB shared by all MVS systems in the Parallel Sysplex. This can make adding and modifying systems much easier.

One detail to remember, especially if you intend to have many DB2 subsystems in the Sysplex, is that each DB2 and each IRLM you define to MVS in the IEFSSN*xx* parmlib member requires an MVS system linkage index (LX). The default number of these indexes that MVS reserves is 55. If you place all your DB2 and IRLM subsystem definitions in a single IEFSSN*xx* member, you might need more than 55 LXs, or risk having subsystems not able to start.

If you need more than 55 LXs, use the NSYSLX option on the MVS IEASYS*xx* parmlib member to increase this number. See *MVS/ESA Initialization and Tuning Guide* for more information.

Connectivity Considerations

DB2 data sharing requires that all DB2-related resources reside on shared DASD. The DB2 catalog and directory, as well as any user data that is to be shared, must be on shared DASD. The integrated catalog for DB2 data sets must also be on shared DASD.

Also, all the members' logs and bootstrap data sets (BSDSs) must be on shared DASD for recovery purposes. A DB2 performing recovery must have access to the logs of other members in the group.

We strongly recommend that you place work files on shared DASD for the following reasons:

- It is required for processing many types of queries when those queries use Sysplex query parallelism. Each assisting DB2 writes to its own work file, and the coordinator can read the results from the assistants' work files.
- It lets you create or drop a work file table space from any other member in the group.
- It keeps DB2 connected to its work file even if you have to restart the DB2 on another processor.

Make sure you have physical connectivity by checking the following connections:

- Verify that there is one user integrated catalog facility for cataloging DB2 system data sets and that this catalog is accessible from each MVS in the Sysplex.
- Verify connectivity from each system on which a DB2 resides to:
  - A set of DB2 target libraries
  - A single DB2 catalog
  - A single DB2 directory
  - All databases that are to be shared
  - All log data sets
  - All BSDS data sets
  - User integrated catalog facility catalogs for shared databases
  - All coupling facilities used by the DB2 data sharing group

Planning a Naming Convention

Carefully consider the naming convention you will use to name the various parts of the data sharing system. Names must be assigned for both IRLM and DB2 groups, and for members within a group. One recommendation is to make names and prefixes unique within the MVS Sysplex. Although this uniqueness is not required for all names, it can help you avoid problems identifying entities and moving them among MVS systems in the Sysplex.

In this section, we describe the names for which you must choose values. Other names are generated during DB2 installation. A complete list of names is in Appendix A, "DB2 and IRLM Names" on page 271. "Naming Recommendations" on page 56 contains a suggested naming convention, which we use when we describe the names in this section. If you want to change the name of an existing DB2 subsystem to conform to your naming convention, see "Renaming the DB2 Subsystem" on page 96.

Choosing Names Used in the DB2 Group

Certain names are shared by all members of the data sharing group.

DB2 group name

The name that encompasses the entire group. The coupling facility structure names are based on the group name, as described in "Coupling Facility" on page 47. The group name must be unique within the Sysplex. If you use this name as a basis for the location name, this name must be unique within the network.

The group name can be up to 8 characters long and can consist of the characters A-Z, 0-9, $, #, and @. The group name must begin with an alphabetic character.

To avoid names that IBM uses for its XCF groups, do not begin with the letters A-I unless the first three characters are DSN. Do not use the string SYS as the first three characters, and do not use the string UNDESIG as your group name.

catalog alias

This name can be up to 8 characters long. This is the name of the MVS catalog alias, which you must place in the MVS master catalog. We recommend that this name be the same as the group name. An example of a DB2 catalog alias name is DSNDB0G.

group attachment name

This name can be up to 4 characters long. This name is used by the TSO/batch attachment, the call attachment facility (CAF), utilities, and the Recoverable Resource Manager Services attachment facility (RRSAF) as a “generic” attachment name. For more information about using the group attachment name, see “Using the Group Attachment Name” on page 149.

An example of a group attachment name is DB0G.

location name

This name is used if the group is going to be processing distributed requests. The group is treated as a single location by remote requesters. This name can be up to 16 characters long.

generic LU name

This name allows remote requesters to configure their systems to treat the data sharing group as a single LU. This name can be up to 8 characters long. See “Chapter 5. Communicating with a Data Sharing Group” on page 123 for more information about the generic LU name.

Choosing Names Used by Members

These are names that must be unique within the data sharing group or, in certain cases, the MVS Sysplex.

member name

This name can be up to 8 characters long and can consist of the characters A-Z, 0-9, $, #, and @. This is the name of an individual member of a particular DB2 group. DB2 uses this name to form its MVS cross-system coupling facility (XCF) member name. An example of a member name is DB1G.

member subsystem name

This name can be up to 4 characters long and is the name used by all the attachment interfaces. It must be unique within the MVS Sysplex. We recommend that the member name and member subsystem name be the same. An example of a member subsystem name is DB1G.

LU name

This name must be unique within the data sharing group and the network. See Section 3 of *Installation Guide* for more information about choosing LU names.

command prefix

This prefix can be up to 8 characters long and is used to direct commands entered from an MVS console to a particular DB2 subsystem. The default is the concatenation of the hyphen character (-) with the subsystem name. See Section 2 of *Installation Guide* for more information about valid characters for a command prefix.

As described in “Registering Command Prefixes and Group Attachment Name” on page 77, this string is specified as a parameter on the IEFSSN*xx* subsystem definition. An example of a command prefix is -DB1G. You can have blanks between the command prefix and the command.

Do not assign a command prefix that is used by another subsystem or that can be interpreted as belonging to more than one subsystem or MVS application. Specifically, do not specify a multiple-character command prefix that is a subset or a superset of another command prefix starting from the first character. For example, it is invalid to assign a hyphen (-) to one subsystem and '-DB2A' to another. Similarly, it is also invalid to assign '?DB2' to one subsystem and '?DB2A' to another. It is valid, for example, to assign '-DB2A' and '-DB2B' to different DB2 subsystems.

work file database

This name can be up to 8 characters long. Each DB2 member has its own work file database (known as DSNDB07 in a non-data sharing environment). One member of the data sharing group can have the name DSNDB07, but you might want to create a work file database with a more meaningful name, such as WRKDB1G, for member DB1G.

You cannot specify a name that begins with DSNDB unless the name is DSNDB07.

load module for subsystem parameters

This name can be up to 8 characters long. Each member has its own subsystem parameters. An example name is DSNZP01G, a naming convention which you can use to associate member DB1G with DSNZP01G.

Choosing Names for Data Sets: When choosing names for member data sets, remember that data set names beginning with *membname* must have a master catalog alias to point to the catalog where the data sets are cataloged. The DB2 installation process does not create this catalog alias. One way to handle this is to begin member data set names with *catalias* and a member-related qualifier. For example, member data set names could have a form *catalias.membname.xxxxx*. This format eliminates the need to have a master catalog alias for *membname*.

member BSDS names

These names can be up to 33 characters long. Sample BSDS names are DSNDB0G.DB1G.BSDS01 and DSNDB0G.DB1G.BSDS02.

active log data set prefixes

This prefix can be up to 30 characters long. Sample active log data prefixes are DSNDB0G.DB1G.LOGCOPY1 and DSNDB0G.DB1G.LOGCOPY2.

archive log data set prefixes

These prefixes can be up to 35 characters long unless you want the data sets time stamped. If they are time stamped with a 2-digit year, only 19 characters can be used. If they are time stamped with a 4-digit year, only 17 characters can be used. Use the TSTAMP subsystem parameter of installation macro DSN6ARVP to determine which timestamp format you use.

Sample archive log data set prefixes are DSNDB0G.DB1G.ARCLG1 and DSNDB0G.DB1G.ARCLG2.

Choosing IRLM Names

Each DB2 in the data sharing group must have its own IRLM. The IRLM group name, subsystem name, and member ID are parameters on the IRLM startup procedure. This means that there must be a separate IRLM procedure for every IRLM in the group. Figure 22 shows the relationship between DB2 and IRLM groups and names.

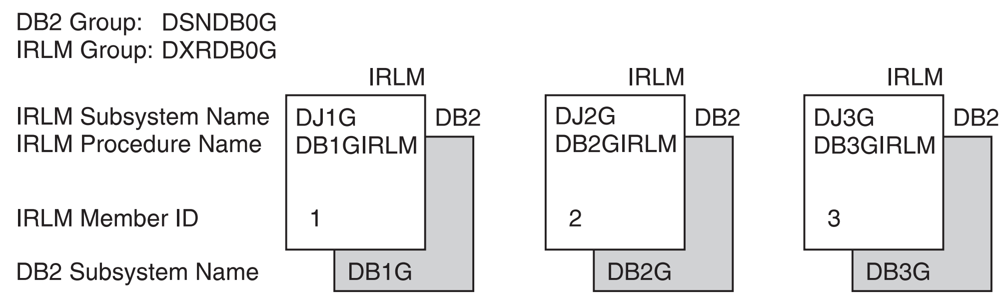

Figure 22. Relationship between DB2 and IRLM Group Names

You must choose the following IRLM names before installing DB2:

group name
The name that encompasses the entire IRLM group. This name can be up to 8 characters long and can consist of the characters A-Z, 0-9, $, #, and @. The group name must begin with an alphabetic character. To avoid names that IBM uses for its XCF groups, do not begin with the letters A-I unless the first three characters are DXR. Do not use the string SYS as the first three characters, and do not use the string UNDESIG as your group name.

The IRLM group name is a parameter (GROUP=) on each DB2 member's IRLM procedure. A sample IRLM group name is DXRDB0G, which is a convention that makes it easy for you to tell which IRLM group is paired with which DB2 group.

subsystem name
Each IRLM must have a subsystem name. This name can be up to 4 characters long and is a parameter on the IRLM procedure. This subsystem name must be unique within the data sharing group so that DB2 does not connect to the wrong IRLM.

A sample subsystem name is DJ1G. The “1G” characters indicate that this IRLM is paired with the DB2 subsystem DB1G. See “Suggested Convention” on page 57 for more information about how this identifier is used.

IRLM procedure name
This name can be up to 8 characters long. We recommend using the DB2 member subsystem name followed by “IRLM.”

Each DB2 member knows its IRLM by the procedure and subsystem name saved in that member's installation parameter load module.

A sample IRLM procedure name is DB1GIRLM.

IRLM member ID
This ID is a number between 1 and 255 (inclusive). The ID uniquely names an IRLM within a group. See the description of START *irlmproc*

in Chapter 2 of *Command Reference* for more information about this value.

Coupling Facility Structure Names

The names for coupling facility structures must conform to a strict naming convention based on the DB2 group name. Sample names are shown here:

Table 2. Sample Coupling Facility Structure Names

Structure Type	Example Name
Cache structure (group buffer pools)	DSNDB0G_GBP0
Lock structure	DSNDB0G_LOCK1
List structure (shared communications area)	DSNDB0G_SCA

Lock Structure Name

The name you must use on the CFRM policy to define the lock structure to the coupling facility is:

*groupname*_LOCK1

Shared Communications Area

The name you must use on the CFRM policy to define the SCA to the coupling facility is:

*groupname*_SCA

Group Buffer Pool Names

The name you must use on the CFRM policy to define the group buffer pool to the coupling facility is:

*groupname*_GBP*xxxx*

Where GBP*xxxx* is the name of the group buffer pool:

- 4KB group buffer pools are named GBP0, GBP1, ..., GBP49
- 32K group buffer pools are named GBP32K, GBP32K1, ... , GBP32K9

Naming Recommendations

You have control over the names you assign to entities during the DB2 installation process. **After installation, some names, such as the group name and member names, cannot be changed.** Because you must name a large number of items, and you might have to add new members or move existing members to different systems in the future, managing all these items is easier if you choose and maintain a consistent naming convention.

If you are enabling the originating member of the group from an existing DB2, it is possible to build a naming scheme around the existing names used in your DB2 subsystem to reduce the disruption for existing applications. However, you might want to change some names now, *before* enabling data sharing, to lay the foundation for a solid naming convention.

Another name to choose carefully is the catalog alias for the group. It is very difficult to change that name. The procedure to do this for a single system is documented in Section 2 (Volume 1) of *Administration Guide*. To change the catalog alias for the group, you have to bring the entire group down and do the procedure for every member of the group.

Assumptions

The DB2 data sharing naming recommendations and installation process support assume the following MVS system Sysplex configuration:

- One SYS1.PARMLIB shared by all MVS systems
- One SYS1.PROCLIB shared by all MVS systems
- One integrated catalog facility master catalog shared by all MVS systems

DB2 data sharing does not require that the MVS Sysplex be configured in this manner. However, the following DB2 naming recommendations support such a configuration, and the DB2 installation process assumes such a configuration.

If the MVS Sysplex is configured differently, you must customize the install process. For example, if you are using different SYS1.PARMLIBs, make sure that the DB2 data sets are in the APF list in each PARMLIB. If you are using different PROCLIBs, modify the JCL to point to the correct libraries during installation. And if you are using more than one integrated catalog facility master catalog, put the DB2 catalog alias in each master catalog.

Suggested Convention

Following are some suggestions to consider when naming various DB2 entities:

- The names with the shortest length are the subsystem names. Subsystem names are limited to four characters. After these names are selected, they can be used as the basis for choosing the other names. 4-character names are needed for the following:

  – Group attachment name; one per group

  – DB2 member subsystem name; one for each DB2 member

  – IRLM member subsystem name; one for each DB2 member

  One possible naming convention is the one suggested in *System/390 MVS Sysplex Application Migration*, which takes a Sysplex-wide approach to naming. That publication suggests assigning subsystem names of the form '*ctmg*' where:

  c Denotes a particular collection of logically related applications or subsystems.

  t Denotes a type of resource, such as “B” for DB2 or “J” for IRLM.

  m Denotes a particular member within a DB2 data sharing group or IRLM group. This identifier is also used to associate subsystems with their MVS for recovery purposes when using automatic restart.

  g Denotes a particular DB2 group.

  Our examples use G as the group identifier. Therefore, our naming scheme has a subsystem name of DB1G for the first member of the DB2 data sharing group. The second member is DB2G, and so on. Table 3 on page 58 shows how the member identifier relates to a particular MVS and how the group identifier associates DB2 members across a Sysplex.

Table 3. Subsystem Names in a Sysplex. Columns are names associated with a particular MVS recovery group (MVS '2', for example). Rows include DB2 and IRLM data sharing groups.

MVS1	MVS2	MVS3	MVS4
DA11 (CICS)	DA21 (CICS)	DA31 (CICS)	DA41 (CICS)
⋮	⋮	⋮	⋮
DB1G (DB2)	DB2G (DB2)	DB3G (DB2)	DB4G (DB2)
DJ1G (IRLM)	DJ2G (IRLM)	DJ3G (IRLM)	DJ4G (IRLM)
DD11 (DBCTL)	DD21 (DBCTL)	DD31 (DBCTL)	DD41 (DBCTL)
⋮	⋮	⋮	⋮

If you are enabling an existing DB2 subsystem to take advantage of data sharing, and have existing applications using that DB2, consider keeping the existing subsystem name and using it for the group attach subsystem name. This allows existing applications to use the data sharing member without changing to use a new subsystem name.

- Make the DB2 group name, the DB2 location name, and the DB2 integrated catalog alias the same.

  A data sharing group uses the same catalog alias name. This catalog alias is used as the high level qualifier for the DB2 directory (DSNDB01), catalog (DSNDB06), default database (DSNDB04), and work file database VSAM data sets.

- Consider making a member name and the member's subsystem name the same.
- For a member's command prefix, use the member's subsystem name with a hyphen (-) in front of it.

DB2 Data Sharing Naming Example

Here is an example of names chosen for a 12-member DB2 data sharing group. The example uses the subsystem naming convention suggested above. We use '0' (a zero) to represent the group identifier. In Table 4 the '#' character is used to denote a character from the set {1,2,3,4,5,6,7,8,9,A,B,C}. These characters are used to denote a particular member's name.

Table 4 (Page 1 of 2). DB2 Data Sharing Naming Example

Entity	Name
Group name	DSNDB0G
Catalog alias	DSNDB0G
Group attach name	DB0G
Member names	DB#G
Member command prefix	-DB#G
Member subsystem name	DB#G
Member BSDS	DSNDB0G.DB#G.BSDS01 DSNDB0G.DB#G.BSDS02
Member active log prefix	DSNDB0G.DB#G.LOGCOPY1 DSNDB0G.DB#G.LOGCOPY2

Table 4 (Page 2 of 2). DB2 Data Sharing Naming Example

Entity	**Name**
Member archive log prefix	DSNDB0G.DB#G.ARCLG1 DSNDB0G.DB#G.ARCLG2
Member work file database	WRKDB#G
Member procedure names	DB#GMSTR DB#GDBM1 DB#GDIST DB#GSPAS
Member subsystem parameters load module	DSNZP0#G
IRLM group name	DXRDB0G
IRLM member subsystem name	DJ#G
IRLM procedure name	DB#GIRLM
IRLM member ID	Number 1-12 corresponding to #

| ***Distributed Naming Convention:*** The example in Table 4 on page 58 does not include names for distributed processing. Those naming conventions are probably part of a much broader convention. See your network administrator for more information about choosing names for distributed access.

Planning for Availability

When planning your data sharing configuration for the highest availability, a primary concern should be for the physical protection of the coupling facility and the structures within the coupling facility. The SCA and lock structure are both necessary for the group to function. If something happens to those structures, they can be dynamically rebuilt on another coupling facility. However, if the rebuild fails, all members of the group fail, and this necessitates a group restart.

Although a loss of a group buffer pool does not require a group restart, availability for users and important applications require that data in a group buffer pool be available as quickly as possible after a failure. Group buffer pools have several availability options, depending on the type of failure to occur. For the highest availability, you can duplex the group buffer pool, allowing DB2 to switch to a secondary group buffer pool if the first one fails.

If a group buffer pool structure fails, the group buffer pool can be recovered automatically from data on the DB2 logs. If members lose connectivity to the group buffer pool, the group buffer pool can be rebuilt onto another coupling facility to which the members can connect.

This section describes the following topics:

- “Using MVS's Automatic Restart Capability” on page 60
- “Coupling Facility Availability” on page 61
- “DB2 Resource Availability” on page 66

Using MVS's Automatic Restart Capability

If you are using MVS/ESA Version 5 Release 2 or later, you can use automatic restart. The purpose of using automatic restart is to reduce the time a particular system is down. When DB2 or IRLM stops abnormally, the surviving MVSs analyze the situation to determine whether MVS failed, too, and where DB2 or IRLM should be restarted. It then restarts DB2 or IRLM appropriately.

Advantage of Automatic Restart

By restarting DB2 and IRLM quickly, locks held by failed members are released quickly, allowing applications running on other DB2 members to access data for which the failed member is holding incompatible locks. Consider using automatic restart in conjunction with the RETLWAIT subsystem parameter described in Table 12 on page 91.

You must have DB2 installed with a command prefix scope of “started” to take advantage of automatic restart. See “Registering Command Prefixes and Group Attachment Name” on page 77 for more information.

Using an Automatic Restart Policy

You control how automatic restart works by using MVS automatic restart policies. When the automatic restart function is active, the default action (for both sharing and non-sharing DB2s and IRLMs alike) is to restart the subsystems when they fail. If this default action is not what you want, then you must create a policy defining the action you want taken.

Creating the Automatic Restart Policy

If the default action of restarting DB2 is the action you want, you do not need to create a policy. However, if you want to change the default, you need to know the automatic restart ELEMENT name. In a data sharing group, the DB2 ELEMENT is the concatenated DB2 group name and member name (such as DSNDB0GDB1G).

For IRLM, the element name is the IRLM group name, concatenated with the IRLM subsystem name and 3-character member ID (such as DXRDB0GDJ1G001).

You can also specify a wild card (such as DSNDB0G*) if you want to use a single policy statement for all members in the group.

To specify that DB2 or IRLM is not to be restarted after a failure, include RESTART_ATTEMPTS(0) in the policy for that DB2 or IRLM element. For IRLM, you can also use the following command to indicate that you want IRLM to deregister from automatic restart manager when it comes down. Deregistering prevents IRLM from automatically restarting after you bring it down:

```
MODIFY irlmproc,ABEND,NODUMP
```

However, if your DB2 has YES for the AUTO START option of installation panel DSNTIPI, and if MVS restarts DB2 automatically, DB2 will restart IRLM, too.

Coupling Facility Availability

For high availability, you should have at least two coupling facilities, and one of those should be a *dedicated* coupling facility, which means that the coupling facility is not running in the same CPC as another MVS system. With two coupling facilities, you can specify that structures be allocated in the second coupling facility in case the first one is damaged.

With more than one coupling facility, you can also consider duplexing the group buffer pools. With duplexing, a secondary group buffer pool is always on standby in another coupling facility, ready to take over should the primary group buffer pool structure fail or if there is a connectivity failure. If you have three coupling facilities, you can even maintain duplexing while performing maintenance on one of the coupling facilities. For more information about duplexing group buffer pools, see “Duplexing group buffer pools” on page 65.

It is also recommended that at least one of the coupling facilities be non-volatile.

See *System/390 MVS Sysplex Hardware and Software Migration* for more information about configuring the coupling facility for high availability.

Recommendations for Placement

We suggest that you physically separate the coupling facilities from each other and from the rest of the MVS images that are using those coupling facilities. The reason for this is that after a major outage, having the lock and SCA structures separated from the systems using them can minimize the chances of having to do a lengthy group restart.

Quick recovery of the group buffer pools uses information in the lock structure and SCA to determine which databases must be recovered. This is known as *damage assessment*. Consider putting the lock structure and SCA in a different coupling facility than important cache structures (such as group buffer pool 0). This makes it less likely that you would lose the lock structure, SCA, and the group buffer pool at the same time.

If you should lose the lock structure or SCA at the same time as one or more group buffer pools, DB2 waits until the lock structure and SCA are rebuilt before doing damage assessment.

Preparing for Structure Failures

For structure failures, DB2 can recover resources in the structures. For the lock and SCA, this is done by rebuilding the information in the structures from information contained in DB2's virtual storage. For group buffer pools, DB2 can automatically recover that information by reading the logs and applying the changes to the data. Or, if you are duplexing the group buffer pool, DB2 just switches over to using the secondary group buffer pool.

Lock and SCA Structure Failures: DB2 can rebuild the lock and SCA in the same coupling facility, or in an alternate coupling facility, assuming:

- You have specified the alternate coupling facility in the CFRM policy preference list.
- You have allocated enough storage in the alternate coupling facility to allow the rebuild of the structures there.

The information used to recover the lock structure and SCA is contained in DB2's virtual storage, not the logs. If the lock structure and SCA cannot be rebuilt, all active members in the group terminate abnormally, and a group restart is required to recover the lost information from the logs.

Group Buffer Pool Structure Failure: Group buffer pools can be recovered from the log when they fail, or you can switch to a secondary group buffer pool if the CFRM policy for the group buffer pool indicates that duplexing is allowed and the group buffer pool is currently running in duplexed mode.

Recovery can occur manually, as the result of a START DATABASE command, or it can occur automatically, because the group buffer pool is defined with the AUTOREC(YES) option. In either case, to reduce the time needed for group buffer pool recovery, use the ALTER GROUPBUFFERPOOL command to make group buffer pool checkpoints more frequent. Weigh the benefit of faster recovery with the processing resources used for the checkpointing. (You can reduce checkpoint costs considerably when OS/390 is at the appropriate level of maintenance. See “Tuning the Group Buffer Pool Checkpoint Interval” on page 250 for more information.) Checkpointing is also used with duplexed group buffer pool, in case both structures fail.

Be sure to specify one or more alternate coupling facilities in the CFRM preference list for the group buffer pools because a group buffer pool can be allocated in an alternative coupling facility when a new connection is made to it. See “Problem: Group Buffer Pool Structure Failure (No Duplexing)” on page 177 for more information about group buffer pool structure failures.

More about Automatic Recovery: Automatic recovery is enabled by the AUTOREC option of the ALTER GROUPBUFFERPOOL command. Automatic recovery is faster than manual recovery because DB2 can optimize internal processing of the recovery. For automatic recovery to be initiated for a group buffer pool, all of the following conditions must be true:

- The AUTOREC option is set to YES for the group buffer pool.
- All active connectors to the group buffer pool are at Version 5 or a subsequent release.
- There was at least one actively connected member at the time that the group buffer pool failed. This member must have successfully completed damage assessment. If the entire Sysplex fails, then no member meets this requirement, and therefore DB2 does not initiate automatic recovery on the subsequent restart.

For a duplexed group buffer pool, DB2 can use automatic recovery if both instances of the group buffer pool are damaged.

Preparing for Connectivity Failures

To prevent having DB2 fail because of a coupling facility channel failure, consider having dual channels (sometimes called *links*) between each CPC and a coupling facility. A channel failure is more likely than a failure in the coupling facility and a loss of connectivity to the SCA or lock structure can bring that particular DB2 member down, unless you specify an alternative coupling facility in the CFRM policy preference list.

A total failure of the coupling facility, such as a power failure to the coupling facility or some problem with coupling facility control code, is also interpreted as a connectivity failure by MVS and DB2. See “Problem: All Members Have Lost Connectivity” on page 174 for more information about recovery from lost connections.

Rebuilding Structures when Connectivity is Lost: This section describes the recovery action when some or all DB2 members have lost their connectivity to the simplexed group buffer pool. See “What if the group buffer pool is actively duplexing?” on page 64 for information about connectivity failures for duplexed group buffer pools.

When connectivity is lost, DB2 can rebuild those structures on the alternate coupling facility that is specified in the CFRM policy. To make this happen, you **must** have an active Sysplex failure management (SFM) policy. When DB2 rebuilds, it attempts to allocate storage on the alternate coupling facility. It uses the value of INITSIZE for the structure on the CFRM policy of the alternate coupling facility. If DB2 cannot allocate the storage for the SCA or lock structure, the rebuild fails. If it cannot allocate the storage for the group buffer pools, the changed pages are written to DASD. See “Connectivity Failure for Lock Structure and SCA” on page 171 and “Connectivity Failure for Group Buffer Pools” on page 172 for more information about failure scenarios.

To control when a rebuild can occur, MVS lets you specify a *rebuild threshold*. This value is a percentage that you specify in the CFRM policy on the REBUILDPERCENT parameter. MVS uses the REBUILDPERCENT value in the CFRM policy to determine whether to initiate a structure rebuild when there is a loss of connectivity to the coupling facility that contains the structure. The percentage is based on the SFM *weights* of all the systems that have active connectors to a structure at the time. You also specify weights on the SFM policy.

MVS calculates the total system weight of (A) all systems with at least one active connection to the structure that have lost connectivity, and (B) all systems with at least one active connection to the structure. If there are multiple connections to a structure from a single system (for example, two DB2 members on the same MVS), then that system weight is counted only once.

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

MVS determines whether or not to rebuild as follows:

- If $(A/B) \times 100$ is greater than or equal to the REBUILDPERCENT value, then rebuild.
- Otherwise, don't rebuild. Instead, the affected DB2 disconnects from the group buffer pool.

In the example above, $(10/80) \times 100$ is the value compared to the REBUILDPERCENT value. If the value of REBUILDPERCENT is 25, then a rebuild is not initiated. See “Problem: A Subset of Members Have Lost Connectivity” on page 175 for more information about this situation.

REBUILDPERCENT(100), the default, means that the structure is not automatically rebuilt unless all the connected members lose connectivity to the structure. Certain types of coupling facility failure, such as a power failure, are treated by MVS as a complete loss of connectivity and are handled by DB2 as a structure failure. See “Problem: All Members Have Lost Connectivity” on page 174 for more information about this situation.

A larger rebuild percent is most appropriate for a group of many small systems. In such a group, you might prefer to have one member lose the use of one group buffer pool rather than temporarily disrupt all of the members using the structure while the structure is rebuilt. If you have high availability requirements, it might be better to always allow the automatic rebuild by specifying a small rebuild percent. For more information, see *MVS/ESA Setting Up a Sysplex*.

What if the group buffer pool is actively duplexing?: A key difference between duplexing and rebuilding is that a normal rebuild treats the 100% lost connectivity failure as a structure failure whereas the duplexed group buffer pool recovery actions are the same for both structure failure and lost connectivity.

If the group buffer pool is duplexed, the recovery action is to use the group buffer pool with good connectivity. This action is better than rebuilding because it is far less disruptive. If your CFRM policy specifies DUPLEX(ENABLED), then automatic reduplexing is attempted (which is about as disruptive as a rebuild).

Monitoring Rebuild Events

You can use a performance class 20 trace (IFCIDs 0267 and 0268), or you can examine the messages returned to the console to monitor how long a rebuild of a structure takes and the reason for the rebuild (such as lost connectivity, operator command, or to establish duplexing).

Coupling Facility Volatility

There are times when a coupling facility can go into a *volatile* state, which means that if the power fails, data in the coupling facility at the time of failure is not saved. If the coupling facility is configured to be non-volatile (using the proper power backup), volatility is generally a transient state that might occur, for example, if you take the battery out.

DB2 issues a warning message if allocation occurs into a volatile coupling facility. A change in volatility after allocation does not have an effect on your existing structures.

Advantages of a Non-Volatile Coupling Facility: If you lose power to a coupling facility that is configured to be non-volatile, the coupling facility enters power save mode, saving the data contained in the structures. When power is returned, there is no need to do a group restart, and there is no need to recover the data from the group buffer pools. For systems requiring high availability, non-volatile coupling facilities are recommended.

For more information about coupling facility volatility options, see *System/390 MVS 9672/9674 System Overview*.

Duplexing group buffer pools

Running some or all of your group buffer pools in duplex mode is one way to achieve high availability for group buffer pools across many types of failures, including lost connections and damaged structures.

How duplexing works

With a duplexed group buffer pool, you have two allocations of the same group buffer pool that use one logical connection. Each structure allocation must be in a different coupling facility. One allocation is called the *primary structure*. DB2 uses the primary structure to keep track of page registration and cross-invalidation, and that primary structure is the one from which changed data is cast out to DASD. From an MVS perspective, duplexing is really an extended rebuild, so OS/390 documentation and commands sometimes call the primary structure the *old* structure.

The other allocation of the structure is called the *secondary structure* (referred to by MVS as the *new* structure). When changed data is written to the primary structure, it is also written to the secondary structure.

MVS commands let you stop and start duplexing and to switch which structure is the primary structure.

We recommend that at least one of the group buffer pool structures be in a non-volatile coupling facility. If power is lost to both coupling facilities and both coupling facilities are volatile, hen the group buffer pool must be recovered from the logs.

Characteristics of the secondary structure: These characteristics are different than those of the primary structure:

- DB2 does not read data from the secondary structure.
- DB2 does not use the secondary structure for cross-invalidation of pages.
- DB2 does not cast out data to DASD from the secondary structure.

When planning for storage, it's a good idea to keep the secondary structure as close to the same size and ratio as the primary structure so that performance is not seriously affected if the secondary must take over.

Coupling facility storage considerations for duplexing: Duplexing does not require any additional coupling facility storage beyond that which you would use for a highly available simplexed group buffer pool. For simplexed structures, you must reserve enough spare capacity in the coupling facilities to be able to absorb the structures of any failed coupling facility. For example, if you have two coupling facilities, each with 1 GB of memory for a total of 2 GB, then you need to ensure that the total size of the structures across the two coupling facilities does not exceed 1 GB (50% of the total coupling facility storage). With duplexing, you are using that previously reserved storage for the secondary structure. You do not need to configure extra coupling facility storage for duplexing.

Requirements

For duplexing to work, the following conditions must be true:

- There must be at least two coupling facilities with a CFLEVEL of 5 or higher in the CFRM policy preference list for the group buffer pool. All members of the data sharing group must have physical connectivity to both coupling facilities in which the primary and secondary structures reside.
- At least one DB2 member must be actively connected to the group buffer pool.
- All connected DB2s must be duplexing-capable. To be capable means that each connected DB2 member must be at Version 5 with APAR PQ17797 applied, or at a subsequent release of DB2.
- All connected DB2s must be running on OS/390 Release 3 or a subsequent release that has APAR OW28460 installed. (The function is included in the base for OS/390 Release 6.)
- The group buffer pool must be defined with GBPCACHE(YES), the default.

Establishing duplexing

There are two options on the CFRM policy for duplexing. DUPLEX(ENABLED) automatically starts duplexing. DUPLEX(ALLOWED) is not automatic. You must issue a command to start duplexing. For more information about starting and stopping duplexing, see “Starting and stopping duplexing for a group buffer pool” on page 189.

Performance of duplexing

The process of establishing duplexing can be somewhat disruptive because access to the group buffer pool is quiesced while the secondary structure is allocated and changed pages are copied from the primary structure to the secondary structure (or cast out to DASD). Transactions that need access to the group buffer pool during this process are suspended until the process is complete. Because of this disruption, it is best to establish duplexing at a time of low activity on the system. By specifying DUPLEX(ALLOWED) on the CFRM policy, you have more control over when to establish duplexing than you do with DUPLEX(ENABLED).

In general, it takes a bit more processor and elapsed time to do duplexed group buffer pool writes and castout processing than it does to do simplexed group buffer pool writes and castout processing. Read performance is unaffected by duplexing.

The statistics and accounting trace classes contain information about group buffer pool duplexing.

Monitoring duplexing rebuilds: When a group buffer pool is duplexed, it is considered to be in an extended rebuild status called a duplexing rebuild. This activity is reported in IFCIDs 0267 and 0268 along with other reasons for rebuilding.

DB2 Resource Availability

DB2 availability considerations for a data sharing group are basically the same as for a single subsystem. For a data sharing group, the catalog and directory data sets are even more important than with a single subsystem because there is a single catalog and directory for all members of the group. Consider placing the catalog and directory behind a 3990 control unit with dual write capability for hardware duplexing. Another possibility is to use the RAMAC Array Subsystem for its improved availability benefits.

Assign data that require high availability to group buffer pools that reside on non-volatile coupling facilities and consider using group buffer pool duplexing.

Estimating Storage

This section gives you information about estimating storage for coupling facility structures and for DB2 resources.

Storage for Coupling Facility Structures

It is difficult to give precise estimates for coupling facility structure sizes used in DB2, partly because every environment is different, and partly because storage allocation is affected by the processor model and level of coupling facility control code you have. Use the information given here for your initial estimates. If you are running on MVS Version 5 Release 2 or subsequent releases, you can use these initial estimates as your initial size values (INITSIZE) and, depending on how much your work load varies, give a larger value for SIZE. You can use an MVS SETXCF START,ALTER command to alter the structure from INITSIZE to SIZE and back again. See “Changing Structure Sizes” on page 72 for more information.

For duplexed group buffer pools, the SIZE and INITSIZE values apply to both instances of the structure.

Don't Overestimate the SIZE Parameter: Coupling facility structures contain some control structures that are static. When the structure is initially allocated, these static structures are allocated to accommodate the potential size of the structure. In other words, the size of the static structures is proportional to the maximum size (the SIZE parameter). If the SIZE is very much larger than the INITSIZE, there might be cases where a large percentage of the INITSIZE is used for these static structures, leaving you little in the way of usable storage for storing items in the structure.

In general, it is a good idea to specify a SIZE that is larger than INITSIZE but to keep the SIZE parameter to no more than 2 to 3 times the INITSIZE the lock and SCA, and to no more than 4 times the INITSIZE for group buffer pools. For example, if INITSIZE for a group buffer pool is 100 MB, specify a SIZE of 400 MB or less. For the lock structure and SCA, it's convenient to have a larger SIZE but not as critical as for group buffer pools, because you can always rebuild the lock and SCA if they run short on storage.

Other Sources of Information: For the cache structures (group buffer pools) we give both “rule of thumb” estimates and input you can use for the storage formulas given in an appendix of *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide*. Consult this guide if you are looking for detailed information about planning for storage in the coupling facility.

When you decide what your structure sizes are, include those values in the CFRM policy definition. See *MVS/ESA Setting Up a Sysplex* for more information about creating CFRM policies.

Group Buffer Pool Sizes

A group buffer pool (coupling facility cache structure) consists of two parts: data pages (sometimes called *data entries*) and directory entries.

Data Pages: This is the place in the group buffer pool in which the data page resides. The size of a data page is either 4KB or 32KB, depending on the page size supported by the corresponding DB2 buffer pools.

If you are caching changed data only, you need enough space to cache changed data plus enough extra for pages that are frequently re-referenced. By having those frequently referenced pages in the group buffer pool, you can decrease the amount of time it takes for any member to refresh that page in its member buffer pool because you are avoiding the DASD I/O.

Directory Entries: A directory entry specifies the location and status of an image of a page somewhere in the data sharing group, whether the image appears in the group buffer pool or in one of the member buffer pools. There is only one directory entry for any given page, no matter in how many places that page is cached.

If you are using coupling facility level 0 (CFLEVEL=0), then the size of a directory entry is:

- 208 bytes for group buffer pools that hold 4KB pages
- 264 bytes for group buffer pools that hold 32KB pages

If you are using a different level of coupling facility, see *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide* for the exact size.

Specifying a Ratio: The space allocated for a group buffer pool is divided into two parts according to the ratio of the number of directory entries to the number of data pages. When you originally define a structure in the CFRM policy for a group buffer pool, you specify its total size. For DB2, the ratio defaults to five directory entries per data page. Later, you can change the ratio with the ALTER GROUPBUFFERPOOL command. See “Making Sure Group Buffer Pools are the Right Size and Ratio” on page 258 for information about detecting problems with the size and ratio of group buffer pools.

Storage Estimate for Group Buffer Pools that Cache Changed Data

The size of a group buffer pool is related to the amount of sharing and the amount of updating. An estimate must be based on the total amount of member buffer pool storage multiplied by a percentage based on the amount of update activity. The more sharing and updating there is, the more pages must be cached in the group buffer pool and the more directory entries are needed to track inter-DB2 buffering.

For installation planning purposes, the following rule-of-thumb is offered as an initial estimate for the size of a DB2 group buffer pool for table spaces, indexes, or partitions that cache only changed data (GBPCACHE CHANGED):

Sum the local buffer pool storage for this buffer pool number (both virtual and hiperpool) across all the DB2s of the group. Then, multiply this amount by one of these factors:

Factor	Condition
10%	for light sharing with a low amount of updating
20%	for medium sharing with a moderate amount of update activity
40%	for a high amount of sharing with a lot of update activity

Bear in mind that the type of workload you run can influence the amount of storage used. For example, if you have “hot spots” in which a single page gets updated frequently rather than having updates spread throughout the table space, then you might not need as much storage for caching.

Example

The total virtual buffer pool storage for all the DB2s of the group is 400MB. A medium amount of read/write sharing is expected in the environment. The calculation is now:

```
400MB x 20% = 80MB
```

Storage Estimate for Caching All Data

For installation planning purposes, the following rule-of-thumb provides an initial estimate of the size of a DB2 group buffer pool when the installation is caching read-only pages with changed pages (GBPCACHE ALL):

Calculate the sum of local buffer pool storage for this buffer pool number (virtual only) across all the DB2s of the group. Then, multiply this amount by one of these factors:

Factor	Condition
50%	Few table spaces, indexes, or partitions specify GBPCACHE ALL
75%	Half the table spaces, indexes, or partitions specify GBPCACHE ALL
100%	Almost all the table spaces, indexes, or partitions specify GBPCACHE ALL for a heavy sharing environment

Example

The local virtual buffer pool storage (do not count hiperpool storage) on all the DB2s of the group adds up to 200 megabytes. Half of the page sets coming into the pool are defined as GBPCACHE ALL. The calculation is now:

```
200MB x 75% = 150MB
```

PR/SM Formulas for Calculating Group Buffer Pool Sizes

Another way to calculate group buffer sizes is to use the coupling facility allocation formulas for cache structures found in *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide* Table 5 on page 70 contains information used in those formulas. The size of cache structures in DB2 can vary greatly based on the amount of data for which there is inter-DB2 R/W interest at any given time. You will most likely have to monitor the use of the group buffer pools and adjust their sizes accordingly.

Table 5. Information for Calculating Cache Structure Sizes

Parameter	DB2 Value	Explanation
MSC	**1**	Maximum storage class
MCC	**1024**	Maximum castout class
MDAS	Dependent on page size being cached: **1** for 4KB pages **8** for 32KB pages	Maximum data area size.
DAEX	**4**	Data area element characteristic.
AAI	**0**	Adjunct assignment indicator.
R_de	Set on ALTER GROUPBUFFERPOOL. Default is **5**. See Table 6 on page 70 for more information on determining this value.	The directory portion of the directory-to-data ratio.
R_data	Set on ALTER GROUPBUFFERPOOL command. See Table 6 on page 70 for more information on determining this value.	The data object portion of the target directory-to-data ratio.

Table 6. Formulas for Determining R_data and R_de. N is the RATIO entered on ALTER GROUPBUFFERPOOL command.

Page Size	N has no decimal point	N has decimal point
4KB R_de	N	N×10
4KB R_data	1	10
32K R_de	N	N×10
32K R_data	8	80

Lock Structure Size

The coupling facility lock structure contains two parts. The first part is a *lock table* used to determine if there is inter-DB2 R/W interest on a particular *hash class* (resources that hash to a particular place in the lock table). The second part is a list of the update locks that are currently held (sometimes called a *modify lock list*). IRLM determines how the space of the lock structure is divided between these two parts, and the total size of the lock structure must be large enough to prevent performance problems.

IRLM reserves space for “must complete” functions (such as rollback or commit processing) so that a shortage of storage does not cause a DB2 subsystem failure. However, if storage runs short in the lock structure, there can be an impact on availability (transactions are terminated), response time, and throughput. See “Monitoring DB2 Locking” on page 229 and “Changing the Size of the Lock Structure” on page 235 for more information.

Specifying the Lock Entry Size

The LOCK ENTRY SIZE field of installation panel DSNTIPJ determines how much space is needed for lock contention control information (that is, individual entries in the *lock table*). The LOCK ENTRY SIZE of the first IRLM to join the group determines the lock entry size for the lock table. The default is 2 bytes, which is probably the number you want, unless you are immediately creating a data sharing group of seven or more members.

By restricting each lock entry to two bytes, you maximize the number of entries in the lock structure. This can help avoid *false contention*, described in “Avoid False Contention” on page 222.

Storage Estimate for the Lock Structure

For installation planning purposes, the following rule-of-thumb is offered as an initial specification of the lock structure total size:

1. Calculate the sum of all the virtual storage used for IRLM control blocks over all the IRLMs of the data sharing group. (See “Estimating a Value for the IRLM MAXCSA Parameter” on page 72 for more information about determining this value.)
2. Divide the result by 2.
3. Multiply the result by 1.1 to allow for space reserved for “must complete” functions.
4. Round to a power of 2.

   If you specify a power of 2, IRLM allocates half the lock structure for the lock table and the other half for the modify lock list. If you do not specify a power of 2, the allocation is not as predictable, although IRLM does try to keep the allocation as close to 1:1 as possible.

Example:

On the DB2 installation panel DSNTIPJ, MAXIMUM ECSA = 9M. If you want 10 DB2 and IRLM pairs in the group, the size estimate is:

```
((9MB x 10) / 2) x 1.1 = 49.5MB   (round to 64MB)
```

SCA Size

The shared communications area (SCA) is a list structure in the coupling facility. A table for estimating the size of the SCA is shown in Table 7.

Table 7. Estimating Storage for the SCA

Site Size	Databases	Tables	SCA Structure Size
Small	50	500	8MB
Medium	200	2000	16MB
Large	400	4000	32MB
Extra Large	600	6000	48MB

Running out of space in the structure can cause DB2 to come down. Because much of the space in the SCA is taken up with exception information, you reclaim space by correcting database exception conditions.

Changing Structure Sizes

You can change the size of structures by changing the CFRM policy and then rebuild by using the SETXCF START,REBUILD command of MVS. (This command does not work on group buffer pools that actively being duplexed.)

MVS attempts to reallocate a new instance of the structure in the same coupling facility, if that coupling facility is healthy. If there is not enough room, MVS looks at the preference list and uses the alternate coupling facility specified there. After the space is allocated, DB2 rebuilds the information into the new structure. Any transactions that need the structure must wait until the rebuild is complete. Thus, it is best to rebuild when other activity in the system is low.

Dynamically Changing the Structure Size: If all members of the data sharing group are running MVS/ESA Version 5 Release 2 or subsequent releases, and the affected structure is allocated in a coupling facility with CFLEVEL greater than 0, you can dynamically change the structure sizes up to a maximum limit specified on the CFRM policy using the MVS SETXCF START,ALTER command.

There are two advantages to this method:

- It allows DB2 to access the structures while a change is taking place.
- It requires less coupling facility storage, because it does not have to allocate enough space for a whole new structure. It dynamically adds or deletes storage from the existing structure.
- It works on duplexed group buffer pools.

A disadvantage to this method is that it cannot be used to increase the size of the lock table portion of the lock structure. You must change the CFRM policy and manually rebuild it for that change to take effect.

For more information, see:

- “Changing the Lock Structure Size Dynamically” on page 235 for the lock structure
- “Problem: Storage Shortage in the SCA” on page 179 for the SCA
- “Dynamic Method” on page 266 for the group buffer pool

Estimating a Value for the IRLM MAXCSA Parameter

The requirements for IRLM storage are described in Section 2 of *Installation Guide*. You can control the amount of storage used for locks with the MAXCSA parameter of the IRLM startup procedure.

Recommendation: Use PC=NO on the IRLM startup procedure and to set the MAXCSA value at the high end of your estimates. IRLM does not take the storage unless it needs it, and you can change the amount of MAXCSA dynamically with the MVS command MODIFY irlmproc,SET,CSA.. If you increase MAXCSA, you might need to increase the CSA value in SYS1.PARMLIB, too.

For data sharing, plan for additional storage because of additional data sharing locks called P-locks. These locks are held on open page sets and on database descriptors (DBDs), skeleton cursor tables (SKCTs), and skeleton package tables (SKPTs). Unlike transaction locks, storage for P-locks is held even when there is no transaction activity, and therefore they consume storage even with no transaction activity. See “P-Locking” on page 242 for more information about P-locks.

Plan also for extra storage that IRLM needs to build retained locks in case other members fail. The variables you need to account for are shown in Table 8.

Table 8. Variables used to Estimate Additional IRLM Storage

Variable	Description	Calculation
X	P-locks	`N = (MAX_OPEN_DATA_SETS × 500)` `X = N + (N × .40)`
Note: The formula assumes that more than one P-lock might be held on a page set occasionally (such as for castout activity) and estimates about 40 percent for P-locks on the EDM pool objects and for short-lived page P-locks. If you know that your EDM pool has relatively few objects in it, you could use a lower percentage for that value. See Section 5 (Volume 2) of *Administration Guide* for more information about estimating the maximum number of open data sets, or use the value specified for the DSMAX subsystem parameter.		
Y	Ability to hold update retained locks for a failed member	Dependent on how update-intensive the workload is. Start with the following: `Y= .25X`

For example, suppose that your non-data-sharing IRLM storage estimate is 5MB. If you estimate that this DB2 member could have as many as 8000 open data sets, you could calculate the IRLM storage as follows:

```
(8000 × 500) + 1600000 = 5.47MB
+
1 MB (approximate for retained locks)
+
5MB (non-data-sharing estimate)
-----------------------
Total IRLM storage = 11.47MB
```

Don't Forget the Proper Dispatching Priority: Be sure to follow the guidelines documented in Section 5 (Volume 2) of *Administration Guide* for setting the dispatching priority of IRLM when using workload manager. If IRLM dispatching priority is too low, storage might not be freed as quickly, and IRLM might run out of storage.

Monitoring Storage: Use the MVS command MODIFY irlmproc,STATUS,STOR to see how much storage IRLM is using. You can see information about accountable storage (that counted agains MAXCSA) and about storage that is not counted against MAXCSA. For more information about the syntax of the command, see Chapter 2 of *Command Reference*.

Increase IRLM Storage for Sysplex Query Parallelism

Sysplex query parallelism uses IRLM Notify messages for the following functions:

- The coordinator uses them to communicate to the assistants.
- The assistants use Notify messages to communicate with the parallelism coordinator.

- The main use of these messages is to pass data between the assistants and the coordinator.

To use Sysplex query parallelism, make sure that you have enough ECSA to handle these messages.

Calculating Storage for the Assistants: To estimate the amount of extra storage required by IRLM on an assisting DB2 (ANY DB2 that receives query work from another DB2), estimate the number of parallel tasks that can run concurrently on the assisting DB2, and divide that by the number of members sharing the query work. Multiply the result by 32KB to get an estimate of the amount of extra storage needed on the assistants.

```
((number of queries × max concurrent tasks) / number of members) × 32KB
```

For example, assume you have a data sharing group in which all four members participate in processing parallel queries. If you have at most 10 queries executing concurrently, and the highest number of parallel tasks is approximately 40, the calculation is:

```
(10 × 40) / 4 = 100
100 × 32KB = 3MB of extra storage on assistant
```

To estimate the number of parallel tasks, you can look at EXPLAIN output or instrumentation records for the concurrent queries.

Calculating Storage for the Coordinator: Any member that can be a coordinator needs approximately 200 KB extra storage for messages that the coordinator sends to the assistants.

Storage for DB2 Objects

In general, you can use the storage estimates in Section 2 of *Installation Guide* for your capacity planning. This section describes some additional information that is specific to data sharing.

Estimating Storage for the EDM Pool

The formula you use to calculate storage for the environmental descriptor manager (EDM) pool is described in Section 2 of *Installation Guide* . For data sharing, you might need to increase that storage estimate by about 10 percent because of the way DB2 cross-invalidates database descriptors (DBDs). This percentage is just an estimate; the actual amount of increase depends on how much you create, drop and alter objects in the data sharing group.

Cross-invalidating Items in the EDM Pool

DB2 does not have a backing EDM pool in the coupling facility for invalidating objects in the EDM pool (DBDs, cursor tables, and so on) because these objects are modified less frequently than database data. So, there is one EDM pool for each DB2 subsystem. When a DBD changes, perhaps because an object descriptor has changed, DB2 uses XCF messages to notify other DB2s using that DBD that new transactions should not use that copy of the DBD. New transactions use the new DBD (which is read into the EDM pool). Thus, it is possible that one transaction is using the new DBD while the old one is still being used by the currently running transactions. In other words, more than one copy of a DBD can exist in the EDM pool.

Storage for Reusing Threads

One of the general recommendations for data sharing is to reuse threads whenever possible and to bind with the option RELEASE(DEALLOCATE). Depending on how much your threads get reused, this bind option can mean more storage is necessary for storing objects used by the plan. Plan for more EDM pool storage if you use RELEASE(DEALLOCATE) and thread reuse.

Planning to Enable Data Sharing

Moving to data sharing is a two-step process, in most cases. First, migrate to Version 5 (or install Version 5), and then "enable" that Version 5 subsystem for data sharing. This process is described more fully in "Enabling DB2 Data Sharing" on page 98. (If you already have a data sharing group, see "Migrating an Existing Data Sharing Group to the New Release" on page 107 for information about migrating that group.)

After data sharing is enabled, disabling data sharing is a very difficult process and one which should be avoided. See "Disabling and Re-enabling Data Sharing" on page 117 for more information.

This section describes things you need to be aware of and to prepare for as you plan your move to data sharing:

- "Decide if Merging is the Right Thing to Do"
- "Connecting IMS and CICS" on page 76
- "Binding Plans and Packages if Moving to a New Machine" on page 77
- "Using Type 2 Indexes" on page 77
- "Registering Command Prefixes and Group Attachment Name" on page 77
- "Shared Read-Only Data" on page 79
- "Applications using CICSPlex SM" on page 80
- "Migrating transactions that have ordered dependencies" on page 82
- "Increasing the Size of the BSDS" on page 83
- "Increasing the Size of the SYSLGRNX Table Space" on page 83
- "Additional Considerations for Existing Subsystem Parameters" on page 84

See also "Tuning deadlock and timeout processing" on page 226 for information about tuning your timeout periods.

Decide if Merging is the Right Thing to Do

Although DB2 does not offer an automated way to "merge" catalogs, it is possible to merge existing DB2s into a data sharing group. Consider carefully a decision to merge existing DB2 subsystems.

Merging is a very complicated process. It involves not just the physical problem of moving data, it includes many other management issues, such as the following:

- Naming conventions for users, plans, packages, databases, tables, views, and so on
- Authorization techniques
- Backup and recovery conventions
- Availability practices

Before you consider merging existing DB2 subsystems into a single data sharing group, ask yourself the following question: **Why are the subsystems separate now?**

Reasons to Not Merge

If the subsystems are separated now because of a different set of user groups that do not need frequent access to each other's data, do not now consider merging them.

For the same reasons that you most likely do not include test and production DB2s in a single MVS, we do not recommend merging test and production subsystems into a single data sharing group.

If you are trying to minimize the number of subsystems because you are running short of subsystem recognition characters, then using 8-character command prefixes offers relief.

| If you have two existing subsystems, and each of those subsystems can grow into
| a separate group, availability is usually better by keeping those groups separate. Administration is simpler by keeping the groups split along the same lines as the users.

Reasons to Consider Merging

If the subsystems were split because of capacity constraints only, then merging might be a good idea, especially if the subsystems already share a common naming convention.

If the subsystems have or need a lot of common data and are using shared read only data, distributed access, or data replication to handle the problem of sharing the data, merging might be a solution. However, this might not be a good approach if the security needs of the two groups are very different. If you try to merge two subsystems with very different security needs, especially if a shared naming convention is not already in place for those separate subsystems, then merging them could be very difficult.

Connecting IMS and CICS

You must define IMS and CICS connections for each DB2 in the data sharing group. See Section 2 of *Installation Guide* for more information about connecting CICS and IMS to DB2.

Connecting CICS to DB2

If you are adding a DB2 member that is connected to CICS Version 4 Release 1, or a subsequent release, then you must use the attachment that is shipped with that CICS release when connecting that CICS region to DB2. DSNTIJSU is not run against the attachment modules shipped with CICS.

The attachment facility that is shipped with CICS 4.1 and subsequent releases has some advantages for use with data sharing:

- The old attachment facility has the subsystem name defined in the RCT, so if you want to attach from 2 DB2s to CICS, you have to assemble two RCTs with different suffixes.

- The new attachment facility allows you to override the subsystem name on startup and with the INITPARM. This eliminates the need for a second instance of the RCT if you wish to connect to a different DB2 subsystem.

Connecting IMS to DB2

For every member that is going to run IMS applications, make sure you attach IMS to that member. IMS must include a separate SSM member for every member DB2. See Section 2 of *Installation Guide* for more information about connecting DB2 to IMS.

Binding Plans and Packages if Moving to a New Machine

Running a plan or package on a data sharing group does not in itself require that you rebind that plan or package. However, if you are using a new machine that has different performance characteristics (from a 9021 711-based processor to the S/390 microprocessor cluster, for example), it is to your advantage to rebind plans and packages on the machine they will be running on. See “Access Path Selection in a Data Sharing Group” on page 268 for more information.

Using Type 2 Indexes

Data sharing has some restrictions on indexes; you can use the “old” indexes (type 1), but these indexes must be defined with SUBPAGES 1 to allow access when there is inter-DB2 read/write interest. If an index has more than one subpage, applications will receive “resource unavailable” SQLCODEs when there is inter-DB2 R/W interest in that index.

Or, you can use type 2 indexes. Type 2 indexes offer many advantages in both the data sharing and non-data-sharing environments. To take full advantage of the improvements offered in Version 4, such as improved partition independence and parallelism, you must convert indexes to type 2. See “Use Type 2 Indexes” on page 222 and Section 5 (Volume 2) of *Administration Guide* for more information about type 2 indexes.

Part of your migration strategy should include converting indexes to type 2 for use in data sharing. Installation job DSNTIJXC converts IBM-defined indexes for you.

Registering Command Prefixes and Group Attachment Name

Use the IEFSSN*xx* parmlib member to register the 1- to 8-character command prefix for a member and the group attachment name for the group, as shown here:

```
ssname,DSN3INI,'DSN3EPX,prefix<,scope<,group-attach>>'
```

When you register the command prefix in the IEFSSN*xx* parmlib member, you also specify the *scope* of the prefix. We recommend that you choose a scope of Started (S), which allows you to have a single IEFSSN*xx* parmlib member to be used by all MVS systems in the Sysplex. It also simplifies the task of moving a DB2 from one system to another; you can stop DB2 on one MVS and start it up on another. There is no need to re-IPL the system.

For more information about information in the IEFSSN*xx* parmlib member, see information about job DSNTIJMV in Section 2 of *Installation Guide*.

Sample Definitions

Here are sample definitions that might appear in the shared SYS1.PARMLIB:

```
DB1G,DSN3INI,'DSN3EPX,-DB1G,S,DB0G'
DB2G,DSN3INI,'DSN3EPX,-DB2G,S,DB0G'
DB3G,DSN3INI,'DSN3EPX,-DB3G,S,DB0G'
DB4G,DSN3INI,'DSN3EPX,-DB4G,S,DB0G'
```

With these definitions, you can start DB1G on MVS1 and that DB2 will be the only one in the Sysplex that can use -DB1G as its command prefix. However, because the DB2 is registered to MVS with a scope of S, it is possible to stop DB1G and restart it on another MVS without having to re-IPL any MVS system.

Changing the Command Prefix

To change the command prefix parameters, you must change the IEFSSN*xx* entry and re-IPL the host system. For example, if you want to change the command prefix scope from system to Sysplex-wide, and you want to register the prefix at DB2 startup, change the M in the entry to S before re-IPLing.

If you want to use multiple-character command prefixes, make sure your automation programs can handle multiple-character prefixes in messages before making that change.

Group Attachment Name

As shown in "Registering Command Prefixes and Group Attachment Name" on page 77, specify the group attachment name in the IEFSSN*xx* member. You can let the DB2 installation process do this for you, or you can update the member yourself. We recommend that you specify the group attachment name at a convenient time, such as during a planned IPL.

Even if you have not yet enabled data sharing, the group attachment name is active after you IPL the system. This is not a problem. Until you are ready to move to data sharing, we recommend that you continue specifying the DB2 subsystem name in your TSO and batch jobs. When ready to move to data sharing, you can then change those jobs to specify a group attachment name without the need for an IPL.

Another option is to keep the group attachment name the same as the originating member's subsystem ID. By doing this, you can avoid changing the name in your jobs when you move to data sharing.

How DB2 Chooses a Subsystem Name: When you submit a job on an MVS system, DB2:

1. Assumes that the name you specified on the DSN command is a subsystem name and attaches to that subsystem if it is started.
2. If *either* of the following is true:
   - The name on the DSN command is not defined as a subsystem
   - A subsystem by that name is not started and that subsystem's group attachment name is the same as the subsystem name

   then DB2 checks to see if the name is a group attachment name.

   a. If the name on the DSN command is a group attachment name, it constructs a list of DB2 subsystems that are defined to this MVS, and tries to attach to each one in order, until it finds a DB2 subsystem that is started on this MVS, or until it reaches the end of the list. DB2 always attaches to the first started one on the list—there is no load balancing.

To create the list, DB2 adds each subsystem when it goes through subsystem initialization. At IPL time, subsystems are intitialized in the order in which they appear in the IEFSSN*xx* member. If you add a subsystem with the MVS SETSSI command, that subsystem is added to the list at that time.

b. If the name on the DSN command is not a group attachment name, then a “not started” message is returned.

Be careful when you begin moving to data sharing that your IEFSSN*xx* definitions are correct. This can be especially troublesome if you have extinct subsystems that are still defined but not used. For example, assume you have the following subsystem definitions on an MVS system. Notice that DB2P does not have the group attachment name specified in its definition:

```
DB2P,DSN3INI,'DSN3EPX,-DB2P,S'  ←—Extinct subsystem
DB1G,DSN3INI,'DSN3EPX,-DB1G,S,DB2P'  ←Active subsystem
```

Because the jobs submitted on this MVS are trying to connect to the name DB2P, DB2 always tries to connect to subsystem DB2P. In most cases, DB2 doesn't use the group attachment name support, and a “not started” message is returned. To avoid this situation, include the group attachment name in DB2P's definition. If the correct group attachment name is present, DB2 does try to attach to DB1G after it discovers that DB2P is not started.

Shared Read-Only Data

DB2 data sharing is intended as a replacement for shared read-only data. Unlike read-only data sharers, all members of a data sharing group have equal and concurrent read/write access to databases. However, there might be a time when data sharing and a shared read-only data configuration have to coexist. For example, you might want an owning subsystem as the originating member of a data sharing group. If the data that is defined as shared read-only still needs to be accessed by other DB2s for a time, you need to maintain that “owner” attribute.

Because a data sharing group shares a single DB2 catalog, the group is defined as either the owner or a reader of a database created with the ROSHARE option. If defined as an owner, the first member to access the data set is the only DB2 allowed to update during that period as shown in Figure 23 on page 80.

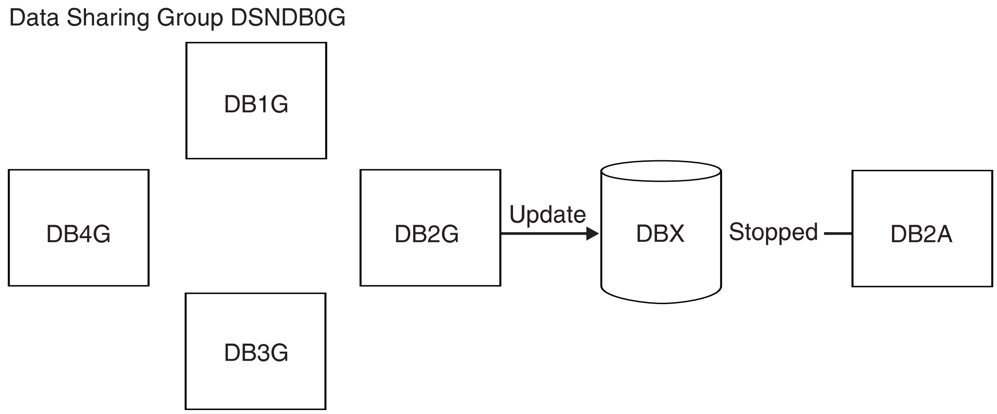

Figure 23. DBX Can Only be Updated by One Member at A Time. The data sharing group has DBX defined as ROSHARE OWNER in its catalog. DB2A has DBX defined as ROSHARE READ.

If defined as the reader, or if the shared read-only database is started with RO access, then all members of the data sharing group can read the database concurrently (as long as the owner is not currently accessing the database in R/W mode).

When you are ready to replace a shared read-only configuration with read/write data sharing, then you must alter the shared read-only databases to ROSHARE NONE to enable them to be shared. Any user-managed data sets must have the SHAREOPTIONS value changed to (3,3).

Applications using CICSPlex SM

CICSPlex System Manager/ESA (CICSPlex SM) is a system-management tool that enables you to manage several CICS systems as if they were one. The dynamic transaction routing program supplied with CICSPlex SM balances the enterprise workload dynamically across the available application owning regions (AORs). CICSPlex SM enables you to manage a variable workload without operator intervention, while maintaining consistent response times. It can do this because it routes transactions away from busy regions and from those that are failing or likely to fail, giving improved throughput and concealing problems from end users.

In some situations, your DB2 applications must be sensitive to a “resource unavailable” condition. For example, assume a database is stopped for planned maintenance, and the application receives a -904 SQLCODE and ends normally. CICSPlex SM might continue to route work to that system because it appears to complete its work rapidly. This is sometimes called the “stormdrain” effect.

Another Cause for Stormdrain: When both of following conditions are true, the stormdrain effect can occur:

- The CICS attachment facility is down
- You are using INQUIRE EXITPROGRAM to avoid AEY9 abends

Again, because there hasn't been an abend, it appears as if work is completing rapidly at that subsystem.

Writing a CICS Exit to Avoid the Stormdrain Effect

The information under this heading, up to “Increasing the Size of the BSDS” on page 83, is Product-sensitive Programming Interface and Associated Guidance Information about Customer Information Control System/Enterprise Systems Architecture (CICS/ESA) and DB2.

If you are using CICS Version 4 Release 1, you can write a resource manager interface program exit, XRMIOUT, to avoid the stormdrain effect caused by a resource unavailable SQLCODE (-904). (This exit does not avoid the stormdrain problem caused by using INQUIRE EXITPROGRAM to avoid AEY9 abends.)

Using XRMIOUT, you can intercept the return from the resource manager. The exit can check whether:

- The resource manager is DB2.
- There is a -904 SQLCODE in the SQL communication area (SQLCA).

If these conditions exist, abend the transaction, instead of ending the transaction normally.

To determine if DB2 is the resource manager, compare `'DSNCSQL'` with the value stored at the address included with the UEPTRUEN parameter passed to XRMIOUT as shown in Figure 24.

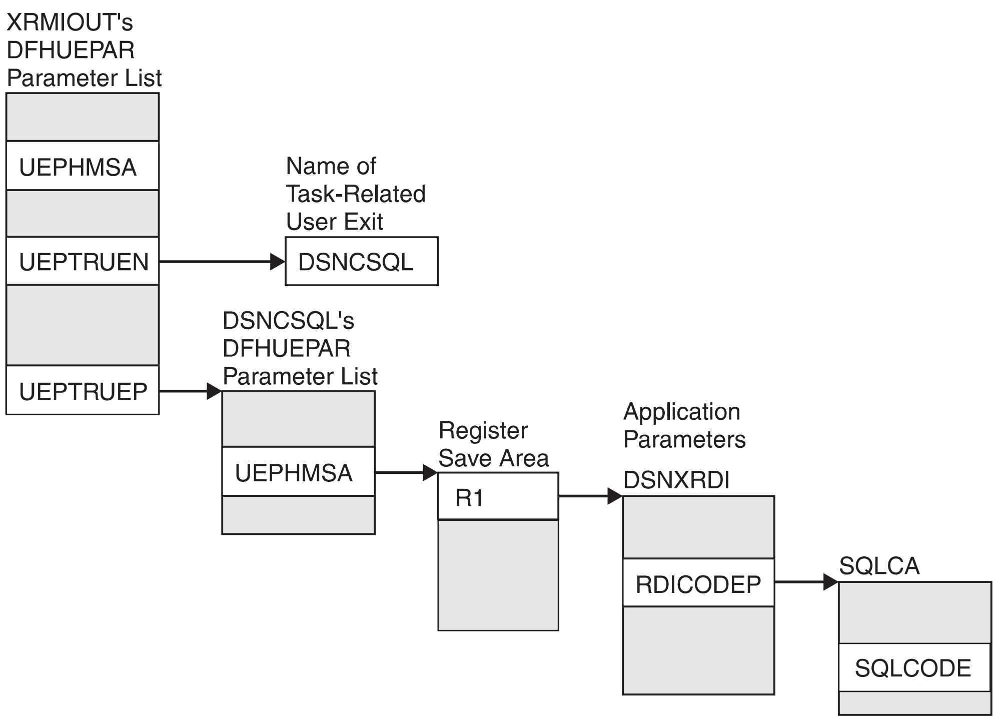

Figure 24. Pointers to Resource Manager Name and SQLCODE

To find the SQLCODE:

1. Find UEPTRUEP in the DFHUEPAR parameter list passed to XRMIOUT. UEPTRUEP contains the address of the DFHUEPAR parameter list passed to DSNCSQL.
2. Find UEPHMSA in DSNCSQL's DFHUEPAR parameter list. UEPHMSA points to the register save area containing the application's registers.

3. Find register 1 in the register save area. Register 1 contains the address of the application parameters.

   The application parameters passed by the precompiler to DB2 are mapped by the DSNXRDI mapping macro. The mapping macro is contained in the data set library *prefix*.SDSNMACS.

4. Find RDICODEP in the DSNXRDI structure. RDICODEP contains the address of the SQL communication area (SQLCA).

5. Find the SQLCODE. The SQLCODE offset is documented in Appendix C of *SQL Reference*.

For more information about the XRMIOUT exit, see *CICS/ESA Customization Guide*.

CICS Enhancement to Avoid the Stormdrain Effect

The CICS Transaction Server for OS/390 Release 1, and subsequent releases, lets you avoid the stormdrain effect. With that release, you will not need to use XRMIOUT to check for resource unavailable conditions.

That follow-on release also gives you new options on the RCT TYPE=INIT macro that allow you to get the benefits of the INQUIRE EXITPROGRAM without the disadvantage of causing the stormdrain effect. Those options are STRTWT=AUTO and STANDBY=SQLCODE. For more information about these options, see Section 2 of *Installation Guide*.

Migrating transactions that have ordered dependencies

This section pertains only for those limited cases in which one transaction, called an *originating* transaction, updates DB2 data using INSERT, UPDATE, or DELETE, and then, *before committing*, spawns a second transaction that is dependent on the updates that were made by the first transaction. This type of relationship is referred to as “ordered dependencies” between transactions.

Statement of the problem

In some situations, the dependent transaction can encounter a “row not found” condition that did not occur in a non-data-sharing environment. The reason that the dependent transaction might start to encounter periodic “row not found” conditions after enabling data sharing, is that in a data sharing environment the dependent transaction might run on a different DB2 member than the DB2 member on which the originating transaction is running, and because of the multi-system buffering effects that are present in data sharing (each DB2 member has its own local buffer pools), the buffered pages that are updated by the originating transaction are not immediately “visible” to the dependent transaction when it runs on a different DB2 member.

When the problem might occur

If all of the following conditions are true, then it is possible that the dependent transaction might periodically encounter a “row not found” condition when attempting to access a row that was updated by the originating transaction:

- The originating transaction spawns the dependent transaction before its phase 2 of commit completes, and

- The dependent transaction runs on a different member than the originating transaction, and

- The dependent transaction is not bound with an isolation level of repeatable read.

Preventing the problem

To prevent these periodic “row not found” situations, use the IMMEDWRITE(YES) option of BIND/REBIND for a plan or package that spawns dependent transactions that might run on other members. DB2 immediately writes to the coupling facility any updated buffers that contain pages that belong to page sets or partitions that are dependent on group buffer pools. An "immediate write" means that the page is written to the group buffer as soon as the buffer update completes. DB2 writes the data to one of the following structures or devices:

- The group buffer pool
- DASD for GBPCACHE NO group buffer pools or GBPCACHE NONE
- SYSTEM page sets

Immediate updates might have some performance impact.

There are several alternative ways to ensure that the dependent transaction runs on the same DB2 member as the originating transaction. If it is necessary to schedule the dependent transaction on a different member, then consider any of the following approaches to the problem:

- Ensure that the originating transaction does not schedule the dependent transaction until the originating transaction has completed phase 2 of commit.
- Run the dependent transaction with an isolation level of repeatable read.
- If the dependent transaction is currently running with an isolation level of cursor stability CURRENTDATA(NO), changing to use CURRENTDATA(YES) can sometimes solve the problem.

Increasing the Size of the BSDS

Data sharing causes additional records to be written to the BSDS for member information. To avoid having the BSDS go into secondary extents, we recommend that you change the record size of the primary space allocation from 70 records to 120 records. This is necessary only for subsystems that have followed a migration path from Version 3 to Version 4 to Version 5 without altering that size. New installations and member installations already do this for you.

To increase the space allocation for the BSDS, you must:

1. Rename existing BSDSs.
2. Define larger BSDSs with the original names.
3. Copy the renamed ones into the new ones.

All this can be done using access method services. To see the definition used for the BSDSs, see the installation job DSNTIJIN.

Increasing the Size of the SYSLGRNX Table Space

The SYSLGRNX directory table space consists of the RBA ranges of when data sets are open for updating. Because there are more members opening and closing data sets, this table space is likely to grow with the addition of each new member to the data sharing group. Consider increasing the frequency with which you remove rows from this table space, or increasing the size. To see the definition used for SYSLGRNX, see installation job DSNTIJIN.

To increase the space allocation for SYSLGRNX, you must use access method services:

1. Rename the existing SYSLGRNX data set.
2. Define a larger SYSLGRNX data set with the original name.
3. Copy the contents of the renamed data set into the new SYSLGRNX data set.

Additional Considerations for Existing Subsystem Parameters

If you are moving your DB2 processing to S/390 microprocessors, some system parameters might need to be changed to accommodate the reduced workload on each individual member:

- System checkpoint

  Fewer log records per DB2 are written when DB2 processing on a large, single system is replaced by many smaller S/390 microprocessor clusters. Consider reducing the number of log records that are written between checkpoints to maintain the same checkpoint frequency. You can do this by lowering the value on the CHECKPOINT FREQ field of installation panel DSNTIPN.

  Frequent checkpointing reduces DB2 restart time and thus reduces the amount of time data might be locked out from other DB2 members. It also reduces the amount of time it takes to recover a group buffer pool if it should fail.

  Be sure to consider also the impact of frequent checkpoints on how often DB2 changes data set status from R/W to R/O state. See "How DB2 Tracks Interest" on page 239 for more information.

- EDM pool size

  See "Estimating Storage for the EDM Pool" on page 74 for information about possibly increasing this value.

Chapter 4. Installing and Enabling DB2 Data Sharing

The purpose of this chapter is to give you an overview of the process of making a data sharing group. Table 9 points you to the procedures you need to create or migrate to a data sharing group. The complete set of installation panels and steps are shown in *Installation Guide* .

Be sure to read “Choosing Subsystem Parameters” on page 88 for guidance on choosing specific subsystem parameters.

Table 9. Data Sharing Options

If you have this...	And you want this...	Read this...
No system.	Version 5, data sharing	“Installing a New DB2 Data Sharing Group” on page 95. (Use this procedure only in low-risk situations. It is best to migrate to Version 5 and then enable data sharing.)
A Version 5 non-sharing subsystem	The originating member of a data sharing group	“Enabling DB2 Data Sharing” on page 98.
One member in the group	More members in the group.	“Adding a New DB2 Data Sharing Member” on page 100.
Separate DB2 subsystems	Merged DB2 subsystems into a single group.	“Merging Existing DB2 Data into the Group” on page 101.
A Version 4 Release 1 data sharing group	A Version 5 data sharing group.	“Migrating an Existing Data Sharing Group to the New Release” on page 107.

The following tasks are also described here:

- “Renaming the DB2 Subsystem” on page 96
- “Merging Existing DB2 Data into the Group” on page 101
- “Testing the Data Sharing Group” on page 105
- “Updating Subsystem Parameters for a Member” on page 107

If you already have a Version 4 data sharing group, read this chapter for new information, and see “Migrating an Existing Data Sharing Group to the New Release” on page 107.

For information about falling back, see “Falling Back and Remigrating” on page 116.

For information about disabling data sharing, which is not a recommended course of action, see “Disabling and Re-enabling Data Sharing” on page 117.

And, in the event that you want to remove a data sharing member, either permanently or temporarily, see “Removing Members from the Data Sharing Group” on page 121.

Choosing Subsystem Parameters

Every member of a DB2 data sharing group must have its own unique load module for subsystem parameters (sometimes called DSNZPARM in a non-data-sharing environment).

The load module for subsystem parameters is built by job DSNTIJUZ and stored in the *prefix*.SDSNEXIT target library. Every member must use a different name for its subsystem parameters load module because the *prefix*.SDSNEXIT target library can be shared among all members of the data sharing group. The installation process requires that you provide the name of the load module for a member.

We recommend that you name each member's load module using the convention DSNZP*xxx*, where *xxx* includes the number in the member name and the group identifier. For example, DB1G's subsystem parameters load module could be named DSNZP01G.

The subsystem parameters load module name for a member is an optional parameter on the EXEC statement in the JCL procedure used to start the *ssnm*MSTR address space. This is provided so that an operator (or automated operations) does not have to specify the subsystem parameter load module name when starting a DB2 member. The format for specifying the parameter on the EXEC statement of the *ssnm*MSTR JCL procedure is:

```
//IEFPROC EXEC PGM=DSNYASCP,PARM='ZPARM(DSNZPxxx)',...
```

The Scope and Uniqueness of DB2 Subsystem Parameters

Even though the various subsystem parameters affect the operation of only a single DB2, some parameters must be the same on all the sharing subsystems. An example is the catalog alias name.

Other subsystem parameters must be unique for each member. An example of this is the name of the IRLM used by each DB2 subsystem.

Most subsystem parameters do not have to be unique. We offer recommendations for some of these parameters. In the tables in the following section, the parameter is indicated by the installation panel field name. However, some of the parameters do not reside in the DSNZP*xxx* load module, and cannot be changed through the installation or migration process.

Subsystem Parameters that Must be Different on Each DB2

These parameters must be different on every data sharing DB2 member in a group. These parameters must be specified when a member is installed.

Table 10 (Page 1 of 2). Subsystem Parameters that Must be Different on Each DB2

Subsystem Parameter Field Name	Panel ID	Comment
SUBSYSTEM NAME	DSNTIPM	The DB2 subsystem identifier.
COMMAND PREFIX	DSNTIPM	The command prefix used to route commands to this subsystem.
MEMBER NAME	DSNTIPK	The member name for this DB2.

Table 10 (Page 2 of 2). Subsystem Parameters that Must be Different on Each DB2

Subsystem Parameter Field Name	**Panel ID**	**Comment**
DB2 NETWORK LUNAME	DSNTIPR	Even if you do not use distributed database, this identifier is required to ensure that logical unit of work IDs (LUWIDs) are unique across the data sharing group.
RESYNC PORT	DSNTIP5	When using TCP/IP network protocols, this is the port used for resynchronization of two-phase commit processes.
WORK FILE DB	DSNTIPK	The name of the work file database for this DB2.
Active Logs: COPY 1 PREFIX	DSNTIPH	Each subsystem writes to its own recovery log.
Active Logs: COPY 2 PREFIX	DSNTIPH	Each subsystem writes to its own recovery log.
Archive Logs: COPY 1 PREFIX	DSNTIPH	Each subsystem writes to its own recovery log.
Archive Logs: COPY 2 PREFIX	DSNTIPH	Each subsystem writes to its own recovery log.
Bootstrap Data Sets (BSDS): COPY 1 NAME	DSNTIPH	Each DB2 has its own BSDS.
Bootstrap Data Sets (BSDS): COPY 2 NAME	DSNTIPH	Each DB2 has its own BSDS.
PROCNAME	DSNTIPI	This is the name of the IRLM procedure that MVS invokes if IRLMAUT=YES.
SUBSYSTEM NAME (IRLM)	DSNTIPI	Specifies the name of the IRLM subsystem associated with a particular DB2. This name must be unique within the Sysplex.
MEMBER IDENTIFIER	DSNTIPJ	The unique identifier for this IRLM.
PARAMETER MODULE	DSNTIPO	The name of the subsystem parameter load module for this DB2.

Subsystem Parameters that Must be the Same on Every DB2

The following subsystem parameters must be the same for every member of the data sharing group.

Table 11 (Page 1 of 2). Subsystem Parameters that Must be the Same

Subsystem Parameter Field Name	Panel ID	Comment
CATALOG ALIAS	DSNTIPA2	Specifies the DB2 catalog alias name.
DB2 LOCATION NAME	DSNTIPR	Specifies the location name for the entire DB2 data sharing group. This name is required.
EXTENDED SECURITY	DSNTIPR	Determines the content of error codes returned to a client when a connection request fails because of security errors. Also allows users to change their passwords if their host passwords expire.
DRDA PORT	DSNTIP5	Specifies the DRDA port number for the entire DB2 data sharing group. This name is required if using TCP/IP network connections.
TCP/IP ALREADY VERIFIED	DSNTIP5	Specifies whether incoming TCP/IP requests are accepted by DB2 without a password or RACF PassTicket. This option must be the same on all members or requesters will have inconsistent results.
SITE TYPE	DSNTIPO	All members of the group must have the same value. All members on the remote site, after they need to become the local site, must have the same value (LOCALSITE).
GROUP ATTACH	DSNTIPK	This is the group attachment name that allows TSO and batch programs, programs using CAF and RRSAF, and utilities to generically attach to any DB2 member of the group. The default is subsystem name of the first installed member of the group.
GROUP NAME	DSNTIPK	The name of the DB2 group.
INSTALL DD CONTROL SUPT. (and related parameters)	DSNTIPZ	All members of the data sharing group must use the same set of data definition control registration tables, or unpredictable results can occur.
MINIMUM DIVIDE SCALE	DSNTIPF	Specifies whether to retain at least three digits to the right of the decimal point after any decimal division.

Table 11 (Page 2 of 2). Subsystem Parameters that Must be the Same

Subsystem Parameter Field Name	Panel ID	Comment
SYSTEM ADMIN 1 SYSTEM ADMIN 2 SYSTEM OPERATOR 1 SYSTEM OPERATOR 2	DSNTIPP	Specifies installation SYSADM and SYSOPR authorities.

Other Recommendations

These parameters can be the same or different on separate members of the group.

Table 12 (Page 1 of 2). Recommended System Parameters

Subsystem Parameter Field Name	Panel ID	Comment
DEALLOC PERIOD	DSNTIPA	This is the length of time during which an archive read tape unit is allowed to remain unused before it is deallocated. We don't recommend archiving to tape. If you *must*, however, we recommend that you specify 0 for this parameter unless you intend to run all RECOVER jobs from the same DB2 member. Specifying a deallocation delay means that the tape is not available to any other DB2 members until the deallocation time expires.
DEVICE TYPE 1 DEVICE TYPE 2	DSNTIPA	This is the device type or unit name for storing archive logs. The recommendation is to archive one copy of the log to DASD.
READ TAPE UNITS	DSNTIPA	This is the maximum number of dedicated tape units that can be allocated to read archive log tape volumes concurrently. We don't recommend archiving to tape. If you *must*, however, it is vital that you have enough tape units allocated to the DB2 doing the recovery to merge the archive logs from *all* members in the group that have updated the object being recovered. Thus, if there are 8 members in the group, make sure you specify at least 8 on this panel for each member. See “The Impact of Archiving Logs in a Data Sharing Group” on page 159 for more information about archiving to tape.
EDMPOOL STORAGE SIZE	DSNTIPC	This is the size of the environmental descriptor manager (EDM) pool in kilobytes. Whatever value is calculated from the installation CLIST, consider adding more storage because of the way data sharing updates database descriptors in the EDM pool.
RECORDING MAX	DSNTIPA	Specifies the maximum number of archive logs to be recorded in the BSDS. We recommend that all members in a DB2 data sharing group use the maximum value of 1000. This makes it easier to transfer a workload from one member to another.

Table 12 (Page 2 of 2). Recommended System Parameters

Subsystem Parameter Field Name	Panel ID	Comment
RETLWAIT	DSN6SPRM macro	This parameter allows you to specify whether a transaction should wait for a lock on a resource if another DB2 has failed and is holding an incompatible lock on that resource. Locks held by failed DB2 members are called *retained* locks. NO, the default, means that a request for a resource which has an incompatible retained lock is immediately rejected, and a "resource unavailable" condition is returned to the requesting application. YES means that DB2 suspends the application for the normal timeout period to wait for the lock to become available. If you have automatic restart, or some other restart automation that quickly restarts failed DB2s, we recommend that you choose YES.

DSNHDECP Parameters

| There is a single DSNHDECP load module for each release of DB2 in a data sharing group. The application programming defaults contained within the DSNHDECP load module are considered global for the group. The load module is created during the installation of the DB2 group. It cannot be modified during the installation of a DB2 member.

The DSNHDECP parameter DECPSSID has a special meaning in a data sharing group. DECPSSID contains the group attachment subsystem name for the data | sharing group This allows utilities, TSO attachment, RRSAF, and CAF applications to attach to any DB2 subsystem in the group.

Creating the DB2 Data Sharing Group

Before enabling data sharing, read the following topics:

- "Recommended Approach for Moving to Data Sharing"
- "Sharing DB2 Libraries" on page 94
- "Ensuring that Installation Jobs Access the Correct JCL Procedures" on page 94
- "Establishing System Affinity for Installation Jobs" on page 95

Recommended Approach for Moving to Data Sharing

Moving to data sharing is a big step. Plan this move carefully, because after you enable data sharing, it is very difficult to disable it. Disabling should only be considered if your long-term plans are to disable data sharing.

| Before enabling data sharing, test other major new functions in the release on a | single system, and then build and try a test data sharing group. When you are ready to begin a move to production, you must avoid having to fall back to the previous release.

Build and Try a Test Data Sharing Group

Here is one approach to testing your data sharing group:

1. Install Version 5 as a single system, and test it with dummy data, or copies of production data, in order to test some of the new functions. Run old applications and begin new application development work.

   In the meantime, prepare the hardware and define the coupling facility structures to enable data sharing.

2. Convert indexes to type 2, if they are not converted already.

   For data sharing, all indexes must be defined as type 2, or with SUBPAGES 1. (Directory indexes must be converted to type 2 because you cannot change the subpages value for those.)

   To take advantage of many important functions, we recommend converting indexes to type 2. You must convert any indexes that are to be used in data sharing, including those for:

   - DB2 catalog and directory
   - communications database
   - data definition control
   - resource limit facility (governor)
   - user data

General-use Programming Interface

To determine which type 1 indexes have a SUBPAGE value greater than 1, use the following query on the DB2 catalog:

```
SELECT NAME, TBNAME, INDEXTYPE, PGSIZE
       FROM SYSIBM.SYSINDEXES
       WHERE (INDEXTYPE = ' ') AND (PGSIZE <> 4096);
```

End of General-use Programming Interface

   These indexes must be converted to type 2, or converted to SUBPAGES 1.

   Installation job DSNTIJXC can be used to convert IBM-defined indexes to type 2. Use ALTER INDEX for user-defined indexes. See Section 2 of *Installation Guide* and Chapter 6 of *SQL Reference*for more information about converting to type 2 indexes.

3. Enable data sharing on this test system.

   You will have a single-system data sharing group at this point, and this can help you find any initial configuration problems. Make sure old applications work in this environment.

4. Install additional members in the test group.

   Run applications from different subsystems in the group to fully exercise the group buffer pools and cross-system locking.

Move to Production

When you are ready to move to production:

1. Migrate your existing Version 4 subsystem to Version 5, but don't use any of the new functions yet.
2. Start using new functions when you are sure the release is stable and you won't need to fall back.

   Make sure you convert all indexes to type 2 (or, alternatively, alter them to SUBPAGES 1). See step 2 on page 93 for a checklist.
3. Tune applications to contain the level of locking and lock contention rates.

   To reduce the effects of locking contention in a data sharing environment, it is best to first control locking costs in a non-data-sharing environment. This will give you a baseline from which to do further tuning after the move to data sharing. See Section 5 (Volume 2) of *Administration Guide* for information about reducing locking contention. See "Improving Concurrency" on page 217 for information about reducing contention in a data sharing environment.
4. Enable data sharing on this originating subsystem, and run applications on this one-member data sharing group.
5. Install additional members as needed.

Sharing DB2 Libraries

DB2 target and distribution library data sets can be shared among all members of a DB2 data sharing group. There is no need for each member of the group to have its own set of target and distribution libraries. Sharing these libraries reduces the effort to install and maintain the different members of the data sharing group. As described in "Administering a Database" on page 38, sharing libraries can also help ensure that members are using the same exit routines.

Sharing libraries also simplifies the tasks of defining a new DB2 member to a data sharing group. The DB2 member installation process supports sharing of the libraries.

Ensuring that Installation Jobs Access the Correct JCL Procedures

If you have more than one procedure library, ensure that your installation jobs access the right set of procedures by using a JCLLIB statement to specify the order in which procedure libraries are searched.

The JCLLIB statement looks like this:

```
//ddname JCLLIB ORDER=(library[,library...])
```

The JCLLIB statement must follow the JOB statement and precede the first EXEC statement in the job. You can have DB2 insert this statement in your JCL for you by entering it on installation panel DSNTIPY.

For more information on the JCLLIB statement, see *MVS/ESA JCL Reference*.

Establishing System Affinity for Installation Jobs

You must ensure that the installation jobs are run on the MVS system on which the appropriate DB2 subsystem is running. There are several MVS installation-specific ways to make sure this happens. These include:

- For JES2 multi-access spool (MAS) systems, use the following JCL statement:

  ```
  /*JOBPARM SYSAFF=cccc
  ```

  Where *cccc* is the JES2 name. You can specify an asterisk (SYSAFF=*) to indicate that the job should run on the system from which it was submitted.

- For JES3 systems, use the following JCL statement:

  ```
  //*MAIN SYSTEM=(main-name)
  ```

  Where *main-name* is the JES3 name.

MVS/ESA JCL Reference describes the JCL statements shown above. You can edit the jobs manually, or you can enter the above statements on installation panel DSNTIPY and have DB2 insert these statements for you.

Your installation might have other mechanisms for controlling where batch jobs run, such as by using job classes.

To create a data sharing group, you add one member at a time. The first member (the *originating* member) can either be created as a new installation or enabled for data sharing from an existing Version 5 subsystem. **We strongly recommend that you use Version 5 before enabling data sharing.** This allows you to test Version 5, convert indexes to type 2, and practice with new Version 5 functions without the additional complexity of defining and tuning coupling facility structures. In addition, this approach helps you avoid falling back after data sharing is enabled. See “Migrating an Existing Data Sharing Group to the New Release” on page 107 for information to help you plan a move to a data sharing environment.

Installing a New DB2 Data Sharing Group

Use this procedure only in low-risk situations. The recommended approach is to migrate to or install a Version 5 subsystem, use it for a while, and then *enable* data sharing as described in “Enabling DB2 Data Sharing” on page 98.

However, if you decide to install and immediately enable data sharing on a new Version 5 subsystem, this new Version 5 subsystem becomes the originating member of the data sharing group. This member's DB2 catalog is used as the DB2 catalog for the data sharing group.

To install the new data sharing group:

1. On panel DSNTIPA1, specify:

   ```
   INSTALL TYPE ===> INSTALL
   DATA SHARING ===> YES
   ```

2. On panel DSNTIPP1, specify 1 to indicate the Group data sharing function.

3. On panel DSNTIPK, specify:

```
GROUP NAME          ===> group name
MEMBER NAME         ===> originating member name
```

   Verify that the *originating member name* is unique within your MVS Sysplex. Installation job DSNTIJMV edits the *ssnm*MSTR startup procedure with the group name and member name you specify here.

4. Complete the installation panels, specifying parameters according to the guidelines in "The Scope and Uniqueness of DB2 Subsystem Parameters" on page 88.
5. Complete all installation steps, as described in Section 2 of *Installation Guide*.
6. Run the installation verification procedures (IVP), as described in Section 2 of *Installation Guide*.

Renaming the DB2 Subsystem

If you want to rename your DB2 subsystem, it is best to do it before enabling data sharing. However, another convenient time to do this is during the process of enabling data sharing. Because DB2 must be shut down during the enabling process, you can perform the tasks necessary to rename the subsystem at the same time. This section is divided into two parts:

- "Tasks that Require an IPL"
- "Tasks at Enable Time" on page 97

Renaming a subsystem is an activity that you should plan for very carefully. Because every installation has a different configuration, we cannot guarantee that this procedure is complete. If you are interested in changing the high level qualifier for data sets, see the procedure in Section 2 (Volume 1) of *Administration Guide*.

In this example procedure, we change the subsystem name for DB2P to DB1G to conform to the naming convention for data sharing that we use in this publication.

Tasks that Require an IPL

If you choose to do any of the following tasks, you must IPL MVS to pick up the changes:

- Modify the IEFSSN*xx* member for this subsystem to include the group attachment name. To avoid having to modify JCL for all your jobs, use the existing subsystem name as the group attachment name. This step is necessary to ensure that there are no problems when the call attachment and TSO attachment facilities try to use the group attachment name.

  For example, if the existing IEFSSN*xx* member looks like this:

```
DB2P,DSN3INI,'DSN3EPX,?'
```

  Change it to look like this:

```
DB2P,DSN3INI,'DSN3EPX,?,S,DB2P'
```

- Add new IEFSSN*xx* definitions for the new names:

```
DB1G,DSN3INI,'DSN3EPX,-DB1G,S,DB2P'
DJ1G
```

Optionally, you can use the MVS command SETSSI to add the **new** IEFSSN*xx* statements without an IPL. You can do this now or later during the enabling process. Don't forget to add these names to the IEFSSN*xx* member before you IPL again.

Attention: The command prefix -DB1G is not a subset or superset of DB2P's command prefix. For example, ?DB1G is invalid in this context.

- Make sure that RACF definitions are in place to handle the new subsystem names. You will have to make changes to the following RACF definitions:
  - Add the correct names to the RACF router table.
  - Add the correct names to the started procedures table (ICHRIN03), if used.

Tasks at Enable Time

You can do the following tasks when you bring down the DB2 subsystem for the enabling procedure described in “Enabling DB2 Data Sharing” on page 98.

1. Define the correct profile names for the DSNR class.
2. Replicate existing PERMIT commands to allow users and groups to access the new profiles.
3. Complete the installation panels for enabling data sharing. (You cannot change the subsystem name on the installation panels.)

   On installation panel DSNTIPH, be sure to modify the archive prefix names to include the member name, like DB2PCAT.DB1G.ARCLG1. Specify the old subsystem name (DB2P) as the group attachment name.
4. Stop DB2 with MODE(QUIESCE).
5. Run job DSNTIJUZ to assemble and link-edit the new subsystem parameter data set (DSNZP*xxx*) and DSNHDECP.
6. Rename the BSDS and active log data sets with the new prefix, like DB2PCAT.DB1G.BSDS01.
7. Update the BSDS with the renamed log data sets. **Be sure to include the same RBA ranges as the original active log data sets.**
   a. Run the utility print log map (DSNJU004) to obtain the start and end RBAs.
   b. Use access method services to rename the log data set.
   c. Run the utility change log inventory (DSNJU003) to delete the active logs with the old names.
   d. Run the utility change log inventory (DSNJU003) to add the renamed active logs with the correct ranges. There is no need to add the archive log data sets, because they are eventually replaced by the archive log data sets you specified on panel DSNTIPH.
8. If necessary, increase the size of the BSDS, as described in “Increasing the Size of the BSDS” on page 83.
9. If necessary, increase the size of the SYSLGRNX table space, as described in “Increasing the Size of the SYSLGRNX Table Space” on page 83.
10. Rename the startup procedures. For example, change DB2PMSTR to DB1GMSTR. Don't forget to change the BSDS names to the new names in the *ssnm*MSTR startup procedure.

11. Rename the IRLM startup procedure, and take this opportunity to increase the value for MAXCSA. See “Estimating a Value for the IRLM MAXCSA Parameter” on page 72 for more information.
12. Make sure CICS can connect to the new subsystem. There are several ways of doing this, depending on which level of CICS you are running. Here are a couple of techniques:
    - Change the CICS RCT TYPE=INIT macro to use the new DB2 subsystem name, reassemble, and link-edit.
    - Modify the JCL for CICS to include the new subsystem name on the INITPARM (for CICS Version 4 and subsequent releases).
13. Make sure IMS can connect to the new subsystem.
14. Enter the command START DB2 and continue with the rest of the enabling process.

Enabling DB2 Data Sharing

After you migrate to Version 5, test your Version 5 subsystem, and convert all indexes to be used for data sharing to type 2 indexes, as described in step 2 on page 93. After you complete this work, you can enable this subsystem for data sharing. The enabling process simply allows this existing subsystem to be the originating member of a data sharing group; it does not allow you to change the subsystem name. (See “Renaming the DB2 Subsystem” on page 96 for information about how to rename the subsystem.) This originating member's catalog is used as the DB2 catalog for the data sharing group.

Jobs that Enable CLIST Tailors:

DSNTEJ0	DSNTEJ2D	DSNTEJ5P	DSNTIJDE
DSNTEJ1	DSNTEJ2F	DSNTEJ6	DSNTIJIN
DSNTEJ1P	DSNTEJ3C	DSNTEJ6P	DSNTIJFT
DSNTEJ1S	DSNTEJ3P	DSNTEJ6S	DSNTIJGF
DSNTEJ1T	DSNTEJ4C	DSNTESA	DSNTIJMV
DSNTEJ2A	DSNTEJ4P	DSNTESC	DSNTIJTM
DSNTEJ2P	DSNTEJ5A	DSNTESD	DSNTIJVC
DSNTEJ2C	DSNTEJ5C	DSNTESE	DSNTIJUZ

The enabling CLIST also edits the DB2I CLISTS.

Procedure: To enable data sharing:

1. On panel DSNTIPA1, specify:

   ```
   INSTALL TYPE ===> INSTALL
   DATA SHARING ===> YES
   ```

2. On panel DSNTIPP1, specify 3 to enable data sharing.
3. On panel DSNTIPK, specify:

   ```
   GROUP NAME           ===> group name
   MEMBER NAME          ===> originating member name
   ```

   Verify that the *originating member name* is unique within your MVS Sysplex. Installation job DSNTIJMV edits the *ssnm*MSTR startup procedure with the group name and member name you specify here.

4. Complete the installation panels, specifying parameters according to the guidelines in “The Scope and Uniqueness of DB2 Subsystem Parameters” on page 88.
5. Complete the following installation steps, as described in Section 2 of *Installation Guide*.
   a. Stop DB2 activity.

      See Migration Step 10 in Section 2 of *Installation Guide* for a detailed list of steps.
   b. Installation Step 1: Define DB2 to MVS: DSNTIJMV
   c. Installation Step 3: Define System Data Sets: DSNTIJIN

      For the enable process, DSNTIJIN only alters the current active log data sets to use SHAREOPTIONS (2 3).
   d. Installation Step 5: Define DB2 Initialization Parameters: DSNTIJUZ
   e. Installation Step 7: Record DB2 Data to SMF (Optional)
   f. Installation Step 9: Connect DB2 to TSO: DSNTIJVC

      This step is necessary only if you are using the group attachment name.
   g. Installation Step 12: IPL MVS

      This step is only necessary if you are changing the command prefix, or adding or changing the group attachment name.
   h. Installation Step 13: Start the DB2 Subsystem
   i. Installation Step 14: Define Temporary Work Files: DSNTIJTM
   j. Installation Step 17: Image Copy the DB2 Directory and Catalog: DSNTIJIC (Optional)

      This is an optional step. If you decide to do this, use the DSNTIJIC job that is generated during the installation or migration of the originating member.

      You do not need to make an image copy for the DB2 catalog or user data sets for recovery, because DB2 uses image copies made before you enabled data sharing.
   k. Installation Step 18: Verify the Installation
6. Optionally, run the installation verification procedures (IVP), or a subset of these, as described in Section 2 of *Installation Guide*.

   | If you ran the complete set of IVP sample jobs after you migrated to Version 5 , you probably don't need to run these jobs again. When you start the originating member, DB2 checks your coupling facility, group, and member definitions and verifies that data sharing is enabled. You can also verify that the group has been correctly established by issuing the DB2 DISPLAY GROUP command after the originating member has completed startup.

Adding a New DB2 Data Sharing Member

You always add more members to the group as **new installations**. You cannot take an independently existing DB2 subsystem and merge it into the group. The new members begin using the DB2 catalog of the originating member.

DB2 does not have an automatic way to “merge” catalogs and resolve naming conflicts. If you have applications that are currently running on several existing DB2s, your migration plan might include procedures for moving the relevant data and applications from those DB2s onto one or more of the group members and for resolving any catalog naming conflicts that result. See “Merging Existing DB2 Data into the Group” on page 101 for more information about this.

After you have installed a new data sharing group or enabled an existing subsystem for data sharing, you can add new data sharing members.

Jobs that Add Member CLIST Tailors:

DSNTIJIN	DSNTIJMV	DSNTIJID	DSNTIJDE
DSNTIJTM	DSNTIJUZ	DSNTIJFT	DSNTIJGF

Procedure: To add a new data sharing member:

1. On panel DSNTIPA1, specify:

   ```
   INSTALL TYPE ===> INSTALL
   DATA SHARING ===> YES
   .
   .
   INPUT MEMBER NAME  ===> originating member's output PDS member
   OUTPUT MEMBER NAME ===> new member's output PDS member
   ```

2. On panel DSNTIPP1, specify 2 for adding a member.
3. On panel DSNTIPK, specify a new member name:

   ```
   MEMBER NAME           ===> new member name
   ```

   Verify that the *new member name* is unique within your MVS Sysplex. Installation job DSNTIJMV edits the *ssnm*MSTR startup procedure with the member name you specify here.

4. Complete the installation panels, specifying parameters according to the guidelines in “The Scope and Uniqueness of DB2 Subsystem Parameters” on page 88.

   We recommend that you **rename the tailored SDSNSAMP** data set for each member. This data set contains tailored JCL for each member, including jobs used for disabling and re-enabling data sharing, if that should ever become necessary. If you do not rename it, it is overwritten when you install a new member. Do this by choosing a new name for the *prefix*.NEW.SDSNSAMP data set on installation panel DSNTIPT. For example, use *prefix.member_name*.SDSNSAMP.

5. Complete the following installation steps, as described in Section 2 of *Installation Guide*.

   a. Installation Step 1: Define DB2 to MVS: DSNTIJMV

   b. Installation Step 3: Define System Data Sets: DSNTIJIN

When you add a new member, DSNTIJIN does not define catalog and directory data sets. It does define the new BSDS and active log data sets to use SHAREOPTIONS (2 3).

c. Installation Step 4: Initialize System Data Sets: DSNTIJID

When you add a new member, DSNTIJID does not initialize the catalog and directory data sets. It does add the new active log data sets to the BSDS.

d. Installation Step 5: Define DB2 Initialization Parameters: DSNTIJUZ

e. Installation Step 7: Record DB2 Data to SMF (Optional)

f. Installation Step 8: Establish Subsystem Security (Optional)

g. Installation Step 10: Connect IMS to DB2 (Optional)

h. Installation Step 11: Connect CICS to DB2 (Optional)

i. Installation Step 12: IPL MVS

If you are using MVS Version 5 Release 2 or later releases, you can use the MVS command SETSSI to dynamically add the new DB2 and IRLM subsystems without having to IPL MVS. For example, the following two commands can be used to add subsystems for DB2 and IRLM to MVS3:

```
RO MVS3,SETSSI ADD,SUB=DB3G,INITRTN=DSN3INI,INITPARM='DSN3EPX,-DB3G,S,DB0G'
RO MVS3,SETSSI ADD,SUB=DJ3G
```

Do take the time to add DB2 to the IEFSSN*xx* member so that it can be used on a subsequent IPL. For more information, see *MVS/ESA Using the Subsystem Interface* for more information.

j. Installation Step 13: Start the DB2 Subsystem

k. Installation Step 14: Define Temporary Work Files: DSNTIJTM

l. Installation Step 17: Image Copy the DB2 Directory and Catalog: DSNTIJIC (Optional)

This is an optional step. If you decide to do this, use the DSNTIJIC job generated during the installation or migration of the originating member.

m. Installation Step 18: Verify the Installation

6. Test the data sharing group, as described in “Testing the Data Sharing Group” on page 105.

Merging Existing DB2 Data into the Group

DB2 provides no way to merge existing DB2 data or subsystems into a data sharing group. Read “Decide if Merging is the Right Thing to Do” on page 75 before attempting to merge subsystems into a single data sharing group.

Merging Subsystems

If, for some reason, you want to merge existing subsystems, you must do the following:

1. Choose a subsystem to be the originating member.
2. Move data and data definitions from the other DB2s to the originating member.

3. Add those other DB2 subsystems to the group using the new member install process.

Merging Data

If you have an application that is now running on independent DB2 subsystems, you might decide that it is an application that will work well in a data sharing group. In that case, you must move the data and merge the catalog definitions for that application from the independent DB2s into the data sharing group. Because those independent DB2s still exist, you cannot reuse their subsystem names when installing new members into the data sharing group.

DB2 does not provide an automated way to move catalog definitions from one DB2 into the catalog of the data sharing group. If you have procedures and tools in place now to do things such as move applications from test to production, or to handle merging databases from enterprise reorganizations or mergers, those same procedures can be used to move applications into the data sharing group.

| ### Existing Distributed Applications

If you move existing data to the data sharing group, it is likely that the location name of objects will change. Existing distributed applications that remotely reference that object by its three-part name must be changed to the new name, and any aliases on that table must also be dropped and recreated with the new location name.

Any application containing explicit SQL CONNECT statements that reference an old location name must be recoded. Any DB2 plan that uses an old location name for the CURRENTSERVER keyword must be bound again.

Procedure for Moving Data

The procedure described here is very difficult. It assumes you are going to change the catalog alias of the data sets to the alias of the data sharing group. If you want to keep the existing catalog alias, you must include steps to protect the data sets (such as by creating dummy data sets) so as not to overlay those data sets when you create the objects in the data sharing group.

In the following procedure, we use the term “target” for the DB2 subsystem you are moving data to (the data sharing group), and “source” for the DB2 you are moving data from.

1. Decide whether you are going to use DSN1COPY to move the data:

   The DSN1COPY utility copies the data and translates OBIDs. This is the simplest way, operationally, but translating OBIDs is an I/O-intensive process. Every record has the table OBID in its header, and this OBID must be translated.

   Another method can be used to avoid the heavy I/O used for translating OBIDs but is more complex. With this method, you use the existing data sets (making no copies), and the same table OBIDs as on the source subsystem. You must use the REPAIR utility to change the OBIDs for index spaces and table spaces in the header pages. REPAIR is also needed to reset the LEVELID of the page sets before you restart them on the target subsystem.

2. Choose a DB2 subsystem's catalog to be the “original” catalog for the data sharing group. In other words, this DB2 is the originating member of the group.

There are many considerations in choosing which DB2 to be the originating member of the group. For example, it probably makes sense to choose the member with the most database objects as the originating member so you can avoid moving as many objects as possible.

However, if all DB2s are pretty much equal, and if you are planning *not* to use DSN1COPY, also consider the log RBA values of the existing subsystems. Compare the end-of-log RBAs with the high order 6 bytes of the time-of-day clock timestamp on each of their systems (this is called the “truncated” timestamp). The usual case is that the RBA will be less than the truncated timestamp. In this case, any DB2 can be chosen as the originating member.

In the event that the current end-of-log RBA in any of the existing DB2 subsystems is higher than the 6-byte truncated timestamp value at the time you are ready to enable sharing, then either:

- Choose the DB2 subsystem that has the highest RBA as the originating member, or
- Use DSN1COPY with the RESET option to reset the log RBAs in each data and index page to 0 when you move databases from other DB2 subsystems to the data sharing group.

3. Resolve name conflicts among the objects and authorization IDs in the data sharing group.
4. Create the objects on the target subsystem.

   If you are using DSN1COPY with OBIDXLAT, you can enter the CREATE statements in any order. DB2 assigns new OBIDs for these objects.

   If you are not using DSN1COPY, or are using DSN1COPY without OBIDXLAT, you must query the DB2 SYSIBM.SYSTABLES catalog on the source DB2 subsystem to get the table OBIDs for tables that are within a database.

   When you run the SQL on the target subsystem, use the OBID clause on the CREATE TABLE statement to specify an OBID the same as the table OBID on the original subsystem. However, you must be sure that the OBID you specify is available. If an OBID is being used for another object within the database (say, for example, an index or referential constraint), DB2 won't allow you to create the table with the OBID you want.

   One way to guarantee the availability of OBIDs for all tables within a database is to defer the creation of all indexes and referential constraints until all tables are created. All CREATE TABLE statements must have the OBID clause to guarantee that they are assigned the correct OBID. If an explicit table space name is specified, then the CREATE TABLESPACE statement must come immediately before the first CREATE TABLE statement for that table space. This helps prevent a group of CREATE TABLESPACE statements from using up OBIDs that are needed for the tables.
5. On the source, run the utility REORG on any table spaces for which the following conditions are both true for a table in the table space:

   - A column has been added (ALTER TABLE ADD COLUMN) with no subsequent REORG
   - The columns are are all fixed-length

   If you are unsure if a table meets this criteria, query SYSIBM.SYSTABLES for those tables in which CREATEDTS does not equal ALTEREDTS. However, there is no way to tell from the DB2 catalog whether the ALTER consisted of adding a column.

6. Stop the database on the target subsystem. If you are not using DSN1COPY, go to step 8.
7. Use DSN1COPY with the OBIDXLAT and RESET options to translate the OBIDs, to reset the level ID, and to copy the data sets.

   After you've completed this step, go to step 12.
8. Delete the data sets you created on the target subsystem when you ran the SQL to create the objects. Rename the source data sets to the high level qualifier of the target.
9. Record the object identifiers of the indexes and table spaces you just created on the target.

   Query the SYSIBM.SYSTABLESPACE catalog table to get the DBID and PSID of the table spaces. Query the SYSIBM.SYSINDEXES catalog table to get the DBID and ISOBID of the indexes.
10. Use the REPAIR utility to change the identifiers in the page set header page (or header pages, if you have a partitioned page set) to match the new identifiers on the target subsystem.

    The identifiers consist of two 2-byte fields: HPGDBID and HPGPSID. You must locate and replace these identifiers as follows:

    - For non-partitioned page sets, locate and replace the 4 bytes starting at `X'0C'` starting on page `X'0'`.
    - For partitioned page sets, locate and replace the 4 bytes starting at `X'0C'` on page `X'n...0'` where *n* is the partition number. The description of the REPAIR utility in Section 2 of *Utility Guide and Reference* contains more information about how to specify partition numbers.
11. Use the REPAIR utility with the LEVELID option to reset the level indicator of the page sets to a neutral value.
12. Start the database on the target subsystem for read-write access.
13. Optionally, drop the database objects on the source subsystem.
14. Make full image copies of all data. This is the earliest time to which data recovery can occur after the merge.
15. Run the RUNSTATS utility on the target.
16. All plans and packages on the source subsystem have to be bound on the target subsystem. Any plans and packages on the target subsystem that change because of name conflict resolution must be bound as well. All plans and packages will have to have the appropriate authorizations granted.

Testing the Data Sharing Group

When you installed DB2, there were sample objects created in job DSNTEJ1. The DSNTESD member of *prefix*.SDSNSAMP contains SQL statements that refer to these objects. These SQL statements can be used to test group buffer pool caching, global lock serialization, and concurrency in the data sharing group. Do these tests after you have installed several data sharing members.

Test Group Buffer Pool Caching

Use the SQL statements in DSNTESD to verify that the group buffer pool operates correctly.

1. Run SPUFI on more than one data sharing member, using DSNTESD as the input data set. Specify AUTOCOMMIT=YES on the SPUFI panel.

   Run SPUFI on the different members serially, a few seconds apart if possible. (The runs must be close enough together to avoid having DB2 close the page set because of infrequent updates. The default amount of time between updates before DB2 switches the page set from read-write to read-only is 10 minutes, as specified on the PCLOSET subsystem parameter. PCLOSEN can also cause the data set to switch to read-only. It is a number of consecutive checkpoints.

   By running the SQL statements serially, DB2 detects inter-DB2 R/W interest on the table space and index and uses the group buffer pool.

   Verify that ITEM_COUNT increases by 5 after each run.

2. After running SPUFI on more than one member, issue the following command to see if the table space and index are using the group buffer pool.

   ```
   -DB1G  DIS DB(DSN8D51A) SPACENAM(DSN8S51S,XPARTS) LOCKS
   ```

   If the P-lock state is IX or SIX, then the table space and index are group-buffer-pool-dependent, as they should be.

3. Issue the following command to display the statistics for GBP0:

   ```
   -DB1G DIS GBPOOL(GBP0) GDETAIL
   ```

   In the group detail statistics, look for non-zero values in the READS and WRITES values of the display. This indicates that DB2 is using the group buffer pool successfully for caching.

Test Global Lock Serialization

Use the SQL statements in DSNTESD to verify that locks are acquired and released correctly across multiple data sharing members.

1. Run SPUFI, using DSNTESD as input, on *member 1*. Specify AUTOCOMMIT=NO.

   Because you have inserted data into DSN8510.PARTS but have not committed, *member 1* holds global locks.

2. Run SPUFI, using DSNTESD as input, on *member 2*. Specify AUTOCOMMIT=NO.

Because *member 1* holds global locks, *member 2* must wait to perform the insert.

3. In less than a minute, commit on *member 1*. (If you wait too long to commit, *member 2* will experience a lock timeout.)

   The global locks should be released, and *member 2* should be able to proceed. Verify that ITEM_COUNT has increased by 5.

Test Concurrency

Use the SQL statements in DSNTESD to test concurrency within the data sharing group.

1. Run SPUFI concurrently on different data sharing members. Specify AUTOCOMMIT=YES.

   Global locking ensures that inserts to DSN8510.PARTS are coordinated across data sharing members.

2. Verify that ITEM_COUNT increases by 5 each time the run completes successfully.

Test Sysplex Query Parallelism

There is no sample procedure to test Sysplex query parallelism, but there is a way you can use your own data to confirm that a single query can be processed on more than one member of the data sharing group. Choose an existing query that you know uses CP parallelism, such as a SELECT COUNT(*) for a table in a large partitioned table space, and use the following procedure that forces DB2 to split the query across multiple DB2s:

1. Decide which DB2 will be the query coordinator for your test, and make sure the COORDINATOR field on installation panel DSNTIPK is set to YES for that DB2.

2. Make sure that all assistants have the ASSISTANT field on installation panel DSNTIPK set to YES.

3. Make sure your statement does not include one of the restrictions listed in Section 5 (Volume 2) of *Administration Guide*.

4. Run EXPLAIN on the statement.

   An X in the PARALLELISM_MODE column of the PLAN_TABLE output indicates that this statement can be split across multiple DB2s.

5. Set buffer pool allocation thresholds on the DB2s that you want considered as possible assistants:

   ```
   ALTER BUFFERPOOL (BPn) VPSIZE(z) VPSEQT(100) VPPSEQT(100) VPXPSEQT(100)
   ```

   Ensure that the VPSIZE is large enough to support parallel processing. Start with at least 50 buffers on each query assistant.

6. Make sure that accounting trace class 3 is active on the coordinating DB2 subsystem.

7. Run the query.

8. Inspect the IFCID 0003 trace record. Field QWA01RBN corresponds to the number of assisting DB2s. This value should be greater than 0.

Updating Subsystem Parameters for a Member

There are no group-wide subsystem parameters that can be updated. To update subsystem parameters for an existing DB2 member within the DB2 data sharing group, specify the following installation options:

```
INSTALL TYPE         ===> UPDATE
```

The created installation job creates new parameters for the DB2 member.

Migrating an Existing Data Sharing Group to the New Release

Migrating a data sharing group requires a carefully thought out plan:

1. Read the information about migration considerations in "Data Sharing Migration Considerations" and also in Section 2 of *Installation Guide*.
2. Read the information in "Considerations for Mixed Releases in a Data Sharing Group" on page 108.
3. Make a plan to migrate the data sharing group over as short a time as possible.
4. Apply the fallback SPE to the Version 4 load library before attempting to migrate any member of the group. Stop and restart each member to pick up the change.
5. Follow the procedure in "Procedure to Migrate the Data Sharing Group" on page 114. Refer to Section 2 of *Installation Guide* for detailed information about migration. To prepare for fallback, keep the subsystem parameter load module used by Version 4.

Data Sharing Migration Considerations

- Plans and packages bound in DB2 Version 4 must be rebound on a Version 5 subsystem to run with Sysplex query parallelism. Any dynamic queries that are contained in these plans can use Sysplex query parallelism, assuming that all other conditions allow it.
- If you are using the resource limit facility to set limits on specific chargeback amounts, a query that runs on multiple DB2s can conceivably accumulate a total that is much greater than when it runs within a single DB2. The theoretical limit is:

  ```
  (number of parallel groups) * (number of DB2s) * SU limit
  ```

  See "Setting Limits using the Resource Limit Facility" on page 213 for more information.
- When using Sysplex query parallelism (or CP-parallelism in a single DB2) an abend of a parallel task returns a -904 SQLCODE and a new resource type rather than an abend reason code.

Considerations for Mixed Releases in a Data Sharing Group

DB2 allows the data sharing group to remain available while you migrate members of the group to the newest release. However, it is best to plan the migration during periods of low activity in the group because the DB2 catalog is unavailable for a brief time during the migration of the first member.

Recommendation

There is no single recommendation that can be made for all DB2 installations. The purpose of coexistence is to allow for migration of individual data sharing members so that your applications can continue to access DB2 data while the members are being migrated. However, you must weigh the benefit of improved availability against the operational costs of running in coexistence mode: many new functions are not available, and there are some system management issues to consider.

If you do not require continuous availability, it is easier to shut down the group for the migration to avoid the coexistence environment. If you need to run in coexistence mode, make a plan to migrate the members within as short a time as possible so that you can take advantage of the new functions available in the current release and minimize the operational complexity.

Determining the Release of the DB2 Group

When the first data sharing member starts Version 5, the entire group is now considered to be at that level, even though not all individual members of the group have migrated. This means that the *group level* is Version 5. You can see the group level by issuing the DISPLAY GROUP command as shown in Figure 25.

```
- DSN7100I -DB1G DSN7GCMD
- *** BEGIN DISPLAY OF GROUP(DSNDB0G ) GROUPLEVEL(510)
-                                      GROUP ATTACH NAME(DB0G)
- --------------------------------------------------------------------
- DB2                                    DB2 SYSTEM    IRLM
- MEMBER   ID  SUBSYS CMDPREF   STATUS   LVL NAME      SUBSYS IRLMPROC
- -------- --- ----   --------  -------- --- --------  ----   --------
- DB1G       1 DB1G   -DB1G     ACTIVE   510 MVS1      DJ1G   DB1GIRLM
- DB2G       2 DB2G   -DB2G     ACTIVE   510 MVS2      DJ2G   DB2GIRLM
- DB3G       3 DB3G   -DB3G     ACTIVE   410 MVS3      DJ3G   DB3GIRLM
- DB4G       4 DB4G   -DB4G     FAILED   510 MVS4      DJ4G   DB4GIRLM
- --------------------------------------------------------------------
- SCA   STRUCTURE SIZE:     1024 KB, STATUS= AC,   SCA IN USE:    11 %

 - LOCK1 STRUCTURE SIZE:     1536 KB
- NUMBER  LOCK ENTRIES:      262144
- NUMBER  LIST ENTRIES:        7353, LIST ENTRIES  IN USE:           0
- *** END DISPLAY OF GROUP(DSNDB0G )
- DSN9022I -DB1G DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
```

Figure 25. DISPLAY GROUP Command Shows Group and Member Release Level

Determining the Function Level of the IRLM Group

IRLM communicates coexistence information by using function levels. A function level is an ever- increasing number that each IRLM can use to tell other IRLMs in the group what level of function it supports. The group function level is the minimum of the individual IRLM function levels for all IRLMs that can coexist. Any IRLM that tries to a data sharing group is prevented from doing so by active members that cannot coexist with the new IRLM's function level.

When the function level for the group changes, that change is serialized by IRLM with lock structure rebuilds. In most cases, however, the lock structure does not actually do a full rebuild. The first phase of rebuild is enough to quiesce the work and cause the function level change to occur. These “partial”take place when an IRLM joins or leaves the group and if that activity causes the group function level to change. For example, if the IRLM group is currently at function level n, and the IRLM member that wants to join the group is at n-1, the partial rebuild occurs to lower the group function level. Conversely, if the lowest level member leaves the group, the partial rebuild might occur if the group can coexist at a higher function level.

To display the function levels, enter the following command:

MODIFY *irlmproc*,STATUS,ALLI

You get output like that shown in Figure 26:

```
 DXR103I LRLM STATUS IRLMID=007
IRLMS PARTICIPATING IN DATA SHARING GROUP FUNCTION LEVEL=013
IRLM_NAME IRLMID STATUS LEVEL SERVICE MIN_LEVEL MIN_SERVICE
  JRLM      005    UP    015  PQ15290    012      PN90337
  KRLM      006    UP    013  PN92893    006      IRLM2.1
  LRLM      007    UP    014  PN09381    006      IRLM2.1
```

Figure 26. Determining IRLM function levels

The IRLMs shown in Figure 26 are at group function level 13, which is the lowest level of any of the individual members (KRLM). The MIN_LEVEL field shows the minimum level with which this IRLM can coexist. MIN_SERVICE indicates the service or release that corresponds with that MIN_LEVEL.

Call Attachment and TSO Attachment Coexistence

While you are in a coexistence environment, you can attach to either release of DB2 with your existing TSO logon procedures or JCL. After you migrate all members of the group to the latest level of DB2, be sure to update those procedures and jobs to point to the latest level of DB2 load libraries.

Avoiding Automatic Rebinds

When developing your migration plan, keep in mind that new functions introduced in this release are not available to members of the group who have not yet migrated. Thus, it is best to either:

- Migrate all members to the new release before attempting to use new utilities or commands, or any new options. Do not allow members to run any applications that include new SQL function until all members have migrated to the new release.

- Don't allow packages and plans bound on Version 5 to execute on members that have not yet migrated. Or, do not allow plans or packages to be bound on Version 5.

  This serves two purposes. First, if those Version 5-bound plans or packages are using new functions, you can avoid the application errors that occur if the plan or package tries to execute an SQL statement that is not allowed in Version 4. Second, it avoids the automatic rebind that occurs when any plan or package that is bound on Version 5 is run on Version 4. It also avoids the automatic rebind that occurs when a Version 5-bound plan or package that was automatically rebound on Version 4 is later run on Version 5.

If it is not possible to enforce on which member a plan or package runs, consider how you want to handle binds and automatic rebinds while two releases are coexisting. One approach is to disallow all binds and disable all automatic rebinds on the Version 5 subsystem. The other approach is to disable only those automatic rebinds that occur on Version 5 in step 3 of the following scenario:

1. Plan or package is bound on Version 5
2. Next, the plan or package is run on Version 4 (automatic rebind occurs on Version 4)
3. Next, the plan or package is run on Version 5 (automatic rebind occurs on Version 5)

   Because DB2 perceives this scenario as a fallback and remigration scenario, the autobind that occurs in step 3 is called a “remigration” rebind.

By disallowing the automatic rebind in step 3, you are avoiding the thrashing that can occur by having the plan or package rebind every time it runs on a member of a different level.

Disallowing all binds: You can do the following on the Version 5 member to avoid automatic rebinds:

- Specify NO on the AUTO BIND field of installation panel DSNTIPO. This disables *all* automatic rebinds on the Version 5 member for any reason. This means you cannot run a plan or package on a Version 5 subsystem if it has gone through the following scenario:

  1. Bind on Version 5.
  2. Run on Version 4. This causes an automatic rebind on that Version 4 subsystem.
  3. Try to run on Version 5. This returns a -908 SQLCODE (SQLSTATE '23510') because DB2 must autobind the plan or package on Version 5 before running it there.

- Use the resource limit facility to disallow BIND operations. Do this by inserting rows in the resource limit specification table (RLST) to set RLFFUNC to “1” and RLFBIND to “N.” This ensures that nobody binds plans or packages on Version 5.

  Here is an example of an INSERT statement for the RLST that disallows all BIND operations for all authorization IDs (except those with installation SYSADM or installation SYSOPR) for all packages and plans:

  ```
  INSERT INTO authid.DSNRLSTxx
    (RLFFUNC,RLFBIND) VALUES('1','N');
  ```

After all the DB2 members in the data sharing group are running at the current release, enable automatic rebinds again by setting AUTO BIND=YES, and allow bind operations by changing the RLST accordingly or by stopping the resource limit facility using the STOP RLIMIT command.

Disallowing only the automatic remigration rebind: To avoid the automatic remigration rebind that occurs because a plan or package was bound on Version 5, run on Version 4, and then rerun on Version 5, specify COEXIST for the ABND subsystem parameter of the Version 5 members. This means that automatic rebind occurs on Version 5 in the following circumstances:

- The plan or package is marked invalid
- You migrate to a future release, bind a plan or package on that release, and then run the plan or package on Version 5.

When all members are at Version 5, you do not need to change the COEXIST value. The behavior is the same as if you had specified AUTOBIND YES.

Recommendations for BIND

The following BIND and REBIND options that are new in Version 5 are rejected if they are used on a Version 4 DB2.

- REOPT for BIND and REBIND PLAN or PACKAGE
- NOREOPT for REBIND PLAN or PACKAGE
- DEFER for BIND PACKAGE
- NODEFER for REBIND PACKAGE
- KEEPDYNAMIC(YES) for BIND and REBIND PLAN or PACKAGE
- KEEPDYNAMIC(NO) for REBIND PLAN or PACKAGE
- OPTIONS(COMMAND) for BIND PACKAGE
- ENABLE(RRSAF) for BIND and REBIND PLAN or PACKAGE
- DISABLE(RRSAF) for BIND and REBIND PLAN or PACKAGE

To avoid problems, make sure the DB2 subsystem named in the DSN subcommand matches the procedure of the right release.

Recommendations for Utilities

Until all members of the data sharing group are running at the current release, avoid using any of the utilities functions available in Version 5. However, as long as you use utility options that are supported in Version 4, utilities can attach to either a Version 4 or Version 5 subsystem.

The utilities batch module (DSNUTILB) is split into two modules: a release-independent module (still called DSNUTILB) and a release-dependent module (DSNUT410 or DSNUT510). To operate in a mixed-release data sharing environment, you must have both DSNUT410 and DSNUT510 available to the utility jobs that operate across the data sharing group. Ways of doing that are in “Changing STEPLIB in DSNUPROC” and “Cross-Copy into Load Libraries” on page 112.

Changing STEPLIB in DSNUPROC: The recommended way of making the release-dependent modules available for utility jobs is to change the STEPLIB in DSNUPROC to include the other release as in the following example:

```
//DSNUPROC PROC LIB='DSN510.SDSNLOAD',
//         SYSTEM=DSN,
//         SIZE=0K,UID='',UTPROC=''
//DSNUPROC EXEC PGM=DSNUTILB,REGION=&SIZE,
//         PARM='&SYSTEM,&UID,&UTPROC'
//STEPLIB  DD   DSN=&LIB,DISP=SHR;
//         DD   DSN=DSN410.SDSNLOAD,DISP=SHR <— coexistence
//SYSPRINT DD   SYSOUT=*
//UTPRINT  DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//*DSNUPROC PEND        REMOVE * FOR USE AS INSTREAM PROCEDURE
```

Cross-Copy into Load Libraries: Another approach, not recommended for long-term use, is to cross-copy the release-dependent modules into the load libraries of the other release. So, copy DSNUT410 into the Version 5 load libraries, and copy DSNUT510 into the Version 4 load libraries. The problem with this approach is that you must repeat this procedure every time you apply maintenance to these modules. Thus, as with coexistence in general, this approach is only for short-term use.

Here is some sample JCL to do the cross-copy:

```
//CROSCOPY PROC D510TPRE='DSN510',
//            D410TPRE='DSN410',
//            RGN=4096K,SOUT='*'
//* *****************************************************************
//*    FOR EXECUTION OF IEBCOPY - DB2 POST-INSTALLATION            ***
//* *****************************************************************
//COPY     EXEC PGM=IEBCOPY,REGION=&RGN
//SYSUT3   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT4   DD  UNIT=SYSDA,SPACE=(CYL,(5,1))
//* **************  DB2 TARGET LIBRARIES  *****************************
//*
//D410LOAD DD  DSN=&D410TPRE..SDSNLOAD,DISP=OLD
//D510LOAD DD  DSN=&D510TPRE..SDSNLOAD,DISP=OLD
//*
//         PEND
//COEXIST  EXEC PROC=DSNTIJCO
//SYSIN    DD *
  COPYMOD INDD=((D510LOAD,R)),OUTDD=D410LOAD
   SELECT MEMBER=(DSNUT510)
  COPYMOD INDD=((D410LOAD,R)),OUTDD=D510LOAD
   SELECT MEMBER=(DSNUT410)
```

Recommendation for Group Restart

If a group restart is necessary while the data sharing group is running with mixed releases, we recommend that you issue the START command only for Version 5 members. Do not start the Version 4 members until the Version 5 members have completed forward log recovery. This recommendation is because if a Version 4 member performs the group restart for a Version 5 member, it adds pages to the logical page list during the peer-forward recovery phase when it tries to apply redo log records against a release-dependent object (table spaces defined with the LARGE keyword, for example).

Coexistence Considerations for Specific Functions

The purpose of this section is to clarify how some of the new Version 5 functions work (or not) when there are mixed releases in the data sharing group. These considerations are similar to those you have to understand for the fallback environment. See the fallback considerations section of Section 2 of *Installation Guide* for more information.

Communications Database: The communications database has extensive changes for Version 5. Not only are there new tables and column names, but it is now shipped as part of the DB2 catalog. The information in the existing CDB is copied into the new CDB during the migration process of the first member, but you must keep the old CDB accessible so that members at Version 4 can process distributed requests using information in that CDB.

Don't forget to update the old CDB when you update the new one, although this is not possible for information that is Version 5-only function.

TCP/IP Requests: Only data sharing members running at Version 5 are eligible to service TCP/IP requests. Requests are routed only to eligible members, so TCP/IP applications can connect to a group that consists of mixed release levels.

Stored Procedures: The recommendation is to use DB2-established stored procedures address spaces until all members are running at Version 5 and with OS/390 Release 3. If a stored procedure that runs on a Version 4 subsystem uses function that is not supported in that release, then the stored procedure is not allowed to execute.

ASCII Data: Do not allow any objects to be created with CCSID ASCII and do not choose the ASCII default subsystem parameter until all members of the group are running at Version 5. Any members that are running at Version 4 cannot access a table created with CCSID ASCII.

Duplexed group buffer pools: Do not start duplexing until all members of the group are at the level of DB2 and OS/390 code that allows them to be duplex-enabled, and until the group buffer pool is allocated in a coupling facility with coupling facility control code level 5 or higher. No DB2 members can connect to a duplexed group buffer pool if they are not duplex-enabled.

If any downlevel members are connected to a group buffer pool, duplexing cannot start for that group buffer pool until all downlevel members are disconnected from the group buffer pool.

Group Buffer Pool Rebuild and Recover Enhancements: You can rebuild group buffer pools and automatically recover them only when *all* active connections to group buffer pools are at the Version 5 level.

Creating Inline Copies with REORG or LOAD REPLACE: Image copies created with REORG or LOAD REPLACE cannot be used for recovery on a Version 4 subsystem. DB2 tries to use a previous copy that is not made with the inline copy option of LOAD and REORG.

Partitioned Table Spaces Created with LARGE Option: Table spaces created as LARGE are not accessible from a Version 4 subsystem. IFCID 0306 can be used from a Version 4 subsystem to retrieve log from a LARGE table space, but compressed data cannot be decompressed.

Partitioned Table Spaces Created with LOCKPART Option: Table spaces created with LOCKPART YES are not accessible from a Version 4 subsystem.

Table Spaces Created with MEMBER CLUSTER Option: A table space that is created with the MEMBER CLUSTER option, is not accessible from a Version 4 subsystem. Neither are its tables or indexes.

Indexes Created with Non-Default PIECESIZE Option: An index that is created or altered to a non-default PIECESIZE (2G, or 4G for indexes created on table spaces defined as LARGE), is not accessible from a Version 4 subsystem.

Terminating an Online REORG: You cannot terminate a stopped REORG SHRLEVEL(REFERENCE) or SHRLEVEL(CHANGE) from a Version 4 subsystem. It can only be terminated from another Version 5 member.

Indexes Created with Non-Default PIECESIZE Option: An index that is created or altered to a non-default PIECESIZE (2G, or 4G for indexes created on table spaces defined as LARGE), is not accessible from a Version 4 subsystem.

Updating Partitioning Keys: You cannot update a column of the partitioning key from a Version 4 subsystem.

Partitioned Index with a Limit Key Greater Than 40 Bytes: A partitioned index with a limit key greater than 40 bytes is not accessible from a Version 4 subsystem.

Procedure to Migrate the Data Sharing Group

Jobs that the Migration CLIST Tailors: Jobs marked with an asterisk (*) are modified only when you migrate the first member.

DSNTIJEX*	DSNTIJGF	DSNTIJSG*	DSNTIJUZ
	DSNTIJIC*	DSNTIJSU*	DSNTIJVC*
DSNTIJFT	DSNTIJIN*	DSNTIJTC*	DSNTIJXC*
DSNTIJFV	DSNTIJMV	DSNTIJTM*	

The DB2I CLISTs also get edited.

Before migrating to Version 5, you must have the fallback and coexistence SPE installed on the Version 4 load library. Version 5 members cannot start if any one of the active Version 4 members do not have the SPE applied. Similarly, if a Version 5 member is started, a Version 4 member cannot start unless it has the fallback and coexistence SPE applied.

Attention: Follow these directions carefully. The first member of the data sharing group uses DSNTIDXA as its input member name. Subsequent members **must** use a previous member's output member name as its input member name.

To migrate the data sharing group:

1. For the **first** member to migrate, specify the following on panel DSNTIPA1:

```
INSTALL TYPE ===> MIGRATE
DATA SHARING ===> YES
.
DATA SET NAME(MEMBER)==> this member's input member from 4.1
.
INPUT MEMBER NAME  ===> DSNTIDXA
OUTPUT MEMBER NAME ===> this member's output PDS member
```

2. On panel DSNTIPP2, specify 1 to indicate that this is the first member of the group to migrate.
3. Complete the installation panels. Specify parameters according to the guidelines in “The Scope and Uniqueness of DB2 Subsystem Parameters” on page 88.
4. Complete all migration steps, as described in Section 2 of *Installation Guide*.
5. Migrate the **next** member of the group. Specify the following on panel DSNTIPA1:

```
INSTALL TYPE ===> MIGRATE
DATA SHARING ===> YES
.
DATA SET NAME(MEMBER)==> this member's input member from 4.1
.
INPUT MEMBER NAME  ===> first member's output PDS member
OUTPUT MEMBER NAME ===> this member's output PDS member
```

6. Specify 2 on panel DSNTIPP2 to indicate that this is not the first member of the group to migrate.
7. For this next member, complete the following migration steps, as described in Section 2 of *Installation Guide*.
   a. Migration Step 5: Connect DB2 to TSO

      You do not have to run job DSNTIJVC for the second and subsequent members.

   b. Migration Step 6: Connect IMS to DB2 (Optional)
   c. Migration Step 7: Connect CICS to DB2 (Optional)
   d. Migration Step 8: Stop Version 4 Activity
   e. Migration Step 10: Define DB2 Initialization Parameters

      Job DSNTIJUZ only contains a subset of the steps contained in the first migrating member's DSNTIJUZ job. DSNHDECP is not reassembled for each subsequent migrating member.

   f. Migration Step 11: Establish Subsystem Security (Optional)
   g. Migration Step 12: Define DB2 Version 5 to MVS

      Job DSNTIJMV contains a subset of the steps contained in the first migrating member's DSNTIJMV job. The steps to update the IEAAPFxx and LNKLSTxx members, and the steps to update the language procedures are not included when subsequent members migrate.

   h. Migration Step 14: IPL MVS

      This step can be done before migrating DB2 if you have made the appropriate updates to the MVS libraries.

   i. Migration Step 15: Start Version 5

j. Migration Step 22: Verify Version 5

k. For each subsequent member migration, repeat all steps beginning with step 5 on page 115.

Falling Back and Remigrating

Which procedure you use to fall back to Version 4 depends on whether Version 4 was enabled for data sharing or not when you migrated to Version 5.

- If you are falling back from Version 5 data sharing to Version 4 data sharing, use the procedure in "Falling Back to Version 4 Data Sharing."
- If you are falling back from Version 5 data sharing to Version 4 non-data-sharing, see "Falling Back to Version 4 Non-Data-Sharing."

Falling Back to Version 4 Data Sharing

Before falling back to Version 4, you must have the fallback and coexistence SPE installed on the Version 4 load library. As each member falls back, you will be running in coexistence mode, so be sure to read "Considerations for Mixed Releases in a Data Sharing Group" on page 108. For complete information about falling back to Version 4, see Section 2 of *Installation Guide*.

You can do the fallback procedure on one member of the data sharing group at a time. Other members can continue to run while another member is falling back.

1. Check the CFRM policy and make sure that any specifications for DUPLEX(ENABLED) are changed to DUPLEX(ALLOWED) or DUPLEX(DISABLED). Then use the MVS command SETXCF STOP,DUPLEX to return any duplexed group buffer pools to simplex mode. If you do not return group buffer pools to simplex mode before falling back, connections receive a resource unavailable condition.
2. Stop DB2 on the member that is falling back.
3. Reactivate Version 4 code for that member (DSNTIJFV).
4. Reconnect TSO, IMS, CICS to Version 4.
5. Start Version 4:
   a. Enter the command START DB2.
      1) Check for indoubt units of recovery
      2) Check for outstanding restrictive states
6. Verify Version 4.
7. Repeat steps 2 through 6 for each member of the data sharing group.
8. When the last member is falling back, run job DSNTEJ0 to free plans and packages and drop objects created by the Version 5 sample programs.

Falling Back to Version 4 Non-Data-Sharing

If you are falling back from Version 5 to a Version 4 non-data-sharing DB2, you must do two procedures:

1. Disable data sharing as described in "Procedure to Disable Data Sharing" on page 118. When you complete this procedure, you have a single member, the *surviving* member.

2. Fall back to Version 4, as described in Section 2 of *Installation Guide*.

Take Special Care when the Survivor is not the Originating Member: If you disable data sharing so that the surviving member is not the originating member, be sure to do the following tasks when you fall back:

1. Tailor the DSNTIJFV job to refer to the procedures used by the surviving member.
2. Modify the *ssnm*MSTR procedure of the surviving member, to include the correct BSDS names.
3. Modify the *ssnm*SPAS procedure of the surviving member to include the correct SUBSYS name.
4. Rerun job DSNTIJUZ to update DSNZP*xxx* and DSNHDECP to refer to the surviving member.

Remigrating

If you followed the procedure in "Falling Back to Version 4 Data Sharing" on page 116, simply remigrate each member of the data sharing group as described in Section 2 of *Installation Guide*.

If you followed the procedure in "Falling Back to Version 4 Non-Data-Sharing" on page 116, remigrate the member as described in Section 2 of *Installation Guide*, and then re-enable data sharing, as described in "Re-enabling Data Sharing" on page 120.

Disabling and Re-enabling Data Sharing

Disabling data sharing is a complex procedure. Do not make a disabling procedure part of your contingency plans for handling recovery situations. For temporary bypasses to data sharing problems, you are much better off moving to one-way data sharing. Some situations can be resolved by doing group restart.

The disabling procedure is included here in case:

- You are in the process of falling back to a non-data-sharing release of DB2, as described in "Falling Back to Version 4 Non-Data-Sharing" on page 116.
- You have made a strategic decision to move off data sharing
- One-way data sharing is not working

Do not attempt to disable data sharing without a thorough understanding of the process.

After you have disabled data sharing, only *one* DB2 from the data sharing group can access the previously shared data. That DB2 is called the *surviving* member.

Disabling Data Sharing

This section describes:

Summary of Disabling Procedure

The procedure to disable data sharing ensures that the most recent versions of all pages are externalized from the group buffer pool to DASD because DB2 does not use the group buffer pool after data sharing is disabled. You must ensure that data is written to DASD, or else **you will lose data when you start DB2 after disabling data sharing.**

You must also ensure that there is no need to recover data from information contained on other members' logs after you disable data sharing, as described in “Data Recovery after Disabling DB2 Data Sharing” on page 120. **Those logs are not available to the surviving member after you disable data sharing.** To prevent the survivor from applying inconsistent updates during recovery processing, a cold start is required to disable data sharing.

If you are planning to re-enable data sharing for this group, do not change any group-wide information in the surviving member's BSDS. This includes the catalog alias name and the database password. It also includes the DDF name and password information, even if you are not going to use DDF when you re-enable. If you change any of this information, you will have to change that value in every member's BSDS before you start the group.

| **Attention:** Do not attempt to go through the installation process to re-enable data sharing after you have disabled it. You must use the procedure described in “Procedure to Re-enable DB2 Data Sharing” on page 121.

Procedure to Disable Data Sharing

The procedure to disable data sharing, and return to the Version 5 non-sharing environment is as follows:

1. Decide which member is going to be the surviving member of the group.
2. Stop all members by entering the following command for each member of the data sharing group:

   ```
   -DB1G STOP DB2 MODE(QUIESCE)
   ```

3. Start the surviving member in maintenance mode by using the following command:

   ```
   -DB1G START DB2 ACCESS(MAINT)
   ```

4. Make sure data is consistent. Enter the following commands from the surviving member of the group. Do not go on to the next step until all problems are resolved.
   - DISPLAY GROUP

     Make sure the STATUS column contains the word QUIESCED for all but the surviving member, which contains the word ACTIVE. If it does not contain QUIESCED, then you must take the following actions, depending on the value in the STATUS column:

     – If the member is FAILED, you must restart the member and stop it successfully.

     – If the member has castout problems, indoubt units of recovery, or outstanding resynchronization problems, you must start that member in maintenance mode and fix the problem.

   - DISPLAY UTILITY(*)

If there is any remaining utility work for any member of the group, you must restart that member with ACCESS(MAINT) and either terminate the utility, or let it finish.

- DISPLAY DATABASE(*) SPACENAM(*) RESTRICT

  If there are any restricted table spaces or index spaces (such as write error ranges, recovery pending status, or logical page list entries), recover them from the surviving member.

5. Stop the surviving member by using the following command:

   ```
   -DB1G STOP DB2 MODE(QUIESCE)
   ```

6. Stop any IRLMs that have not stopped, using the following command:

   STOP *irlmproc*

7. Dismantle the data sharing group.

   a. Enter the following command to display the structures for this data sharing group:

      D XCF,STRUCTURE,STRNAME=*grpname**

   b. For all structures that are still allocated (STATUS:ALLOCATED) and still have connections (which appear as FAILED PERSISTENT) enter the following command to force the connections off those structures:

      SETXCF FORCE,CONNECTION,STRNAME=*strname*,CONNAME=ALL

   c. Delete all the DB2 coupling facility structures by using the following command for each structure:

      SETXCF FORCE,STRUCTURE,STRNAME=*strname*

   d. Edit the JCL in job DSNTIJGF to point to the correct BSDS data sets.

      DSNTIJGF is a change log inventory job that sets up the surviving member for a cold start. **Attention: Do not change the hex values that appear in the change log inventory CRESTART control statement. They are not real RBA values.**

   e. Run job DSNTIJGF.

      After you run this job, do not try to restart any of the non-surviving members. None of those members can start successfully.

8. Change the IRLM procedure to SCOPE=LOCAL.

9. Start the surviving DB2 subsystem with ACCESS(MAINT). Specify the old DSNZP*xxx*, from the non-data-sharing environment. If the surviving member is not the originating member, then you must reassemble the surviving member's subsystem parameters specifying the subsystem parameter DSHARE=NO in the invocation of the DSN6GRP macro. Also, comment out all steps from the DSNTIJUZ job except for those that reassemble and link-edit the subsystem parameters.

   When you start DB2 after having run DSNTIJGF, you are required to respond with Y to a cold start prompt (message DSNJ246I on the MVS console).

   This is a cold start, because DB2 increases the log RBA to a value higher than any LRSN used while sharing data. From now on, your RBAs will look like LRSNs.

10. Edit and run job DSNTIJFT, if necessary, to ensure that the surviving member's work file database is DSNDB07.

    The surviving member must use DSNDB07 as its work file database. If the work file database for the surviving member is not DSNDB07, drop that work file database and run job DSNTIJFT.

11. Verify that the surviving member works by running a subset of the Version 5 installation verification sample jobs.

    See the step for “Verifying Your Version 5 Subsystem” in Section 2 of *Installation Guide* for a list of these jobs.

12. To establish a new recovery point, take a full or incremental image copy or non-DB2 backup of all data. Run job DSNTIJIC to image copy the DB2 catalog and directory.

    We recommend that you do this step as soon as possible after data sharing is disabled. See “Data Recovery after Disabling DB2 Data Sharing” for more information about recovery.

13. Stop and restart DB2 for normal unrestricted access.

Data Recovery after Disabling DB2 Data Sharing

After data sharing is disabled, you cannot recover to the current point or to a previous point in time if that recovery depends on any portion of the log made before you disabled data sharing. Therefore, if there are any updates to the table space between the time of the copy and the time you disabled data sharing, DB2 does not let you use that copy as the basis for recovery. This is why we recommend that you create a new recovery point as soon as possible after you have disabled data sharing.

Using the Group Attachment after Disabling Data Sharing

After disabling data sharing, you can continue to use the group attachment name. There is no need to change this to the surviving member's subsystem ID.

Re-enabling Data Sharing

Perform a subset of the original procedure to enable data sharing. **You cannot use the enabling process of installation to re-enable data sharing**.

The following output from the original data sharing enabling procedure remains intact after disabling data sharing and does not need to be recreated or re-specified:

- Data sharing subsystem parameters (output from the CLIST processing when enabling data sharing)
- XCF definitions
- Coupling facility definitions
- RACF definitions
- DB2 catalog and directory

Procedure to Re-enable DB2 Data Sharing

The procedure to re-enable data sharing is as follows:

1. Edit the JCL in job DSNTIJGF.

   DSNTIJGF is a change log inventory job that sets up the BSDS of the surviving member for data sharing.

2. Run job DSNTIJGF.

3. Change the IRLM procedure to SCOPE=GLOBAL.

4. Start the surviving DB2 subsystem with the subsystem parameters used when data sharing was originally enabled.

   During startup, you will be asked to start all other members that were not quiesced at the time you disabled data sharing. You must start all these members.

5. Edit and run job DSNTIJFT on the surviving member to recreate the work file database for data sharing.

   See the directions in the prologue of job DSNTIJFT for information about editing this job to re-enable data sharing.

Removing Members from the Data Sharing Group

One of the features of DB2 data sharing is incremental growth, being able to add members to an existing group. However, there might be a situation in which you want to remove members from the group permanently or temporarily. For example, assume your group does the job it needs to do 11 months of the year. However, you get a surge of additional work every December that requires you to expand your capacity. It is possible to quiesce some members of the group for those 11 months. Those members are “dormant” until you restart them.

The same principle is used to “remove” a member of the group forever. Make sure a member is quiesced cleanly, and that member can remain dormant forever. In effect, it is removed from the group.

A quiesced member (whether you intend for it to be quiesced forever or only temporarily) still appears in displays and reports. It appears in DISPLAY GROUP output with a status of QUIESCED.

What Data Sets to Keep

When you quiesce a member, you must keep the log data sets until such time as they are no longer needed for recovery (other members might need updates that are recorded on that member's log). You must keep the BSDS, too, because it contains information that points to those log data sets.

The BSDS is also needed for group restart. However, if you are confident that logs for the quiesced member are no longer necessary, because that member has been quiesced for a long time or is permanently quiesced, it is possible to delete the BSDS data sets. However during group restart, you must expect the following message:

```
DSNR020I -DB2G csect-name START MEMBER DB1G, OR REPLY 'NO' or QUIESCED'
```

When you respond with QUIESCED, then DB2 issues the following message:

```
DSNR030I -DB2G csect-name WILL CONTINUE WITHOUT THE DB1G MEMBER'S LOG,
                          REPLY 'YES' OR 'NO'
```

In summary, you must do one of the following things to ensure that you have group restart capability:

- Keep the quiesced member's BSDS data sets (thus avoiding the WTOR messages above)
- Update your group restart procedure to ensure that operators know how to respond to the DSNR020I and DSNR030I messages.

Procedure to Quiesce a Member

In summary, to quiesce a member of the group, you must:

1. Stop the DB2 you are going to quiesce. Our example assumes you want to quiesce member DB3G.

   ```
   -DB3G STOP DB2 MODE(QUIESCE)
   ```

2. From another member, enter the following commands:

   ```
   -DB1G DISPLAY GROUP
   -DB1G DISPLAY UTILITY (*) MEMBER(member-name)
   -DB1G DISPLAY DATABASE(*) RESTRICT
   ```

   See step 4 on page 118 for more information about using these commands. If there is no unresolved work, you can stop now. However, if you want to create an archive log, continue to the next step.

3. If there is unresolved work, or if you want to do optional step 4 to create a disaster recovery archive log, start the quiesced member with ACCESS(MAINT).

   ```
   -DB3G START DB2 ACCESS(MAINT)
   ```

   If there is unresolved work, resolve any remaining activity for the member, such as resolving indoubt threads, finishing or stopping utility work, and so on.

4. Optionally, to create an archive log that can be sent to a disaster recovery site, archive the log for the member by entering the following command:

   ```
   -DB3G ARCHIVE LOG
   ```

5. Stop DB2 again with MODE(QUIESCE).

   ```
   -DB3G STOP DB2 MODE(QUIESCE)
   ```

Chapter 5. Communicating with a Data Sharing Group

A data sharing group can be a powerful server in your client/server environment. The group can be part of an SNA network, a Transmission Control Protocol/Internet Protocol (TCP/IP) network, or part of a network that uses both protocols. The data sharing group has a single-system image to requesting applications, whether those requests are coming in through TCP/IP or SNA network protocols. Requesting applications use the LOCATION NAME of the data sharing group to direct their SQL requests to that group.

A data sharing group can support many more threads than a single DB2 subsystem. The DDF thread limit for a DB2 group is “$n \times 25\,000$,” where n is the number of DB2 subsystems in the group. Thus, a group with 16 DB2 members can support 400 000 DDF threads.

Before Reading this Chapter: Be sure to read Section 3 of *Installation Guide*.

Included in this Chapter: The following topics are described in this chapter:

An Overview of the Ways to Access a DB2 Data Sharing Group

This section contains an overview of the ways to access a data sharing group.

SNA Network Alternatives: Each DB2 subsystem within the data sharing group has a unique DB2 NETID.LUNAME value. To provide a single-system image and to provide balancing of requests in the data sharing group, there are two ways to configure a DB2 for distributed access to the DB2 group using SNA:

- One alternative uses DB2-provided support and associates a given location name with a list of LU names at the requester location. We call this *member routing* because the requester defines a list of LUs for the data sharing group members. This is the recommended method of support unless your requester configuration does not support it.
- The other alternative uses VTAM-supplied support for generic resources, which allows a generic LU name for the group. We call this setup *group-generic* because a single LU name serves for the entire group.

TCP/IP Network: TCP/IP is supported only for access using DRDA protocols. When using TCP/IP, all members of the data sharing group use the same DRDA port number used to receive incoming SQL requests, but each member must have a resynchronization port number that is unique within the Sysplex. In the event of a failure, this unique number allows a client to reconnect to the correct member so units of work requiring two-phase commit can be resolved.

Clients connect to a data sharing group using a domain name. (It is possible to connect using an IP address, but this is not recommended because it does not work if DB2 is restarted on a different CPC.) After a client has connected to a member of the group the first time, that client receives a list of all eligible members in the data sharing group and subsequently can connect to any eligible member. This is functionally equivalent to member routing, but the requester does not have to define a name for each member of the data sharing group. See “TCP/IP Workload Balancing for DB2 Data Sharing” on page 131 for more information about balancing workloads in a TCP/IP network.

Mixed SNA and TCP/IP Networks: You can have the group send or receive requests using either or both network protocols. It can receive TCP/IP requests if a DRDA port and resynchronization port are defined for each member (any member that does not have this information cannot receive TCP/IP requests). When sending a request, DB2 uses the LINKNAME column of the SYSIBM.LOCATIONS catalog table to determine which protocol to use, as shown in Figure 27.

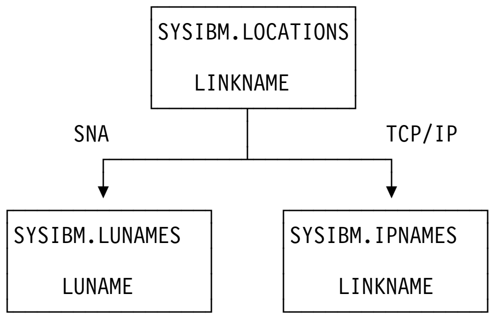

Figure 27. The LINKNAME Column of SYSIBM.LOCATIONS Determines which Protocol to Use. If the LINKNAME value is found in the SYSIBM.IPNAMES table, TCP/IP is used. If it is not, then the SYSIBM.LUNAMES table is checked. If the value is found in SYSIBM.LUNAMES, then SNA is used.

Attention: A requester cannot connect to a given location using both SNA and TCP/IP protocols. For example, if your SYSIBM.LOCATIONS specifies a LINKNAME of LU1, and if LU1 is defined in both the SYSIBM.IPNAMES and SYSIBM.LUNAMES table, TCP/IP is the only protocol that will be used to connect to LU1 from this requester. (TCP/IP is never chosen for access using DB2 private protocol, so it is possible to switch between SNA and TCP/IP.)

SNA Alternatives

There are two methods that use SNA protocols. A given DB2 subsystem can support both setups at the same time, but only one can be used for a given partner. Use member routing whenever possible.

Using A Single Location Name with a List of LU Names (Member Routing)

A requester can define a server location name that is associated with many LU names. Unlike with the generic LU name, a requester can establish sessions with one or more subsystems in the group.

Requirements: The following requesters can use member routing:

- DB2 for MVS/ESA Version 4 or a subsequent release
- DB2 Connect Version 5.1 Enterprise Edition (In DB2 Connect, this support is called *Sysplex workload balancing*.)

  To enable DB2 Connect EE for Sysplex workload routing, enter the SYSPLEX parameter in the correct position in the DCS directory. See the README file shipped with the enabled version of DB2 Connect for more information.

For member routing, the group must have workload management support that is available in MVS/ESA Version 5 Release 2 or a later release. You can be running in goal mode or compatibility mode; incoming requests are distributed to members of the data sharing group based on their capacity

Using a Generic LU Name for the Data Sharing Group (Group-generic)

If you have VTAM Version 4 Release 2 or a subsequent release on the server, you can use VTAM's support for generic resources and have a generic 8-character name to represent a group of VTAM LU names. For ease of migration, we recommend choosing the LU name of the originating member of the data sharing group as the generic LU name. For example, generic name LUDB2A might represent three DB2 subsystems in the group whose real LU names are LUDB2AR, LUDB2B, and LUDB2C. (In this example, the "real" LU name for DB2A is renamed to LUDB2AR.)

Requesters outside the DB2 group set up their communications directories to refer to DB2 by a generic name, and VTAM selects the real DB2 LU name to be used by the requester. VTAM makes this choice based on the number of active DDF sessions or the result of a user-written VTAM or MVS workload manager exit routine.

After a requester is associated with a particular LU in the data sharing group, all future requests from that requester are sent to the same member of the data sharing group until all connections between the two LUs are terminated. In addition, if the connection between the client and server is enabled for two-phase commit processing, the mapping between the client and data sharing LUs are preserved until you issue the DB2 command RESET GENERICLU.

Comparing the SNA Alternatives

Table 13 describes the differences between using the group-generic setup and member routing. "Communicating with a Release of DB2 before Version 4" on page 129 describes how communication works if you do not implement one of these setups, or if you are communicating with an earlier release of DB2.

Table 13 (Page 1 of 2). Comparing Use of Group-Generic and Member-Routing Setups

Consideration	Group-generic	Member-routing
A requester sends SQL to a DB2 group	For any requesting LU, all SNA sessions must be directed to one and only one DB2 subsystem in the group. If the requester supports multiple concurrent threads, all threads must be directed to the same member of the DB2 group.	Each requester can establish sessions with one or more DB2 subsystems in the group. If the requester supports multiple concurrent threads, the threads need not be directed to a single member of the DB2 group.
DB2 group member sends an SQL request to another server.	Only one of the DB2 subsystems in the group is known to the remote server by the generic LU name at any given point in time. Other members of the DB2 group can send SQL requests to the remote server, but these members of the DB2 group must be known by their real VTAM LU names rather than the generic name. (For a DB2 remote server, a blank LUNAME column in SYSIBM.LUNAMES can serve this purpose.)	Any member of the DB2 group can issue SQL requests to the target server.

Table 13 (Page 2 of 2). Comparing Use of Group-Generic and Member-Routing Setups

Consideration	Group-generic	Member-routing
DB2 server workload balancing	All SQL requests from a given LU are directed to only one DB2 server in the group. That DB2 server is chosen when the first VTAM session is established. After all sessions are terminated, VTAM can choose a different DB2 server for future communications, unless the sessions require two-phase commit. If sessions require two-phase commit, the same DB2 server must be used.	SQL requests can be sent to different DB2 servers in the group. The DB2 server informs the requester of the desired LU to service future SQL threads. DB2 uses special support from MVS's workload management (only available in MVS/ESA Version 5 Release 2 or subsequent releases) to determine which DB2 subsystems have the greatest available capacity.
Reconnection after a communications failure	If the session uses two-phase commit, VTAM must always connect the requester to the DB2 LU chosen for the first session established by VTAM. If that DB2 LU is unavailable, a communication error is returned. If the session uses single-phase commit, VTAM can connect the requester to any DB2 LU in the group.	The requester can connect to any LU in the group. If resynchronization must be performed due to previous communication or system failures, the resynchronization request is routed to the DB2 LU involved in the failure. This activity is performed asynchronously, and does not prevent new sessions from being established with other members of the DB2 group.
Software requirements for partner	Existing DRDA products are sufficient.	DB2 for MVS/ESA Version 4 or DB2 Connect EE Version 5.1, or subsequent releases of either of those products.
Network definitions at partner site	For remote requesters, only the generic LU name must be defined. For remote servers, the generic LU name and the real LU names for each member of the group must be defined. See “Definitions Required for the Group-Generic Setup” on page 137 for more information.	At the remote server, no special definitions are required. At the remote requester, definitions must be created for each LU name in the DB2 group. See “Definitions Required for Member Routing” on page 134 for more information.

Recommendation: Use member routing unless your requester does not support it. It provides better workload balancing and two-phase commit.

Planning for DB2 Restart on Another MVS

If you are running with a release of VTAM before Version 4 Release 3, the APPL definition statements are not shared among members of the sysplex. This means if one system fails and you want to restart DB2 on another system in the sysplex, you must deactivate the APPL definition on the failed system and reactivate on the new system.

With VTAM Version 4 Release 3 and subsequent releases you can use a model application program definition for DB2. With a model definition, you use wild card characters for the application name (LU name). For example, you might have a model definition that looks like this:

```
LUDB2*  APPL   APPC=YES, ATNLOSS=ALL, ...
```

If you have a model definition like that shown above, and if one of your DB2s, say LUDB2A, fails on MVS1 and is restarted on MVS2, VTAM can use the above model definition for LUDB2A. For more information about using model application program definitions, see *VTAM for MVS/ESA Network Implementation Guide*.

Communicating with a Release of DB2 before Version 4

A release of DB2 before Version 4 can communicate with a data sharing group, but not with the flexibility offered in Version 4 and subsequent releases. (There is no TCP/IP access to the group from any release before Version 5.) In releases before Version 4, administrative overhead is increased and there is no dynamic workload balancing.

As you can see from Table 14, one of the major restrictions in having a data sharing group request data from a DB2 server before Version 4 is that only one member of the data sharing group can communicate with that server. All requests for the server have to be routed to the member of the data sharing group that is communicating with the server of the early release.

Table 14. DDF Behavior when Communicating with Releases of DB2 before Version 4

Consideration	DDF Behavior
A requester sends SQL to the DB2 group	Each requester must be directed to one of the available DB2 LU names to gain access to the DB2 server. If the requester supports multiple concurrent threads, all threads must be directed to the same member of the DB2 group. If the chosen LU name is not available, a communication failure is returned.
A DB2 group member sends SQL to another server	If the server is a DB2 subsystem, only one LU in the DB2 group is allowed to send SQL to the DB2 server. If two or more DB2 subsystems in the group require access to the remote data, the installation must have some mechanism for ensuring that all applications requiring access run on a single member of the DB2 group. If the chosen LU name is not available, a communication failure is returned. If the server is not a DB2 subsystem, each DB2 system in the group can send SQL statements to the server.
DB2 server workload balancing	Because DDF is not using any special support for data sharing, you are responsible for balancing the data sharing group workload. This is achieved by assigning some number of requesters to each DB2 server in the group. No dynamic workload balancing can occur.
Reconnection considerations after a communication failure	The requester is statically assigned to a single member of the DB2 group. If that member of the DB2 group is unavailable, a communication error is returned.
Network definitions at partner site	Only a single DB2 LU name must be defined at the requester. You are responsible for determining which LU name should be used at each requester to achieve acceptable workload balancing.

DRDA Access through TCP/IP

A requester can define a server location name that is associated with many IP addresses; it is simpler to associate a server location name with a domain name rather than the many IP addresses for the systems in the Sysplex. The domain name is case-sensitive. To allow a requester to connect to any member in the group, each member uses the same DRDA port number; 446 is the port number we recommend. Only one member on a particular CPC can claim that DRDA port number for the purpose of serving incoming DRDA requests.

For resynchronization, however, each member has its own unique resync port number, which allows that member to handle outstanding resync requests, even if the member is started on another CPC.

Requirements

Requesters must support DRDA level 3. To be eligible to service TCP/IP requests, the members of the DB2 data sharing group must run on OS/390 Release 3 or subsequent releases.

Planning for Availability

Using Virtual IP Addresses: We recommend using a virtual IP address (VIPA) on each CPC to minimize the impact of a network controller failure. If you have more than one network controller on a single CPC, and if you route users to a particular controller's address, that becomes a single point of failure for the connection.

If you use the VIPA, end users can connect to the VIPA instead of the IP address associated with any single network controller. If a network controller fails, TCP/IP can automatically route the user's data to a different path.

To have DB2 send the virtual IP address to DRDA clients, you can use two methods:

- Specify the VIPA as the first entry for the TCP/IP HOME statement. For example:

```
HOME
  12.23.34.45  VIPA1
  12.23.34.46  CTC1
  12.23.34.47  T3172A
  12.23.34.48  T3172B
```

- Specify the VIPA on the TCP/IP PRIMARYINTERFACE statement. This overrides the ordering specified on the HOME statement. For example:

```
PRIMARYINTERFACE VIPA1
```

Planning for Resynchronization after Failure

All DB2 members must have the same DRDA port number for incoming DRDA SQL requests. If DB2 is automatically restarted on another CPC, the TCP/IP on that system must be able to allow that DB2 to use the DRDA port. The best way to do this is to assign the DRDA port to every member that could conceivably start on a particular system. On each system, you can replicate the TCP/IP PORT statement shown here. By explicitly assigning these port numbers, you can prevent some other program from using DB2's port number.

```
PORT
⋮
  446 TCP OMVS
```

In OS/390 Release 4, you can specify the PORT statement as follows, which reserves the DRDA port just for the DB2 DDF address space:

```
PORT
⋮
  446 TCP DB1GDIST
  446 TCP DB2GDIST
  446 TCP DB3GDIST
  446 TCP DB4GDIST
```

Only one of the DB2 members associated with a given DRDA port number is selected by TCP/IP to process the incoming requests. The other member can fulfill any resynchronization responsibilities it has using its unique resync port number. However, don't forget to register each member in the domain name server as described in “Registering Names in the Domain Name Server” on page 142 .

The chosen DB2 member receives all of the DRDA server's workload for that TCP/IP instance, which leaves the other DB2 members with no TCP/IP server threads for DRDA. This is transparent to the DRDA clients if the DB2 member processing the TCP/IP requests doesn't reach the MAX REMOTE CONNECTED thread limit. If the MAX REMOTE CONNECTED limit is reached, the connection request of the DRDA client is rejected.

After you resolve a failure situation, it is a good idea to move the DB2 member to its original IP address.

TCP/IP Workload Balancing for DB2 Data Sharing

Consider the following methods for balancing the workload of DRDA clients connecting to a DB2 data sharing group:

- Enter the IP address of the desired DB2 member in each client's TCP/IP configuration. This approach has some drawbacks:
  - Balancing client workloads in this fashion is very labor-intensive.
  - The DB2 member's IP address changes if it is restarted on another CPC, so this approach doesn't work if your automatic restart configuration restarts DB2 on another CPC.
  - It doesn't adjust to changes in the work load.
- Use the domain name server (DNS) to balance the workload.

  This approach, sometimes called *connection optimization for a Sysplex*, is recommended for single-user workstation connections, such as DB2 Connect Personal Edition Version 5.1 or subsequent releases. It's also useful for directing the first connection through a gateway.

  The steps for using the DNS to balance the workload are, in summary:

  1. Make sure that all members of the data sharing group are running in workload manager goal mode.
  2. Set up your DNS with a bootfile entry for the Sysplex that includes the cluster keyword. The high level qualifier for the bootfile entry must match the Sysplex name, which you can obtain from the MVS COUPLE*xx* data set. An example bootfile entry is:

```
;  /etc/named.boot
;
;       boot file for name server
;
;
; type        domain                 source file or host
;
directory     /etc/dnsdata
primary       sysplex1.ibm.com       named.wlm.for     cluster
primary       183.65.121.in-addr.arpa named.rev
primary       0.0.127.in-addr.arpa   named.lbk
cache         .                      named.ca
options query-log
```

For more information about setting up the DNS to handle Sysplex domain workload balancing, see *OS/390 eNetwork Communications Server: IP Configuration*.

3. WLM registers the DB2 group and server domain names to the domain name server. In OS/390 Release 4 (with APAR PQ06340), and subsequent releases, WLM provides information to the domain name server that identifies which IP address has the most capacity. For more information, see “Registering Names in the Domain Name Server” on page 142.

- Take advantage of the Sysplex workload balancing information built into the DRDA architecture. If the requester is DB2 Version 5 or DB2 Connect Enterprise Edition Version 5.1, or subsequent releases of those products, it can use this workload balancing information to spread connections across the available members in the data sharing group.

  To enable DB2 Connect EE for Sysplex workload routing, enter the SYSPLEX parameter in the correct position in the DCS directory. See the README file shipped with the enabled version of DB2 Connect for more information.

  Other DRDA application requesters might also provide this capability; refer to the documentation associated with your DRDA requester product to determine whether this approach can be used. If you cannot use DRDA workload balancing, use the domain name server to balance the workload.

Recommendation: For the highest availability and workload balancing, use DNS workload balancing and DRDA Sysplex routing together where possible. Because the Sysplex routing information is returned only after the first connection is made to the host, the DNS can be used to route that first connection to an available member with the most capacity. After that first connection is made, the client can use DRDA Sysplex routing to choose where subsequent connections are made.

Defining a DB2 Data Sharing Group in an SNA Network

When you have decided whether to use group-generic or member routing for a particular partner, you must specify communications definitions for the group and for any partner LUs. To illustrate, assume a data sharing group whose original member is already defined to the partner LUs shown in Figure 28. The data sharing group has a location name of DB2A (it is using the original member's location name so that existing applications do not have to be changed). As described in "Communicating with a Release of DB2 before Version 4" on page 129, requesters of DB2A's data can access its data only through the DB2 defined as LUDB2A. If LUDB2A is unable to service requests, the requesters have to change their communication definitions to connect to a different member of the data sharing group.

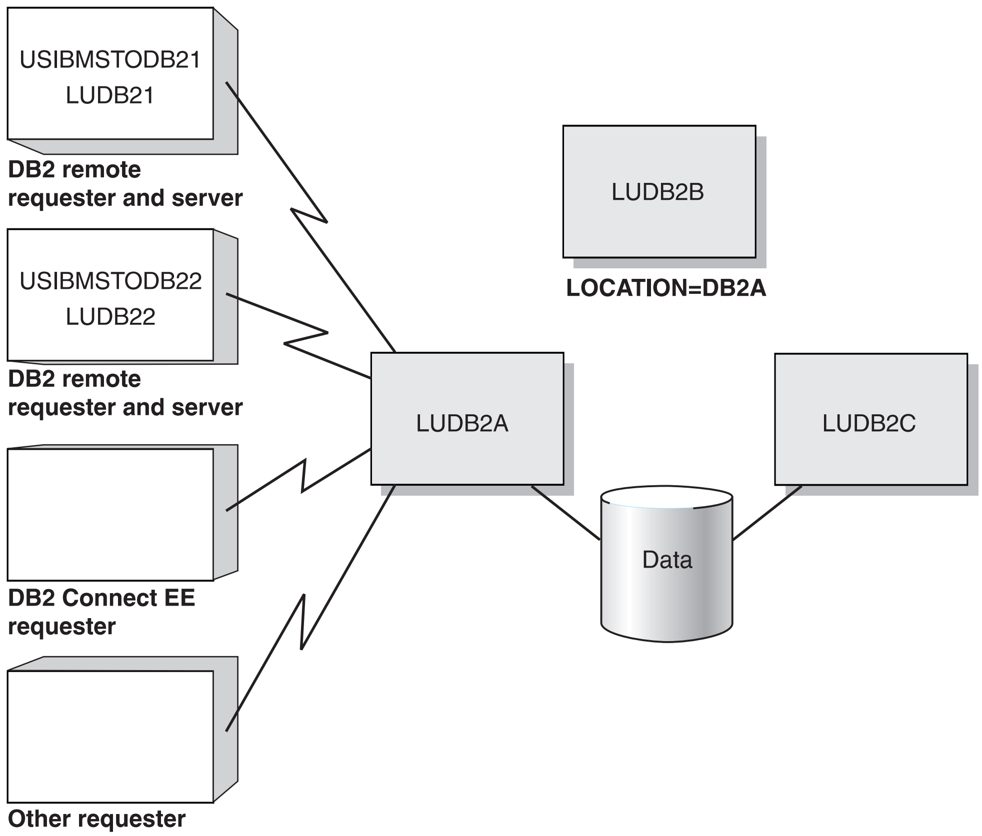

Figure 28. Example Configuration before Distributed Group Support Is Enabled. Access is limited to a single member of the data sharing group.

Example Configuration with Enhanced Distributed Support

Now assume that you want the configuration shown in Figure 29. In that configuration, both remote DB2s and DB2 Connect EE are implementing member routing; the other DRDA requester is using group-generic processing because it does not have support for member routing.

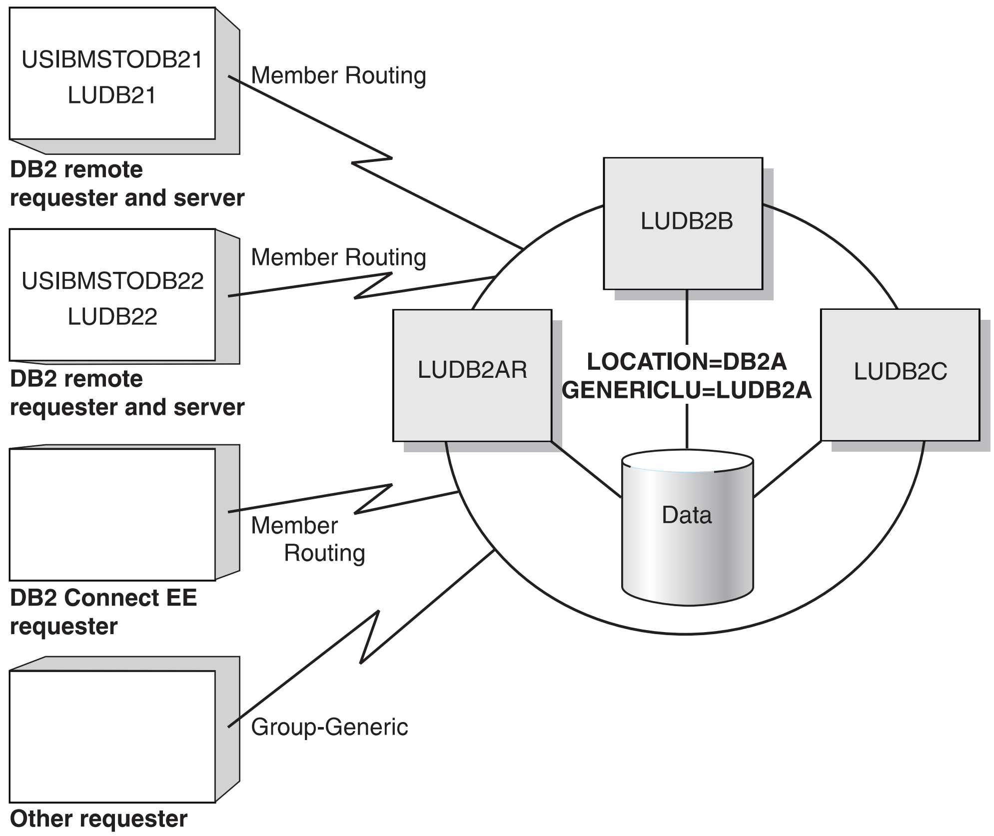

Figure 29. Configuration after Enabling Group-Generic or Member Routing Support. All members of the group can potentially service requests.

Definitions Required for Member Routing

This section describes how to update the CDBs of DB2 requesters to use member routing as shown in Figure 30.

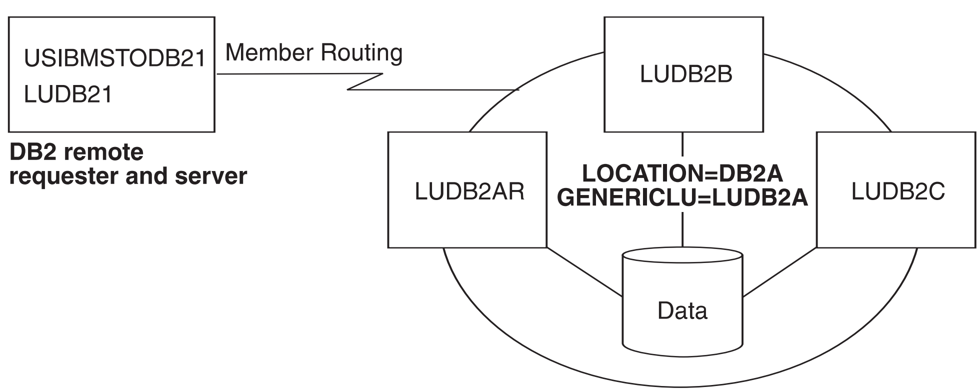

Figure 30. DB2s Using Member Routing.

Communications Database for a DB2 Requester

To indicate that you want DB2 to use a list of LU names for a data sharing group, you must use a token, which we call the link name, that corresponds a list of LUs in the SYSIBM.LULIST table with a single LU definition in the SYSIBM.LUNAMES table.

A DB2 Requester's SYSIBM.LUNAMES Table: Assume that you choose a link name of LUGROUP. You must enter that name in the SYSIBM.LUNAMES table of the requester. The real LU names must not be included in SYSIBM.LUNAMES. To enable member routing, USIBMSTODB21's SYSIBM.LUNAMES table must look like Table 15. (The blank row is optional, but recommended.)

Table 15. A DB2 Requester's SYSIBM.LUNAMES Table to Enable DDF Support

LUNAME	...	GENERIC
LUGROUP		
(blank)		

A DB2 Requester's SYSIBM.LOCATIONS Table: Now, you must map the data sharing location name to the link name. USIBMSTODB21's SYSIBM.LOCATIONS table looks like Table 16.

Table 16. A DB2 Requester's LOCATIONS Table for Member Routing Setup. Not all columns are shown.

LOCATION	LINKNAME	TPN
DB2A	LUGROUP	

Use the following SQL statements to populate SYSIBM.LUNAMES and SYSIBM.LOCATIONS:

General-use Programming Interface

1. This statement inserts the link name (LUGROUP) into the SYSIBM.LUNAMES table:

```
INSERT INTO SYSIBM.LUNAMES (LUNAME)
   VALUES ('LUGROUP');
```

2. The following statement updates the LINKNAME column in SYSIBM.LOCATIONS to point to LUGROUP:

```
UPDATE SYSIBM.LOCATIONS
  SET LINKNAME='LUGROUP'
  WHERE LOCATION='DB2A';
```

3. The following statement must be used to delete the existing LU name from the SYSIBM.LUNAMES table. You enter the real LU names for the DB2 members in the SYSIBM.LULIST table in a later step.

```
DELETE FROM SYSIBM.LUNAMES
  WHERE LUNAME='LUDB2A';
```

End of General-use Programming Interface

A DB2 Requester's SYSIBM.LULIST Table: When DB2 acts as a requester, DB2 makes use of a communications database table called SYSIBM.LULIST. This table makes it possible for you to specify more than one LUNAME value for any given server. Populate this table only if DB2 is acting as a requester of data that resides in a data sharing group. You can specify a subset of members in list; those members that are not included in the list are not considered as servers for member routing.

The table is used by a DB2 requester as follows:

- When access is requested to a particular remote location, the SYSIBM.LOCATIONS table is searched to find a matching row. The LINKNAME column of that row identifies the corresponding rows in the SYSIBM.LUNAMES table.
- If one or more rows exist in the SYSIBM.LULIST table for the specified LINKNAME, the LUNAME values of the SYSIBM.LULIST rows represent the available network destinations (LUNAMEs) for the LOCATION. The values of the columns in the SYSIBM.LUNAMES row (SECURITY_IN, SECURITY_OUT, USERNAMES, and so on) apply to each of these destinations.

  Similarly, the rows (if any) in the SYSIBM.USERNAMES and SYSIBM.MODESELECT tables apply to each of the destinations. A blank row in the requester's SYSIBM.LUNAMES table lets DB2 use without error whatever list of LUs the data sharing group server returns to the requester when the requester makes a request to the data sharing group.
- If no matching row exists in SYSIBM.LULIST, the LUNAME column of the SYSIBM.LUNAMES table provides the VTAM LUNAME of the remote LOCATION.

SYSIBM.LULIST has the following columns:

LINKNAME CHAR(8)
The value of the LINKNAME column of SYSIBM.LOCATIONS with which this row is associated. This is also a value in column LUNAME for some row in table SYSIBM.LUNAMES. The values of the other columns of that row of LUNAMES (SECURITY_IN, SECURITY_OUT, USERNAMES, and so on) apply to the LU identified in column LUNAME of this row of SYSIBM.LULIST.

LUNAME CHAR(8)
The VTAM LU name of the remote database system. This LUNAME must not exist in the LUNAME column of SYSIBM.LUNAMES.

A DB2 requester, such as USIBMSTODB21, wants a SYSIBM.LULIST table that looks like Table 17 on page 137.

Table 17. A DB2 Requester's SYSIBM.LULIST Table

LINKNAME	LUNAME
LUGROUP	LUDB2AR
LUGROUP	LUDB2B
LUGROUP	LUDB2C

General-use Programming Interface

Use the following SQL statements to associate the LUNAMES of the all members of the group (LUDB2AR, LUDB2B, and LUDB2C) with DB2A (the location name of the data sharing group) by creating a new LUNAME (LUGROUP) that is composed of a list of the real LUNAMES.

```
INSERT INTO SYSIBM.LULIST
   VALUES ('LUGROUP', 'LUDB2AR');

INSERT INTO SYSIBM.LULIST
   VALUES ('LUGROUP', 'LUDB2B');

INSERT INTO SYSIBM.LULIST
   VALUES ('LUGROUP', 'LUDB2C');
```

End of General-use Programming Interface

When Updates to SYSIBM.LULIST Take Effect: Changes to SYSIBM.LULIST take effect as follows

- If DDF has not yet tried to communicate with a particular location, rows added to LULIST take effect when DDF attempts to communicate with the remote location.
- If DDF has already attempted communication with a particular location, rows added to LULIST take effect the next time DDF is started.

Definitions Required for the Group-Generic Setup

This section describes how to set up the data sharing group to service requests from a DRDA requester, as shown in Figure 31 on page 138. “DB2 for OS/390 Remote Requester and Remote Server Definitions” on page 140 describes how you would set up a DB2 for OS/390 to request and serve using the group-generic setup, but in most cases, you will want to use member routing when both partners are DB2 for OS/390 (or MVS/ESA).

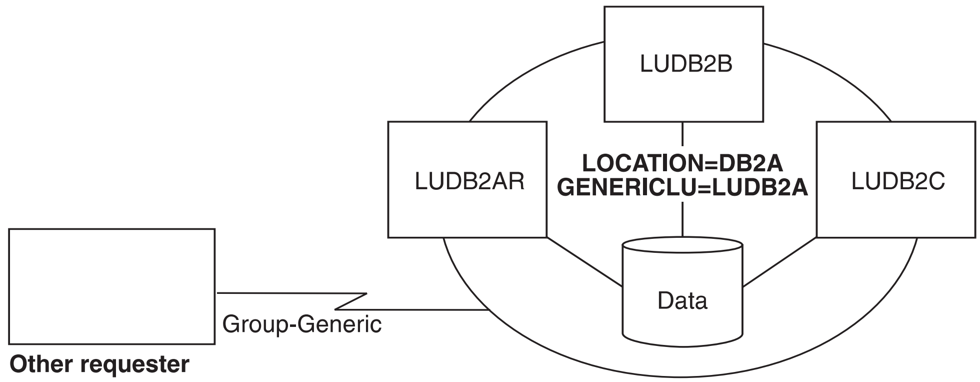

Figure 31. DRDA Requester using Group-generic Processing.

To set up the group for group-generic processing, specify the generic LU name on an installation panel for each member of the group. If the group is to be used as a server (the most likely configuration) then you must specify Y in column GENERIC for the data sharing group's SYSIBM.LUNAMES table entry for requesters that use the generic name to connect to the group. (A blank LUNAME will do fine for this purpose.)

You must also include information in the coupling facility for support of VTAM generic resources (the ISTGENERIC structure). For more information about using VTAM's generic resources, see *VTAM for MVS/ESA Network Implementation Guide*. To calculate storage for that structure, you also need information from *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide*.

Specifying the Generic LU Name for the Group at Installation Time

To specify the 1- to 8-character generic LU name for the data sharing group, use the DB2 GENERIC NAME field on the DSNTIPR installation panel.

To avoid extensive changes to the communications directories of requesting systems, **we recommend that you choose the existing LU name of the originating member of the data sharing group**. Before starting communications, you must change the real LU name of this originating member by changing the value in the VTAM APPL statement and the LUNAME value in the BSDS. In our sample configuration, the originating member of the group has changed its LU name from LUDB2A to LUDB2AR.

Each member of a data sharing group must choose the same name. The generic LU name is stored in each member's BSDS.

If this DB2 is not a member of the data sharing group but is running as part of an MVS Sysplex, you can still choose a generic LU name for the DB2. This might be useful, for example, during a transition when network names are being changed.

Updating the Group's SYSIBM.LUNAMES Table

Use the GENERIC column of the SYSIBM.LUNAMES communications database table to specify whether the VTAM generic LU name or the DB2 real LU name is used by DB2 to identify itself to a given remote server. **This column is used only when DB2 is initiating contact with a given partner LU.**

This column does not determine whether you use group-generic or member-specific access for a partner. The rows, or lack of rows, in SYSIBM.LULIST determine whether group-generic or member routing access is used for a particular partner.

The description of the GENERIC column is as follows:

GENERIC CHAR(1)
The values have the following meanings:

N or blank The real VTAM LU name of the DB2 subsystem is used for CNOS operations and SQL requests to the partner LU identified in this row.

Y The VTAM generic LU name is used for CNOS operations and SQL requests to the partner LU when DB2 initiates processing. The partner must be able to recognize the generic LU name. If one member of the data sharing group is already using the generic LU name for a given partner, another member of the group cannot, and the real LU name is used instead. This means if the data sharing group is requesting data from a system that only accepts generic LU names, all requests must be routed through one member of the data sharing group.

If the partner starts CNOS processing first, VTAM uses the name with which the partner connected, whether the real LU name or the generic LU name. Because the behavior is not always predictable, it is best if the system that is servicing requests from the data sharing group can accept either the generic LU name or the real LU name when group-generic processing is used.

The GENERIC column is ignored if the field DB2 GENERIC NAME, on installation panel DSNTIPR, is blank.

DB2A wants a SYSIBM.LUNAMES table that looks like Table 18. A value of 'Y' is inserted into the GENERIC column for all partners that use a generic LU name to communicate with the data sharing group.

Table 18. The Data Sharing Group's SYSIBM.LUNAMES Table for Group-Generic

LUNAME	...	GENERIC
(blanks)		Y

General-use Programming Interface

The following statement can be used to update the SYSIBM.LUNAMES table of the data sharing group:

```
UPDATE SYSIBM.LUNAMES
   SET GENERIC='Y'
   WHERE LUNAME=' ';
```

End of General-use Programming Interface

DB2 for OS/390 Remote Requester and Remote Server Definitions

Although we recommend using member routing when both partners are DB2 for OS/390, this section is included in case you must use group-generic processing.

For a DB2 **requester,** use the generic LU name to access a DB2 data sharing group. Place the generic LU name of the DB2 group in the LUNAME and LINKNAME columns of the CDB entries associated with the DB2 group's location. If you use the original member's LU name as the generic LU name, then no changes are necessary.

For a remote DB2 **server**, include the generic LU name but also include all the LU names of all members of the requesting DB2 data sharing group. You can use the default row (blanks for LUNAME) in the SYSIBM.LUNAMES table as we have shown here:

Table 19. A Remote DB2 Server's SYSIBM.LUNAMES Table

LUNAME	...	GENERIC
(blank)		

Switching from Group-generic to Member Routing

If you are currently using group-generic for your connections between DB2 for OS/390 (or MVS/ESA) partners, and you want to switch to member routing because of the additional benefits it brings, you must do the following procedure. This procedure is necessary because of two-phase commit connections that have a real LU name or generic LU name registered in the coupling facility. To switch which name gets used, you must unregister the name.

1. Shut down the network connections between DB2 and the partner system that you are unregistering from the coupling facility. Some ways to do this include using the VTAM command VARY NET,INACT,ID=*luname*, or entering the DB2 command STOP DDF.

2. From an active member of the data sharing group, enter the command RESET GENERICLU *luname*.

3. Modify the requester system's CDB to ensure that member routing is used. You do this by populating the SYSIBM.LULIST table, as described in “Definitions Required for Member Routing” on page 134. Make sure the requesting DB2's GENERIC column of SYSIBM.LUNAMES contains N or blank for the row to this data sharing group server.

4. Enable network connections.

Defining a DB2 Data Sharing Group in a TCP/IP Network

This section gives an overview of one possible way to configure a data sharing group to operate in a TCP/IP network. See your network administrator and the information in Section 2 of *Installation Guide* to work out the details for your specific configuration.

Example TCP/IP Configuration

The configuration in Figure 32 shows two DB2 for OS/390 requesters and one other DRDA requester. When a requesting application uses an SQL statement that specifies DB2A as the location name, the requester uses the LINKNAME column of SYSIBM.LOCATIONS to find out whether or not it is using TCP/IP to connect to the server known as DB2A.

Because the LINKNAME value appears in the SYSIBM.IPNAMES table, DB2 knows that it is using TCP/IP to send the DRDA request to the server. (The requester is using a domain name to represent the server. That way, only the domain name server must be updated with specific IP addresses.) The application is directed to the member of the group that is best able to service the request based on its current work load. For more information about workload balancing, see "TCP/IP Workload Balancing for DB2 Data Sharing" on page 131.

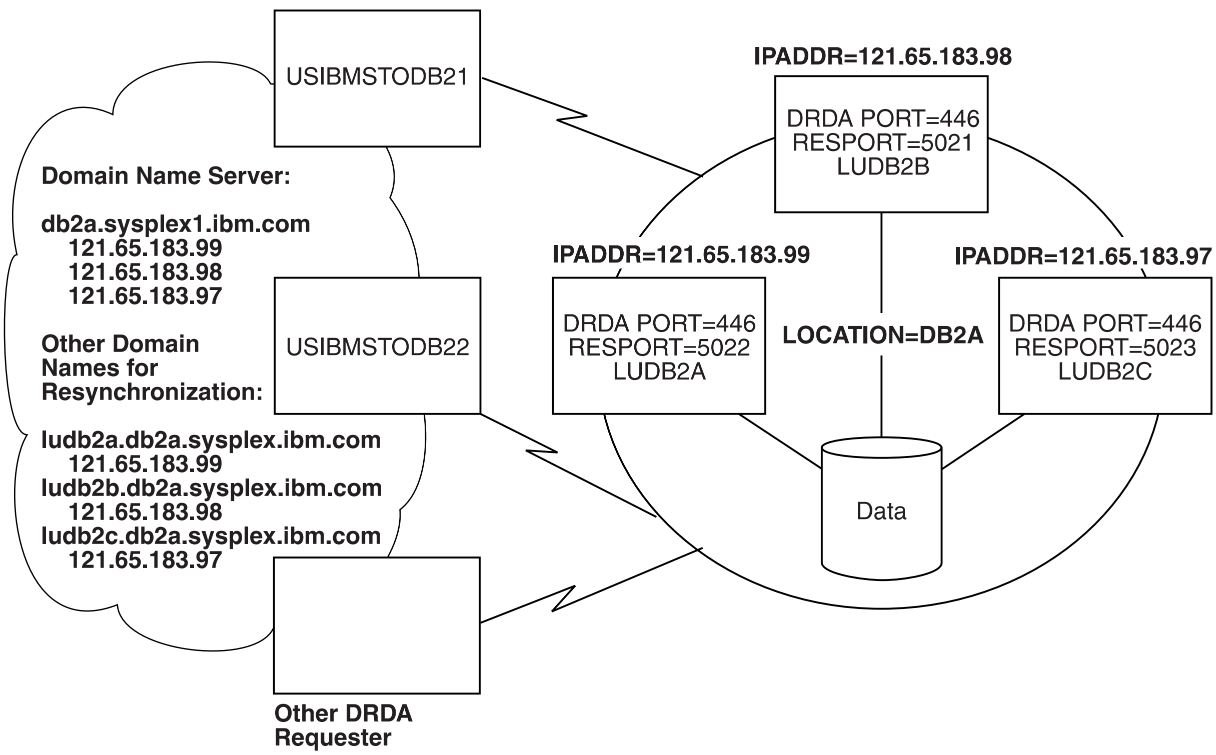

Figure 32. Example TCP/IP Network Configuration. Applications use the location name to direct requests to the group. The network addresses of the various members of the group are provided in the requester communications definitions. Use a domain name to simplify this process.

Registering Names in the Domain Name Server

To take advantage of workload balancing and correct resynchronization for two-phase commit processing if a DB2 is restarted on another MVS, you must register DB2 names in the domain name server (DNS). (In OS/390 Release 4 with APAR PQ06340, workload manager registers these names for you automatically when you set up the DNS for connection optimization in a Sysplex domain, as described in *OS/390 eNetwork Communications Server: IP Configuration*.)Register the following names:

- location.sysplex.domainname

  This is the group name that represents the entire data sharing group. It contains the DB2 location name and the sysplex name (from the COUPLE*xx* data set, among other places). The domain name comes from the socket calls gethostid for the host address, and gethostbyaddr for the host name. DB2 removes the high level qualifier returned by the gethostbyaddr call and replaces it with the location name and sysplex name. You can verify that you have the correct name to register in the DNS by starting DDF. The name you must register appears in the 'domain name' field of the DSNL004I message.

- luname.location.sysplex.domainname

  This is the server name. Register one of these names for each member of the data sharing group. It includes the real LU name for the member. This name is used by DB2 to resolve indoubt threads.

Server Definitions

Defining a data sharing group as a TCP/IP server does not require any CDB definitions.

Specify a Generic LU Name if You Use RACF PassTickets

If you use RACF PassTickets for security, define a generic LU name for the data sharing group by using the DB2 GENERIC NAME field of installation panel DSNTIPR, or use the change log inventory utility to do that. The generic LU name is used to generate the RACF PassTicket application names. See Section 3 (Volume 1) of *Administration Guide* for more information about using RACF PassTickets.

Specify the Port Numbers for the Group

Use installation panel DSNTIP5 or the change log inventory utility to specify a DRDA port number on each member that accepts TCP/IP requests. Specify the same DRDA port number on all members of the group.

To resynchronize two-phase commit units of work, specify a unique RESYNC port number for each member of the group. (RESYNC port numbers must be unique within the Sysplex.) See *Utility Guide and Reference* for more information about the change log inventory utility.

The network administrator must register the port numbers on the TCP/IP that is associated with each member's MVS system.

Remote Requester Definitions

This section describes the CDB entries and the BSDS entries needed at the requesting DB2 for OS/390.

Remote Requester CDB Entries

To access the DB2A data sharing group server, the DB2 for OS/390 requester has a SYSIBM.LOCATIONS table that looks like Table 20. The example specifies a generic LU name for the link name, because that generic LU name is used to generate the application name for the RACF PassTicket.

If you do not use RACF PassTickets, and you have no SNA communication at all, you can use any name as the link name; the link name relates rows in SYSIBM.LOCATIONS to rows in SYSIBM.IPNAMES.

Table 20. USIBMSTODB21's LOCATIONS Table. Not all columns are shown.

LOCATION	LINKNAME	PORT
db2a	GENERICLU	446

Use the following SQL statement to populate SYSIBM.LOCATIONS.

General-use Programming Interface

```
INSERT INTO SYSIBM.LOCATIONS
            (LOCATION, LINKNAME, PORT)
      VALUES ('db2a', 'GENERICLU', '446');
```

End of General-use Programming Interface

If you prefer, you can use a case-sensitive *service name* instead of a hard-coded port number in the PORT column. This service name is another way to refer to a port number. For more information, see *IBM TCP/IP for MVS: Customization & Administration Guide*.

The DB2 for OS/390 requester has a SYSIBM.IPNAMES table that looks like Table 21.

Table 21. SYSIBM.IPNAMES at the DB2 for OS/390 Requester. Not all columns are shown.

LINKNAME	SECURITY_OUT	USERNAMES	IPADDRESS
GENERICLU	R	blank	db2a.sysplex1.ibm.com

Use the following SQL statement to populate SYSIBM.IPNAMES:

General-use Programming Interface

```
INSERT INTO SYSIBM.IPNAMES
            (LINKNAME, SECURITY_OUT, USERNAMES, IPADDR)
      VALUES ('GENERICLU', 'R', ' ', 'db2a.sysplex1.ibm.com');
```

End of General-use Programming Interface

Bootstrap Data Set Entries

Use installation panel DSNTIP5 or the change log inventory utility to enter a DRDA port and resynchronization port into the BSDS. If you do not enter values for these ports, you cannot use TCP/IP.

The Data Sharing Group as a Requester

To use a data sharing group as a TCP/IP requester, you need no special configuration other than that shown in “Remote Requester Definitions” on page 143. If you use RACF PassTickets, you must use the name (either generic LU name or real LU name) that is defined at that server. Be aware, however, that only one member of the data sharing group can use the generic LU name as a requester.

Excluding a Member from Processing Remote Requests

Transparently to end users, you can exclude one or more members from DDF server processing while still letting it make DDF requests. To exclude a member from DDF server processing, set the MAX REMOTE ACTIVE option of installation panel DSNTIPE to zero for that member. The effects of setting this parameter to zero are:

- DB2 does not register the member's LU name with the VTAM generic LU name during DDF startup. Connections that use the generic LU name are directed to those members for which MAX REMOTE ACTIVE is greater than zero.
- DB2 does not register the member to MVS workload manager for member routing. The member can continue to use workload manager for setting priorities on work, but its name is not included in the list of available LUs that is sent to remote sites. Therefore, DDF server work is never routed to that member.
- DB2 does not listen on the DRDA SQL port, which means that SQL TCP/IP requests are accepted only by members for which MAX REMOTE ACTIVE is greater than zero.

Using the Change Log Inventory Utility to Update the BSDS

Use the change log inventory utility (DSNJU003) to update the following information in the BSDS related to distributed data processing:

- generic LU name
- resynchronization port (for TCP/IP)
- DRDA port (for TCP/IP)

If you change the generic LU name or DRDA port for one member of the data sharing group, you must change it for all members. This requires that you stop and restart DDF to pick up the change.

See Section 2 of *Utility Guide and Reference* for more information about using DSNJU003.

Chapter 6. Operating with Data Sharing

Most data sharing operations are accomplished by means of commands to the licensed program DB2 for OS/390. This chapter describes:

Entering Commands

This section describes the following:

Routing Commands

Operations on an individual member of a data sharing group can be controlled from any MVS console by entering commands prefixed by the appropriate *command prefix*. For example, with the appropriate choice of command prefix, you can start a DB2 statistics trace on member DB1G by entering this command at any console in the Sysplex:

```
-DB1G START TRACE (STAT)
```

This routing of commands requires that the command prefix scope is registered as S or X on the IEFSSN*xx* parmlib member. For specifications of command prefixes, see "Registering Command Prefixes and Group Attachment Name" on page 77.

Operations on certain objects are controlled by commands or command options that affect an entire group. These, also, can be entered from any MVS console. For example, and again with the appropriate choice of command prefixes and assuming DB1G to be active, you can start database XYZ by entering this command at any MVS console of the Sysplex:

```
-DB1G START DATABASE (XYZ)
```

Command Scope

The breadth of a command's impact is called the *scope* of that command.

Many commands used in a data sharing environment affect only the DB2 for which they are issued. For example, a DISPLAY THREAD command displays only those threads that exist for the member identified by the command prefix. Such commands have *member* scope.

Other commands have *group* scope because they affect an object in such a way that affects all members of the group. For example, a STOP DATABASE command, issued from any member of the group, stops that database for all members of the group. See *Command Reference* for information about the scope of each command.

Entering Commands from an Application Program

You can enter commands from an application program attached to a DB2 subsystem through any of the attachment facilities: IMS, CICS, TSO, CAF, and RRSAF. Commands entered in this way are executed by the DB2 subsystem to which the application program is attached. The application cannot send a command to a different DB2 subsystem.

Authorizing Commands

Data sharing does not introduce any new techniques for establishing and checking authorization IDs. Because all members of a data sharing group share the DB2 catalog, any ID has the same privileges and authorities on every member.

It is your responsibility to use the same connection or sign-on exit routines on every member of the data sharing group to avoid authorization anomalies.

Receiving Messages

You can receive messages from all members at a single console. Hence, a message must include a member identifier as well as a message identifier. The member's command prefix appears in messages to identify the source of a message.

Starting and Stopping DB2

To start members of a data sharing group, you must enter a START DB2 command for each subsystem in the group. If this is the first startup of the group, **you must start the originating member first**.

Impact of Command Prefix Scope: If DB2 is installed with a command prefix scope of STARTED (the default and recommended value), you must either issue the command from the MVS system you want to start DB2 on or route the command to that MVS. Here is an example of routing the command to the MVS on which DB1G is to be started:

```
ROUTE MVS1,-DB1G START DB2
```

After DB2 is started, all other commands can be issued from any MVS in the Sysplex, and the commands are routed to the appropriate DB2 subsystem.

Stopping DB2: You stop DB2 members using the STOP DB2 command as described in Chapter 2 of *Command Reference* .

Submitting Work to be Processed

The methods you use to submit work need not change for data sharing. However, you might find it to your advantage to use the group attachment name to direct jobs that are not submitted through CICS or IMS. This section describes more about the group attachment name and how it works.

Running CICS and IMS Applications

There is no change in the process of running CICS and IMS applications for data sharing. You cannot use the group attachment for CICS and IMS applications because those transaction managers must be aware of the particular DB2 subsystem to which they are attached so they can resolve indoubt units of recovery in the event of a failure. See Section 4 (Volume 1) of *Administration Guide* for more information about running applications.

Using the Group Attachment Name

Utilities and applications that use TSO, batch, or the RRSAF to connect to DB2 have two methods for specifying the DB2 to which they want to connect. The first method is to specify the particular subsystem name in the job for TSO and batch, RRSAF, and on an explicit CONNECT for CAF. The second method is to use the *group attachment name* instead of a specific subsystem name.

The group attachment name acts as a generic name for the DB2 subsystems in a data sharing group. It substitutes for the DB2 subsystem name running on the MVS the job was submitted from.

By using the group attachment name, the TSO user or batch job does not have to be sensitive to the particular subsystem, which makes it easier to move jobs around the Sysplex as needed.

The group attachment name is specified at DB2 installation on the DSNTIPK installation panel, which places the name in the IEFSSN*xx* member and in the DSNHDECP load module for the group. The group attachment name appears on the output from the command DISPLAY GROUP.

If you do not explicitly specify a subsystem name or group attachment name, DB2 uses the group attachment name as the intended subsystem. As with any application program, make sure you are accessing the set of DB2 libraries with the correct DSNHDECP programming defaults.

DB2I (DB2 Interactive) can also use the group attachment name.

For more information about submitting applications, see *Application Programming and SQL Guide*.

Submitting Online Utilities

When you submit a utility job, you must specify the name of the DB2 subsystem to which the utility is to attach, or you can specify the group attachment name. For example, the EXEC statement might look like the following:

```
//stepname  EXEC PGM=DSNUTILB,PARM='group-attach-name,[uid],[utproc]'
```

Establishing Affinity: If you don't use the group attachment name, the utility job must run on the MVS system where the specified DB2 subsystem is running. To do that, you must be sure that MVS runs the utility job on the appropriate MVS system. There are several MVS installation-specific ways to make sure this happens. These include:

- For JES2 multi-access spool (MAS) systems, use the following JCL statement:

  ```
  /*JOBPARM SYSAFF=cccc
  ```

Where *cccc* is the JES2 name. You can specify an asterisk (SYSAFF=*) to indicate that the job should run on the system from which it was submitted.

- For JES3 systems, use the following JCL statement:

```
//*MAIN SYSTEM=(main-name)
```

Where *main-name* is the JES3 name.

MVS/ESA JCL Reference describes the above JCL statements. Your installation might have other mechanisms for controlling where batch jobs run, such as by using job classes.

| ***Stopping and Restarting Utilities:*** In a data sharing environment, active utilities can be stopped (with the TERM UTILITY command) only on the DB2 subsystem on which they are active. You can terminate a stopped utility from any active member of the group. If a DB2 fails while a utility is executing, you must restart that DB2 on either the same or another MVS before stopping the utility. For remote site recovery from a disaster at the local site, utilities that were active at the local site can be terminated from any restarted member of the group at the remote site.

| You can restart a utility on any member, but that member must be able to access all required data sets. We recommend that you define all work data sets used by utilities on shared DASD. Use the same utility (UID) to restart the utility. That UID is unique within a data sharing group. However, if a DB2 fails while a utility is executing, you must restart that DB2 on either the same or another MVS before restarting the utility.

| ***Altering Utilities:*** In a data sharing environment, the REORG utility can be altered (with the ALTER UTILITY command) only on the DB2 subsystem on which it is active.

Submitting Stand-alone Utilities

DB2 stand-alone utilities (such as DSN1COPY) run as MVS jobs that have no direct connection to DB2 services. Therefore, a DB2 system has no indication that one of these utilities is running.

In a data sharing environment, if a table space has inter-system R/W interest, then its most recently updated pages might be in the coupling facility and a stand-alone utility might not be running with current data. If it is important that the data is in a consistent state, then you must stop or quiesce the table space. Also, the data must not be in the RECP or GRECP status nor have any logical page list (LPL) entries. Use DISPLAY DATABASE with the RESTRICT option to find out if there are exception statuses for a given table space or index.

Monitoring the Group

This section describes the commands you can use to do the following tasks:

"Obtaining Information about the Group" on page 151
"Obtaining Information about Structures and Policies" on page 152
"Obtaining Information about Group Buffer Pools" on page 154
"Monitoring Databases" on page 154

The section on monitoring databases includes information about the logical page list and how to clear entries from that list. It also includes information about detecting retained locks.

Obtaining Information about the Group

The information under this heading, up to "Obtaining Information about Structures and Policies" on page 152 is General-use Programming Interface and Associated Guidance Information, as defined in "Notices" on page v.

To obtain general information about all members of a particular group, use the DISPLAY GROUP command, shown here:

```
-DB1G DISPLAY GROUP
```

The command can be issued from any active member of the group, and displays the following output:

```
- DSN7100I -DB1G DSN7GCMD
- *** BEGIN DISPLAY OF GROUP(DSNDB0G ) GROUPLEVEL(510)
-                                      GROUP ATTACH NAME(DB0G)
- --------------------------------------------------------------------
- DB2                                    DB2 SYSTEM    IRLM
- MEMBER   ID  SUBSYS CMDPREF   STATUS   LVL NAME      SUBSYS IRLMPROC
- -------- --- ----   --------  -------- --- --------  ----   --------
- DB1G       1 DB1G   -DB1G     ACTIVE   510 MVS1      DJ1G   DB1GIRLM
- DB2G       2 DB2G   -DB2G     ACTIVE   510 MVS2      DJ2G   DB2GIRLM
- DB3G       3 DB3G   -DB3G     ACTIVE   410 MVS3      DJ3G   DB3GIRLM
- DB4G       4 DB4G   -DB4G     FAILED   510 MVS4      DJ4G   DB4GIRLM
- --------------------------------------------------------------------
- SCA   STRUCTURE SIZE:     2560 KB, STATUS= AC,   SCA IN USE:    48 %
- LOCK1 STRUCTURE SIZE:    16384 KB
- NUMBER  LOCK ENTRIES:     4194304
- NUMBER  LIST ENTRIES:       59770, LIST ENTRIES  IN USE:         719
- *** END DISPLAY OF GROUP(DSNDB0G )
- DSN9022I -DB1G DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
```

| *Figure 33. Output of DISPLAY GROUP Command*

Figure 33 shows the following information:

- The DB2 group name and group release level, and the member names and release levels
- | The group attachment name
- The IRLM subsystem names to which members are connected
- The command prefix for each member
- The status of each member (ACTIVE, QUIESCED with or without additional conditions, or FAILED)
- The MVS system names where the member is running, or was last running in cases when the member status is not active
- The procedure names of the connected IRLMs
- The SCA structure size in kilobytes and the percentage currently in use
- Lock structure size and the percentage of the structure size in use. This percentage is based on the maximum percentage used of either the lock table or the modify lock list.

The display also shows the following:

- The maximum number of lock entries possible for the lock table and how many of those lock entries are currently in use. This number is an approximate value.
- The maximum number of modify lock list entries and how many of those list entries are currently in use.

For more information about the lock table and the list of modify locks, see "Avoid False Contention" on page 222.

| ***Parallel Query Information:*** To see the COORDINATOR and ASSISTANT subsystem parameters for all active members of the group, use the DETAIL option of DISPLAY GROUP. See Figure 46 on page 204 for an example.

Obtaining Information about Structures and Policies

Use the MVS command D XCF,STR to display information about coupling facility structures and policy information. For more information about the D XCF,STR command, see *MVS/ESA System Commands*.

Displaying All Structures

The following command displays summary information about all structures:

```
D XCF,STR
```

Output similar to the following is produced:

```
STRNAME          ALLOCATION TIME  STATUS
DSNDB0G_GBP0     02/23/95 08:46:03 ALLOCATED
DSNDB0G_GBP1     02/23/95 08:23:12 ALLOCATED
DSNDB0G_GBP3     02/23/95 08:46:00 ALLOCATED
DSNDB0G_GBP32K      --       --    NOT ALLOCATED
DSNDB0G_GBP4     02/23/95 08:45:42 ALLOCATED
DSNDB0G_GBP5     02/23/95 08:26:26 ALLOCATED
DSNDB0G_GBP6     02/23/95 08:24:05 ALLOCATED
DSNDB0G_GBP7     02/23/95 08:24:21 ALLOCATED
DSNDB0G_GBP8     02/23/95 08:24:47 ALLOCATED
DSNDB0G_LOCK1    02/23/95 08:18:17 ALLOCATED
DSNDB0G_SCA      02/23/95 08:18:01 ALLOCATED
```

Figure 34. Ouput from D XCF,STR Command

Displaying Information about Specific Structures

You can also display more detailed information about specific structures. The following command displays information about the lock structure for group DSNDB0G:

```
D XCF,STR,STRNAME=DSNDB0G_LOCK1
```

Output similar to the following is produced:

```
 STRNAME: DSNDB0G_LOCK1
  STATUS: ALLOCATED
   POLICY SIZE    : 64000 K
   REBUILD PERCENT: N/A
   PREFERENCE LIST: LF01
   EXCLUSION LIST : DSNDB0G_GBP0

  ACTIVE STRUCTURE
  ----------------
   ALLOCATION TIME: 02/23/95 08:18:17
   CFNAME         : LF01
   COUPLING FACILITY: 009674.IBM.00.000000040016
                      PARTITION: 1   CPCID: 00
   ACTUAL SIZE    : 64000 K
   STORAGE INCREMENT SIZE: 256 K
   VERSION        : AAACC81F 863B0C02
   DISPOSITION    : KEEP
   ACCESS TIME    : 0
   MAX CONNECTIONS: 7
   # CONNECTIONS  : 2

  CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
  ---------------- -- -------- -------- -------- ---- ----------------
  DXRDB0G$$DJ1G001 01 00010105 STLABC1  DB1GIRLM 0017 ACTIVE
  DXRDB0G$$DJ2G002 02 000200AC STLABC2  DB2GIRLM 001A ACTIVE
```

Figure 35. Ouput from D XCF,STR,STRNAME Command

States of Connections and Structures

When DB2 allocates its coupling facility structures, it specifies a disposition for the structures and for connections to the structures after a normal or abnormal termination. When you display the structures, then, you can see different states for the connections and structures based on how the disposition is defined. Table 22 summarizes the information that you see after a normal termination:

Table 22. States of Structures and Connections after Normal DB2 Termination

Structure	Connections	Structure
SCA	---	Allocated
Lock	Failed-persistent	Allocated
Note: If a given DB2 member has no retained locks, its failed-persistent connection to the lock structure is removed when it shuts down either normally or abnormally. If this is the last member to shut down, the connection remains in a failed-persistent state.		
Group buffer pools	---	Unallocated
Note: If castout failure occurs during shutdown, group buffer pool connections show as failed-persistent, even though DB2 terminates normally.		

For abnormal terminations, the table remains the same except that group buffer pool connections are 'failed-persistent' and the group buffer pool structure remains allocated.

Obtaining Information about Group Buffer Pools

To obtain information about group buffer pools, you can use the MVS command D XCF,STR as described in “Displaying Information about Specific Structures” on page 152. However, DB2 provides a DISPLAY GROUPBUFFERPOOL command that is useful for displaying statistical information about group buffer pool use.

Depending on the options you choose for the command, the display output contains the following information:

- A list of all connections to the group buffer pools. For duplexed group buffer pools, there is only one set of connections for both instances of the group buffer pool. For example, if there are 3 connections to duplexed structure GBP0, there are just 3 connections, not 6 connections.
- Statistical reports on group buffer pool use, either by a specific member or by the whole group. Some statistical information is also available for the secondary allocation of a duplexed group buffer pool.

See “Using the DISPLAY GROUPBUFFERPOOL Command” on page 256 for more information about the DISPLAY GROUPBUFFERPOOL command.

Monitoring Databases

Data sharing introduces the GRECP and LPL statuses. These statuses can appear on the output from the DISPLAY DATABASE command.

GRECP “Group buffer pool recovery pending.” The group buffer pool failed, and the changes that are recorded in the log must be applied to the page set. When a page set is placed in the GRECP state, DB2 sets the starting point for the merge log scan to the LRSN of the last complete group buffer pool checkpoint.

DB2 automatically recovers GRECP page sets when the group buffer pool is defined with AUTOREC (YES).

LPL “Logical page list.” Some pages were not read from or written to the group buffer pool because of some failure, such as a channel failure between the group buffer pool and the processor. Or perhaps pages could not be read from or written to DASD because of a transient DASD problem.

For page sets or partitions that have LPL or GRECP status, either start the page set or partition using the START DATABASE command with SPACENAM and ACCESS (RW) or (RO), or run the RECOVER utility. For more information about removing LPL status, see “Recovering Pages on the Logical Page List” on page 169.

Obtaining Information about Pages in Error (LPL)

The logical page list (LPL) is a new construct in DB2 that contains a list of pages (or a page range) that could not be read or written for some reason, such as transient disk read and write problems that can be fixed without redefining new disk tracks or volumes.

Specific to data sharing, the LPL also contains pages that could not be read or written for “must-complete” operations, such as a commit or a restart, because of some problem with the coupling facility. For example, pages can be added if there is a channel failure to the coupling facility or DASD or if locks are held by a failed subsystem, thus disallowing access to the desired page.

The LPL is kept in the SCA and is thus accessible to all members of the group.

If an application tries to read data from a page that is on the LPL, it receives a “resource unavailable” SQLCODE. In order to be accessible, pages in the LPL must first have their logged changes applied to the page set.

General-use Programming Interface

To verify the existence of LPL entries, issue the DISPLAY DATABASE command. The LPL option of DISPLAY DATABASE can then be used to see the specific list of pages:

```
-DB1G DIS DB(DSNDB01) SPACENAM(*) LIMIT(*) LPL
```

Output similar to the following is produced:

```
DSNT360I -DB1G
************************************************************
DSNT361I -DB1G *  DISPLAY DATABASE SUMMARY
               *  GLOBAL LPL
DSNT360I -DB1G
************************************************************
DSNT362I -DB1G    DATABASE = DSNDB01  STATUS = RW
               DBD LENGTH = 8000
DSNT397I -DB1G
NAME     TYPE PART STATUS            LPL PAGES
-------- ---- ---- ------------------ ------------------
DBD01    TS        RW,LPL,GRECP      000001,000004,00000C,000010
----                                 000039-00003C
SPT01    TS        RW
SCT02    TS        RW
SYSLGRNG TS        RW
SYSUTILX TS        RW
SYSLGRNX TS        RW,LPL,GRECP      000000-FFFFFF
DSNSCT02 IX        RW
DSNSPT01 IX        RW
DSNSPT02 IX        RW
DSNLUX01 IX        RW
DSNLUX02 IX        RW
DSNLLX01 IX        RW
DSNLLX02 IX        RW
******* DISPLAY OF DATABASE DSNDB01  ENDED      **********************
DSN9022I -DB1G DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

If LPL entries exist, LPL recovery begins when you start the table space, index, or partition by using the START DATABASE command with ACCESS(RW) or ACCESS(RO).

End of General-use Programming Interface

Physical R/W Errors: In previous releases of DB2, physical read and write errors were recorded in an error page range. This is still the case; however, if a read or write problem is of undetermined cause, the error is first recorded in the LPL. If recovery from the LPL is unsuccessful, the error is then recorded on the error page range.

Obtaining Information about Locks Held during DB2 Failure

When a lock is used to allow an object to be changed (this is called a *modify* lock), the lock is kept in a list in the coupling facility lock structure to allow for recovery in case a DB2 subsystem fails. If a DB2 subsystem fails, modify locks become *retained* locks, which means they are held until the failed subsystem is restarted. For more information about retained locks, see "Active and Retained Locks" on page 182.

To determine if there are retained locks, use the DISPLAY DATABASE command with the LOCKS option as shown here:

```
-DB1G DISPLAY DATABASE(TESTDB) LOCKS
```

General-use Programming Interface

The following output is generated:

```
DSNT360I -DB1G ***********************************************************
DSNT361I -DB1G *  DISPLAY DATABASE SUMMARY
               *    GLOBAL LOCKS
DSNT360I -DB1G ***********************************************************
DSNT362I -DB1G     DATABASE = TESTDB STATUS = RW
DBD LENGTH = 4028
DSNT397I -DB1G
NAME     TYPE PART STATUS            CONNID   CORRID       LOCKINFO
-------- ---- ---- ------------------ -------- ------------ ---------
TBS43    TS        RW                BATCH    SELEC        H-IS,S,C
-                  MEMBER NAME DB1G
TBS43    TS        RW                                      R-IX,S
-                  MEMBER NAME DB2G
TBS43    TS    01  RW                                      H-IS,PP,I
-                  MEMBER NAME DB1G    (CO)
TBS43    TS    01  RW                                      R-IX,PP
-                  MEMBER NAME DB2G
⋮
```

\# *Figure 36. Partial DISPLAY DATABASE Output Showing Retained Locks. From this display, it appears that DB2G has gone down. You can tell if a lock is retained if there is an R in the LOCKINFO field of the report.*

End of General-use Programming Interface

DB2G must be restarted to clear the retained locks.

Removing Retained Locks: A normal restart of DB2 resolves and removes retained locks held by that DB2 with the full data integrity control that DB2 restart provides. However, if for some reason you cannot get DB2 restarted and the failed DB2 has retained locks that are severely affecting transactions on other DB2s, consider the following actions:

- Defer the restart processing of the objects that have retained locks.

  When you defer restart processing, the pages that locks are protecting are placed in the logical page list (LPL). Those pages are still inaccessible. However, this approach still has the advantage of removing any retained page set P-locks, which have the potential of locking out access to an entire page set. See Section 4 (Volume 1) of *Administration Guide* for more information about deferred restart.

- Cold start the failed member.

  This approach causes DB2 to purge the retained locks, but **data integrity is not protected.** When the locks are released after the cold start, DB2 will be looking at data whose status is unclear.

Controlling Connections to Remote Systems

This section describes how controlling DDF connections is changed for data sharing. The following topics are described:

General-use Programming Interface

Starting and Stopping DDF

The distributed data facility (DDF) is controlled on a member basis. This gives you more control over DDF processing in the group. Say, for example, that you want to devote DB1G to batch processing for some period of time without disrupting other connections. You can enter the following command to disallow any further distributed connections from coming into this member:

```
-DB1G STOP DDF MODE(QUIESCE)
```

To stop *all* DDF processing for the group, you have to enter the STOP DDF command for every member of the group. You might need to do this when, for example, you change the SYSIBM.SYSLOCATIONS table of the CDB.

End of General-use Programming Interface

General-use Programming Interface

Monitoring Connections to Remote Systems

There are no new commands for monitoring connections to or from a data sharing group. The DISPLAY LOCATION and DISPLAY THREAD commands show information only for the member on which it is issued.

If your data sharing group is defined with a generic LU, you must use the *real* LU name for *luwid*, if you are requesting information by *luwid*.

When a remote DB2 issues a DISPLAY LOCATION command to obtain information about connections to a data sharing group, the output displays information about every LU at that location.

End of General-use Programming Interface

Resetting Generic LU Information

If you are using a generic LU name to connect to a member of the data sharing group using 2-phase commit, VTAM permanently records information in the coupling facility about which member of the DB2 group was involved in the communication. This permanently recorded information is required to guarantee that future VTAM sessions are always directed to the same DB2 group member, making it possible to provide access to the correct DB2 subsystem log for resolution of indoubt threads.

There might be times when you need to break that affinity between the group member and the other system. You would need to do this, for example, if you want to start using the member-specific method, or if you want to remove a member from the data sharing group.

General-use Programming Interface

To break this affinity, use the RESET GENERICLU command. The command must be issued from the member with the affinity to the particular LU. Here is an example that removes information about USIBMSTODB22 from DB1G:

```
-DB1G RESET GENERICLU(LUDB22)
```

Great care should be taken when using this command, because it can potentially cause the specified partner LUs to connect to different DB2 subsystems on later sessions. This can cause operational problems if indoubt threads exist at a partner LU when this command is issued. This command can be issued from any member of the data sharing group.

End of General-use Programming Interface

For more information about using the RESET GENERICLU command, see Chapter 2 of *Command Reference*.

Establishing the Logging Environment

In a data sharing environment, the member subsystems still maintain separate recovery logs. Each manages its own active and archive log data sets and records those in its own bootstrap data set (BSDS). The shared communications area (SCA) in the coupling facility contains information about all members' BSDSs and log data sets. In addition, every member's BSDS also contains information about other members' BSDS and log data sets, in case the SCA is not available.

Figure 37 illustrates a typical logging environment.

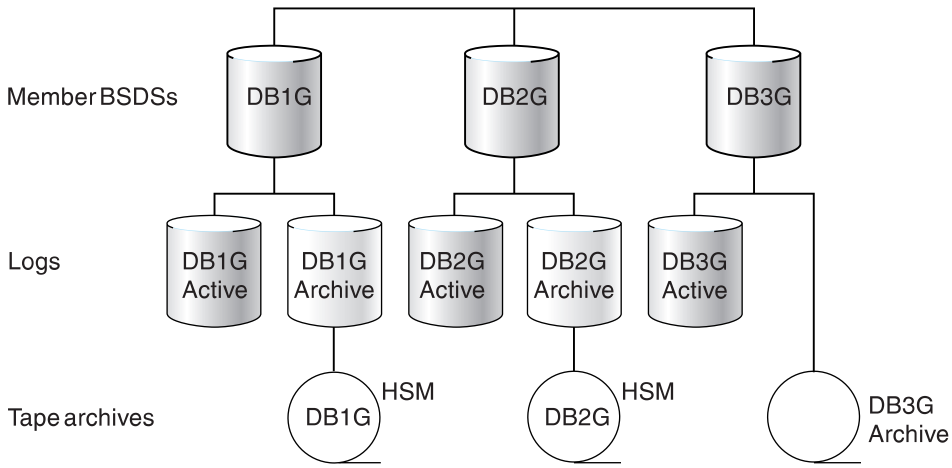

Figure 37. Member BSDSs and Logs in the Data Sharing Environment

The Impact of Archiving Logs in a Data Sharing Group

In data sharing, DB2's RECOVER utility needs log records from every member that has changed the object needing to be recovered. More information about how RECOVER uses these logs is described in "How Recovery Works in a Data Sharing Group" on page 161. If the logs are archived, the impact on RECOVER depends on how the log data sets are archived:

- Archive to DASD without DFSMShsm to migrate the data sets from DASD to tape.

  There is no major impact on performance. But you need enough DASD to hold archive logs, and the DASD devices must be shared (accessible) by all DB2 members in a data sharing group. Because DFSMShsm or its equivalent is not used, you must manage DASD carefully to avoid running out of space.

- Archive to DASD with DFSMShsm.

  DFSMShsm can do automatic space and data availability management among storage devices in a system. DFSMShsm can migrate the archive on DASD to less expensive storage (such as tape), and recall back to DASD when needed.

  Using DFSMShsm, a particular RECOVER job needs only one tape unit to recall migrated archive data sets. If the archive data sets have been migrated, recovery time might be adversely affected, because the recalls of the migrated archive data sets are done one at a time from the member running the RECOVER job. For example, a RECOVER job started on DB1G might need log data sets from DB1G, DB2G, and DB3G. DB1G sends the recall requests to DFSMShsm one at a time for the tapes needed for recovery.

- Archive to tape

  The RECOVER job needs at least one tape unit for each DB2 member whose archived log records are to be merged. (More might be needed if you run more than one recover at the same time for different partitions of a partitioned table space.) Therefore, **do not archive logs from more than one system to the same tape**.

  Archiving to tape is not recommended because there can be negative consequences to not having enough tape units allocated: **If there are not enough tape units to do the recovery, DB2 can possibly deadlock.** If this happens, use the command SET ARCHIVE to increase the number of tape units that can be used.

  If you must archive to tape, make sure the value for READ TAPE UNITS on installation panel DSNTIPA for each member is high enough to handle anticipated recovery work. For example, if you have 8 members, each member should specify at least 8 tape drives. You'll need more if you run more than one recovery job at the same time on a given member, or if multiple members run recovery jobs at the same time.

  Also, make sure you specify 0 for the DEALLOC PERIOD field on installation DSNTIPA to avoid making an archive tape inaccessible to other members of the data sharing group. (If you intend to run all RECOVER jobs from a single DB2, this suggestion does not apply.)

For data sharing, we recommend that you avoid using tape archive logs for data recovery.

How to Avoid Using the Archive Log

A recovery cycle for a table space is defined by how often its image copy is taken. A RECOVER job needs the latest image copy, the optional incremental copies, and the log records since the last incremental copy (or image copy, if there is no incremental copy). A RECOVER job needs no archived log records if all the log records since the last incremental image copy are still in active log data sets. This is to your advantage, because reading log records from the active log is much faster than reading from archive logs, even if those archives are on DASD.

There are several ways to minimize the need to go to the archive log:

- **Increase the total active log space.**

  The total amount of active log space is the number of active log data sets multiplied by its size. Currently, DB2 limits the maximum number of active log data sets to 31. Because each DB2 member can have up to 31 active log data sets, the total number of active log data sets is effectively increased by the number of DB2 members in a data sharing group.

  The size of an active log data set is up to 2 gigabytes but is usually limited by the size of a tape cartridge. Most installations prefer not to have an archived data set on more than one volume. With the ever-increasing capacity of the new tape devices, the size of the active log can also increase. However, some of the increased capacity is due to a tape compression algorithm. We do not recommend using tape compression for the DB2 archive log, because DB2 needs to read the log backwards for backout operations. Performance for backout can be severely degraded it if there is compression.

(This is not to be confused with the DB2 data compression, which compresses the data portion of a DB2 log record. With DB2 data compression, the log record header is not compressed and causes no extra performance degradation for backward scans.)

- **Increase the frequency of incremental image copies.**

  Because only the log records generated since the last incremental image copy are needed for recovery, the more often you make incremental copies, the less chance there is that archive log records will be needed. Of course, this consideration needs to be weighed against the time it takes to make the incremental image copies and their effects on SQL transactions.

  See “Preparing for Faster Recovery” on page 163 for more information about improving recovery performance.

- **Make sure applications commit frequently.**

  To avoid having to mount an archive log for backing out changes, ensure that applications are committing frequently. Consider using the UR CHECK FREQ field of installation panel DSNTIPN to help you track when applications are not committing to the frequency set at your site.

Backing Up and Recovering Databases

This section describes the changes in data recovery that are required by data sharing, including data affected by the failure of the coupling facility or structures within the coupling facility.

The procedures for data recovery are fundamentally the same for data sharing as for non-data-sharing. Data sharing involves one catalog, but there are now many logs and BSDSs. In addition to DASD and cache controllers, a new medium, the coupling facility, is introduced. This adds a possible point of failure and necessitates appropriate procedures for data recovery. In planning for data sharing, it is important to consider having more than one coupling facility. Should a structure failure occur, recovery for the SCA and lock structure can proceed automatically if a second coupling facility is available.

How Recovery Works in a Data Sharing Group

This section describes how the recovery process works when a shared object needs to be recovered.

Determining the Logs Needed for Recovery

Let's assume there are three members making updates to table space TS1 as shown in Figure 38 on page 162.

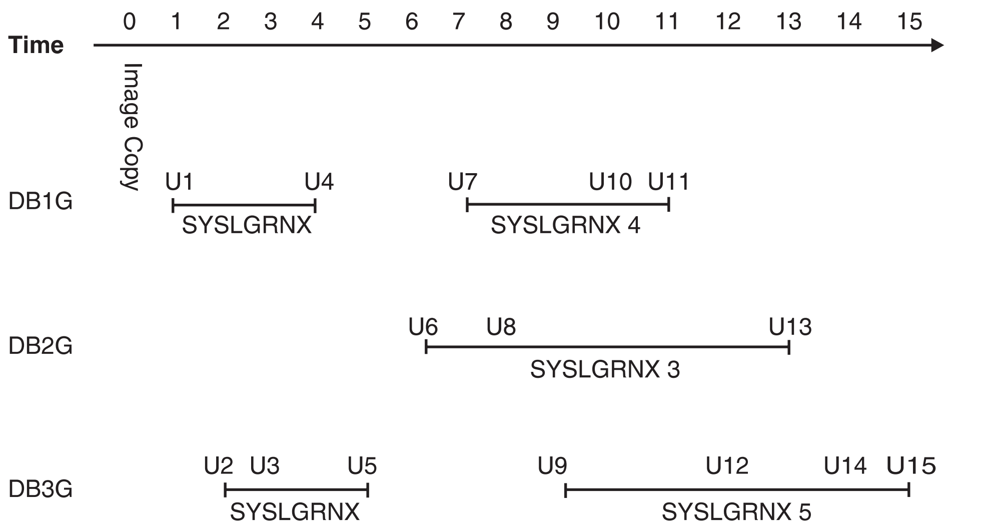

Figure 38. Three DB2 Subsystems Updating Table Space TS1

Here is the sequence of updates leading up to the time of recovery:

- DB1G updated TS1 between Time 1 and 4 (SYSLGRNX record 1) with updates 1 and 4 (U1 and U4 in the figure). DB1G updated TS1 again between Time 7 and 11 (SYSLGRNX record 4) with U7, U10, and U11.
- DB2G updated TS1 between Time 6 and 13 (SYSLGRNX record 3) with U6, U8, and U13.
- DB3G updates TS1 between Time 2 and 5 (SYSLGRNX record 2) with U2, U3, and U5. DB3G updates TS1 again between Time 9 and 15 (SYSLGRNX record 5) with U9, U12, U14 and U15.

Now, assume you want to recover TS1 to time 9. The full image copy taken at T0 is used as the recovery base. All the SYSLGRNX records mentioned above are selected to determine the log ranges of each system for the log scan. Updates U1 through U9 are applied in order.

Applying the Log Records

DB2 can access the logs of other DB2 systems in the group and merge them in sequence. Hence, the merge process needs a unique, group-wide identifier. The log record sequence number (LRSN), a 6-byte hex value derived from a store clock timestamp, is that identifier. Figure 39 on page 163 shows the structure of the log record.

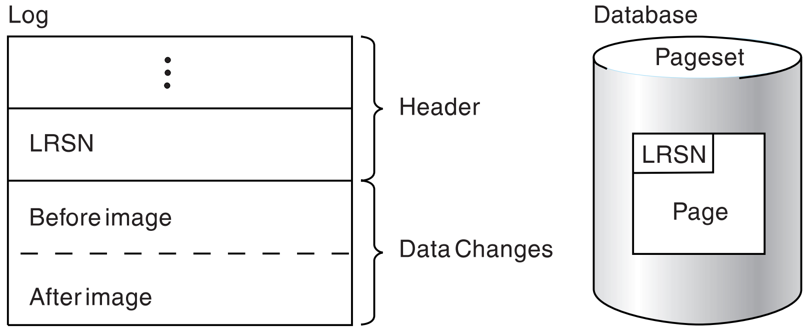

Figure 39. The Log and LRSN in the Data Sharing Environment. During recovery, DB2 compares the LRSN in log record with LRSN in the data page to determine whether the log record must be applied to the data on DASD.

The log record header contains a log record sequence number (LRSN). The LRSN is a 6-byte value that is equal to or greater than the timestamp value truncated to 6 bytes. This value also appears in the page header. During recovery, DB2 compares the LRSN in the log record to the LRSN in the page header before applying changes to DASD. If the LRSN in the log record is larger than the LRSN on the data page, the change is applied.

Preparing for Faster Recovery

One way to prepare for a quicker recovery is to increase the frequency of copies. You might want to limit this activity by determining which table spaces most need this fast recovery. The following guideline is provided as a starting point to help you determine how often you must do incremental image copies. The assumption is that you are already familiar with using the COPY utility and with all the ramifications of using incremental versus full image copies, as described in Section 2 of *Utility Guide and Reference*. As with a single subsystem, doing frequent enough image copies can help you avoid going to the archive log for recovery.

Use the guideline below for each member of the data sharing group. Use the output of the print log map utility (DSNJU004) for each member.

1. Find the starting timestamp of the active log data set with the lowest STARTRBA.
2. Find the ending timestamp of the active log data set with the highest ENDRBA.
3. Calculate the time interval:

   ```
   time_interval = end_TIMESTAMP − start_TIMESTAMP
   ```

4. Calculate the interval at which to perform incremental image copies:

   ```
   interval of copy = time_interval * (n-1) / n
   ```

   Where n is the number of active log data sets.

5. Take the smallest interval for the group and, to account for fluctuations of log activity, decrease the interval by 30 percent. (30 percent is an arbitrary figure; you might have to adjust this interval based on your system's workload.)

   This is the recommended interval for doing incremental image copies. If the interval is too small to be realistically accomplished, consider increasing the size or number of active log data sets.

   | Be sure to periodically run MERGECOPY utility with incrementals. The
   | RECOVER utility attempts to mount tape drives for all the incrementals at the

same time. If it runs out of tape drives, it then switches to log apply. MERGECOPY merges what it can and then mounts more incrementals.

Using the RECOVER Utility

Use the RECOVER utility to recover to currency or to a prior point in time. The details of RECOVER are described in Section 2 of *Utility Guide and Reference*.

Recovery to Currency

This process is used to recover from damaged data by restoring from a backup and applying all logs to the current time. The recovery process operates similarly in the data sharing and non-data sharing environments. Image copies are restored and subsequently updated based on changes recorded in the logs. In the data sharing group, multiple member logs are read concurrently in log record sequence.

Point-in-Time Recovery

This process discards potentially corrupt data by restoring a database to a prior point of consistency. Such problems with the data might result from a logical error. The following point-in-time recovery options are available:

TORBA This option is used to recover to a point on the log defined by an RBA. In a data sharing environment, TORBA can only be used to recover to a point prior to defining the data sharing group.

TOLOGPOINT This option is used to recover to a point on the log defined by a *log record sequence number* (LRSN). The TOLOGPOINT keyword must be used when you recover to a point on the log after the data sharing group was defined. However, you can also use TOLOGPOINT in a non-data sharing environment.

The LRSN is a 6-byte hexadecimal number derived from a store clock timestamp. LRSNs are reported by the DSN1LOGP stand-alone utility.

TOCOPY This option is used to recover data to the values contained in an image copy without subsequent application of log changes.

Successful recovery clears pending recovery conditions and brings data to a point of consistency. In a data sharing environment, all pages associated with the recovered data entity are removed from the group buffer pool and written to DASD.

Recovering a Data Sharing Group in Case of a Disaster

This section presents an overview of how to recover a data sharing group at a remote site. To develop a procedure, you can use as a base the disaster recovery procedure documented in Section 4 (Volume 1) of *Administration Guide* . With a couple of exceptions, you must perform those steps for each member of the data sharing group. The following topics describe how to prepare for recovery of a data sharing group at a recovery site:

- “Configuring the Recovery Site” on page 165
- “What to Send to the Recovery Site” on page 166

“Recovery Procedure” on page 167 is the procedure you use to prepare the data sharing group at the recovery site for a group restart.

Configuring the Recovery Site

The recovery site must have a data sharing group that is identical to the group at the local site. It must have the same name, the same number of members and the names of the members must be the same. The CFRM policies at the recovery site must define the coupling facility structures with the same names (although the sizes can be different).

You can run the data sharing group on as few or many MVS systems as you want.

The hardware configuration can be different at the recovery site, as long as it supports data sharing. Conceptually, there are two ways to run the data sharing group at the recovery site. Each way has different advantages that can influence your choice:

- **Run a multi-system data sharing group.**

  This is the way the local site is most likely configured. You have a Parallel Sysplex containing many CPCs, MVS systems, DB2s. This configuration requires a coupling facility, the requisite coupling facility channels, and the Sysplex Timer.

  The advantage to this is you have the same availability and growth options that you had on the local site.

- **Run a single-system data sharing group.**

  In this configuration, you centralize all your DB2 processing within a single, large CPC, such as a 711-based processor model. As Figure 40 shows, you must *install* a multi-member data sharing group. After the group starts up, you shut down all but one of the DB2s and access data through that single DB2.

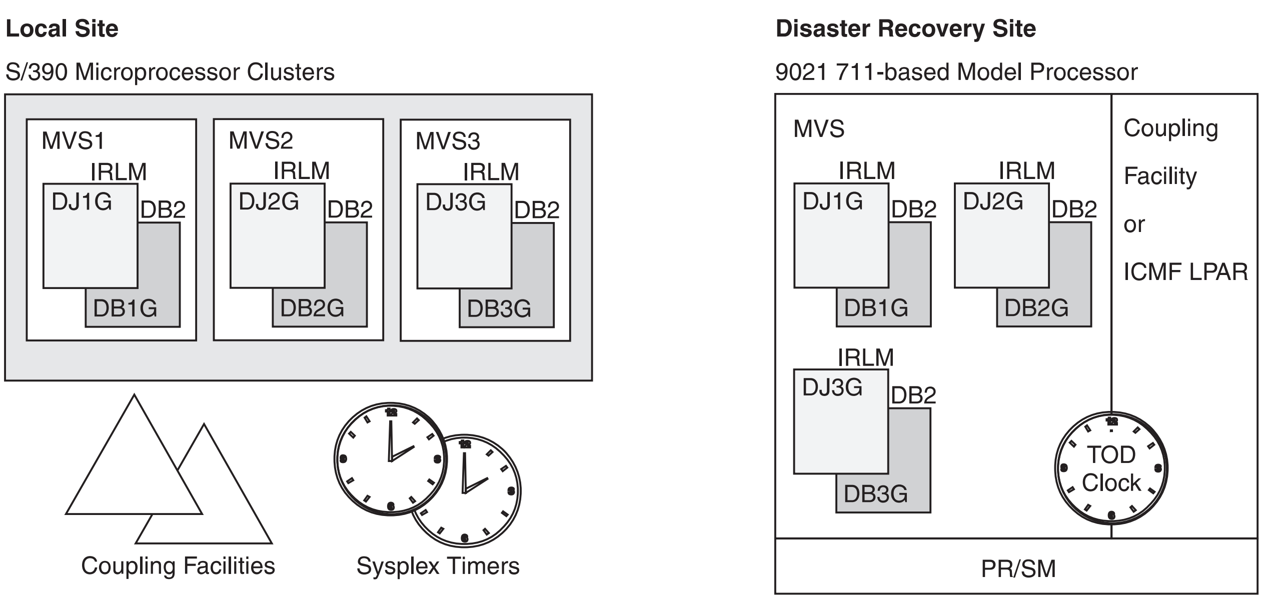

Figure 40. Example Local and Disaster Recovery Site Configurations. With this configuration, the recovery site can be a single-system data sharing group. After the data sharing group is started on the recovery site, all but one of the members can be stopped.

Obviously, you lose the availability benefits of the Parallel Sysplex, but the single-system data sharing group has fewer hardware requirements:

– The Sysplex Timer is not needed; the time-of-day clock of the CPC can be used.

– The coupling facility emulator, the Integrated Coupling Migration Facility (ICMF), can be used instead of a regular coupling facility.

  Although ICMF is not recommended for data sharing environments where you expect lots of coupling facility traffic, such traffic is greatly reduced in a single-system data sharing environment. With a single-system data sharing group, there is no longer inter-DB2 R/W interest, and the requirements for the coupling facility are as follows:

  - A lock structure (which can be smaller)
  - An SCA

Group buffer pools are not needed for running a single-system data sharing group. However, you do need to have at least small group buffer pools for the initial startup of the group so that DB2 can allocate them and do its damage assessment processing. When you are ready to do single-system data sharing, you can remove the group buffer pools by stopping all members and then restarting the member that is handling the workload at the disaster recovery site.

Before choosing ICMF for your recovery site, be aware of the following considerations, all of which are minimized when you run a single-system data sharing group:

– There is additional overhead associated with ICMF
– ICMF emulator channels are not as fast as coupling facility channels
– All partitions in the Sysplex must be on the same CPC
– If the CPC fails, you lose both the DB2 and the coupling facility.

For more information about ICMF, see *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide*.

What to Send to the Recovery Site

You must send the same information as documented for single-system remote recovery: logs and BSDSs, image copies, and so on. To prepare the logs for the remote site, you have two options:

- Use the command ARCHIVE LOG with the MODE (QUIESCE) option to ensure a point of consistency for each of the log data sets. If the quiesce is not successful, the command fails and the logs are not archived.

- Use the command ARCHIVE LOG SCOPE(GROUP). This version of command does not ensure a point of consistency for all members' logs, but the logs are archived on each of the active members of the data sharing group. You can use the ENDLRSN option of the change log inventory utility on the remote site to truncate all logs to the same point in time.

  To determine the truncation value, you can look at the print log map output from the latest copies of the archived BSDS, as described in 5 on page 168.

  Another way to determine the truncation value is to ship the SYSLOG containing message DSNJ003I with your archive log data sets to the recovery site. This message is issued when archive log data sets are created as a result of someone issuing the ARCHIVE LOG command. The message contains the

| starting and ending LRSN and RBA values for the archive log data set. For example, the following messages appear when the command ARCHIVE LOG SCOPE(GROUP) is issued from one of the members at the local site:

```
DSNJ003I -DB1G DSNJOFF3 FULL ARCHIVE LOG VOLUME
DSNAME=DSNC510.ARCHLOG1.A0000003, STARTRBA=000001C68000,
ENDRBA=000001D4FFFF, STARTLRSN=ADFA208AA36C, ENDLRSN=AE3C45273A77,
UNIT=SYSDA, COPY1VOL=SCR03, VOLSPAN=00, CATLG=YES

DSNJ003I -DB2G DSNJOFF3 FULL ARCHIVE LOG VOLUME
DSNAME=DSNC518.ARCHLOG1.A0000001, STARTRBA=000000000000,
ENDRBA=0000000D6FFF, STARTLRSN=ADFA00BB70FB, ENDLRSN=AE3C45276DD7,
UNIT=SYSDA, COPY1VOL=SCR03, VOLSPAN=00, CATLG=YES
```

Compare the ending LRSN values for all members' archive logs, and choose the lowest LRSN as the truncation point; for the two members here, the lowest LRSN is AE3C45273A77. To get the last complete log record, you must subtract 1 from that value, so you would enter AE3C45273A76 as in the conditional restart control records on each of the members at the remote site. All log records with a higher LRSN value are discarded during the conditional restart.

Attention: Make sure that all members of the group are active when you archive the logs. If you have a quiesced member whose logs are necessary for a recovery base at the disaster recovery site, you must start that member with ACCESS(MAINT) to archive its log.

For read-only members, DB2 periodically writes a log record to prevent those members from keeping the LRSN value too far back on the log.

Recovery Procedure

The procedure at the recovery site differs for data sharing in that there are extra steps for cleaning out old information in the coupling facility. Old information is in the coupling facility from any practice startup. (If you do not have old information in the coupling facility, then you can omit steps 1 through 3.)

In addition, you must prepare each subsystem for conditional restart rather than just a single system.

1. Enter the following MVS command to display the structures for this data sharing group:

   D XCF,STRUCTURE,STRNAME=*grpname**

2. For group buffer pools and the lock structure, enter the following command to force the connections off those structures:

   SETXCF FORCE,CONNECTION,STRNAME=*strname*,CONNAME=ALL

   Connections for the SCA are not held at termination, so there are no SCA connections to force off.

3. Delete all the DB2 coupling facility structures by using the following command for each structure:

   SETXCF FORCE,STRUCTURE,STRNAME=*strname*

   This step is necessary to clean out old information that exists in the coupling facility from your practice startup when you installed the group.

4. Do your initial recovery steps, including creating and importing the integrated facility catalog and restoring DB2 libraries. See Section 4 (Volume 1) of *Administration Guide* for more information.

5. Recover the BSDS. This includes using the REPRO command of access method services to restore the contents of the BSDS data sets from the latest archive, and then using the change log inventory utility (DSNJU003) to register the latest archive log tape data set. After these archive logs are registered, use the print log map utility (DSNJU004) with the GROUP option to list the contents of all the BSDSs. You get output that includes the start and end LRSN and RBA values for the latest active log data sets (shown as NOTREUSABLE). If you did not save the values from the DSNJ003I message, you can get those values from here, as shown in Figure 41 and Figure 42. Remember to subtract 1 from the lowest ENDLSRN value before entering it in the conditional restart control records of each of the data sharing members.

```
ACTIVE LOG COPY 1 DATA SETS
  START RBA/LRSN/TIME   END RBA/LRSN/TIME      DATE    LTIME  DATA SET INFORMATION
 --------------------  --------------------  --------  -----  --------------------
   000001C20000          000001C67FFF       1996.358  17:25  DSN=DSNC510.LOGCOPY1.DS03
     ADFA0FB26C6D          ADFA208AA36B                       STATUS=TRUNCATED, REUSABLE
 1996.361  23:37:48.4  1996.362  00:53:10.1
   000001C68000          000001D4FFFF       1996.358  17:25  DSN=DSNC510.LOGCOPY1.DS01
     ADFA208AA36C          AE3C45273A77                       STATUS=TRUNCATED, NOTREUSABLE
 1996.362  00:53:10.1  1997.048  15:28:23.5
   000001D50000          0000020D3FFF       1996.358  17:25  DSN=DSNC510.LOGCOPY1.DS02
     AE3C45273A78          ............                       STATUS=NOTREUSABLE
 1997.048  15:28:23.5  ........  ..........
```

Figure 41. Member DB1G's BSDS Contents (Partial)

```
ACTIVE LOG COPY 1 DATA SETS
  START RBA/LRSN/TIME   END RBA/LRSN/TIME      DATE    LTIME  DATA SET INFORMATION
 --------------------  --------------------  --------  -----  --------------------
                EMPTY DATA SET              1996.361  14:14  DSN=DSNC518.LOGCOPY1.DS03
     000000000000          000000000000                       STATUS=NEW, REUSABLE
 0000.000  00:00:00.0  0000.000  00:00:00.0
   000000000000          0000000D6FFF       1996.361  14:14  DSN=DSNC518.LOGCOPY1.DS01
     ADFA00BB70FB          AE3C45276DD7                       STATUS=TRUNCATED, NOTREUSABLE
 1996.361  22:30:51.4  1997.048  15:28:23.7
   0000000D7000          00000045AFFF       1996.361  14:14  DSN=DSNC518.LOGCOPY1.DS02
     AE3C45276DD8          ............                       STATUS=NOTREUSABLE
 1997.048  15:28:23.7  ........  ..........
```

Figure 42. Member DB2G's BSDS Contents (Partial)

If there is a discrepancy among the print log map reports as to the number of members in the group, record the one that shows the highest number. (This is an unlikely occurrence.) This is the DB2 that must be started first.

6. Do all other preparatory activities as you would for a single system. Do these activities for *each* member of the data sharing group, including preparing conditional restart control records.

   If you are using an LRSN value to truncate the logs at the remote site, use the lowest end LRSN value, minus 1, in the CRCR records of all the members. This avoids a spanned log record. (In our example, that is AE3C45273A77 - 1, which is AE3C45273A76.)

7. Start one DB2 with ACCESS(MAINT).

DB2 will prompt you to start each additional DB2 subsystem in the group.

8. After all additional members are successfully restarted, and if you are going to run single-system data sharing at the recovery site, stop all DB2s but one using the STOP DB2 command with MODE(QUIESCE).

9. Continue with the recovery actions as documented for a single non-data-sharing system.

Recovering Pages on the Logical Page List

In some cases, DB2 can automatically recover pages on the logical page list when group buffer pools are defined with AUTOREC(YES), the default. However, there are many situations where pages are put on the LPL that require you to do manual recovery. There are several ways to do this:

- Start the object with access (RW) or (RO). That command is valid even if the table space is already started.

  When you issue the command START DATABASE, you see message DSNI006I, indicating that LPL recovery has begun. Message DSNI022I might be issued periodically to give you the progress of the recovery. When recovery is complete, you see DSNI021I.

- Run the RECOVER utility on the object.

  The only exception to this is when a logical partition of a type 2 non-partitioned index has both LPL and RECP status. If you want to recover the logical partition using RECOVER INDEX with the PART keyword, you must first use the command START DATABASE to clear the LPL pages.

- Run the LOAD utility with the REPLACE option on the object.
- Issue an SQL DROP statement for the object.
- Use the utility REPAIR SET with NORCVRPEND. This can leave your data in an inconsistent state.
- Use START DATABASE ACCESS(FORCE). This can leave your data in an inconsistent state.

None of the items in the above list work if there are retained locks held on the object. You must restart any failed DB2 that is holding those locks.

Recovery from Coupling Facility Failures

Failures of the coupling facility can be classified into two main groups:

- Connectivity failures

  Connectivity failures can be caused by a problem with the attachment of the MVS system to the coupling facility. They can also occur when the following types of failures occur:

  – Power failure that affects the coupling facility but leaves one or more MVSs running.

  – Deactivation of the coupling facility partition.

  – Coupling facility control code failure.

  – Coupling facility CPC or LPAR failure.

- Structure failure

A structure failure is a rare event in which structures are damaged in some way but the coupling facility continues to operate.

This section also includes information about allocation failure and problems caused by not enough storage.

Preparing for Structure and Connectivity Failures

Coupling facility failures can mean serious outages for users. Not having a lock structure or SCA can cause the entire group to come down abnormally. Group buffer pool failure does not cause the group to come down, but it can still mean loss of availability for applications depending on the data in that group buffer pool.

Careful preparation can greatly reduce the impact of coupling facility outages on your end users. To best prepare yourself for both types of failures, you must have the following:

- An active system failure management (SFM) policy with system weights specified.

  This is strongly recommended. Descriptions of failure scenarios in this section assume you have done this. If you have not, it is not possible to automatically rebuild coupling facility structures. If the SCA and lock structure cannot be rebuilt, DB2 abnormally terminates the members affected by the loss of those structures, or the loss of connectivity to those structures. If the group buffer pool cannot be rebuilt, which is only attempted when a subset of members lose connectivity, then those members disconnect from the group buffer pool.

- Alternative coupling facility information provided on the preference list of each of the structures in the CFRM policy.
- Adequate storage in an alternate coupling facility to rebuild or reallocate structures as needed.

  For rebuild, MVS normally uses the INITSIZE value of the CFRM policy on the alternate coupling facility to allocate storage. If MVS cannot allocate enough storage to rebuild the SCA or lock structure, the rebuild fails. If it cannot allocate enough storage for the group buffer pool, DB2 must write the changed pages to DASD instead of rebuilding them into the alternate group buffer pool. For more information about how structure allocation works, see *OS/390 MVS Programming: Sysplex Services Guide*.

- For page sets requiring very high availability, use of group buffer pool duplexing.

For more information about planning, see “Coupling Facility Availability” on page 61.

Summary of Failure Scenarios

The tables in this section summarize connectivity and structure failure situations.

- “Connectivity Failure for Lock Structure and SCA” on page 171
- “Connectivity Failure for Group Buffer Pools” on page 172
- “Structure Failures” on page 173
- “Summary of failure scenarios for duplexed group buffer pools” on page 173

For more information about recovery scenarios, see “Coupling Facility Recovery Scenarios” on page 174.

Connectivity Failure for Lock Structure and SCA: Table 23 summarizes what happens when there are connectivity failures to the lock structure or SCA.

Table 23. Summary of Connectivity Failures for the SCA and Lock Structure

Situation	Without Active SFM Policy OR weighted loss < REBUILDPERCENT	Operational Response	Active SFM Policy where weighted loss ≥ REBUILDPERCENT	Operational Response
Lost connectivity to SCA	Each affected member: DSN7501A 00F70600 DB2 comes down. Connection is deleted; structure remains allocated.	Options include: • Fix problem • Restart failed member on system that is connected to coupling facility • Manually rebuild onto another coupling facility.	Automatic rebuild DSN7503I	None needed.
Lost connectivity to lock structure.	Each affected member: DXR136I 00E30105 DB2 comes down. Connection is failed-persistent; structure remains allocated.	Same as SCA.	DXR143I Automatic rebuild DXR146I	None needed.

Connectivity Failure for Group Buffer Pools: Table 24 summarizes what happens when there are connectivity failures to the group buffer pools.

Table 24. Summary of Connectivity Failures for Group Buffer Pools

Situation	Weighted loss < REBUILDPERCENT	Operational Response	Weighted loss ≥ REBUILDPERCENT	Operational Response
Lost connectivity from all connected members	Not applicable	Not applicable	Each affected member: DSNB303E DSNB228I Add pages to LPL, if necessary. DSNB250E DSNB314I* rsn=100% DSNB304I* Damage assessment, GRECP page sets. DSNB320I DSNB321I DSNB353I DSNI006 DSNI021 DSNB354I DSNB305I*	None needed if the group buffer pool is defined with AUTOREC(YES), and DB2 successfully recovers the page sets. Otherwise, enter START DATABASE commands.
A subset of connected members lost connectivity to some or all group buffer pools.	Each affected member: DSNB303E DSNB228I DSNB313I rsn=LOSSCONN Quiesce applications that use the group buffer pool. Add pages to LPL if necessary. DSNB250E DSNB311I, DSNB312I Disconnect GBPx failed-persistent. DSNB309I	Options include: • Fix the problem. • Manually rebuild the structure onto another coupling facility. • Stop and restart DB2 on a system that is connected to the coupling facility. Enter START DATABASE commands to recover LPL entries.	Automatic rebuild. DSNB331I DSNB332I DSNB333I* DSNB338I	None needed.

Note: *Issued by the structure owner.

Structure Failures: Table 25 summarizes what happens to each structure if there is a structure failure. For more information, see:

"Problem: Group Buffer Pool Structure Failure (No Duplexing)" on page 177
"Problem: Lock Structure Failure" on page 178
"Problem: SCA Structure Failure" on page 178

Table 25. Summary of Structure Failures, by Structure Type

Failed Structure	DB2 Response	Operational Response
SCA	DSN7502I Automatic rebuild DSN7503I	None needed.
Lock Structure	DXR143I Automatic rebuild DXR146I	None needed.
Group buffer pool	DSNB228I DSNB314I rsn=STRFAIL Add pages to LPL, if necessary. DSNB250E Damage assessment, GRECP page sets. DSNB304I DSNB320I DSNB321I DSNI006 DSNI021 DSNB305I	None needed if the group buffer pool is defined with AUTOREC(YES) and DB2 successfully recovers the page set. Otherwise, enter START DATABASE commands.

Summary of failure scenarios for duplexed group buffer pools

For duplexed group buffer pools, a failure response is the same for both structure failures and for lost connectivity.

Table 26. Summary of scenarios for both structure failure and lost connectivity for duplexed group buffer pools

Failure occurred for which structure?	DB2 response	Operational response
Primary	Switch to secondary in simplex mode DSNB744I DSNB745I If DUPLEX(ENABLED) then reduplexing is attempted.	If the system did not automatically reduplex, correct the problem with the failed coupling facility. If you want to restart duplexing, use the SETXCF command.
Secondary	Revert to primary in simplex mode DSNB743I DSNB745I If DUPLEX(ENABLED) then reduplexing is attempted.	If the system did not automatically reduplex, correct the problem. If you want to restart duplexing, use the SETXCF command.
Both (structure failure or 100% LOSSCONN)	Damage assessment, GRECP page sets.	None needed if the group buffer pool is defined with AUTOREC(YES) and DB2 successfully recovers the page set. Otherwise, enter START DATABASE commands.

Coupling Facility Recovery Scenarios

Problem: All Members Have Lost Connectivity

This scenario explains what you might see if there was a failure that is treated by MVS and DB2 as total loss of connectivity to the coupling facility. It assumes you have an active SFM policy.

Symptom: Some or all of the following messages appear, depending on which structures DB2 tries to access:

```
DSNB303E -DB1G csect-name A LOSS OF CONNECTIVITY
WAS DETECTED TO GROUP BUFFER POOL gbpname.

DSNB228I csect-name GROUP BUFFER POOL gbpname
CANNOT BE ACCESSED FOR function
MVS IXLCACHE REASON CODE=reason

DSNB314I csect-name DAMAGE ASSESSMENT TO BE TRIGGERED FOR
GROUP BUFFER POOL gpbname REASON=100%LCON

DSNB250E csect-name A PAGE RANGE WAS ADDED TO THE
LOGICAL PAGE LIST
DATABASE NAME = dbn
SPACE NAME = spn
DATA SET NUMBER = dsno
PAGE RANGE = lowpg TO highpg
START LRSN = startlrsn
END LRSN = endlrsn
START RBA = startrba

DXR143I irlmx REBUILDING LOCK STRUCTURE BECAUSE IT HAS FAILED OR AN IRLM
        LOST CONNECTION TO IT
```

System Action: When all active members lose connectivity to the **SCA** or **lock structure**, these structures are rebuilt:

DSN7503I is issued for a successful rebuild of the SCA.
DXR146I is issued for a successful rebuild of the lock structure.

Important: If the lock structure or SCA cannot be rebuilt, the lost connectivity causes the members of the group to abend with abend code 00E30105 or 00F70600. If they cannot be rebuilt, such as if both coupling facilities are volatile and lose power, a group restart is required. Group buffer pools cannot be automatically recovered during group restart, and you have to recover those group buffer pools with START DATABASE commands.

To avoid situations in which group restart is necessary, put structures in nonvolatile coupling facility structures. See “Advantages of a Non-Volatile Coupling Facility” on page 64 for more information.

For the lost connectivity to the **group buffer pools**, applications needing access to group buffer pool data are rejected with a -904 SQL return code and can see any of the following reason codes:

00C20204
00C20205
00C20220

- DB2 puts the group buffer pool in “damage assessment pending” status. The following message appears:

  ```
  DSNB304I -DB1G  csect-name GROUP BUFFER POOL
  gbpname WAS SET TO 'DAMAGE ASSESSMENT PENDING' STATUS
  ```

- DB2 adds entries to the logical page list, if necessary.
- DB2 marks the affected table spaces, indexes, or partitions as group buffer pool recovery pending (GRECP) (DSNB320I or DSNB321I) and initiates recovery for the page set (DSNI006I).
- As each page set is recovered, the castout owner for the page set issues DSNI021I. After the last page set is recovered, any DB2 that has issued a DSNB353I now issues a DSNB354I.
- After damage assessment is complete, the structure owner issues DSNB305I.

  The first new connection to the group buffer pool causes MVS to reallocate the group buffer pool in the same or an alternate coupling facility as specified on the preference list in the CFRM policy.

System Programmer Action: The problem causing the loss of connectivity must be fixed. If the problem is not an obvious one (such as a power failure), call IBM service. After the problem is fixed, restart any failed members as quickly as possible to release retained locks.

- For the SCA or lock structure, if the automatic rebuild occurred normally, processing can continue while you wait for the coupling facility problem to be fixed.
- For lost connectivity to group buffer pools, DB2 automatically recovers data that was in any group buffer pool defined with AUTOREC(YES). If the automatic recovery is not successful or if any pages remain in the LPL after recovery, issue START DATABASE commands for all affected page sets. You must issue separate START DATABASE commands in the following order.

  1. DSNDB01
  2. DSNDB06

Problem: A Subset of Members Have Lost Connectivity

This scenario describes what happens if one or more members have lost connectivity to the coupling facility, but some systems are still connected. This might happen if a link is detached between a system and the coupling facility. **This scenario assumes that the combined system weights of the systems that have lost connectivity is less than that required to trigger an automatic rebuild of the structure; that is, less than the value specified on the REBUILDPERCENT parameter of the CFRM policy.** If the combined system weight is greater than or equal to that required to cause rebuild, automatic rebuild occurs. When DB2 rebuilds a group buffer pool, it writes changed pages from the group buffer pool to the alternate structure specified in the CFRM policy. If DB2 determines that there is not enough space to hold the changed pages, it casts the pages out to DASD, instead.

Operator intervention is usually not required unless DB2 is required to add pages to the LPL for some reason while the rebuild occurs. In this case, a START DATABASE command is needed recover the pages on the LPL.

Symptom: Some or all of the following messages appear, depending on which structures DB2 tries to access:

```
DSNB303E -DB1G csect-name A LOSS OF CONNECTIVITY
WAS DETECTED TO GROUP BUFFER POOL gbpname.
```

```
DSNB301E -DB1G csect-name GROUP BUFFER POOL
gbpname CANNOT BE CONNECTED
DB2 REASON CODE =  reason1
MVS IXLCONN REASON CODE =  xxxx0C06
```

```
DSNB228I csect-name GROUP BUFFER POOL gbpname
CANNOT BE ACCESSED FOR function
MVS IXLCACHE REASON CODE=reason
```

```
DSNB313I -DB1G csect-name GROUP BUFFER POOL gbpname
TO BE DISCONNECTED
REASON=LOSSCONN
LOSSCONN PERCENTAGE=percentage
```

```
DSNB250E csect-name A PAGE RANGE WAS ADDED TO THE
LOGICAL PAGE LIST
DATABASE NAME = dbn
SPACE NAME = spn
DATA SET NUMBER = dsno
PAGE RANGE = lowpg TO highpg
START LRSN = startlrsn
END LRSN = endlrsn
START RBA = startrba
```

```
DSNB311I csect-name DBNAME database SPACE NAME spacename
HAS PAGES IN THE LOGICAL PAGE LIST
```

```
DSNB312I csect-name DBNAME database SPACE NAME spacename
PARTITION part-number
HAS PAGES IN THE LOGICAL PAGE LIST
```

```
DSNB309I csect-name GROUP BUFFER POOL gbpname
HAS BEEN DISCONNECTED WITH
A REASON OF 'FAILURE'
```

```
DXR136I irlmx HAS DISCONNECTED FROM THE DATA SHARING GROUP
```

```
DSN7501A -DB1G csect-name SCA STRUCTURE sca-structure-name
               CONNECTIVITY FAILURE.
```

System Action: For a loss of connectivity to the **SCA** or **lock structure**, DB2 abnormally terminates on the members affected by the loss of connectivity. Either abend code 00E30105 or 00F70600 is issued for those members.

For a loss of connectivity to the **group buffer pool**:

1. DB2 adds entries to the logical page list, if necessary.
2. Applications running on the members with the lost connectivity continue processing until the next COMMIT point. On the next attempt to access a GBP-dependent page set associated with a disconnected group buffer pool, applications receive a -904 SQLCODE (reason code 00C20204). For inflight units of recovery, the loss of connectivity is detected immediately, and the application receives a -904 SQLCODE (reason code 00C20220).
3. DB2 disconnects from the group buffer pool and issues message DSNB309I with a reason of FAILURE. The connection enters a failed-persistent state.

System Programmer Action: The problem causing the loss of connectivity must be fixed. If the problem is not an obvious one (a disconnected link, for example), call IBM service. After the problem is fixed, restart any failed members as quickly as possible to release retained locks.

Consider the following options:

- If connectivity was lost to the SCA or lock structure causing the member to fail, restart the failed member on another system that has connectivity to the structure.

  If connectivity is lost to a group buffer pool, DB2 continues to run. If you want to start the member on another system, you must stop DB2 before you restart it.

- Rebuild the structure manually onto another coupling facility using the MVS SETXCF command:

  ```
  SETXCF START,REBUILD,STRNM=strname,LOC=OTHER
  ```

  After you rebuild the group buffer pool, there might still be LPL entries for the page sets. To recover these, enter the command START DATABASE after receiving a DSNB311I or DSNB312I for the page set or partition. Issue the command from a DB2 member that is connected to the group buffer pool.

  To avoid manual intervention next time, lower the REBUILDPERCENT value in the CFRM policy so that the next time you lose connectivity, DB2 can automatically rebuild.

Problem: Group Buffer Pool Structure Failure (No Duplexing)

Symptom: The following message appears on the console of the member who will be coordinating damage assessment:

```
DSNB314I csect-name DAMAGE ASSESSMENT TO BE TRIGGERED FOR
GROUP BUFFER POOL gpbname REASON=STRFAIL
```

System Action: A group buffer pool failure restricts access to the data assigned to that group buffer pool; it does not cause all members of the group to fail. Applications needing access to group buffer pool data are rejected with a -904 SQL return code and can see any of the following reason codes:

00C20204
00C20205
00C20220

See “Problem: All Members Have Lost Connectivity” on page 174 for a description of what DB2 does during automatic recovery.

System Programmer Action: Correct the coupling facility failure. For any page sets that were not automatically recovered by DB2, notify the database administrator to recover the data from the group buffer pool by using the command START DATABASE (*dbname*) SPACENAM (*spacename*) to remove the GRECP status.

Problem: Lock Structure Failure

Symptom: Locking requests are suspended until the lock structure is rebuilt. If the lock structure cannot be rebuilt, the following message appears:

```
DXR136I irlmx HAS DISCONNECTED FROM THE DATA SHARING GROUP
```

System Action: If the structure cannot be rebuilt, all active members of the group terminate abnormally with a 00E30105 abend code.

System Programmer Action: See message DXR135E for the root cause of the problem and the corrective procedure.

Problem: SCA Structure Failure

Symptom: The following message appears:

```
DSN7502I -DB1G csect-name SCA STRUCTURE FAILURE,
ATTEMPT TO REBUILD IS IN PROGRESS.
```

DB2 suspends processing until the SCA is rebuilt, using information contained in DB2's memory.

If the SCA cannot be rebuilt, the following message appears:

```
DSN7504I -DB1G csect-name SCA STRUCTURE structure-name
               REBUILD UNSUCCESSFUL.
```

System Action: If the rebuild is unsuccessful, all DB2s in the group terminate abnormally.

System Programmer Action: Check the termination code for the reason the rebuild was unsuccessful. Correct the problem and then restart the members of the group.

Problem: Allocation Failure of the Group Buffer Pool

Symptom: The following message appears:

```
DSNB301E -DB1G csect-name GROUP BUFFER POOL
gbpname CANNOT BE CONNECTED
DB2 REASON CODE =  reason1
MVS IXLCONN REASON CODE =  xxxx0C08
```

System Action: Applications needing access to group buffer pool data are rejected with a -904 SQL return code (reason code 00C20204).

If the group buffer pool cannot be allocated in an alternate coupling facility as specified on the preference list of the CFRM policy, then there can be no inter-DB2 R/W activity on the table spaces, indexes, or partitions that are assigned to this buffer pool. If the group buffer pool that cannot be allocated is group buffer pool 0, there can be no update activity on the DB2 catalog and directory.

System Programmer Action: Use IFCID 0250 in performance class 20 to determine the reason for the allocation failure. If the trace indicates that the reason for the allocation failure is inadequate storage in the coupling facility, you can:

- Change the CFRM policy to decrease the amount of storage for the group buffer pool, or redefine that group buffer pool to a different coupling facility that has more storage. See “Changing the Size of the Group Buffer Pool” on page 266 for more information.

- Have the database administrator reassign some of the table spaces or indexes using that group buffer pool to a different group buffer pool.

Problem: Storage Shortage in the Group Buffer Pool

Symptoms: The following message appears:

```
DSNB319A -DB1G csect-name THERE IS A SHORTAGE OF SPACE IN GROUP
BUFFER POOL gbpname
```

If you don't do anything to relieve the shortage, the following message appears when the group buffer pool is 90 percent full:

```
DSNB325A -DB1G csect-name THERE IS A CRITICAL SHORTAGE OF
          SPACE IN GROUP BUFFER POOL gbpname
```

If the group buffer pool is full, DB2 cannot write to the group buffer pool, and the following message appears:

```
DSNB228I csect-name GROUP BUFFER POOL gbpname
CANNOT BE ACCESSED FOR function
MVS IXLCACHE REASON CODE=xxxx0C17
```

Performance problems are evidence that the group buffer pool is not large enough. See "Group Buffer Pool Size is Too Small" on page 259 for more information about such symptoms and how to avoid having writes to the group buffer pool fail because of a lack of storage.

System Action: DB2 initiates castout processing if it isn't already in progress. DB2 then tries again to write to the group buffer pool. For simplexed group buffer pools, or for the primary of a duplexed group buffer pool, pages that cannot be written to the group buffer pool are added to the logical page list and message DSNB250E is issued.

If it is the secondary group buffer pool that is too full, DB2 does not add pages to the logical page list; instead, it takes the structure out of duplexing mode.

System Programmer Action: Do the following steps:

1. Issue START DATABASE commands with the SPACENAM option to remove pages from the logical page list (LPL).
2. Increase the size of the group buffer pool, as described in "Changing the Size of the Group Buffer Pool" on page 266.

   If you cannot increase the size of the group buffer pool, use the ALTER GROUPBUFFERPOOL command to decrease the castout thresholds. If decreasing the castout threshold negatively impacts performance, this should be used as a temporary solution.

Problem: Storage Shortage in the SCA

Symptoms: The following message appears:

```
DSN7505I -DB1G csect-name THERE IS A SHORTAGE OF FREE STORAGE
          IN SCA STRUCTURE sca-structure-name.
```

If you don't do anything to reclaim space, such as recovering pages from the LPL, the following message appears when the SCA is 90 percent full:

```
DSN7512A -DB1G csect-name THERE IS A CRITICAL SHORTAGE OF
         FREE STORAGE IN SCA STRUCTURE sca-structure-name.
```

System Action: Some critical functions that cannot be completed can cause one or more members of the group to come down with reason code 00F70609.

System Programmer Action: Do the following steps:

1. Reclaim space in the SCA by removing exception conditions.

   You can issue START DATABASE commands with the SPACENAM option or use the RECOVER utility to remove pages from the logical page list (LPL).

2. Restart any failed DB2s.

If your actions do not free up enough space, or if this problem continues to occur, you have the following options, depending on what level of MVS and coupling facility you have.

- If all of the following conditions are true:

  – All members of the group are running with MVS Version 5 Release 2 or above
  – The SCA is allocated in a coupling facility with a CFLEVEL greater than 0
  – The currently allocated size of the SCA is less than the maximum structure size as defined by the SIZE parameter of the CFRM policy

  Then you can enter the following command to increase the size of the SCA (this example assumes the group name is DSNDB0G):

  ```
  SETXCF START,ALTER,STRNAME=DSNDB0G_SCA,SIZE=newsize
  ```

  This example assumes that *newsize* is less than or equal to the maximum size defined in the CFRM policy for the SCA structure.

- If **any** of the following conditions are true:

  – Any member of the group is not running with MVS Version 5 Release 2 or above
  – The SCA is allocated in a coupling facility with CFLEVEL=0
  – The allocated size of the structure is already at the maximum size defined by the SIZE parameter in the CFRM policy

  Then you must:

  1. Increase the storage for the SCA in the CFRM policy SIZE parameter

  2. Use the MVS command SETXCF START,POLICY to start the updated policy

  3. Use the following MVS command to rebuild the structure:

     ```
     SETXCF START,REBUILD,STRNAME=DSNDB0G_SCA
     ```

- If **all** members are down, and you cannot alter the SCA to a larger size, you must do the following:

  1. Delete the SCA structure by using the following command:

     ```
     SETXCF FORCE,STRUCTURE,STRNAME=DSNDB0G_SCA
     ```

  2. Increase the size of the SCA in the CFRM policy.

  3. Restart DB2 to rebuild the SCA using group restart, as described in “Group Restart” on page 183.

Problem: Storage Shortage in the Lock Structure

Symptom: A DXR170I message indicates when storage reaches 50, 60, and 70% full. The following message appears at increasing thresholds starting at 80% full:

```
-DB1G DXR142I irlmx THE LOCK STRUCTURE structure-name
      IS zzz%  IN USE
```

System Action: DB2 continues processing, but some transactions might obtain a "resource unavailable" code because they are unable to obtain locks.

System Programmer Action: First, make sure no DB2s are down and holding retained locks. Restarting any failed DB2s can remove the locks retained in the coupling facility lock structure and release that space.

If a failed DB2 is not the problem, you have two courses of action:

- Lower the lock escalation values to get fewer locks. You do this either by lowering the value on the LOCKS PER TABLE(SPACE) of installation panel DSNTIPJ or by using the LOCKMAX clause of CREATE TABLESPACE.
- Increase the size of the lock structure, as described in "Changing the Size of the Lock Structure" on page 235.

Deallocating Structures by Force

If you cannot restart DB2 because the lock structure or SCA has been corrupted, you must force the structure to be deallocated. Then, DB2 can reconstruct the SCA or lock structure from the logs during group restart.

To deallocate structures, you must use MVS SETXCF FORCE commands to delete persistent structures or connections. Each DB2 structure requires a different set of commands.

- For the group buffer pools:

  ```
  SETXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL
  ```

- For the SCA:

  ```
  SETXCF FORCE,STRUCTURE,STRNAME=strname
  ```

- For the lock structure:

  ```
  SETXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL

  SETXCF FORCE,STRUCTURE,STRNAME=strname
  ```

Restarting DB2 after Termination

After a failure or after a normal shutdown of DB2, you can restart DB2 with the command START DB2. You can also choose to have DB2 automatically restart after a failure by using the automatic restart manager of MVS. See "Using MVS's Automatic Restart Capability" on page 60 for more information.

During restart, DB2 resolves inconsistent states. Restart is changed for data sharing because of the following:

- Database exception states, which exist solely on the log in a non-data-sharing environment, are on both the SCA and the log in data sharing.

- Locks that are retained in the event of a failure must be processed.
- If the SCA or the lock structure is lost and cannot be rebuilt on another coupling facility, all members of the group come down. If this unlikely event occurs, then DB2 must perform *group restart*. Group restart is distinguished from normal restart by the activity of rebuilding the information that was lost from the SCA or lock structure. Group restart does not necessarily mean that all DB2s in the group start up again, but information from all nonstarting DB2s must be used to rebuild the lock structure or SCA.

Normal Restart for a Data Sharing Member

Normal restart for a member of a data sharing group is very much the same as for a non-data-sharing DB2. In this section we describe some additional information about locks, because locks that are held by a failed member can affect the availability of data to the other members of the group that are still running DB2 applications.

Active and Retained Locks

When a DB2 member is active, the locks it holds are called *active locks*. For transaction locks (L-locks), the normal concurrency mechanisms apply, including suspensions and timeouts when incompatible locks are requested for a resource. For physical locks (P-locks), DB2 uses a negotiation process to control access. For more information about locking mechanisms, see “Improving Concurrency” on page 217.

The particular concern for availability is what happens to locks when a DB2 subsystem fails. For data sharing, active locks used to control updates to data (modify locks) become *retained* in the event of a failure. This means that information about those locks is stored in the coupling facility until they are released during restart. Retained locks are necessary to protect data in the process of being updated from being accessed by another active DB2 member of the group.

DB2 has various types of retained locks. Among them are L-locks, page set P-locks, or page P-locks. As long as an incompatible lock is held by failed member, another member cannot lock the resource in a mode that incompatible with the mode of the retained lock on that resource. Incompatible requests from other members are suspended if the RETLWAIT subsystem parameter is YES; otherwise, they are immediately rejected. See Table 12 on page 91 for more information about RETLWAIT.

In the event of a page set P-lock, it is conceivable that an entire page set could be unavailable (an X mode page set P-lock is retained if the page set was non-GBP-dependent at the time of the failure). Incompatible lock requests from other members can be processed after the retained locks are removed, which occurs during restart. To keep data available for all members of the group, **it is important to restart failed DB2s as soon as possible**, either on the same or another MVS.

| ***Utility ID Locks:*** When a member is running a utility, it is holding a lock on the utility ID (UID) for that utility. That lock, too, is retained should the member fail. This means you cannot restart a utility until that member is restarted and the retained lock is converted to an active lock.

When Retained Locks are Reacquired or Purged

During the restart process, DB2 will remove its retained locks in either of two ways:

- Convert the lock to active, called *reacquiring* the lock. This is what DB2 does for page set P-locks.
- Purge the lock. This is what DB2 does for page P-locks and for L-locks.

This process of reacquiring or purging locks can happen at different times in the restart process, depending on the type of retained lock as shown in Table 27.

Table 27. Restart Processing for Locks

Lock Type	Processing
Page set P-lock	Reacquired when page sets are opened for log apply. This generally happens during forward recovery.
Note: The earliest that a page set P-lock can become negotiable is at the end of forward recovery. See “Page Set P-Locks” on page 242 for more information about negotiation. If no log apply is needed, it can happen later, such as the end of restart processing.	
Page P-lock	Purged at the end of forward recovery.
L-lock	Purged at the end of restart processing.

If the DB2 requesting an incompatible retained lock has RETLWAIT set to YES, its applications can be suspended waiting for those retained locks. Those requests can go through as soon as the retained locks are purged or become negotiable. For example, if an application is suspended because of an incompatible retained page set P-lock, that retained lock most likely becomes active and available for negotiation at the end of forward log recovery.

Group Restart

Group restart requires scanning the logs of each member to rebuild the SCA or retained lock information. This is why we recommend that you have an alternate coupling facility on which these vital structures can be automatically rebuilt in the event of a coupling facility failure. That automatic rebuild does not require the log scans that group restart does.

During group restart, all restarting DB2s update the SCA or lock structure from information contained on their logs. If you don't enter a START DB2 command for all members of the group, then the started DB2 subsystems carry out group restart on behalf of the nonstarting subsystems by reading their logs.

Although one DB2 can perform restart on behalf of the group, we recommend that you restart all of the nonquiesced members together, perhaps by using an automated procedure. This shortens the total restart time. Also, because retained locks are held for nonstarting DB2s, it is best for data availability to start all members of the group.

Because all members must synchronize at the end of current status rebuild (CSR) and at the end of forward log recovery, the time taken for group restart done in parallel is determined by the member that has the longest CSR and, if the lock structure is lost, by the member that has the longest forward log recovery.

Once the members are synchronized after forward log recovery, backward log read proceeds in parallel for the started subsystems.

Phases of Group Restart

The phases of restart are generally the same as in a non-data-sharing environment, with the addition of function for group restart. The phases of group restart vary based on whether the SCA or lock structure is lost (or both), and whether information is needed from the logs of inactive members. Table 28 summarizes the phases, depending on which structure is lost.

Table 28. Summary of Phases Based on Which Structure Is Lost

SCA Lost	Lock Structure Lost
Initialization	Initialization
CSR (rebuild SCA)	CSR (reacquire page set P-locks)
Peer CSR (rebuild SCA)	Peer CSR (rebuild page set P-locks)
Forward recovery (rebuild locks)	Forward recovery (rebuild locks)
	Peer forward recovery (rebuild locks)
Backward Recovery	Backward Recovery

In the message output shown in this section, we are showing a group restart controlled by member DB3G on behalf of members DB2G and DB1G.

DB2 Initialization

This phase verifies BSDSs, logs and the integrated catalog facility catalog. The RBA of the last log record is retrieved and logging is set to begin at the next CI following the last RBA. Also during this phase, DB2 determines if the lock structure or SCA is lost and needs to be recovered.

During initialization, you see messages similar to the following:

```
 $HASP373 DB3GMSTR STARTED
⋮
 DSNJ127I @DB3GDB2 SYSTEM TIMESTAMP FOR BSDS= 95.040 13:03:05.32
 DSNJ001I @DB3GDB2 DSNJW007 CURRENT COPY 1 ACTIVE LOG 753
 DATA SET IS DSNAME=DSNC410.THIRD.LOGCOPY1.DS01,
 STARTRBA=000000000000,ENDRBA=000000167FFF
 DSNJ001I @DB3GDB2 DSNJW007 CURRENT COPY 2 ACTIVE LOG
 DATA SET IS DSNAME=DSNC410.THIRD.LOGCOPY2.DS01,
 STARTRBA=000000000000,ENDRBA=000000167FFF
 DSNJ099I @DB3GDB2 LOG RECORDING TO COMMENCE WITH
 STARTRBA=000000010000

⋮
 $HASP373 DB3GDBM1 STARTED
```

Current Status Rebuild (CSR)

During current status rebuild, the SCA is rebuilt from the log by reading it forward from the last checkpoint. In addition, DB2 determines all outstanding units of recovery (UR) that were interrupted by the previous termination. If the lock structure is lost, all partition and page set P-locks are reacquired by reading information from the log. These locks are “retained” locks until the end of restart.

When a restarting member has completed its own CSR, it checks and waits for every other DB2 member to finish CSR. If there are non-starting DB2 subsystems, then peer CSR is performed.

During current status rebuild, you see messages similar to these (The phrase in parentheses is not part of the output.):

```
DSNR001I @DB3GDB2 RESTART INITIATED
DSNR003I @DB3GDB2 RESTART...PRIOR CHECKPOINT RBA=00000000DC4E
DSNR004I @DB3GDB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
(End of current status rebuild for member DB3G)

DSNR021I @DB3GDB2 DSNRRGRC DB2 SUBSYSTEM MUST PERFORM
GROUP RESTART FOR PEER MEMBERS
```

Peer CSR (used only when some DB2s are not starting)

This activity is skipped unless it is necessary to perform group restart on behalf of non-starting members. (Peer CSR is not used when the non-starting DB2s are normally quiesced.)

A restarting DB2 can select an inactive member on which to perform peer initialization and peer CSR:

- If the SCA is lost, the restarting DB2 rebuilds SCA information from the information contained in the nonstarting member's logs.
- If the lock structure was lost, the restarting DB2 reacquires page set and partition P-locks (as retained locks) for the nonstarting member. Those locks are now retained locks.

When all members have completed current status rebuild, either by doing it on their own or by having a peer do it for them, the SCA has been rebuilt, and page set and partition P-locks have been reacquired.

During peer current status rebuild, you see messages similar to these:

```
DSNR023I @DB3GDB2 DSNRRGRC GROUP RESTART INITIATED TO
RECOVER THE SCA FOR GROUP MEMBER DB1G
DSNR003I @DB3GDB2 RESTART...PRIOR CHECKPOINT RBA=00000201CC4E
DSNR004I @DB3GDB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
DSNR024I @DB3GDB2 DSNRRGRC GROUP RESTART COMPLETED TO
RECOVER THE SCA FOR GROUP MEMBER DB1G
(End of peer current status rebuild for member DB1G)

DSNR023I @DB3GDB2 DSNRRGRC GROUP RESTART INITIATED TO
RECOVER THE SCA FOR GROUP MEMBER DB2G
DSNR003I @DB3GDB2 RESTART...PRIOR CHECKPOINT RBA=000000009C4E
DSNR004I @DB3GDB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
DSNR024I @DB3GDB2 DSNRRGRC GROUP RESTART COMPLETED TO
RECOVER THE SCA FOR GROUP MEMBER DB2G
(End of peer current status rebuild for DB2G)

DSNR022I @DB3GDB2 DSNRRGRC DB2 SUBSYSTEM HAS
COMPLETED GROUP RESTART FOR PEER MEMBERS
(End of peer processing)
```

Forward Log Recovery

In this phase DB2 applies log records and completes the database write operations which were outstanding at the time of the failure. It also rebuilds retained locks during this phase by reading that information from the log. Restart time is longer when lock information needs to be recovered than during a normal restart, because DB2 needs to go back to the earliest begin_UR for an inflight UR belonging to that subsystem. This is necessary to rebuild locks that member has obtained during the inflight UR. (A normal restart goes back only as far as the earliest RBA that is needed for database writes or is associated with the begin_UR of indoubt units of recovery.)

If there is a problem applying a log record for an object (such as if the DASD version of the data could not be allocated or opened), or if the page set is deferred, DB2 adds the relevant pages and page ranges to the logical page list. Only pages affected by the error are unavailable.

When each restarting member has completed its own forward log recovery, it checks and waits for every other DB2 member to finish. If there are non-starting DB2 subsystems, then peer forward log recovery is performed.

At the end of forward log recovery, the rebuild of the lock structure is complete.

During forward log recovery, you see messages similar to these:

```
DSNR005I @DB3GDB2 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR021I @DB3GDB2 DSNRRGRH DB2 SUBSYSTEM MUST PERFORM
GROUP RESTART FOR PEER MEMBERS
```

Peer Forward Log Recovery (used only when some DB2s are not starting)

This activity is skipped unless it is necessary to rebuild lock information from information contained in inactive, non-quiesced members' logs. Peer retained lock recovery requires that DB2 do a peer initialization, a partial CSR phase to rebuild UR status, and then do the forward log recovery for the nonstarted member.

During peer forward log recovery, you see messages similar to these:

```
DSNR025I @DB3GDB2 DSNRRGRH GROUP RESTART INITIATED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB1G
DSNR003I @DB3GDB2 RESTART...PRIOR CHECKPOINT RBA=00000201CC4E
DSNR004I @DB3GDB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
(End of peer partial current status rebuild for DB1G)

DSNR005I @DB3GDB2 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR026I @DB3GDB2 DSNRRGRH GROUP RESTART COMPLETED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB1G
(End of peer forward log recovery for member DB2G)

DSNR025I @DB3GDB2 DSNRRGRH GROUP RESTART INITIATED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB2G
DSNR003I @DB3GDB2 RESTART...PRIOR CHECKPOINT RBA=000000009C4E
DSNR004I @DB3GDB2 RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
(End of partial current status rebuild for member DB2G)

DSNR005I @DB3GDB2 RESTART...COUNTS AFTER FORWARD
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR026I @DB3GDB2 DSNRRGRH GROUP RESTART COMPLETED TO
RECOVER RETAINED LOCKS FOR GROUP MEMBER DB2G
DSNR022I @DB3GDB2 DSNRRGRH DB2 SUBSYSTEM HAS
COMPLETED GROUP RESTART FOR PEER MEMBERS
(End of peer forward log recovery for member DB1G)
```

Backward Log Recovery

At this point, all forward log applies have been performed and inflight, indoubt, and in-abort transactions are protected by locks. During this phase, DB2 completes recovery processing by reversing all changes performed for inflight and in-abort units of recovery.

Any updates that cannot be externalized to the group buffer pool or DASD cause the affected pages to be added to the logical page list.

Backward log recovery can occur in parallel for all the started subsystems. There is no peer backward log recovery; all members must be started to complete backward log recovery and release the locks held by in-flight and in-abort transactions.

During backward log recovery, you see messages like the following:

```
DSNR006I @DB3GDB2 RESTART...COUNTS AFTER BACKWARD
RECOVERY
INFLIGHT=0, IN ABORT=0
(End of backward log recovery for member DB3G)
DSNR002I @DB3GDB2 RESTART COMPLETED
DSN9022I @DB3GDB2 DSNYASCP 'START DB2' NORMAL COMPLETION
```

The backward log recovery messages for the other members do not appear until those members are actually started.

Restarting a DB2 Member with Conditions

As described in Section 4 (Volume 1) of *Administration Guide*, you might, in unusual circumstances, choose to make inconsistent data available for use without recovering it. This might be the case for certain test groups, for example, where data consistency is not important.

Some installations use conditional restart to bypass a long active UR backout, such as might occur when a long-running batch job fails without having issued interim commits. In data sharing, this use of conditional restart is not recommended. It is safer, and you have better availability, to reroute work to another DB2rather than suffering the total outage necessary for a conditional restart.

If you do use conditional restart, it is necessary to stop all members of the group other than the one that is conditionally restarting to ensure that applications on those other members do not change the data that is not locked by the restarting member.

This section describes two procedures:

- "Procedure for cold starting a member (STARTRBA = ENDRBA)"
- "Procedure for other conditional restarts (STARTRBA ≠ ENDRBA)" on page 189

Procedure for cold starting a member (STARTRBA = ENDRBA)

Use this procedure for cold start; that is, when the STARTRBA = ENDRBA on the change log inventory's CRESTART statement.

1. Stop all other members of the data sharing group.
2. Cold start the chosen member using ACCESS(MAINT).

   The cold start deallocates the group buffer pools to which this DB2 was connected. All GBP-dependent page sets are then placed in GRECP status.
3. Use START DATABASE to recover the GRECP page sets. This step is not necessary if you are planning to recover to a prior point-in-time.
4. Resolve all data inconsistency problems resulting from the cold start. For more information about how to do this, see Section 4 (Volume 1) of *Administration Guide*.

5. After you have resolved the data inconsistencies resulting from the cold start, you can start all the other members and restart this one without ACCESS(MAINT).

Procedure for other conditional restarts (STARTRBA ≠ ENDRBA)

Use this procedure when you are truncating the log but are not doing a cold start:

1. Stop all other members of the data sharing group.
2. Conditionally restart the chosen member using ACCESS(MAINT).
3. Resolve all data inconsistency problems resulting from the cold start. For more information about how to do this, see Section 4 (Volume 1) of *Administration Guide*.
4. After you have resolved the data inconsistencies resulting from the conditional restart, you can start all the other members and restart this one without ACCESS(MAINT).

Deferring Recovery during Restart

It is possible to defer recovery for an object during restart. If you use the DEFER installation option, that defers the log apply processing of the object for only the member who specified that option. All the pages that would have been applied to DASD or the group buffer pool are instead added to the logical page list. This can affect the rest of the group; any member who needs a page that is on the LPL will not be able to access that page until the object is restarted.

You have to use the START DATABASE command with the SPACENAM option to make those pages available after DB2 has restarted.

Deferring recovery does not change restart time significantly.

Starting and stopping duplexing for a group buffer pool

This section describes how you can start and stop duplexing for a particular group buffer pool.

Starting duplexing

When duplexing starts, there is a period in which activity to the group buffer pools is quiesced. Thus, it is best to start duplexing during a period of low activity in the system.

There are two ways to start duplexing for a group buffer pool:

- Activate a new CFRM policy with DUPLEX(ENABLED) for the structure. If the group buffer pool is currently allocated, then MVS can automatically initiate the process to establish duplexing as soon as you activate the policy. If the group buffer pool is not currently allocated, then the duplexing process can be initiated when the group buffer pool is allocated.

  For MVS to automatically initiate duplexing, all CFRM policy parameters other than DUPLEX must be the same as they were before. For example, if you change both the DUPLEX value and the SIZE value, neither value takes effect until you manually rebuild the group buffer pool.

- Activate a new CFRM policy with DUPLEX(ALLOWED) for the structure. If the group buffer pool is currently allocated, use the following command to start the duplexing rebuild:

  ```
  SETXCF START,REBUILD,DUPLEX,STRNAME=strname
  ```

  If the group buffer pool is not currently allocated, wait until it is allocated before starting the duplexing rebuild.

While duplexing is being established, and for the entire time duplexing is in effect, a display of the structure shows structure as being in DUPLEXING REBUILD. The rebuild phase is called DUPLEX ESTABLISHED, which means that duplexing is truly active for the structure. See Figure 73 on page 255 for an example.

Stopping duplexing

To stop duplexing, you must first decide which instance of the group buffer pool is to remain as the surviving simplexed group buffer pool. If you have a choice in the matter, it is preferable to use the primary one as the surviving one. The primary group buffer pool has intact page registration information, whereas to switch to the secondary one, all the page registration is lost. Therefore, all locally cached pages revert to an invalid state and must be refreshed from the group buffer pool or DASD.

If you want to switch temporarily to one of the structures as a simplexed group buffer pool, use the following method:

1. Optional: If DUPLEX(ENABLED) is specified for the active CFRM policy, activate a new policy specifying DUPLEX(ALLOWED). For the new DUPLEX value to take effect, all other CFRM policy parameters must be the same as before.

   This first step is necessary only if you have at least 3 coupling facilities and you do not do not want to automatically reestablish duplexing after stopping it.

2. Use the SETXCF STOP,REBUILD command, specifying KEEP=OLD to revert to using the primary structure as the primary, or KEEP=NEW to switch to the secondary instance as the simplexed structure. For example, the following command reverts to using the primary instance as the simplexed group buffer pool:

   ```
   SETXCF STOP,RB,DUPLEX,STRNAME=strname,KEEP=OLD
   ```

If you do not plan on reestablishing duplexing for the group buffer pool in the near future, then activate a new CFRM policy specifying DUPLEX(DISABLED) for the structure.

Shutting down the Coupling Facility

Make a plan for those cases where it is necessary to shut down a coupling facility to apply maintenance or perform some other type of reconfiguration. For the least disruptive shutdown, move all your structures to another coupling facility before shutting it down. This section gives some recommendations for how to handle this event for DB2. For other structures in the coupling facility, see the appropriate product documentation.

Consider the following:

1. Prepare for the move:
   - Make sure that you have enough room on the alternate coupling facility for all structures you intend to move.
   - Make sure that the preference list for the group buffer pool, SCA, and lock structures contains the alternate coupling facility information.
2. Move simplexed structures to the **new** coupling facility:

   SETXCF START,REBUILD,CFNAME=*newcf*,LOC=OTHER

   That command rebuilds all structures that allow rebuild onto the alternate coupling facility.
3. For duplexed structures, issue the command to deallocate the duplexed structures on the **target** coupling facility:

   SETXCF STOP,REBUILD,DUPLEX,CFNAME=*targetcf*

   If the CFRM policy specifies DUPLEX(ALLOWED) or DUPLEX(DISABLED), the structure goes into simplex mode.

   If the CFRM policy specifies DUPLEX(ENABLED), MVS tries to automatically restart duplexing. If you have a third coupling facility specified in the CFRM policy, it is possible to continue duplexing during the outage of the target coupling facility. When an operator command causes the structure to drop from duple to simplex mode, MVS avoids automatially reduplexing the structure back into the same coupling facility from which one of the duplexed instances of the structure was just deallocated. Instead, it duplexes into that third coupling facility.

Chapter 7. Performance Monitoring and Tuning

One of the main objectives of the data sharing function of the licensed program DB2 for OS/390 is to increase processing capacity while using the lower cost S/390 Parallel Sysplex technology. Work load capacity is increased by allowing many DB2 subsystems to access shared DB2 databases with full integrity. DB2 data sharing has been designed to address this objective while providing balanced performance for a broad range of SQL applications.

DB2 gives you the power of data sharing while avoiding overhead whenever possible for such things as global locking and data caching. However, there are also actions you can take to reduce the performance cost of data sharing. The purpose of this chapter is to describe briefly how data sharing locking and buffer management work and to offer some guidance about how to improve performance in the group.

This chapter describes the following topics:

- “Monitoring Tools”
- “Improving the Performance of Data Sharing Applications” on page 197
- “Improving the Response Time for Read-Only Queries” on page 199
- “Improving Concurrency” on page 217
- “Tuning Group Buffer Pools” on page 237
- “Access Path Selection in a Data Sharing Group” on page 268

Monitoring Tools

This section describes the following tools:

“Using Resource Measurement Facility (RMF) Reports”
“Using DB2 Trace” on page 196
“Using DB2 PM” on page 196

Using Resource Measurement Facility (RMF) Reports

The Resource Measurement Facility Version 5 (RMF) provides single system and Sysplex views by reporting on collected resource usage data:

- The Sysplex Summary report provides an integrated view of the entire Sysplex on one screen.
- The Response Time Distribution report gives details about the distribution of response times on a Sysplex level with the capability of zooming into a single system that indicates problems.
- The Coupling Facility reports have information about storage allocation, structure activity, and subchannel activity to allow you to plan for better resource utilization. See Figure 62 on page 231 and Figure 74 on page 256 for examples of Coupling Facility Activity reports.
- The Shared Device report provides information about how DASD and tape resources are shared among the different systems in the Sysplex.

Using DB2 Trace

DB2 writes trace records to help you monitor events in the DB2 group. Many trace classes now include information about locking and use of the group buffer pool.

A single DB2 trace does not gather information about all the members of the group. All the trace commands (such as START TRACE and MODIFY TRACE) take effect only on the DB2 subsystem on which they are issued; they do not affect tracing activity on any other DB2 subsystems. In addition, the statistics interval for each subsystem is set by each member and the interval is not synchronized across the members of the group.

To gather data about group activity, as when auditing access to a particular table or detecting locking contention from processes on different DB2 subsystems, you must merge trace records in time sequence. To do this, use the value of the Sysplex Timer store clock contained in the standard header. Every trace record from a member of a data sharing group includes a data sharing header, which gives the DB2 group name and member name. This header is mapped by mapping macro DSNDQWHA. For more information about the format of trace records, see Appendix D (Volume 2) of *Administration Guide*.

Using DB2 PM

DB2 PM can monitor the use of shared DB2 resources such as the catalog and directory, group buffer pools, and databases for entire data sharing groups as well as for individual members of the group.

- Reports and traces with *member scope* present a group's instrumentation data member by member, without merging the data. These reports and traces are similar to those generated in a non-data-sharing environment, where each report or trace is produced for an individual DB2 subsystem.
- Reports and traces with *group scope* merge instrumentation data for individual members and present it for the entire group. The traces show events in chronological order and indicate which member is reporting the event, and the reports show events summarized for the entire group.

The following report sets provide both group-scope and member-scope reporting:

- Accounting
- Locking
- Statistics
- Audit

Group-scope reporting is also available in exception processing and graphics: you can define exception thresholds for groups, and you can create graphs showing performance trends for an entire data sharing group.

With DB2 PM, you can have processor times reported in service units. This allows times from different CPC models in the group to be normalized before they are reported.

See Figure 63 on page 232 and Figure 64 on page 233 for examples of DB2 PM reports.

Improving the Performance of Data Sharing Applications

Many of the things you currently do for a single DB2 to improve response time or reduce processor consumption also hold true in the Parallel Sysplex. For example, most of the recommendations for reducing locking overhead described in “Improving Concurrency” on page 217 are the same for a single system.

However, if the processing speed of the S/390 microprocessor cluster is slower than that on which you are currently running, be sure to plan carefully for applications or DB2 utility processes where processor resource consumption represents the significant part of the elapsed time. Some examples are some batch applications, complex queries, and many DB2 utilities. This section describes ways to make these resource-intensive applications run at their best in the Parallel Sysplex. The topics are:

See “Improving the Response Time for Read-Only Queries” on page 199 for information about using query parallelism in a data sharing group. That section also includes information about governing resources in a data sharing group.

General Recommendation

We suggest that you take advantage of data partitioning and that you design for parallel processing whenever possible. DB2 can use parallel processing effectively only when the data is partitioned. See Section 5 (Volume 2) of *Administration Guide* for more information about parallel processing for queries and utilities.

Migrating Batch Applications to Data Sharing

If the significant portion of the elapsed time of a long-running batch application is because of processor resource consumption, consider the relative processing speed of the processor on which you are currently running to that of the S/390 microprocessor clusters. If this causes a scheduling problem, consider running more than one copy of the same program in parallel. Each copy can be run on a different key range, typically the partitioning key of the main table.

To avoid DASD contention, make sure the data partitions are placed on DASD devices with separate I/O paths. Consider using the PIECESIZE option of CREATE or ALTER INDEX to give you more control over the data set size of nonpartitioning indexes. See Chapter 6 of *SQL Reference* for more information about the PIECESIZE option.

If your batch application cannot be redesigned to run on separate partitions, consider running batch jobs on a CPC in the group that has a faster single-CP speed. This approach is recommended only if the application does not access the coupling facility at a high intensity.

If your application does heavy sequential insert processing from multiple members in the group, consider putting the data in a table space that is defined with the

MEMBER CLUSTER option. The MEMBER CLUSTER option causes DB2 to insert the data based on available space rather then respecting the clustering index or the first index. For sequential insert applications, specifying MEMBER CLUSTER can reduce P-lock contention and give better insert performance at the cost of possibly higher query times. If you must also run query work that scans many rows on this data, run REORG to put the data in clustering order before running the query workload. See “Reducing Space Map Page Contention” on page 243 for more information about the MEMBER CLUSTER option of CREATE TABLESPACE.

Migrating Online Applications

The response times for online applications are usually not degraded solely by limited processing power. They can most effectively use the Parallel Sysplex to run many transactions in parallel. Any slight increase in transaction elapsed time caused by running on the S/390 microprocessors is offset by running transactions across many CPCs. This high level of parallelism can use the available capacity of the Sysplex to deliver equivalent or higher levels of throughput compared to running on a high-end processor.

The release of MVS and coupling facility level can affect performance of applications that use sequential or list prefetch. See “Prefetch Processing” on page 245 for more information.

Running Utilities

For most DB2 utilities, processor resource consumption represents the significant portion of the elapsed time when running on a S/390 microprocessor cluster. Two of the most significant examples are:

LOAD with many columns
RUNSTATS with non-index column statistics

Consider using the SAMPLE option of RUNSTATS to reduce processing costs. The key to letting processor-intensive utilities perform well is to partition the data. For more information about running utilities on separate partitions of data, see *Utility Guide and Reference*.

Where partitioning the data is not an option, consider running utilities on a CPC with a faster single-CP speed.

Using the Resource Limit Facility (Governor)

The resource limit facility (governor) controls the amount of processor time that any dynamic, manipulative SQL statement (SELECT, INSERT, UPDATE, DELETE) can consume in DB2. Different members of a data sharing group can use the same or different resource limit specification tables (RLSTs). Each RLST must have a unique name within the data sharing group. For information about using the resource limit facility to control queries using Sysplex query parallelism, see “Setting Limits using the Resource Limit Facility” on page 213.

Controlling the RLST

The commands used to control the RLST (DISPLAY RLIMIT, START RLIMIT, and STOP RLIMIT) affect only the DB2 on which the command is issued. The same holds true if you start the RLST at DB2 startup by using a system parameter.

Dropping Objects in the Resource Limit Facility (Governor)

While an RLST is active in any DB2 in the group, you cannot drop any object associated with any active RLST.

Restrictions for STOP and START DATABASE

While an RLST is active in any DB2, you cannot enter a STOP DATABASE command for a database or table space that contains the active RLST, nor can you start the database or table space with ACCESS(UT).

Improving the Response Time for Read-Only Queries

The response time of a complex SQL query can be constrained by processor resources when it runs on a single central processor (CP). For complex queries, run the query in parallel within a member of a data sharing group, and, with Sysplex query parallelism, use the full power of the data sharing group to process individual complex queries on many members of the data sharing group. The information in this section is based upon information about query parallelism in Section 5 (Volume 2) of *Administration Guide*, but this section emphasizes how you can use the full power of the data sharing group to run your complex queries.

Applications that are primarily read-only and processor-intensive benefit from Sysplex query parallelism. If a query qualifies for parallel processing, DB2 determines the optimal degree of parallelism at bind time. DB2 can distribute parts of the query to be processed on other members of the data sharing group.

Terminology: The DB2 that is attached to the application that issued the query is the *parallelism coordinator*. A member who helps process one of these parallel queries is an *assisting DB2* or *parallelism assistant*.

How Data is Returned to the Parallelism Coordinator: Data is returned to the parallelism coordinator in one of two ways:

- By work file

  If the query requires the use of a work file (because the data needs to be sorted, for example), then the parallelism coordinator can access the data from the parallelism assistants' work files. Each assistant writes to its own work file, and the coordinator can read the results from the assistants' work files.

- By cross-system coupling facility (XCF) links

  For data that does not require the use of a work file, XCF is used. XCF traffic can be transported using channel-to-channel connections between the CPCs, or by using the coupling facility.

The follow topics are described in this section:

- “Planning for Sysplex Query Parallelism” on page 200
- “Enabling Sysplex Query Parallelism” on page 203
- “Monitoring and Tuning Parallel Queries” on page 207
- “Disabling Sysplex Query Parallelism” on page 215

Planning for Sysplex Query Parallelism

This section describes the following tasks:

- "Configuring the Systems"
- "Setting Workload Management Goals"
- "Designing the Database" on page 203

Configuring the Systems

Before you can use Sysplex query parallelism, you must configure the members of the data sharing group to allow it. In summary:

- The original query must execute on a member of a data sharing group that is running DB2 Version 5 and MVS/ESA Version 5 Release 2. Specify YES in the COORDINATOR field of installation panel DSNTIPK for that DB2 to allow it to send query work to other members. The COORDINATOR subsystem parameter controls whether this DB2 can send parallel tasks out to other DB2s in the data sharing group. If the COORDINATOR parameter is not YES at run time, the query runs within a single DB2. This is reported in the DB2 PM accounting report in Figure 53 on page 210 in field **C**.
- To be considered as candidates for assisting the parallelism coordinator, the other members must run DB2 Version 5 and MVS/ESA Version 5 Release 2. YES must be specified in the ASSISTANT field of installation panel DSNTIPK. The ASSISTANT subsystem parameter controls whether this DB2 can receive parallel tasks from another DB2 in the data sharing group. If this DB2 is the coordinator for a particular query, then its ASSISTANT parameter is not relevant.
- Work files must be on shared DASD that is accessible from all members that can process parallel queries. Keep work files in a separate buffer pool that is not backed by a group buffer pool. Use the same buffer pool number for all members of the data sharing group. DB2 assumes at run time that all work files are in the same numbered buffer pool.
- Group-wide goals for workload management should be defined for both work that originates on a particular DB2 and work that is processed by that DB2 on behalf of another. See "Setting Workload Management Goals."
- Buffer pools that can assist with processing queries from other members must have a VPXPSEQT buffer pool threshold greater than 0. Because VPXPSEQT is a subset of VPPSEQT and VPSEQT, these values must be greater than 0, too. See "Buffer Pool Threshold for Parallelism Assistants" on page 205 for more information.
- For optimal use of processor resources, run only one DB2 at a time on a given CPC.

Setting Workload Management Goals

It is important to define how you want MVS to handle the work for Sysplex query parallelism. The task of classifying work that runs on the parallelism coordinator is the same as in the past. However, you must also classify work that runs on the assistant DB2. If you do not classify DB2 as an assistant, the following outcomes can occur:

- If you are running in compatibility mode, the part of the query that runs on the assistant has the same priority as the DB2 database services address space. This is probably not what you want.

- If you are running in goal mode, the part of the query that runs on the assistant is discretionary work. Discretionary work runs at the priority usually reserved for very low-priority batch work.

To set workload management goals for parallel queries that this DB2 assists, you must:

1. Define a service class that is used for queries that run on an assistant. This service class can have the same goals as those for the coordinator, but the name must be different unless you are running with OS/390 Release 3 or subsequent releases. Alternatively, you can apply modified goals to the assistants to take into account the numerous amount of enclaves per assistant for a single query. See “Example: Setting Goals for the Parallelism Assistants” on page 202 for information about defining a service class.
2. If you run in compatibility mode, use the IEAIPS*xx* PARMLIB member to indicate which performance group should control the query work. Also, use the IEAICS*xx* PARMLIB member to set the SRVCLASS parameter to the service class defined in the policy, and associate it with a performance group. The subsystem type you use is DB2.

   As described in “How Period Switches Work on Parallelism Assistants,” any work coming into an assistant starts in period 1.
3. Install the service definition and activate the service policy.

For more information about using workload management, see *MVS/ESA Planning: Workload Management*.

How Period Switches Work on Parallelism Assistants

For work performed within a single system, work is governed by specifying *periods*. Each period has a service unit value and a priority attached to it. If you run in WLM compatibility mode, you must specify these values in the IEAIPS*xx* PARMLIB member. If you run in WLM goal mode, those values are assigned by the system. When a piece of work exceeds the service units for a given period, then a period switch occurs and the piece of work is switched to the next period, which probably has a lower priority attached to it. The dispatching priority is reevaluated after a period switch.

With Sysplex query parallelism, the query work for a particular query always starts out in the first period on each assistant with no accumulated service units.

Recommendation: Initially, we recommend that you classify work for the assistants using only one performance period. You can add more performance periods after you monitor the system.

Example: Setting Goals for the Parallelism Coordinator

Although this work has probably already been done for the coordinating DB2, this section illustrates the relationship between service classes for the parallelism coordinator and for the assistant. Figure 43 on page 202 is a an example of how you might classify complex queries that originate from TSO. Note that SUBSYSTEM TYPE is TSO.

```
**********************************************************************
Subsystem-Type Xref Notes Options Help
----------------------------------------------------------------------
Modify Rules for the Subsystem Type       Row 1 to 5 of 5
 Command ===> ____________________________________________ SCROLL ===>
 Subsystem Type . : TSO        Fold qualifier names?   Y  (Y or N)
 Description  . . . Decision support TSO
 Action codes:  A=After    C=Copy        M=Move     I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
-------Qualifier-------------           -------Class--------
Action    Type       Name     Start               Service     Report
                                       DEFAULTS: RTDISC     ________
 ____ 1 UI        EMMES    ___                  RTVEL       ________
 ____ 2    AI        D44*     ___                ________    PROJ1
 ____ 2    AI        D88*     ___                ________    PROJ2
 ____ 1 UI        BRYAN    ___                  RTONLY      ________
 ____ 2    AI        P73*     ___                ________    PROJ1
***************************** BOTTOM OF DATA *************************
```

Figure 43. Defining Workload Management Information for the Parallelism Coordinator

The service classes shown in Figure 43 are used for queries that originate and run on a particular DB2. If you are running on a release of OS/390 before release 3, any part of the same query that is shipped to another DB2 must use a different service class name, because WLM cannot manage enclave work and address space work with the same name. Report classes can be the same.

Example: Setting Goals for the Parallelism Assistants

Figure 44 shows one way to classify work on the assistant. In Figure 44, all query work originating from a given environment (such as TSO) has the same goals, but each work environment has its own goals.

```
**********************************************************************
Subsystem-Type Xref Notes Options Help
----------------------------------------------------------------------
Modify Rules for the Subsystem Type       Row 1 to 2 of 2
 Command ===> ____________________________________________ SCROLL ===>
 Subsystem Type . : DB2       Fold qualifier names?   Y  (Y or N)
 Description  . . . Participants in complex queries
 Action codes:  A=After    C=Copy        M=Move     I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
-------Qualifier-------------           -------Class--------
Action    Type       Name     Start               Service     Report
                                       DEFAULTS: Z          ________
 ____ 1 SI        TSO      ___                  ZTSODB2     ________
 ____ 1 SI        JES      ___                  ZJESDB2_    ________
***************************** BOTTOM OF DATA *************************
```

Figure 44. Classifying Query Work on the Assistants. The DB2 in the SUBSYSTEM TYPE field specifies that these classifications are for query work originating from another member of the data sharing group.

Figure 45 on page 203 shows another way that work can be classified. It more clearly illustrates the relationship between the work classified on the coordinator in Figure 43 and work classified on the assistant. The SUBSYSTEM TYPE is DB2 and the service class names are different.

```
************************************************************************
Subsystem-Type Xref Notes Options Help
------------------------------------------------------------------------
Modify Rules for the Subsystem Type       Row 1 to 7 of 7
 Command ===> ____________________________________________ SCROLL ===> PAGE
 Subsystem Type . : DB2         Fold qualifier names?   Y  (Y or N)
 Description  . . . Participants in complex queries
 Action codes:  A=After    C=Copy         M=Move     I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
-------Qualifier-------------            -------Class--------
Action    Type       Name     Start                Service     Report
                                       DEFAULTS: Z            ________
 ____ 1 SI        TSO      ___                  ZRTDISC       ________
 ____ 2   UI         EMMES    ___               ZRTVEL        ________
 ____ 3     AI         D44*     ___              ________     PROJ1
 ____ 3     AI         D88*     ___              ________     PROJ2
 ____ 2   UI         BRYAN    ___                ZRTONLY      ________
 ____ 3     AI         P73*     ___              ________     PROJ1
 ____ 1 SI        JES      ___                   ZRTDISC_     PROJ2___
****************************** BOTTOM OF DATA **************************
```

Figure 45. Classifying Work for the Parallelism Assistants

Designing the Database

Ideally, you want as much partitioning of table spaces as possible and as much separation of I/O as possible. Section 5 (Volume 2) of *Administration Guide* contains some detailed guidance on how to partition to achieve your performance goals.

Define Table Spaces and Indexes with GBPCACHE CHANGED: Because it is unlikely that DB2s processing a large query will repeatedly read the same pages, there is no need to cache those pages in the group buffer pool. Define the relevant table spaces and indexes with GBPCACHE CHANGED (the default).

Enabling Sysplex Query Parallelism

There is no change for how to enable parallel processing within your application. To **allow** parallel processing:

If query is:	**Use:**	**In:**
Static	DEGREE(ANY)	BIND or REBIND subcommand
	`PARALLEL DEGREE ==> ANY`	DB2I DEFAULTS panel for BIND or REBIND PACKAGE or PLAN
Dynamic	`SET CURRENT DEGREE = 'ANY'`	A previous SQL statement

Note: Installation panel DSNTIPF allows you to set the default degree (1 or ANY) for the CURRENT DEGREE special register.

There is also a new subsystem parameter to allow DB2s to coordinate queries and another parameter for allowing them to assist. There is also a new *assisting parallel sequential threshold* to determine how much of the buffer pool parallel sequential threshold applies to assisting other members with processing a queries in parallel.

Not all Queries are Eligible: Even if you turn on parallelism, not all read-only queries are eligible. For example, DB2 does not choose Sysplex query parallelism if the plan or package is bound with RR or RS isolation. If you want to use RR or RS isolation with with Sysplex query parallelism, bind with CS or UR and then use a LOCK TABLE *table-name* IN SHARE MODE statement before the query.

See Section 5 (Volume 2) of *Administration Guide* for more general parallelism restrictions.

Subsystem Parameters

For a DB2 to be a coordinator of parallelism, you must specify YES in the COORDINATOR field of installation panel DSNTIPK. To allow a DB2 to assist with parallelism, you must specify YES in the ASSISTANT field of the same panel.

At run time, assistants must have buffer pools defined to allow parallelism, or DB2 does not send work to those members.

Easy Way to See Coordinators and Assistant Parameters: The command DISPLAY GROUP now has a DETAIL option that allows you to see the COORDINATOR and ASSISTANT subsystem parameters for all active members of the group. For example, the following command:

```
-DB1G DISPLAY GROUP DETAIL
```

Displays output similar to Figure 46.

```
DSN7100I -DB1G DSN7GCMD
*** BEGIN DISPLAY OF GROUP(DSNDB0G ) GROUPLEVEL(510)
                                     GROUP ATTACH NAME(DB0G)
--------------------------------------------------------------------
DB2                                     DB2  SYSTEM   IRLM
MEMBER   ID  SUBSYS CMDPREF   STATUS    LVL  NAME     SUBSYS IRLMPROC
-------- --- ----   --------  -------- ---  -------  ----   --------
DB1G       1 DB1G   -DB1G     ACTIVE   510  MVS1     DJ1G   DB1GIRLM
DB2G       2 DB2G   -DB2G     ACTIVE   510  MVS2     DJ2G   DB2GIRLM
DB3G       3 DB3G   -DB3G     QUIESCED 510  MVS3     DJ3G   DB3GIRLM
DB4G       4 DB4G   -DB4G     ACTIVE   510  MVS4     DJ4G   DB4GIRLM
--------------------------------------------------------------------
DB2      PARALLEL    PARALLEL
MEMBER   COORDINATOR ASSISTANT
-------- ----------- ---------
DB1G             YES        NO
DB2G             YES       YES
DB3G            ****      ****
DB4G              NO        NO
--------------------------------------------------------------------
SCA   STRUCTURE SIZE:     1024 KB, STATUS= AC,   SCA IN USE:    11 %
LOCK1 STRUCTURE SIZE:     1536 KB
NUMBER  LOCK ENTRIES:      262144
NUMBER  LIST ENTRIES:        7353, LIST ENTRIES  IN USE:           0
*** END DISPLAY OF GROUP(DSNDB0G )
DSN9022I -DB1G DSN7GCMD 'DISPLAY GROUP ' NORMAL COMPLETION
```

Figure 46. DISPLAY GROUP with DETAIL Option

See *Command Reference* for more information.

Buffer Pool Threshold for Parallelism Assistants

Use the VPXPSEQT value to tell DB2 how much of the buffer pool resources can be used for assisting a coordinator with processing parallel queries. Figure 47 shows the relationship among the various buffer pool thresholds.

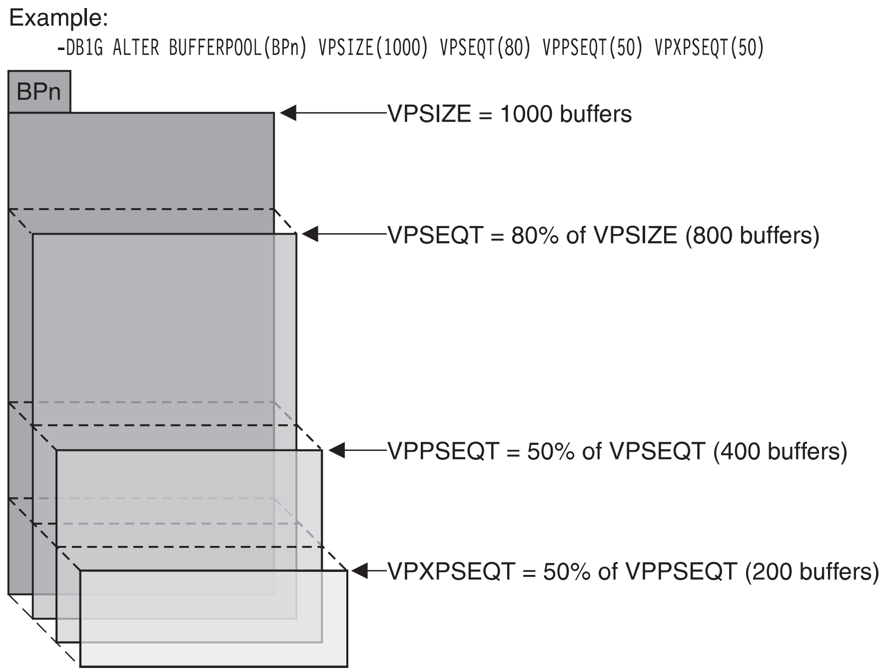

Figure 47. Relationships of Buffer Pool Thresholds. The threshold used to determine the allocation of resources for Sysplex query parallelism is determined by VPXPSEQT, which is a subset of the VPPSEQT threshold.

Sample Configurations: Let's look at how the buffer pool parallel allocation threshold values might be set up for two different configurations.

Figure 48 shows how the online systems (IMS and CICS) are configured to send all of their parallel queries to the query processor subsystems on CPC7 and CPC8, but they do not assist other members.

- The COORDINATOR subsystem parameter is YES
- The ASSISTANT subsystem parameter is NO

The TSO and BATCH systems can run their own queries in parallel and can send query work to the query processors, but they do not assist other members.

- The COORDINATOR subsystem parameter is YES
- The ASSISTANT subsystem parameter is NO

The query processors have both ASSISTANT and COORDINATOR set to YES.

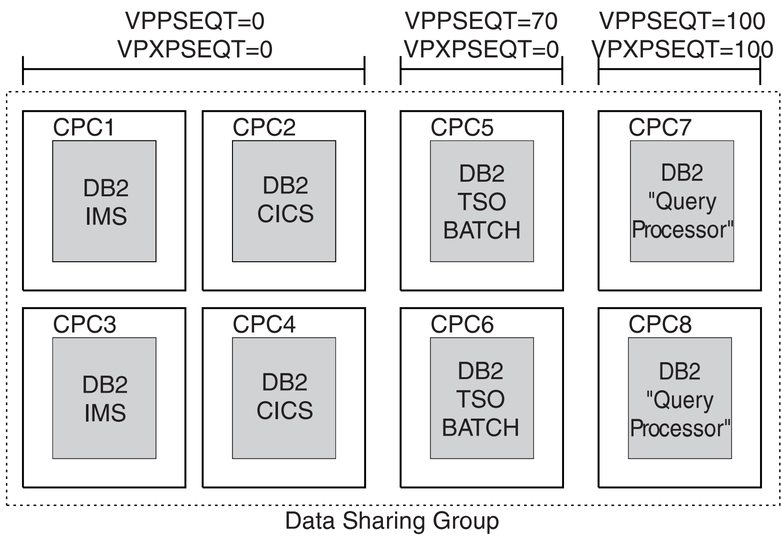

Figure 48. Separate Query Processor Configuration. In this configuration, the transaction systems (IMS and CICS) are configured for maximum transaction throughput. They do not assist in processing queries for other members (ASSISTANT=NO), and no query parallelism takes place on those subsystems.

Figure 49 shows a data sharing group in which all members can coordinate and assist with parallel queries. All COORDINATOR and ASSISTANT subsystem parameters are set to YES.

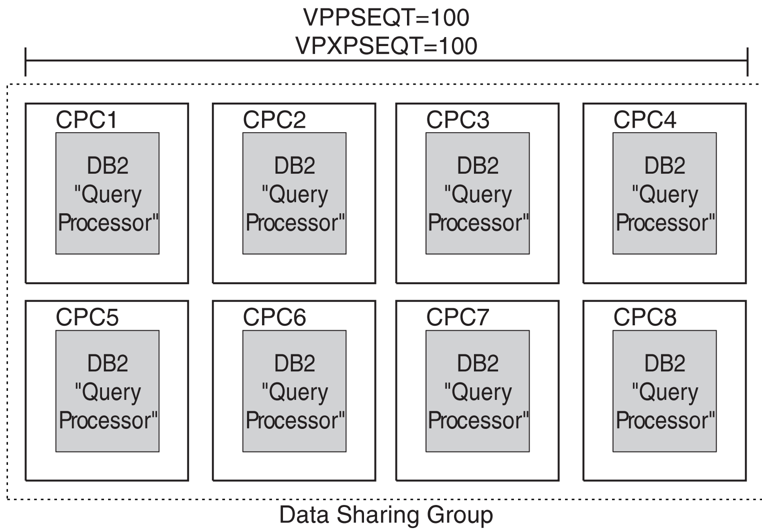

Figure 49. Dedicated Query Processor Configuration. In this configuration, the entire data sharing group is a set of "query processors." The virtual pool sequential threshold (VPSEQT) is presumably set very high (perhaps 100).

Displaying the Buffer Pool Thresholds: The DB2 command DISPLAY BUFFERPOOL displays all buffer pool thresholds, including the assisting parallel sequential threshold, as shown in Figure 50.

```
-DB1G DISPLAY BPOOL(BP1)
DSNB401I -DB1G BUFFERPOOL NAME BP1, BUFFERPOOL ID 1, USE COUNT 76
DSNB402I -DB1G VIRTUAL BUFFERPOOL SIZE = 20000 BUFFERS
             ALLOCATED       =    20000   TO BE DELETED   =        0
             IN-USE/UPDATED  =    16345
DSNB403I -DB1G HIPERPOOL SIZE = 1000 BUFFERS, CASTOUT = YES
             ALLOCATED       =        0   TO BE DELETED   =        0
             BACKED BY ES    =        0
DSNB404I -DB1G THRESHOLDS -
            VP SEQUENTIAL       =100   HP SEQUENTIAL          = 75
            DEFERRED WRITE      = 85   VERTICAL DEFERRED WRT  = 80
            PARALLEL SEQUENTIAL = 50   ASSISTING PARALLEL SEQT=100
DSN9022I -DB1G DSNB1CMD '-DISPLAY BPOOL' NORMAL COMPLETION
```

Figure 50. Displaying Buffer Pool Thresholds.. In this particular buffer pool, 50 percent of the buffer space is available for parallel processing. All of that parallel space is available to assist with processing queries from other members of the data sharing group.

Monitoring and Tuning Parallel Queries

This section describes the following:

"Using DISPLAY THREAD" on page 208
"Using a Performance Monitor" on page 209
"How DB2 Reports What It Did" on page 209
"Monitoring Processor Use" on page 209
"Improving Response Time" on page 210
"Controlling Resources Used by Parallel Operations" on page 213

Using DISPLAY THREAD

Although DISPLAY THREAD is not a group-wide command, it can display information about parallel tasks associated with their originating task. If a thread is using Sysplex query parallelism, issuing a DISPLAY THREAD on any assisting DB2 gives you the thread token of the originating task and the query coordinator's member name. The status PT is used in the display to indicate parallel tasks. All parallel tasks are displayed immediately after their corresponding originating thread.

See Chapter 2 of *Command Reference* for information about the syntax of the command DISPLAY THREAD.

Display on the Parallelism Coordinator: This example shows an allied, nondistributed originating thread (TOKEN=30) that is established (plan allocated) along with all of its parallel tasks, which are running on DB2 members DB1G and DB2G. Because the originating thread is running on DB1G, it is the coordinating DB2.

```
- 17.08.44           -DB1G DISPLAY THREAD(*)
- 17.08.44 STC00090  DSNV401I -DB1G DISPLAY THREAD REPORT FOLLOWS -
- 17.08.44 STC00090  DSNV402I -DB1G ACTIVE THREADS -
- NAME     ST A   REQ ID           AUTHID   PLAN     ASID TOKEN
- BATCH    T  *     1 PUPPYDML     USER001  DSNTEP2  0025    30
-          PT *   612 PUPPYDML     USER001  DSNTEP2  002A    35
-          PT *   545 PUPPYDML     USER001  DSNTEP2  002A    34
-          PT *   432 PUPPYDML     USER001  DSNTEP2  002A    33
-          PT *   443 PUPPYDML     USER001  DSNTEP2  002A    32
-          PT *   252 PUPPYDML     USER001  DSNTEP2  002A    31
- DISPLAY ACTIVE REPORT COMPLETE
- 17.08.45 STC00090  DSN9022I -DB1G DSNVDT '-DB1G DISPLAY THREAD' NORMAL COMPLETION
```

Figure 51. Display on Coordinator

Display on the Assistant: The PARALLELSIM COORDINATOR field tells you which DB2 is the coordinator. The ORIGINATING TOKEN field tells you lets you associate a parallel task with its originating task on the coordinator.

```
- 17.10.12           -DB2G DISPLAY THREAD(*)
- 17.10.12 STC00044  DSNV401I -DB2G DISPLAY THREAD REPORT FOLLOWS -
- 17.10.12 STC00044  DSNV402I -DB2G ACTIVE THREADS -
- NAME     ST A   REQ ID           AUTHID   PLAN     ASID TOKEN
- BATCH    PT *   641 PUPPYDML     USER001  DSNTEP2  002D     6
-  V443-PARALLELISM COORDINATOR=DB1G, ORIGINATING TOKEN=30
- BATCH    PT *    72 PUPPYDML     USER001  DSNTEP2  002D     7
-  V443-PARALLELISM COORDINATOR=DB1G, ORIGINATING TOKEN=30
- BATCH    PT *   549 PUPPYDML     USER001  DSNTEP2  002D     8
-  V443-PARALLELISM COORDINATOR=DB1G, ORIGINATING TOKEN=30
- BATCH    PT *   892 PUPPYDML     USER001  DSNTEP2  002D     9
-  V443-PARALLELISM COORDINATOR=DB1G, ORIGINATING TOKEN=30
- BATCH    PT *    47 PUPPYDML     USER001  DSNTEP2  002D     10
-  V443-PARALLELISM COORDINATOR=DB1G, ORIGINATING TOKEN=30
- DISPLAY ACTIVE REPORT COMPLETE
- 17.10.12 STC00044  DSN9022I -DB2G DSNVDT '-DISPLAY THREAD' NORMAL
-  COMPLETION
```

Figure 52. Display on Assisting DB2

Using a Performance Monitor

To see parallel tasks, you can use an online performance monitor such as RMF Monitor III, which displays information about enclaves. Parallel tasks on assistants execute within an enclave. The RMF monitor shows the classification attributes such as the plan name, package name, SQLID, and so on.

How DB2 Reports What It Did

DB2 reports information about parallelism at two times: At bind time (what it *plans* to do) and at run time (what it actually did).

Bind Time Activity: As with parallelism in general, DB2 determines a parallel degree and a parallel mode (I/O, CP, Sysplex) at bind time. DB2 assumes that the number of DB2 members available at BIND time will be available at run time, so plan your binds accordingly.

The access path that DB2 plans to use can be determined by the EXPLAIN output. The PARALLELISM_MODE column contains "X" to indicate that Sysplex query parallelism is chosen for the explainable statement.

Run Time Activity: DB2 can use a different parallel degree at run time, depending on the amount of buffer resources available.

The accounting trace record indicates whether this originating task sent work to assisting DB2s. (See **A** in Figure 53 on page 210.)

To determine the actual degree that was used, use IFCIDs 0221, IFCID 0222, and 0231 in performance trace class 8 (same as with CP parallelism). The QW0221XC field of IFCID 0221 indicates on how many members the parallel group executed, and it also indicates when a particular buffer pool is constrained on a particular member. IFCIDs 0222 and 0231 now include DB2 member names for members that participated in processing that parallel group.

When a member does not have enough buffer pool resources to participate in processing a query, the coordinator must "skip" that member when it distributes work around the group. For more information, see "Is the Buffer Pool Too Small?" on page 210.

Monitoring Processor Use

As with CP parallelism, the accounting trace record fields for processor execution time for the originating task and all of the parallel tasks must be added together to yield the total processor time used by a DB2 thread. In order to perform this same function with Sysplex query parallelism, the accounting trace records from all members involved in the query must be assembled and used. DB2 PM reports this total time in its accounting report and normalizes the time to the processor size of the originating task. See Figure 53 on page 210 for an example of an accounting report from DB2 PM.

```
TIMES/EVENTS  APPL (CLASS 1)   DB2 (CLASS 2)   CLASS 3 SUSP.   ELAPSED TIME    HIGHLIGHTS
------------  --------------  --------------   --------------  ------------    ----------------
ELAPSED TIME        1.106203        0.998704   LOCK/LATCH          0.173352    THREAD TYPE   : ALLIED
CPU TIME            0.167582        0.093132   SYNCHRON. I/O       0.022122    TERM.CONDITION: NORMAL
 TCB                0.134694        0.060444   OTHER READ I/O      0.000000    INVOKE REASON : DEALLOC
 TCB-STPROC         0.000000        0.000000   OTHER WRTE I/O      0.000000    COMMITS       :          1
 PAR.TASKS          0.032888        0.032688   SER.TASK SWTCH      0.652961    ROLLBACK      :          0
SUSPEND TIME             N/A        1.488755   ARC.LOG(QUIES)      0.000000    INCREM.BINDS  :          0
 TCB                     N/A        0.459259   ARC.LOG READ        0.000000    UPDATE/COMMIT :       0.00
 PAR.TASKS               N/A        1.029497   DRAIN LOCK          0.000000    PROGRAMS      :          0
NOT ACCOUNT.             N/A        0.479002   CLAIM RELEASE       0.000000    PARALLELISM   : SYSPLEX
DB2 ENT/EXIT             N/A             791   PAGE LATCH          0.000000
EN/EX-STPROC             N/A               0   STORED PROC.        0.000000
DCAPT.DESCR.             N/A             N/A   NOTIFY MSGS         0.640320
LOG EXTRACT.             N/A             N/A   GLOBAL CONT.        0.000000
                                               TOTAL CLASS 3       1.488755

⋮
QUERY PARALLEL.     TOTAL
---------------  --------
MAXIMUM MEMBERS A       2
MAXIMUM DEGREE         13
GROUPS EXECUTED         1
 RAN AS PLANNED         0
 RAN REDUCED    B       1
 ONE DB2 COOR=N  C      0
 ONE DB2 ISOLAT         0
 SEQ - CURSOR           0
 SEQ - NO ESA           0
 SEQ - NO BUF D         0
 SEQ - ENCL.SER         0
MEMB SKIPPED(%) E       0
PARALL.DISABLED  NO
⋮
```

Figure 53. Partial Accounting Trace That Shows Sysplex Query Parallelism

Improving Response Time

When DB2 executes a query on more than one DB2 member, the query elapsed time should improve when compared to executing the query within one member. The actual elapsed time improvement is affected by dynamic factors such as processor utilization, DB2 buffer pool availability, I/O contention, and XCF capacity.

As with CP parallelism, to determine whether performance tuning is needed to further improve elapsed time of a query, look at the elapsed time and CP execution time of each parallel task (via the IFCID 0231 class 8 performance trace record), especially the ones with the largest times.

Here are some things to look for:

- “Is the Buffer Pool Too Small?”
- “Is There I/O Contention?” on page 211
- “Is There Lock Contention?” on page 212
- “Is There XCF Signalling Contention?” on page 212
- “Is There Inter-DB2 Read/Write Interest?” on page 213

Is the Buffer Pool Too Small?: A shortage of buffer pool resources can cause specific members to be bypassed when distributing tasks around the group, leading to a possibly less-than-optimal degree of distribution. It can also cause a query to run at a degree that is less than the optimal degree determined at bind time.

To determine the percentage of unsuccessful distributions (when members are skipped) see **E** in Figure 53. Use performance class 8 and inspect the IFCID

0221 trace record to pinpoint which buffer pools cause DB2 members to be skipped.

Non-zero values in the QXREDGRP (**B** in Figure 53 on page 210) and QXDEGBUF (**D**) fields in the statistics or accounting trace record can indicate a buffer pool shortage. Consider increasing the size of the buffer pool or increasing the number of buffers available to assist with processing query work. For more information about determining buffer pool size for parallel work, see Section 5 (Volume 2) of *Administration Guide*.

Don't forget to set the parallel sequential threshold for work file buffer pools. DB2 assumes that you have all your work files of the same size (4KB or 32KB) in the same buffer pool number on all DB2s and makes run time decisions based on a single buffer pool. A lack of buffer pool resources for the work files can lead to a reduced degree of parallelism or cause the query to run sequentially.

Is There I/O Contention?: One possible cause of poor response time is contention for I/O resources. Contention can occur in any place in the I/O subsystem. Here are some ways to determine if and where I/O contention is causing the problem:

1. **Monitor**

   If you have previous accounting reports, look for changes in those reports. If the application has not changed (that is, the SQL Profile and the number of GETPAGEs per commit are relatively constant), then move on to the next step.

2. **Analyze**

   If you see increased class 3 I/O times from the accounting reports (specifically the SYNCHRON. I/O and OTHER READ I/O fields from a DB2 PM report), look to see if the number of I/Os per commit has increased. If so, consider some form of buffer pool tuning. By doing such things as increasing the size of the pool, isolating the data into a separate pool, or by tuning thresholds, you might be able to reduce the number of I/Os and speed up the remaining I/Os in the system.

   If that doesn't solve the problem, move on to pinpointing the I/O trouble spot. Use the following reports:

   - SMF type 42 records, subtypes 5 and 6 show the response times by data set for an interval.
   - The RMF Direct Access Device Activity Report shows response time by volume for an interval. It also breaks down where in the I/O subsystem the time is spent. The breakdown is of the components of the average response time to the volume. Use this information to find the bottlenecks in the setup or capacity of the I/O subsystem.

3. **Relieve the constraint**

   When you've determined where the problem is occurring, you can take steps to bring relief. This could be a matter of adding to the I/O subsystem by doing such things as adding extra channel paths to the DASD controller, adding storage directors in the DASD controller, adding extra cache/NVS in the controller, or adding extra DASD volumes. More typically, it means you have to move data sets from one volume to another.

Where to Place Data Sets: In general, you want frequently used data sets or partitions to be allocated across your available DASD so that I/O operations are distributed as evenly as possible. Make sure that device and control unit utilization is balanced. This helps balance I/O operations across the I/O configuration and takes maximum advantage of parallel I/O operations.

To determine whether the partitioning of a table or the physical placement of the partitions are reasonable, see performance trace records, IFCIDs 0221 and 0222.

The IFCID 0221 record describes how the tables within a parallel group are partitioned by specifying the key range or page range for each partition. The IFCID 0222 record shows the elapsed time statistics for each parallel operation. The elapsed time for each parallel operation within a parallel group should be comparable. An operation with a much higher elapsed time means that DB2 is doing more than desirable I/O operations on a particular logical partition, or there is significant contention on the DASD volume or channel path.

If there is an uneven distribution of work that is causing the I/O problems, consider moving data that has low activity with data that is more frequently accessed. Also be sure that your high-priority work is not sharing I/O resources with work that ties up the I/O subsystem.

If the elapsed times of the parallel tasks are comparable but are still too high, consider repartitioning the table space to more parts, assuming that processor time is not a bottleneck.

Is There Lock Contention?: To avoid lock contention run with an isolation level of UR, if that's possible. If it is not possible to run with UR, the next best thing is to take advantage of lock avoidance by running with ISOLATION (CS) and specifying CURRENTDATA (NO). This can minimize the locking effect.

Is There XCF Signalling Contention?

The resource measurement facility (RMF) of MVS provides an XCF Activity Report that contains useful measurement data for analyzing the performance of the XCF signalling service and doing capacity planning for an MVS system in a sysplex. The report shows the data collected on a system during sysplex processing. Each system collects its own data, and RMF on each system produces reports only about that system's data. It might be necessary to run the RMF reports on two or more systems to get data for corresponding outbound and inbound signalling paths in order to better understand the message traffic patterns.

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

IRLM will report the number of retries in case it cannot deliver a message for Sysplex query parallelism via XCF. This information is reported in the IRLM Exception (EXP) trace and can be used to tune the XCF performance.

For more information about monitoring XCF activity, see *MVS/ESA Setting Up a Sysplex*.

Is There Inter-DB2 Read/Write Interest?: Because DB2 can split query processing onto different DB2 subsystems, updates of the same page set cause inter-DB2 read/write interest as part of the normal data sharing integrity process. To ensure that updates on a page set are seen by the query, DB2 flushes all changed pages from the virtual pool before processing the query on other DB2 members.

Inter-DB2 read/write interest can also cause additional locking overhead. Child locks (page or row) are propagated to XES and the coupling facility based on the inter-DB2 interest on the parent (table space or partition) lock. If all the TS locks are IS, then the child S locks are not propagated. However, if there is an IX lock on the table space, all the S children must be propagated. To avoid locking overheads, use isolation level UR, or try to limit the table space locks to IS on all data sharing members to avoid child lock propagation.

For partitioned table spaces, DB2 can avoid locking all partitions in a table space, thus reducing the number of child locks that must be propagated. To take advantage of this, you must define the table space with LOCKPART YES and bind the plan or package with ACQUIRE(USE). For more information, see "Using the LOCKPART Option for Partitioned Table Spaces" on page 224.

Controlling Resources Used by Parallel Operations

There are several different ways to control how much system resource is used for processing queries using Sysplex query parallelism:

- "Dispatching Priority"
- "Amount of Buffer Space Allocated for Parallel Processing"
- "Setting Limits using the Resource Limit Facility"

Dispatching Priority: MVS workload management can be used to control the priority of parallel query work within a subsystem. You must pay special attention to the priority of query work coming into an assisting DB2. "Setting Workload Management Goals" on page 200 has more information about how to do this.

Amount of Buffer Space Allocated for Parallel Processing: You can control how much of your total buffer pool space can be used for parallel processing in general, and for Sysplex query parallelism specifically.

- The parallel threshold (VPPSEQT), determines the amount of space for all parallel processing.
- The assisting parallel sequential threshold (VPXPSEQT) determines what subset of the parallel space can be used to assist queries originating from another member of the data sharing group.

Response time can degrade if these thresholds are set too low.

Setting Limits using the Resource Limit Facility

DB2's resource limit facility can help control dynamic queries that use Sysplex query parallelism. Resource limits are local in scope, which lets DB2 ensure that a single dynamic query does not overrun any DB2 subsystem. Members of a data sharing group can share the same RLST, or each member can have its own RLST. In either case, DB2 honors the limits that are specified on that DB2, no matter how many tasks are included in the statement.

Figure 54 shows how this works.

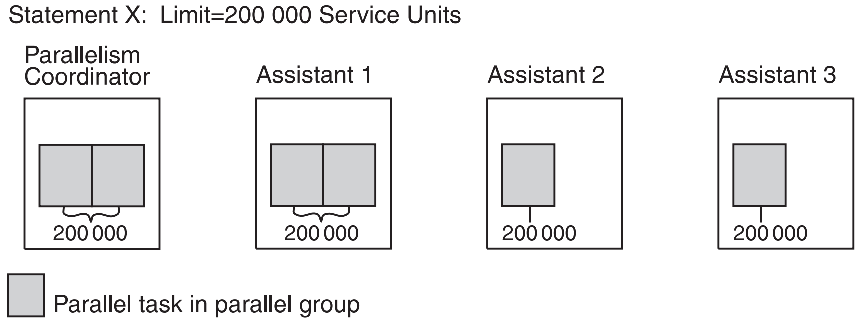

Figure 54. Governing a Dynamic Query Running on Four Members. This examples assumes that all members share the same RLST. Statement X is not allowed to consume over 200 000 service units on any DB2.

Governing Statements with More than One Parallel Group: If more than one parallel group processes the work for a query on a member, each parallel group can run up the service unit limit set for that member. On the coordinator, things work differently because *all* parallel groups, including the originating task, are governed by the limit on the coordinating member. Figure 55 shows how this works.

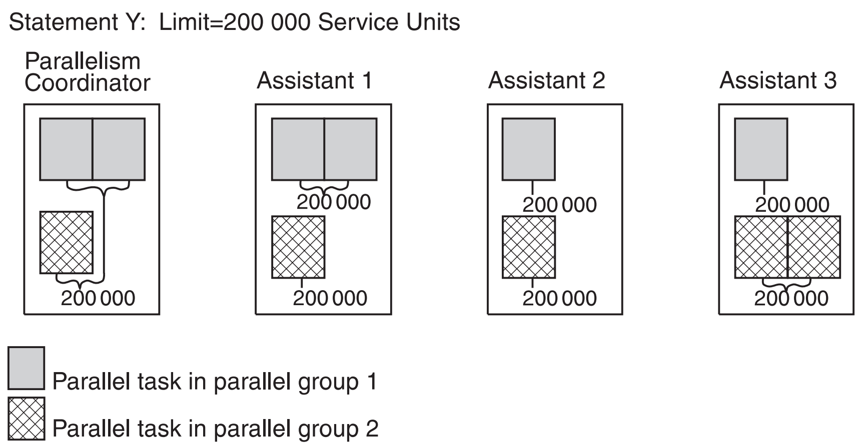

Figure 55. A Query with Two Parallel Groups Running at the Same Time. This example assumes that members are sharing the same RLST. No parallel group that is part of statement Y can run beyond the limit of 200 000 service units. On the parallelism coordinator, all tasks in all parallel groups run up to the limit specified for the statement.

Queries using INSTALL SYSADM or SYSOPR Authorities: If a query is submitted by an authorization ID with INSTALL authority, none of the parallel tasks is governed, regardless of where the parallel tasks run.

A Statement Executed More than Once: If the same statement executes more than once in an application, the limit is accumulated only on the coordinating DB2. On the assisting DB2s, the limit is reset each time the statement executes.

Disabling Sysplex Query Parallelism

In addition to existing controls that enable or disable parallelism, you can control Sysplex query parallelism specifically. You can disable Sysplex query parallelism on a system-wide basis by:

- Specifying COORDINATOR=NO and ASSISTANT=NO on installation panel DSNTIPK.
- Using buffer pool threshold controls that are described in "Buffer Pool Threshold for Parallelism Assistants" on page 205.

You can also disable Sysplex query parallelism for a single dynamic query by using a value of '5' for the resource limit facility (governor).

Table 29 on page 216 summarizes these controls.

Table 29. Controls for Sysplex Query Parallelism

Control	Checked at...	Effect
COORDINATOR subsystem parameter	Bind and run time	If COORDINATOR is NO, parallelism is restricted to a single DB2. (See QXCOORNO field of IFCID 0002 for cases in which the parameter was changed between bind and run time.) Changes from NO to YES require that plans or packages be rebound before they are considered for Sysplex query parallelism.
ASSISTANT subsystem parameter	Bind and run time	If ASSISTANT is NO, this member is not considered as a parallelism assistant. (At run time, the assistant's buffer pool must be defined to allow parallelism; otherwise, the coordinator does not send work there.) Changes from NO to YES require that plans or packages be rebound for this assistant's processing capability to affect the planned parallel degree.
Parallel buffer pool threshold (VPPSEQT) of the coordinator	Run time	If 0, no type of query processing is allowed on this DB2.
Assisting parallel sequential threshold (VPXPSEQT)	Run time	If 0, this DB2 is not considered to be an assistant for any query using that buffer pool. To disallow the entire subsystem from assisting with Sysplex query parallelism, all buffer pools must have VPXPSEQT=0 or VPPSEQT=0.
Governor (RLFFUNC='5')	Run time	Affects dynamic queries only. The query is not considered for Sysplex query parallelism.
BIND option DEGREE (1)	Bind time	Disables query parallelism for static queries.
Special register SET CURRENT DEGREE='1' or a 1 in the CURRENT DEGREE field of installation panel DSNTIPF	Run time	Affects dynamic queries only. Disables all query parallelism.

Improving Concurrency

This section describes briefly how transaction locking works in DB2 data sharing and some actions you can take to reduce locking overhead:

- “Global Transaction Locking”
- “Tuning deadlock and timeout processing” on page 226 #
- “Tuning Your Use of Locks” on page 221
- “Monitoring DB2 Locking” on page 229
- “Changing the Size of the Lock Structure” on page 235

Data sharing also introduces a new type of lock, called a *physical lock* (P-lock). But, P-locks are related to caching, not concurrency, and they use different mechanisms than the transaction locks you are familiar with in DB2. See “P-Locking” on page 242 for information about P-locks. Transaction locks are often called *logical locks* (L-locks) in data sharing.

Global Transaction Locking

With data sharing, there is concurrency control both within a specific DB2 and among all DB2s in the DB2 data sharing group. This means that locks used in data sharing are global in scope. Many global locks are processed not only by the local IRLM but also by MVS's cross-system extended services (XES) and the lock structure in the coupling facility.

Locking Optimizations

DB2 has optimizations in place to reduce the necessity to go beyond the local IRLM whenever possible:

- **Explicit hierarchical locking**, described in “Explicit Hierarchical Locking” on page 218 makes certain optimizations possible. When there is no inter-DB2 R/W interest in an object, it is possible to avoid processing certain locks beyond the local IRLM.
- If there is a single DB2 with update interest and multiple DB2s with read-only interest, DB2 propagates fewer locks than when all DB2s have update interest in the page set.
- All locks that go beyond the local IRLM are owned by the subsystem, not by an individual work unit. This allows for another optimization. Only the most restrictive lock mode for an object on a given subsystem must be propagated to XES and the coupling facility. A new lock that is equal to or less restrictive than one currently being held is not propagated. “A Locking Scenario” on page 220 gives an example of this type of optimization.
- When running on MVS 5.2 *and* the lock structure is allocated in a coupling facility of CFLEVEL=2 or higher, IRLM can release many locks with just one request to XES. This can occur after a transaction commits and has two or more locks that need to be unlocked in XES. It also can occur at DB2 shutdown or abnormal termination when the member has two or more locks that need to be unlocked.

  To see if DB2 is using this optimization, the statistics trace should show that there are significantly fewer unlock requests propagated to XES than lock requests, as shown in Figure 63 on page 232.

Explicit Hierarchical Locking

When sharing data, DB2 uses explicit hierarchical locking to determine whether it is necessary to propagate L-locks beyond the local IRLM to XES and the coupling facility. There is a performance advantage to granting lock requests locally. Explicit hierarchical locking allows IRLM to grant child locks locally when there is no inter-DB2 R/W interest on the parent. Explicit hierarchical locking is based on the lock hierarchy shown in Table 30.

Table 30. Lock Hierarchy. Type 2 indexes are not included in the hierarchy because their index pages are protected by locks on the corresponding data.

Simple Table Space	Partitioned Table Space	Segmented Table Space
Table Space Data Page Row	Data Partition Data Page Row	Table Space Table Data Page Row
Type 1 Index: Table Space Index Page	**Type 1 Partitioned Index:** Last Data Partition Index Page	**Type 1 Index:** Table Space Index Page
	Type 1 Nonpartitioned Index: Last Data Partition Nonpartitioned Index Page	

The top object in the hierarchy is a *parent*, and everything below a parent is a *child*. (A child can be the parent of another child.) While the lock is held, the first lock on the top parent is always propagated to XES and the lock structure. Thereafter, only more restrictive locks are propagated. When the lock is released, the process begins again. For partitioned table spaces, if you use LOCKPART YES, each locked partition is a parent of the child locks held for that partition. If you use LOCKPART NO (the default), the last data partition is the parent lock for all child locks. For more information about using the LOCKPART clause, see "Using the LOCKPART Option for Partitioned Table Spaces" on page 224.

Locks on the children are propagated depending on the compatibility of the maximum lock held on a parent with locks requested by other DB2s in the group for that parent.

Table 31 on page 219 shows which conditions cause the child locks to be propagated. "A Locking Scenario" on page 220 describes explicit hierarchical locking in action.

Table 31. Determining when Child Locks are Propagated to XES

Maximum lock mode of my member is ...	And the maximum lock mode of other members is...	Are X children propagated?	Are S children propagated?	Are U children propagated?
IS, S	None, IS, S	N/A	No	N/A
X	None	N/A	N/A	N/A
IS	IX, SIX	N/A	Yes	N/A
IX, SIX	IS	Yes	No	No
IX	IX	Yes	Yes	Yes
IX, SIX	None	No	No	No

Relationship between L-locks and P-locks: L-locks and P-locks are managed independently of one another. It is possible to have inter-DB2 interest on an L-lock but not on a P-lock that is held on the same object, and it is also possible to have inter-DB2 interest on the P-lock but not the L-lock. The inter-DB2 interest on the parent *L-lock* controls the propagation of the child locks. The inter-DB2 interest on the *P-lock* controls GBP-dependency.

Therefore, it is possible to propagate child L-locks for an object that is not GBP-dependent. For example, it is possible that an application bound with ACQUIRE(ALLOCATE) could have an IS L-lock on a table space without yet having opened the table space. The table space therefore has no P-lock yet on that one member. If another member opens the table space with an IX lock, its child locks have to be propagated, even though the table space is not GBP-dependent (no P-lock on the first member).

Conversely, it is also possible to have a GBP-dependent object for which child locks are not propagated. If an application is bound with RELEASE(COMMIT), for example, it might have released all its L-locks and write claims to a table space. However, until the table space switches from R/W to R/O, that table space is considered GBP-dependent. But child locks from another member can be resolved by its local IRLM.

A Locking Scenario

Figure 56 shows an example of locking activity between two members of a data sharing group.

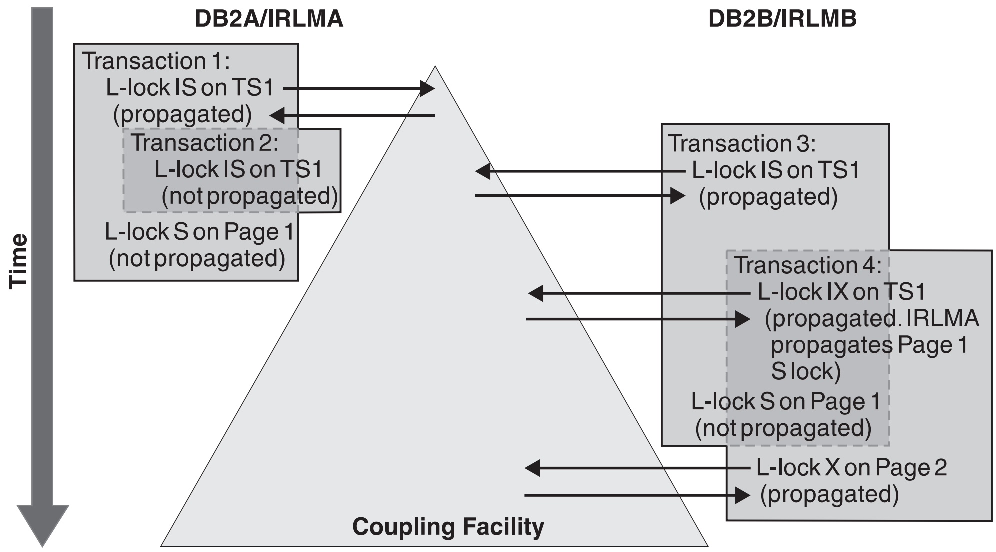

Figure 56. A Scenario Showing Lock Propagation

In the figure:

1. Transaction 2's L-lock on TS1 is not propagated because that DB2 already holds a lock on that object of an equal restriction. (Transaction 3's L-lock on TS1 is propagated, because that lock is from another DB2 subsystem.)
2. Transaction 1's child L-lock on Page 1 is not propagated at the time it was requested because its parent lock is IS on both subsystems. That is, there is no inter-DB2 R/W interest on the parent.
3. When Transaction 4 upgrades its lock to IX on TS1, its X lock on Page 2 must be propagated because there is now inter-DB2 R/W interest on the parent. Also, Transaction 1's child lock on Page 1 must be propagated.

   Transaction 1's child lock is propagated under an IRLM SRB, not the transaction's TCB. This propagation is counted in the statistics report as an asynchronous propagation, as shown in **B** in Figure 63 on page 232.

Determining Whether Locks Have Been Propagated

The statistics and accounting traces indicate the number of global locks that have been propagated to XES. The ratio of this number to the total number of global locks requested reflect the effects of explicit hierarchical locking and other locking optimizations. See “Checking for Transaction Locking Optimizations” on page 234 for more information.

Tuning Your Use of Locks

Most recommendations for reducing lock contention and locking costs in a single system hold true when sharing data, as well. This section reiterates some general recommendations and emphasizes the following:

"Use Type 2 Indexes" on page 222
"Avoid False Contention" on page 222
"Reduce the Time Needed to Resolve Contentions" on page 224
| "Using the LOCKPART Option for Partitioned Table Spaces" on page 224

There is also information about:

"Monitoring DB2 Locking" on page 229
"Changing the Size of the Lock Structure" on page 235

General Recommendations

To reduce locking contention, use the same tuning actions that are in place for single-DB2 processing today, as described in Section 5 (Volume 2) of *Administration Guide*. Here is a summary:

- Use partitioned table spaces.
- Use page locking.

  Although row locking can be used to increase concurrency in some cases, you must weigh that benefit against the increase in locking overhead that row locking might incur. (The amount of overhead depends on how well your application can avoid taking locks.)

  | One way to achieve the concurrency of row locking but to avoid the additional data sharing lock overhead is to define the table space with MAXROWS 1. See *SQL Reference* for more information.
- If your applications can tolerate reading uncommitted data, use an ISOLATION level of uncommitted read (UR).
- If you cannot use ISOLATION(UR), take advantage of lock avoidance whenever possible by binding with an isolation level of cursor stability (CS) and CURRENTDATA(NO). These are *not* the defaults.
- Reduce the scope of BIND operations by using packages. This reduces DB2 catalog and directory contention.
- Design for thread reuse, and use the bind option RELEASE(DEALLOCATE).

  For objects that don't have a lot of concurrent activity within a member, this helps to avoid the cost of releasing and reacquiring the same parent locks again and again. It can also reduce the amount of false contention, described in "Avoid False Contention" on page 222 for those transactions that use the thread.
- | Design for selective partition locking, and bind with ACQUIRE(USE). For more information about selective partition locking, see Section 2 of *Administration Guide*.

Use Type 2 Indexes

Data sharing lets you use either type 1 or type 2 indexes. However, type 1 indexes have limitations. After you enable data sharing, the index cannot be accessed with inter-DB2 R/W interest unless it is defined with SUBPAGES 1.

In any event, type 2 indexes are a better choice because they rely on data-only locking, which can reduce the total number of locks taken. Also, operations that access partitioned table spaces through a type 2 index can take advantage of selective partition locking, which can significantly reduce the number of locks propagated to the coupling facility.

Type 2 indexes are also required for many other functions, such as improved partition independence and the ability to run complex queries as parallel tasks. Again, the fewer locks that are needed, the fewer locks that need to be propagated beyond the local IRLM. For more information about type 2 indexes, see Section 5 (Volume 2) of *Administration Guide*.

Avoid False Contention

The coupling facility lock structure has two parts to it: a lock table used to determine whether there is inter-DB2 R/W interest on a particular resource, and a list of the update locks that are currently held. When considering false contention, it is the size of the lock table that you must be concerned with. It is the total size of the lock structure that determines the size of the lock table. Assuming you specify a SIZE (MVS 5.1) or INITSIZE (MVS 5.2) value on the CFRM policy that is a power of 2, the lock table is allocated one-half the total size of the lock structure. The number of members in the group determines the size of each entry in that lock table. IRLM uses the value you specify in the LOCK ENTRY SIZE field of installation panel DSNTIPJ to determine the initial size of the lock table entries. See Section 2 of *Installation Guide* for details.

IRLM assigns locked resources to an entry value in the lock table. This is how it can quickly check to see if a resource is already locked. If the lock structure is too small (thereby making the lock table portion too small), many locks can be represented by a single value. Thus, it is possible to have “false” lock contention. This is where two different locks on different resources hash to the same lock entry. The second lock requester is suspended until it is determined that there is no real lock contention on the resource.

False contention can be a problem for work loads that have heavy inter-DB2 R/W interest.

Monitoring for False Contention: You can determine the amount of false contention by using the RMF Coupling Activity reports as described in “Using the Coupling Facility Structure Activity Report of RMF” on page 230. DB2 also provides necessary information in its accounting and statistics trace classes. See “Using the DB2 Statistics Trace” on page 231 and “Using the DB2 Accounting Trace” on page 233 for more information. More detailed information can be found in the performance trace, as shown described in “Using the DB2 Performance Trace” on page 234.

How Much Contention is Acceptable: For the best performance, you want to achieve the least possible amount of global lock contention, both real and false. Aim for total global lock contention of less than 5 percent, preferably less than 1 percent. In the descriptions of the various reports, we show you how to figure out the contention percentages.

How to Reduce False Contention: Some things to do when false contention becomes a problem:

- As much as possible, reduce the amount of real lock contention in your applications.
- Specify a larger size for the lock structure and manually rebuild it as described in “Changing the Size of the Lock Structure Size by Rebuilding” on page 236.
- Make sure the value for LOCK ENTRY SIZE is not too large for the number of members in your group.

  The LOCK ENTRY SIZE parameter for the first IRLM to join the group determines the size of each lock entry in the lock table.

  A lock entry size of 2 allows twice as many lock entries as compared to a lock entry size of 4, as shown in Figure 57.

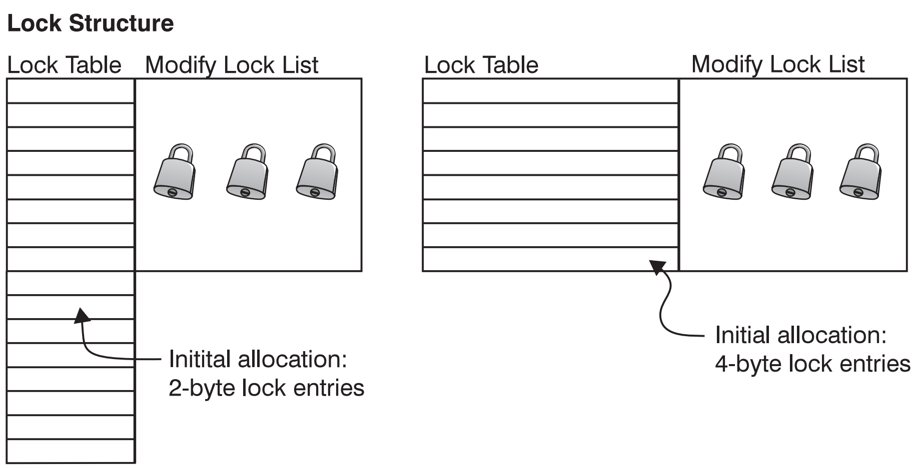

Figure 57. Initial Lock Entry Size

IRLM automatically rebuilds the structure when it needs to. For example, when the 7th member joins the group, the lock structure automatically rebuilds to create the 4-byte lock entries. This prepares the lock structure to handle an 8th member joining the group.

For this reason, even if you anticipate your group growing beyond 7 members, you can start out with a lock entry size of 2 to make the most efficient use of lock structure storage and to reduce false contention.

How to Decrease Lock Entry Size: IRLM does not automatically rebuild if the number of members goes down such that a smaller lock entry size is optimal. To decrease the lock entry size, you must:

1. Quiesce all members of the group, using the following command:

   ```
   -DB1G STOP DB2 MODE(QUIESCE)
   ```

2. If IRLM is not automatically stopped along with DB2, enter the MVS command:

   STOP *irlmproc*

3. Force off all connections from the lock structure by issuing the following MVS command:

   ```
   SETXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL
   ```

4. Force the deallocation of the lock structure by issuing the following MVS command:

   ```
   SETXCF FORCE,STRUCTURE,STRNAME=strname
   ```

5. Change the lock entry size for at least one IRLM. (We recommend changing the value for all of them.)

   If you change the IRLM startup procedure directly, the parameter you change is called MAXUSRS. See the description of MAXUSRS in the description of the command START IRLMPROC in *Command Reference*; the value of LOCK ENTRY SIZE is translated during the DB2 installation or migration process. The value you enter on the parameter directly is not the same as the value you put in the LOCK ENTRY size field of DSNTIPJ.

6. Start the DB2 and IRLM that has the updated value. (You must be sure to start the updated member first.)

7. Start all other members.

A group restart occurs when you restart the members. Because you quiesced work before changing the lock entry size, the group restart should be relatively quick. Nonetheless, this is a disruptive procedure. Consider decreasing the lock entry size only in situations when it is set too high for the number of members in the group, and you cannot alleviate the false contention within the storage boundaries you have.

Reduce the Time Needed to Resolve Contentions

When there is contention on a hash class, MVS uses XCF messages to resolve the conflict. This is how it determines which specific resources are involved in the contention, or if the contention is false. For speedy resolution of contention situations, make sure there is no queuing of messages for XCF message buffers. You can use the XCF Activity Report of RMF to detect this queuing. See *MVS/ESA Setting Up a Sysplex* for more information about tuning the XCF message buffers.

Using the LOCKPART Option for Partitioned Table Spaces

For those special cases in which you are purposefully creating affinity between data partitions and DB2 members, you can use the LOCKPART option of CREATE and ALTER TABLESPACE to indicate that you want individual partitions locked only as they are accessed.

With LOCKPART NO, the default, DB2 uses the table space lock as the lock parent in the locking hierarchy (the table space lock is represented by a single lock on the last partition of the table space). With LOCKPART YES, each locked partition is a separate lock parent; therefore, DB2 and IRLM can detect when no inter-DB2 R/W interest exists on that partition and thus does not propagate child locks unnecessarily.

With LOCKPART YES, you have the option of using the LOCK TABLE with the PART option to exclusively lock individual partitions. See Section 5 (Volume 2) of *Administration Guide* for more information about the LOCK TABLE statement.

Restrictions: If any of the following conditions are true, DB2 cannot use selective partition locking; instead, it locks *all* partitions:

- A type 1 index is used in the access path
- The plan is bound with ACQUIRE(ALLOCATE)
- The table space is defined with LOCKSIZE TABLESPACE
- When LOCK TABLE IN EXCLUSIVE MODE is used (without the PART option)

Figure 58 shows partition locks when LOCKPART YES. A partition lock is taken only when the partition is accessed.

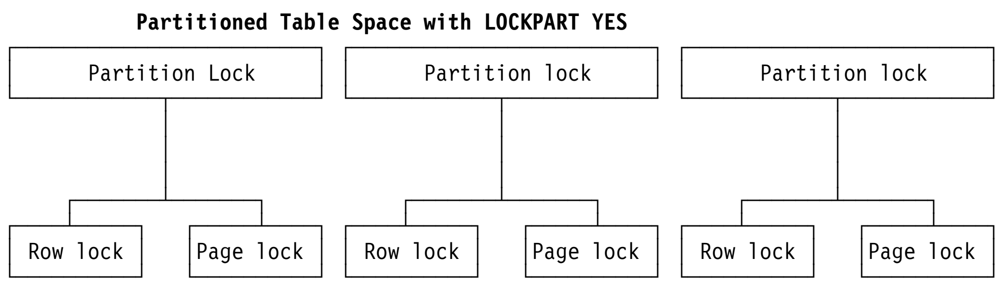

Figure 58. Partition Locks

The Duration of a Partition Lock: Partition locks follow the same rules as table space locks, and *all* partitions are held for the same duration. Thus, if one package is using RELEASE(COMMIT) and another is using RELEASE(DEALLOCATE), all partitions use RELEASE(DEALLOCATE). A partition lock can be held past commit points if using CURSOR WITH HOLD.

The Mode of a Partition Lock: Partition locks have the same possible states as table space locks (IS, IX, S, U, SIX, and X).

Lock Escalation: Lock escalation occurs when the number of locks per table space exceeds the threshold specified at installation or on the LOCKMAX clause of CREATE or ALTER TABLESPACE. Lock counts are not kept on a partition basis. When the maximum number of locks for a table space is reached, the locks on all partitions are escalated to S or X. Partitions that have not yet been locked are not affected by lock escalation; they remain unlocked. Any partitions already holding S, U, or X locks remain unchanged.

After lock escalation occurs, any unlocked partitions that are subsequently accessed use a gross lock.

Monitoring Selective Partition Locking: The performance trace and DISPLAY DATABASE give information about selective partition locking.

Use performance trace class 6 (IFCID 0020) to see whether you have taken advantage of selective partition locking for your partitioned table spaces. If you are using DB2 PM, you can see this in the locking summary portion of the SQL Activity Report.

Tuning deadlock and timeout processing

This section describes how deadlock detection and resource timeouts work in a data sharing environment. This information can help you choose the appropriate times for deadlock detection intervals and for determining a resource timeout value.

In this section:

Global deadlock processing

In a data sharing environment, deadlocks can occur between transactions on different DB2 members. The term *global deadlock* refers to the situation where two or more DB2 members are involved in the deadlock. *Local deadlock* refers to the situation where all of the deadlocked transactions reside on a single DB2 member.

Controlling deadlock detection: Use the DEADLOK parameter in the IRLM startup procedure to control how often IRLM does its deadlock detection processing. Specify the parameter as follows:

```
DEADLOK='x,y'
```

where

x
: The number of seconds between two successive scans for a local deadlock (DEADLOCK TIME value on installation panel DSNTIPJ). The default is 5 and can be no larger than 5.

y
: The number of local scans that occur before a scan for global deadlock starts (DEADLOCK CYCLE value on install panel DSNTIPJ). The value that is always used by IRLM is 1.

Global deadlock detection requires the participation of all IRLM members in the data sharing group. Each IRLM member has detailed knowledge of the locks that are held and being waited for by the transactions on its associated DB2 member. However, to provide global detection and timeout services, each IRLM is told about all requests that are waiting globally so that the IRLM can provide information about its own blockers. That IRLM also provides information on its own waiters. The IRLM members use XCF messages to exchange information so that each member has this global information.

The global deadlock manager: To coordinate the exchange of information, one IRLM member assumes the role of the global deadlock manager. The global deadlock manager is the IRLM member with the lowest member ID value (as specified in its IRLM procedure). As IRLM members join or leave the group, the global deadlock manager might change.

The local deadlock detector: Each IRLM member in the group must participate in the global deadlock detection process. Each IRLM member (including the one designated as the global deadlock manager) has the role of local deadlock detector.

Relationship between local and global deadlock detection: There are four XCF messages required to gather and communicate the latest information from the local deadlock detectors:

1. The local deadlock detector sends its information about lock waiters to the global deadlock manager.
2. The global deadlock manager gathers that information from all local deadlock detectors and then sends messages to each of the IRLMs in the group. (Because the global deadlock manager is also a local deadlock detector, it receives the same information, although somewhat quicker than the rest of the IRLMs do.)
3. Each local deadlock detector looks at the global view of resources and determines if it has blockers for other waiters. It passes that information along to the global deadlock manager with its list of waiters.
4. The global deadlock manager, from the information it receives from the local deadlock detectors, determines if a global deadlock or timeout situation exists. If a global deadlock situation exists, it chooses a victim for the deadlock. The global deadlock manager also determines if any timeout candidate is blocked by an incompatible waiter or holder and, if so, presents that candidate to the owning IRLM, along with any deadlock victims belonging to that IRLM. When DB2 receives this information, it makes the decision whether to request IRLM to reject any given timeout candidate waiter.

These four messages represent one global detection cycle, and it usually takes 2-4 *x*-second intervals to complete (where *x* is local cycles). Figure 59 illustrates an example where the deadlock time value is set to 5 seconds.

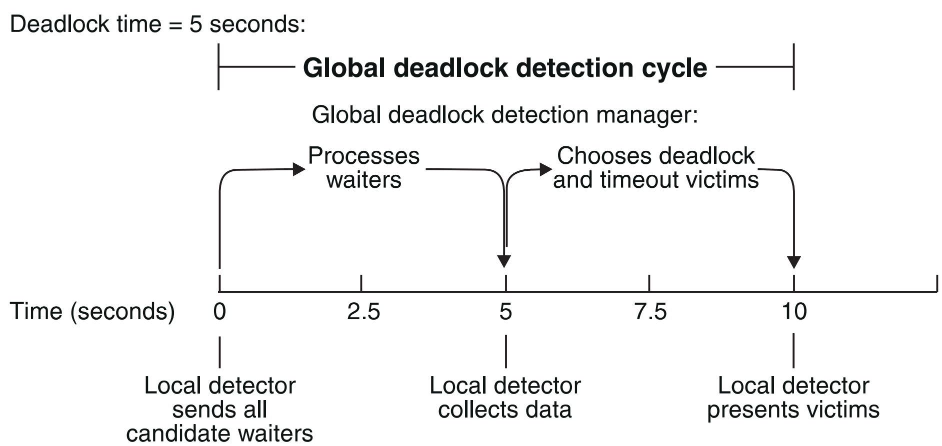

Figure 59. Global deadlock detection cycle

Deadlock detection might be delayed if any of the IRLMs in the group encounter any of the following conditions:

- XCF signalling delays
- IRLM latch contention (could be encountered in systems with extremely high IRLM locking activity)
- A large number of global waiters

Global timeout processing

Just as in a non-data-sharing environment, DB2 calculates the timeout period based on the RESOURCE TIMEOUT and DEADLOCK TIME installation parameter values. DB2 calculates the timeout period as follows:

1. Divide RESOURCE TIMEOUT by DEADLOCK TIME
2. Round up to the next largest integer
3. Multiply that integer by DEADLOCK TIME

As described in Section 5 (Volume 2) of *Administration Guide*, in non-data-sharing systems, the actual time that a transaction waits on a lock before timing out varies between the timeout period and the timeout period plus one DEADLOCK TIME interval.

For example, if the timeout period for a given transaction is 60 seconds and the DEADLOCK TIME value is 5 seconds, then the transaction waits between 60 and 65 seconds before timing out, with the average wait time being about 62.5 seconds. This is because timeout is driven by the deadlock detection process, which is activated on a timer interval basis.

Elapsed time until timeout, non-data-sharing: The actual time a process waits until timing out usually falls within the following range:

```
MIN LOCAL TIMEOUT = timeout period
MAX LOCAL TIMEOUT = timeout period + DEADLOCK TIME value
AVERAGE LOCAL TIMEOUT = timeout period + DEADLOCK TIME value/2
```

However, the maximum or average values can be larger, depending on the number of waiters in the system or if there is a heavy IRLM workload.

Elapsed time until timeout, data sharing: In a data sharing environment, because the deadlock detection process sends inter-system XCF messages, a given transaction typically waits somewhat longer before timing out when compared to what you might normally experience in a non-data-sharing environment. How much longer a transaction waits depends on where in the global deadlock detection cycle that the timeout period actually expired. However, the length of time a process waits until timing out generally falls within the following range:

```
MIN GLOBAL TIMEOUT = timeout period + DEADLOCK TIME value
MAX GLOBAL TIMEOUT = timeout period + 4*DEADLOCK TIME value
AVERAGE GLOBAL TIMEOUT = timeout period + 2*DEADLOCK TIME value
```

Again, the maximum or average values might be larger.

Recommendations

Quick detection of deadlocks and timeouts is necessary in a data sharing environment to prevent a large number of waiters on each system. These large numbers of waiters can cause much longer wait times for timeouts and deadlocks. Based on this assumption, here are two recommendations:

- If your DB2 non-data-sharing subsystem has a problem with deadlocks, consider reducing the deadlock time to prevent these long lists of waiters from developing. (If you don't have a problem with deadlocks, it is likely that you won't have to change any parameters for data sharing.)

- If you have stringent timeout limits that must be honored by DB2, consider decreasing the deadlock time before moving to data sharing, as illustrated in this example:

  Assume that you have set your timeout period for your non-data-sharing DB2 to 55 seconds because you want the wait time for timeout to be at or before 60 seconds. (This assumes that your deadlock time value is 5.) In a data sharing environment, reduce the value of DEADLOCK TIME so that the timeout period is 40 seconds. This makes it more likely that your actual wait time for timeouts is at or before 60 seconds.

Monitoring DB2 Locking

With data sharing, it is essential to control the volume of global lock requests that are propagated to the coupling facility and to control the amount of lock contention, both real and false. You must monitor both the amount and type of locking your applications are doing, but you must also make sure that any locking problems are not caused by data sharing resources, such as an undersized lock structure, or the over-utilization of the coupling facility or coupling facility channels (links).

This section describes the following ways to watch locking activity and lock structure activity:

"Using the Command DISPLAY DATABASE"
"Using the Coupling Facility Structure Activity Report of RMF" on page 230
"Using the DB2 Statistics Trace" on page 231
"Using the DB2 Accounting Trace" on page 233
"Using the DB2 Performance Trace" on page 234

There's also the MVS command D XCF,STRNAME, described in "Displaying Information about Specific Structures" on page 152.

General-use Programming Interface

Using the Command DISPLAY DATABASE

Use the LOCKS option on DISPLAY DATABASE to display information about page set, partition, or table locks held on resources. The "lock" column of the display describes the type and duration of locks used by corresponding agents.

If the table space is defined with LOCKPART NO, the display looks like Figure 60 on page 230. The LOCKINFO field shows a value of S, indicating that this is a table space lock.

```
NAME     TYPE PART STATUS            CONNID   CORRID       LOCKINFO
-------- ---- ---- ----------------- -------- ------------ ---------
TSPART   TS                          LSS001   DSN2SQL      H-IS,S,C
    -                 MEMBER NAME DB1G
TSPART   TS                          LSS002   DSN2SQL      H-IS,S,C
    -                 MEMBER NAME DB2G
TSPART   TS    01  RO                                      H-S,PP,I
    -                 MEMBER NAME DB1G
TSPART   TS    02  RW                                      H-S,PP,I
    -                 MEMBER NAME DB2G
TSPART   TS    03  RW                                      H-S,PP,I
    -                 MEMBER NAME DB2G
TSPART   TS    04  RW
```

Figure 60. Example DISPLAY DATABASE LOCKS for a Table Space Defined with LOCKPART NO

Figure 61 is an example of output of DISPLAY DATABASE when the table space is defined with LOCKPART YES. The application identified as LSS001 on member DB1G has locked partitions 1 and 2. LSS002 on member DB2G has locked partitions 1 and 3. Partition 4 has no locks held on it.

```
NAME     TYPE PART STATUS            CONNID   CORRID       LOCKINFO
-------- ---- ---- ----------------- -------- ------------ ---------
TSPART   TS    01  RO                LSS001   DSN2SQL      H-IS,P,C
    -                 MEMBER NAME DB1G
TSPART   TS    01  RO                                      H-S,PP,I
    -                 MEMBER NAME DB1G
TSPART   TS    01  RO                LSS002   DSN2SQL      H-IS,P,C
    -                 MEMBER NAME DB2G
TSPART   TS    01  RO                                      H-S,PP,I
    -                 MEMBER NAME DB2G
TSPART   TS    02  RW                LSS001   DSN2SQL      H-IS,P,C
    -                 MEMBER NAME DB1G
TSPART   TS    02  RW                                      H-S,PP,I
    -                 MEMBER NAME DB1G
TSPART   TS    03  RW                LSS002   DSN2SQL      H-IS,P,C
    -                 MEMBER NAME DB2G
TSPART   TS    03  RW                                      H-S,PP,I
    -                 MEMBER NAME DB2G
TSPART   TS    04  RW
```

Figure 61. Example DISPLAY DATABASE LOCKS for a Table Space Defined with LOCKPART YES

End of General-use Programming Interface

Using the Coupling Facility Structure Activity Report of RMF

The Coupling Facility Activity Report of RMF describes activity to all structures in the coupling facility for a given time period. Figure 62 on page 231 shows a partial report, giving information about:

- Total number of lock-related requests (**A**)
- Number of requests that were delayed because of contention (**B**).
- The number of delayed requests that were caused by false contention (**C**).

```
------------------------------------------------------------------------------------------------------------------------
                                          COUPLING  FACILITY  STRUCTURE  ACTIVITY
------------------------------------------------------------------------------------------------------------------------

 STRUCTURE NAME = DSNDB0G_LOCK1     TYPE = LOCK
           # REQ    -------------- REQUESTS -------------   -------------- QUEUED REQUESTS --------------
 SYSTEM    TOTAL             #     % OF  -SERV TIME(MIC)-          (ARRIVAL  RATE) --- QUEUE TIME(MIC)--       REQUEST
 NAME      AVG/SEC          REQ    ALL     AVG   STD_DEV           MIN   AVG   MAX          AVG  STD_DEV     CONTENTIONS
 STLABC1    2207K   SYNC   2207K   100%   206.0     91.0    HPRIO     0   0.0     0  NO SCH   0.0     0.0 # REQ  A    2194K
                                                                                                              B
            1719    ASYNC     0    0.0%     0.0      0.0    LPRIO     0   0.0     0                       # REQ DELAYED  14K
                    CHNGD     0    0.0%  INCLUDED IN ASYNC                                                   -CONT       14K
                                                                                                              C
                                                                                                             -FALSE CONT  9318
------------------------------------------------------------------------------------------------------------------------
     TOTAL      2207K   SYNC   2207K   100%   206.0     91.0                         NO SCH   0.0     0.0 # REQ         2194K
                1719    ASYNC     0    0.0%     0.0      0.0                                               # REQ DELAYED   14K
                        CHNGD     0    0.0%                                                                   -CONT         14K
                                                                                                              -FALSE CONT 9318
```

Figure 62. Partial RMF Coupling Facility Activity Report for Lock Structure

Determining the Contention Percentages: Use the following calculations:

- Total contention is the number of delayed requests (**B**) divided by the total number of requests (**A**), multiplied by 100. So, for our example:

  `(14000  / 2194000) × 100 = .638%`

  This indicates that the global contention rate is approximately .64 percent (a good figure).

- False contention is the number of false contentions (**C**) divided by the total number of requests (**A**) multiplies by 100. For our example, then:

  `(9318 / 2194000) × 100 = 0.4%`

  Thus, the rate of false contention is 0.4 percent, a very good figure.

Using the DB2 Statistics Trace

The DB2 statistics trace provides counters that track the amount of global locking activity and contention that each DB2 in the data sharing group is encountering. This trace runs with low overhead. It is therefore a good idea to keep this trace turned on to allow continuous monitoring of each subsystem.

See Figure 63 on page 232 to see what kind of information you can get from a statistics trace.

```
DATA SHARING LOCKING         QUANTITY  /MINUTE  /THREAD  /COMMIT
---------------------------  --------  -------  -------  -------
GLOBAL CONTENTION RATE (%)       0.41 F
FALSE CONTENTION RATE (%)        0.17  G

LOCK REQUESTS (P-LOCKS)      10515.00  1050.53  2103.00     0.72
UNLOCK REQUESTS (P-LOCKS)     9183.00   917.45  1836.60     0.63
CHANGE REQUESTS (P-LOCKS)      101.00    10.09    20.20     0.01
A
SYNCH.XES - LOCK REQUESTS      484.5K    48.4K    96.9K    33.40
SYNCH.XES - CHANGE REQUESTS  50557.00  5051.02    10.1K     3.49
SYNCH.XES - UNLOCK REQUESTS    242.5K    24.2K    48.5K    16.70
ASYNCH.XES - RESOURCES     B     2.00     0.20     0.40     0.00

SUSPENDS - IRLM GLOBAL CONT C 1829.00   182.73   365.80     0.13
SUSPENDS - XES GLOBAL CONT. D    4.00     0.40     0.80     0.00
SUSPENDS - FALSE CONTENTION E 1342.00   134.08   268.40     0.09
INCOMPATIBLE RETAINED LOCK       0.00     0.00     0.00     0.00

NOTIFY MESSAGES SENT            30.00     3.00     6.00     0.00
NOTIFY MESSAGES RECEIVED       867.00    86.62   173.40     0.06
P-LOCK/NOTIFY EXITS ENGINES     10.00      N/A      N/A      N/A
P-LCK/NFY EX.ENGINE UNAVAIL      0.00     0.00     0.00     0.00

PSET/PART P-LCK NEGOTIATION      2.00     0.20     0.40     0.00
PAGE P-LOCK NEGOTIATION        553.00    55.25   110.60     0.04
OTHER P-LOCK NEGOTIATION       104.00    10.39    20.80     0.01
P-LOCK CHANGE DURING NEG.      655.00    65.44   131.00     0.05
```

Figure 63. Data Sharing Locking Block of DB2 PM Statistics Report

Explanation of Fields

A These counters indicate the total numbers of lock-related requests (both L-locks and P-locks) that are propagated to XES synchronously.

B This number indicates the number of lock-related requests that were propagated to XES asynchronously. DB2 uses the term *asynchronous* to mean that the request was done under a system execution unit, asynchronous to the allied work unit.

This particular counter can be incremented when, for example, one DB2 has an IS lock on a particular table space and another DB2 requests an IX lock. The S child locks held by the first DB2 must be propagated under a system execution unit to XES and the coupling facility. See Figure 56 on page 220 for an example of this.

It is possible for these asynchronous "child lock" propagations to encounter false contention. If so, the false contention is counted in RMF statistics, but not in DB2.

C The number of real contentions, as detected by IRLM.

D The number of real contentions, as detected by XES, that were not IRLM-level contentions. IRLM has knowledge of more lock types than XES. Thus, IRLM often resolves contention that XES cannot. The most common example of XES-level contention is usually the intent locks (IS and IX) on the parent L-locks. IS and IX are compatible to IRLM but not to XES. Another common example is the U and S page L-locks; U and S are compatible to IRLM, but not to XES.

E The number of false contentions.

Calculating Global Contention Percentages: To figure the global contention percentages, using the statistic report shown in Figure 63 on page 232. Our calculations only account for synchronous lock requests.

- Total contention is:

  The total number of suspends because of contention (**C** + **D** + **E**)
  divided by
  The total number of requests that went to XES (excluding asynchronous requests): ((three fields under **A**) + **C** + **D** + **E**)
  Multiplied by 100.

  So, for our example:

  ```
  (3175 / 780732) × 100 = .41%
  ```

  This indicates that the contention rate is approximately .41 percent (**F**). Because this is such a low rate of contention, it is probably not necessary to determine the amount of false contention. However, here is the calculation you would use:

- False contention is the number of false contentions (**E**) divided by the total number of requests that went to XES (excluding asynchronous requests) ((three fields under **A**) + (**C** + **D** + **E**)) multiplied by 100. For our example, then:

  ```
  1342 / 780732 × 100 = .17%  ( G )
  ```

  Thus, the approximate rate of false contention is less than .2 percent, a very low figure.

Using the DB2 Accounting Trace

Use the accounting trace to determine which users or plans are experiencing global lock contention. The DB2 accounting trace provides a summary of thread resource usage within DB2. DB2 threads experiencing global lock contention are shown in accounting trace class 1, as shown in Figure 64. The accumulated elapsed time of the suspensions are shown in accounting trace class 3.

LOCKING		AVERAGE	TOTAL	DATA SHARING		AVERAGE	TOTAL
TIMEOUTS		0.00	0	GLOBAL CONT RATE(%)		0.43	N/A
DEADLOCKS		0.00	0	FALSE CONT RATE(%)	M	0.20	N/A
ESCAL.(SHARED)		0.00	0	LOCK REQ - PLOCKS	C	2.11	2434
ESCAL.(EXCLUS)		0.00	0	UNLOCK REQ - PLOCKS	K	1.53	1769
MAX LOCKS HELD		20.43	28	CHANGE REQ - PLOCKS	D	0.03	36
LOCK REQUEST	A	42.12	48570	LOCK REQ - XES	E	41.82	48220
UNLOCK REQUEST	J	24.59	28348	UNLOCK REQ - XES	L	24.70	28481
QUERY REQUEST		0.00	0	CHANGE REQ - XES	F	11.10	12796
CHANGE REQUEST	B	11.10	12803	SUSPENDS - IRLM	G	0.18	205
OTHER REQUEST		0.00	0	SUSPENDS - XES	H	0.00	0
LOCK SUSPENS.		0.02	19	SUSPENDS - FALSE	I	0.16	179
LATCH SUSPENS.		1.17	1344	INCOMPATIBLE LOCKS		0.00	0
OTHER SUSPENS.		0.00	0	NOTIFY MSGS SENT		0.00	3
TOTAL SUSPENS.		1.18	1363				

Figure 64. Portion of DB2 PM Accounting Report Showing Locking Activity

Checking for False Contention: The false contention rate (**M**) is a percentage of the total number of requests that went to XES (**E** + **L** + **F**).

In our example:

```
(179 / 89497) × 100 = .2%    M
```

False contention is approximately .2 percent of the total number of requests that went to XES.

Checking for Transaction Locking Optimizations: To see how well global transaction locking optimizations are working in your application, you must:

1. Determine the total number of L-lock LOCK requests propagated to XES (call that X). This requires subtracting P-lock LOCK requests:

   ```
   E - C = X
   ```

   In our example:

   ```
   48220 - 2434 = 45786
   ```

2. Divide the total number of L-lock LOCK requests that were propagated to XES (X) by the total number of LOCK requests (**A**).

   In our example:

   ```
   45786 / 48570 = 94%
   ```

In this particular work load of 100 percent R/W sharing, 6 percent of LOCK requests were not propagated to XES and the coupling facility because of locking optimizations.

Using the DB2 Performance Trace

The DB2 performance trace gives more detail about which shared resources are experiencing contention. Performance traces are generally activated on an “as needed” basis because of their added overhead. Performance trace class 6 (specifically, IFCID 0045) indicates whether the suspension is because of contention.

This trace causes DB2 to write a trace record every time a lock is suspended and every time it is resumed. Part of the data that is recorded is the resource name that is experiencing the contention. By determining which shared resources are experiencing the lock contention, you might be able to make some design changes or other tuning actions to increase concurrency.

Figure 65 on page 235 shows an example of a performance trace formatted by DB2 PM. (Consider using the DB2 PM Lock Suspension Trace report as a more usable alternative.)

```
                    16:58:23.16100650   3655   7  44 LOCK SUSPEND   NETWORKID:  'BLANK'   LUNAME:  'BLANK'   LUWSEQ:     1
                    N/P
|-------------------------------------------------------------------------------------------------------------------------
|LOCK RES TYPE: INDEX PAGE LOCK                      DBID: 259                 OBID: 46          RESOURCE ID: X'0000A300'
|IRLM FUNC CODE: LOCK (NAME)        STATE: SHARED  A                        DURATION: MANUAL   C REASON SUSP: IS
|REQ TOKEN: X'00000000'   LOCK ATTRIBUTES: L-LOCK GLOBAL NOMODIFY NOFORCE   PROP TO XES: YES      ASYN TO XES: YES
|PARENT TOKEN: X'0FCD5A48'     LOCK HASH VALUE: X'0103A22D'                 B
|QW0044CL  X'00'   QW0044FL  X'B0'
|-------------------------------------------------------------------------------------------------------------------------
0PRIMAUTH CONNECT
 ORIGAUTH CORRNAME  CONNTYPE      RECORD TIME     DEST  ACE IFC  DESCRIPTION                            DATA
 PLANNAME CORRNMBR  INSTANCE      TCB CPU TIME   SEQ NO      ID
 -------- -------- ----------- ----------------- ------ --- --- ------------- ----------------------------------------------------
 T3270A   IMSA               16:58:23.17671462   3658   7  45 LOCK       <-- NETWORKID:  'BLANK'   LUNAME:  'BLANK'   LUWSEQ:   1
 T3270A   D2PWA    IMS-MPP   N/P                              RESUME         REASON FOR RESUME             :  NORMAL RESUME
 D2PWA    0044     AAACCE41F07D                                              REASON FOR SUSPEND            :  X'08'
                                                                               IRLM LATCH CONTENTION       :   NO
                                                                               IRLM QUEUED REQUEST         :   NO
                                                                               LOCAL RESOURCE CONTENTION   :   NO
                                                                           D   GLOBAL RESOURCE CONTENTION   :   YES
                                                                               RETAIN LOCK CONTENTION      :   NO
                                                                               INTER-SYSTEM MESSAGE SENDING  :   NO
                                                                             GLOBAL CONTENTION EXTENT      :  X'C0'
                                                                             XES GLOBAL CONTENTION         :   NO
                                                                               IRLM GLOBAL CONTENTION      :   NO
                                                                            E FALSE CONTENTION             :   YES
                                                                             QW0045W3   NO   QW0045W5  YES   QW0045W7   NO
                                                                             QW0045X1  YES   QW0045X2  YES   QW0045X5   NO
                                                                             QW0045X6   NO   QW0045X7   NO   QW0045X8   NO
```

| *Figure 65. Performance Trace Records*

Explanation of Fields: IFCIDs 0044 and 0045 show an index page lock. Other relevant fields of the report are:

A These fields indicate that the lock is an L-lock and that it is global.

B This lock was propagated to XES.

C The “IS” means that inter-system communication was required to resolve the lock request.

D This indicates that there was global contention.

| **E** This gives more detail about the extent of the contention. In this case, the contention was false (YES for FALSE CONTENTION). The contention could also have occurred at the XES level (XES GLOBAL CONTENTION) or at the IRLM level (IRLM GLOBAL CONTENTION). Because IRLM has the real knowledge of true contention, that is the level where you know there is true contention over resources.

Changing the Size of the Lock Structure

This section describes two possible ways of changing the size of the lock structure. One method is dynamic and changes the storage of only the modify lock list portion of the lock structure. The other method requires a CFRM policy change and a rebuild, but it can change the storage of both the lock table and lock table in the lock structure.

Changing the Lock Structure Size Dynamically

If *all* of the following conditions are true:

- All members of the group run with MVS Version 5 Release 2 or subsequent releases.
- The lock structure is allocated in a coupling facility with CFLEVEL greater than zero.

- The currently allocated size of the structure is less than the maximum size that is defined in the SIZE parameter of the CFRM policy.
- You do not want to change the size of the lock table portion of the lock structure.

Then you can enter the following command (this example assumes the group name is DSNDB0G):

```
SETXCF START,ALTER,STRNAME=DSNDB0G_LOCK1,SIZE=newsize
```

This example assumes that *newsize* is less than or equal to the maximum size defined the CFRM policy for the lock structure.

If the maximum size (SIZE in the CFRM policy) is still not big enough, you must increase the lock storage in the CFRM policy and rebuild the lock structure.

Changing the Size of the Lock Structure Size by Rebuilding

If *any* of the following conditions are true:

- Any member of the group does not run with MVS Version 5 Release 2 or subsequent releases.
- The lock structure is allocated in a coupling facility at CFLEVEL=0.
- The allocated size of the structure is already at the maximum size defined by the SIZE parameter of the CFRM policy.
- You want to change the size of the lock table portion of the lock structure.

Then you must do the following procedure:

1. Increase the storage for the lock structure in the CFRM policy to the next power of 2 to allow the lock table portion of the structure to be increased. For example, if the lock structure was 16MB, increase the size to 32MB.
2. Use the MVS command SETXCF START,POLICY to start the updated policy.
3. Issue the following MVS command to rebuild the structure (this example assumes the group name is DSNDB0G):

   ```
   SETXCF START,REBUILD,STRNAME=DSNDB0G_LOCK1
   ```

You can use a performance class 20 trace (IFCIDs 0267 and 0268) to monitor how long the rebuild of the lock structure takes.

Tuning Group Buffer Pools

This chapter describes the following about how DB2:

- Ensures that DB2 does not read down-level data that is cached in its member buffer pools (*cache coherency*).
- As much as possible, enables a quick refresh of a down-level page without having to go to DASD.

With data sharing, group buffer pools are a key component of cache coherency as are the subsystem locking mechanisms, the *P-locks*, used in that process. Your understanding of these processes is helpful when tuning the data sharing group for best performance.

With data sharing, a database page can reside:

In a virtual buffer pool
In a hiperpool
In a group buffer pool
On DASD

Database pages continue to be cached in each sharing member's buffer pool before they can be referenced or updated. Each sharing member can control its own buffer pool configurations (that is, the size and number of buffer pools). However, if there is inter-DB2 R/W interest in the data, then the group buffer pool is also used for caching data.

It is also possible to cache clean pages in the group buffer pool as a mutually exclusive alternative to hiperpools.

The group buffer pool contains information necessary for maintaining cache coherency. Pages of GBP-dependent page sets are *registered* in the group buffer pool. When a changed page is written to the group buffer pool, all DB2 subsystems that have this page cached in their buffer pools are notified that the page has been invalidated (this notification does not cause a processing interruption on those systems). This is called *cross-invalidation*. When a member needs a page of data and finds it in its buffer pool, it tests to see if the buffer contents are still valid. If not, then the page must be refreshed, either from the group buffer pool or DASD.

This section describes the following topics:

- “Assigning Page Sets to Group Buffer Pools” on page 238
- “Inter-DB2 Interest and GBP-Dependency” on page 239
- “P-Locking” on page 242
- “Read Operations” on page 245
- “Write Operations” on page 247
- “Group Buffer Pool Thresholds” on page 252
- “Monitoring Group Buffer Pools” on page 254
- “Making Sure Group Buffer Pools are the Right Size and Ratio” on page 258
- “Changing Group Buffer Pools” on page 265

Assigning Page Sets to Group Buffer Pools

Any data sharing group can have up to 50 4KB-page size group buffer pools (named GBP0-GBP49) and 10 32KB-page size group buffer pools (named GBP32K-GBP32K9). Different group buffer pools can be on different coupling facilities. Because of a strict naming convention you must use, DB2 maps the group buffer pools to the buffer pools. For example, buffer pool BP0 maps to group buffer pool GBP0. Thus, it is your choice of buffer pool that determines which group buffer pool is used. GBP0 is the default.

General-use Programming Interface

For example, to assign table space DSN8S51D to GBP2, you must:

1. Stop all access to the table space by issuing the following command:

   ```
   -DB1G STOP DATABASE(DSN8D51A) SPACENAM(DSN8S51D)
   ```

2. Change the buffer pool assignment by executing the following SQL statement:

   ```
   ALTER TABLESPACE DSN8D51A.DSN8S51D
    BUFFERPOOL BP2;
   ```

3. Allow access to the table space by issuing the following command:

   ```
   -DB1G START DATABASE(DSN8D51A) SPACENAM(DSN8S51D)
   ```

End of General-use Programming Interface

Recommendation for Performance

For best performance, it is best to keep GBP-dependent page sets in separate buffer pools from non-GBP-dependent page sets. For example, it is a good idea to keep work file table spaces, which are always non-GBP-dependent, in different buffer pools than that used by GBP-dependent page sets. This means you must assign work file table spaces to a buffer pool other than BP0.

This separation helps DB2 more efficiently handle the registration and unregistration of pages to the group buffer pool.

How to Keep Data from Being Shared

It is possible, although not necessarily recommended, to restrict access to data to a single member. If you choose to do this, there are operational issues to consider:

- You cannot do workload balancing for that data, because the other DB2s in the group are not aware of that data. Thus, it is possible for the member that has access to the data to become overloaded if access to that data increases over time.
- Availability is compromised, because if the member that owns the data goes down, no other member can access that data.
- You might have to set up special affinities to allow the application access to that data. Work cannot be automatically routed around the group to find the data.

Defining Private Data: If you want access to table space named NOSHARE limited only to DB2C, you could assign NOSHARE to a previously unused buffer pool, such as BP25, using the ALTER TABLESPACE statement. Do not define a group buffer pool that corresponds to BP25, and assign BP25 a size of zero on any other DB2 in the group. This prevents the other members of the group from attempting to use this buffer pool and therefore from accessing table space NOSHARE.

Inter-DB2 Interest and GBP-Dependency

The concepts of *inter-DB2 R/W interest* and *group buffer pool dependency* (GBP-dependency) are closely related. Whenever there is inter-DB2 R/W interest on a page set or partition, that object is GBP-dependent. Conversely, if there is no inter-DB2 R/W interest, the object is *usually* not GBP-dependent. Sometimes an object still has pages cached in the group buffer pool, and it can remain GBP-dependent even after the inter-DB2 R/W interest has gone away.

Table 32. Determining Group Buffer Pool Dependency

Your DB2's Interest	Other DB2's Interest	Is Page Set GBP-Dependent?
R/O	None, R/O	No
R/O	R/W	Yes
R/W	None	No
Exception: The page set remains GBP-dependent for some time before DB2 removes the dependency. DB2 might not be able to remove the GBP-dependency if applications update the page set without issuing periodic commits.		
R/W	R/O	Yes
R/W	R/W	Yes

How DB2 Tracks Interest

The mechanism DB2 uses to express interest in an object is a global lock called *physical locks* (abbreviated to *P-locks*). These locks are “physical” to contrast them with transaction locks, sometimes called “logical locks” (or L-locks, for short). Although you do not have as much control over physical locks as over transaction locks, they play an important part in how DB2 tracks inter-DB2 interest. P-locks are described in more detail in “P-Locking” on page 242.

P-lock operations occur on each member, and reflect that member's level of interest in a page set. Even if only one data sharing member is active, P-lock operations still occur. Table 33 shows when P-lock operations occur on each member.

Table 33. When P-locks Operations Occur on Each Member

Event	P-Lock Operation
Page set or partition data sets are physically opened.	P-lock is obtained in a read-only state.
Page set or partition is first updated.	P-lock is changed to a read-write state.
There was no update within an installation-specified time period or number of checkpoints (R/O switching).	P-lock is changed to a read-only state.
Page set or partition data sets are closed.	P-lock is released.

In addition to the events mentioned in Table 33, there is also a special case that occurs in the following conditions:

- There is a single DB2 with R/W interest and any number of DB2s with R/O interest.
- All members have R/O interest and the page set or partition has been GBP-dependent since the it was physically opened.

In those conditions, if the R/O DB2s do not reference the page set again in a certain amount of time, DB2 physically closes the page set for the R/O DB2s to remove the inter-DB2 R/W interest. See Figure 67 for an example of this.

Scenarios Showing P-Lock Operations

Figure 66 shows a typical sequence of events for P-locking and P-lock negotiations between two members of a data sharing group.

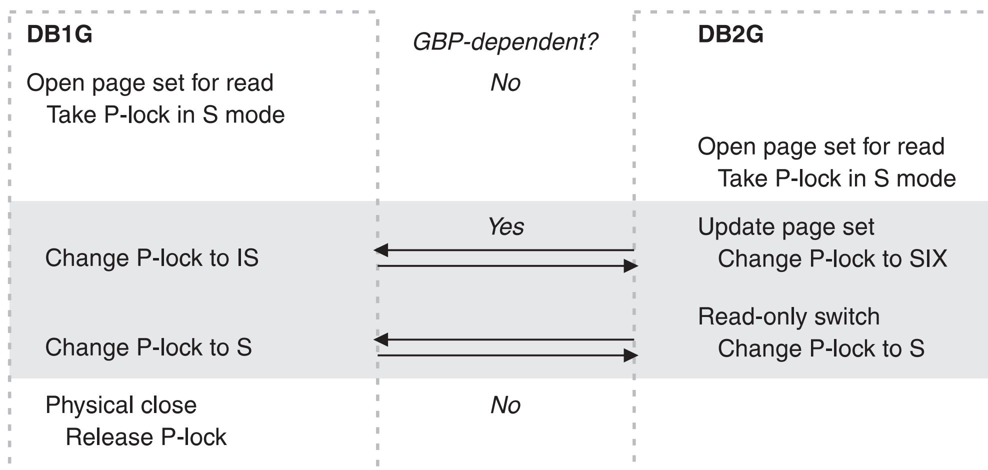

Figure 66. P-lock Operations between Two Members. The arrows indicate that the members are negotiating P-lock modes.

Figure 67 shows what happens when a single updater remains.

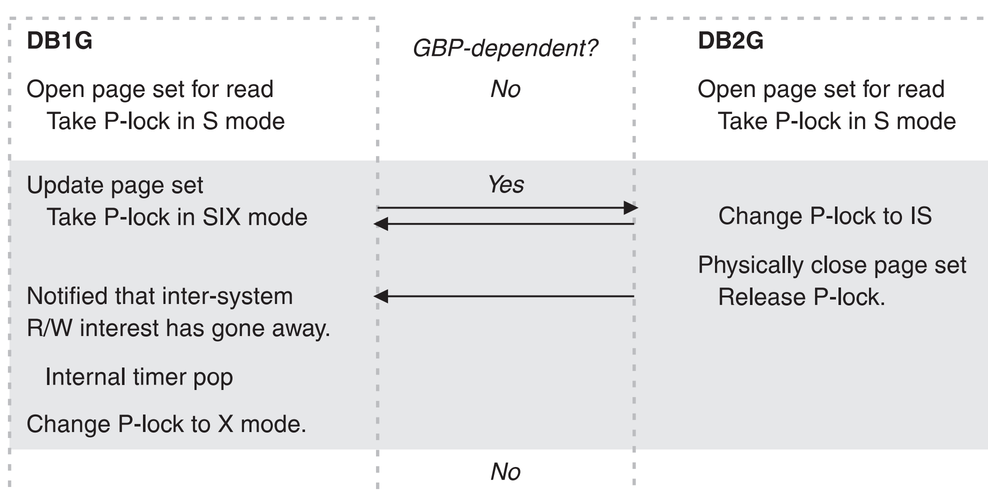

Figure 67. P-lock Operations when Single Updater Remains. When the reader physically closes the page set, the updater does not remove the GBP-dependency immediately.

Tuning Recommendation

To avoid having DB2 go in and out of GBP-dependency too often, tune the following subsystem parameters that affect when data sets are switched to a different state:

- LOGLOAD (CHECKPOINT FREQ field of installation panel DSNTIPN)
- PCLOSEN (DSN6SYSP macro)
- PCLOSET (DSN6SYSP macro)

See Section 5 (Volume 2) of *Administration Guide* for more information about these parameters.

General-use Programming Interface

Displaying GBP-dependent Page Sets

To find out if a particular page set is GBP-dependent, use the DISPLAY DATABASE command with the LOCKS option:

```
-DB1G  DISPLAY DB(DSN8D51A) SPACE(DSN8S51D) LOCKS
```

Output similar to the following is generated:

```
NAME     TYPE PART STATUS            CONNID   CORRID       LOCKINFO
-------- ---- ---- ------------------ -------- ------------ ----------
DSN8S51D   TS       RW                                        H-IX,PP,I
                                      MEMBER NAME DB1G (CO)
DSN8S51D   TS       RW                                        H-IX,PP,I
                                      MEMBER NAME DB2G
```

Figure 68. Sample DISPLAY DATABASE Output Showing GBP-dependent Table Spaces

Page set P-locks are identified by a member name rather than a correlation ID. They are also identified by 'PP' as the lock unit. If any of the P-locks shown in the output have a lock state of NSU, SIX, or IX, then the identified page set is GBP-dependent. Thus, the above output shows that DSN8S51D is GBP-dependent.

Determining GBP-Dependency for a Particular Member: There might be a time when you need to know what the impact is to bring down a particular member or disconnect a particular member from the group buffer pool. You can use the DISPLAY BUFFERPOOL command with the LIST options to discover whether this DB2 member has any page sets opened that are GBP-dependent:

```
-DB1G DISPLAY BUFFERPOOL(BP0) LIST(*)
```

A report like the following is produced:

```
DSNB401I -DB1G BUFFERPOOL NAME BP0, BUFFERPOOL ID 0, USE COUNT 4
DSNB402I -DB1G VIRTUAL BUFFERPOOL SIZE = 1000 BUFFERS
             ALLOCATED       =     1000   TO BE DELETED   =       0
             IN-USE/UPDATED  =       15
DSNB403I -DB1G HIPERPOOL SIZE = 100000 BUFFERS, CASTOUT = YES
             ALLOCATED       =        0   TO BE DELETED   =       0
             BACKED BY ES    =        0
DSNB404I -DB1G THRESHOLDS -
             VP SEQUENTIAL         = 80   HP SEQUENTIAL         = 75
             DEFERRED WRITE        = 85   VERTICAL DEFERRED WRT = 80
             PARALLEL SEQUENTIAL   = 50   ASSISTING PARALLEL SEQT=50
DSNB450I -DB1G TABLESPACE = DSNDB01.DBD01, USE COUNT = 1, GBP-DEP=Y
DSNB450I -DB1G TABLESPACE = DSNDB06.SYSDBASE, USE COUNT = 1, GBP-DEP=Y
DSNB450I -DB1G TABLESPACE = DSNDB04.TS1, USE COUNT = 1, GBP-DEP=N
DSNB451I -DB1G INDEXSPACE = DSNDB06.DSNDLX01, USE COUNT = 1, GBP-DEP=Y
```

Figure 69. Display Indicating Page Sets are Group-Buffer-Pool-Dependent

End of General-use Programming Interface

P-Locking

This section describes P-locks on page sets and on pages, and it describes how to monitor and tune those locks.

Page Set P-Locks

P-locks are used on physical objects stored in buffers (table spaces, index spaces, and partitions). They have complete partition independence; that is, it is possible to have a P-lock on one partition of a page set and not on another. A P-lock on a page set does not necessarily mean that there is a P-lock on the corresponding index.

P-locks do not control concurrency but they do help DB2 track the level of interest in a particular page set or partition and determine the need for cache coherency controls.

P-locks differ from L-locks (transaction locks) in the following important ways:

- P-locks are owned by a subsystem. After a P-lock is obtained for the subsystem, later transactions accessing the locked data do not have to incur the expense of physical locking.
- The mode of a P-lock can be negotiated. If one DB2 subsystem requests a P-lock that another DB2 holds in an incompatible mode, the existing P-lock can be made less restrictive. The negotiation process usually involves registering pages or writing pages to the GBP, and then downgrading the P-lock to a mode that is compatible with the new request.

Displaying Retained P-Locks

Just as with transaction locks, certain P-locks can be retained because of a system failure. A retained P-lock means other DB2s cannot access the data that the P-lock is protecting if the accessing DB2 requests a P-lock in an incompatible state. Thus, if a DB2 fails holding an IX page set P-lock, it is still possible for another DB2 to obtain an IX page set P-lock on the data. See “Active and Retained Locks” on page 182 for more information about retained locks and when they are released.

Use the DISPLAY DATABASE command with the LOCKS option to determine if there are retained locks on a table space, index, or partition. An “R” in the LOCKINFO column indicates that a lock is retained.

Table 34 shows the possible modes of access for a page set and the P-lock state that is retained should DB2 fail.

Table 34. Determining Retained P-Lock State

Your DB2's Interest	Other DB2s' Interest	Retained P-Lock State of My DB2
R/O	None, R/O	None
R/O	R/W	None
R/W	None	X or NSU
Note: NSU stands for “non-shared update.” It acts like an X lock, but is only used during P-lock negotiation from an X to an SIX.		
R/W	R/O	IX
Note: The P-lock is retained in SIX mode if the page set or partition is a type 2 index that is not on a DB2 catalog or directory table space.		
R/W	R/W	IX

Page P-locks

There are times when a P-lock must be obtained on a page to preserve physical consistency of the data between members. These locks are known as *page P-locks*. Page P-locks, are used, for example, when two subsystems attempt to update the same page of data and row locking is in effect. They are also used for GBP-dependent space map pages and GBP-dependent leaf pages for type 2 indexes, regardless of locking level. Page P-locks can also be retained if DB2 fails.

If a type 2 index page set or partition is GBP-dependent, DB2 does not use page P-locks for that index page set or partition if all of the following are true:

- Only one DB2 member is updating the index page set or partition
- There are no repeatable read claimers on the read-only DB2 members for the index page set or partition
- The index is not on a DB2 catalog or directory table space

Because of the possible increase in P-lock activity with row locking, evaluate carefully before using row locking in a data sharing environment. If you have an update-intensive application process, the amount of page p-lock activity might increase the overhead of data sharing. To decrease the possible contention on those page P-locks, consider using page locking and a MAXROWS value of 1 for the table space to simulate row locking. You can get the benefits of row locking without the data page P-lock contention that comes with it. A new MAXROWS value does not take effect until you run REORG on the table space.

Reducing Space Map Page Contention: For applications that do heavy sequential insert processing from multiple members, the contention on the space map can be considerable. The MEMBER CLUSTER option of CREATE TABLESPACE causes DB2 to manage space for inserts on a member-by-member basis instead of by using one centralized space map. Table spaces defined with MEMBER CLUSTER have the following characteristics:

- Data that is inserted by the SQL INSERT statement is not clustered by the implicit clustering index (the first index) or the explicit clustering index.
- DB2 chooses where to locate the data in such a way that avoids lock and latch contention. In general, it tries to insert data in a place that is covered by the locally cached space map page. If it can't find space there, it continues to search through space map pages until it can find a place for which the space map page is available. As a result, space in a data set might not be fully used. But when the data set reaches the maximum number of extents, lock contention can increase and DB2 does use the entire space.
- Each space map covers 199 data pages (either 4KB or 32KB). Because there are more space map pages and some might be partially used, table spaces that are defined with MEMBER CLUSTER can use more DASD.
- To reduce the overhead of reacquiring page P-locks, a page P-lock is held longer for MEMBER CLUSTER table spaces.

The downside to using MEMBER CLUSTER is that data is not inserted in clustering order, so if you have a query application that performs best when data is in clustering order, run REORG on the table space before starting the query application.

Monitoring P-Locks

There is more overhead when there is inter-DB2 R/W interest in a page set than when there is not. Although DB2 does dynamically track inter-DB2 R/W interest, which helps avoid data sharing overhead when it is not needed, you will pay some cost for the advantages of data sharing.

Monitoring P-lock activity, especially page P-locks, can help you determine if it is necessary to take steps to control inter-DB2 R/W interest. If there is excessive global contention that cannot be resolved by any tuning measures, it might be necessary to reduce the locking overhead by keeping some transactions and data together on a single system.

How to Find Information about Page Set P-Locks: You can use the DISPLAY DATABASE command with the LOCKS option to find out information about page set P-locks, including what DB2 subsystem is holding or waiting for P-locks, and whether there are P-locks being held because of a DB2 failure. Figure 68 on page 241 has a sample of output obtained from the command. A “PP” in the LOCKINFO field of the output indicates that a particular lock is a page set or partition P-lock.

Information about P-locks can be obtained by the statistics and accounting traces, along with information about transaction locking. Performance class 20 (IFCID 0251) also contains information about P-lock negotiation requests. IFCID 0251 is mapped by DSNDQW04.

How to Find Information about Page P-Locks: Page P-locking activity is recorded along with the rest of the data sharing locking information in the statistics and accounting trace classes. You can find more detail about those page P-locks in performance trace class 21 (IFCID 0259). IFCID 0259 allows you to monitor page P-locking without having to turn on a full DB2 lock trace. IFCID 0259 is mapped by DSNDQW04.

Read Operations

This section describes how the process of reading data is changed for data sharing.

Where DB2 Looks for a Page

DB2 searches in this order:

1. In the virtual buffer pool. If the page is invalid, it refreshes the page from the group buffer pool (or DASD).
2. In the hiperpool. If the page is invalid, it refreshes the page from the group buffer pool (or DASD). (This step is skipped for GBPCACHE ALL page sets.)
3. In the group buffer pool. If it's not there, it refreshes the page in the virtual buffer pool from DASD.

Testing the Page Validity

Part of the process of controlling cache coherency is testing to see if a page that is referenced in the buffer pool must be refreshed from the group buffer pool or DASD because it might no longer be the most current version of the data. This is known as testing the page *validity*. Because DB2 tracks the level of interest in a page set across the group, it is not always necessary to make this test. Table 35 indicates when this test is performed.

For duplexed group buffer pools, only the primary structure is used for cross-invalidations.

Table 35. Determining when Page Validity Must be Tested

Your DB2's Interest	Other DB2s' Interest	Test Page Validity?
R/O	None, R/O	No
R/O	R/W	Yes
R/W	None	No
R/W	R/O	No
R/W	R/W	Yes

Prefetch Processing

DB2's prefetch processing for GBP-dependent page sets and partitions varies depending on what release of MVS you are running and what level (CFLEVEL) of coupling facility the group buffer pool is allocated in.

When running on MVS 5.1, or if the group buffer pool is allocated in a coupling facility with CFLEVEL=0 or 1, then DB2 reads and registers one page at a time in the group buffer pool.

When running on MVS 5.2 *and* the group buffer pool is allocated in a coupling facility with CFLEVEL=2 or higher, DB2 can register the entire list of pages that are being prefetched with one request to the coupling facility. This can be used for sequential prefetch (including sequential detection) and list prefetch.

DB2 does not include on the list any valid pages that are found in the local virtual buffer pool or hiperpool.

For those pages that are cached as “changed” in the group buffer pool, or those that are locked for castout, DB2 still retrieves the changed page from the group buffer pool one at a time. For large, sequential queries, there most likely won't be any changed pages in the group buffer pool.

For pages that are cached as “clean” in the group buffer pool, DB2 can get the pages from the group buffer pool (one page at a time), or can include the pages in the DASD read I/O request, depending on which is most efficient.

Determining if DB2 Registered a List of Pages: If DB2 registers a list of pages during prefetch, there will be a non-zero value in field QBGLAX in IFCID 0002 (see **J** in Figure 78 on page 263). You can also use the DISPLAY GROUPBUFFERPOOL command with the MDETAIL option (see message DSNB789I).

Caching Pages that Are Read in from DASD

You can cache pages in the group buffer pool as they are read in by specifying GBPCACHE ALL when you create or alter a table space or index. When you choose ALL, pages are copied to the group buffer pool as they are read in from DASD, even if there is no inter-DB2 R/W interest in those pages.

However, when there is only a single DB2 that has exclusive R/W interest in the page set, pages are not cached in the group buffer pool when they are read in. | They can, however, be cached in the hiperpool, if one exists.

Choosing GBPCACHE ALL does not prevent DB2 from continuing to cache changed pages in the group buffer pool before writing them to DASD (the function provided by the default GBPCACHE CHANGED).

General-use Programming Interface

Example: Here is an example of using the GBPCACHE clause to cache read-only page sets in the group buffer pool:

```
ALTER TABLESPACE DSN8D51A.DSN8S51D
  GBPCACHE ALL;
```

See Chapter 6 of *SQL Reference* for more information about the GBPCACHE clause.

End of General-use Programming Interface

Why Choose GBPCACHE ALL?: GBPCACHE ALL avoids having several different members read the same page in from DASD. It is most useful for workloads where there is primarily inter-DB2 read interest. To prevent double buffering for clean pages, hiperpools are not used for page sets or partitions that | are defined with the GBPCACHE ALL optionunless the DB2 is an exclusive updater . In a non-data-sharing environment, hiperpools are still used regardless of the GBPCACHE option.

Planning Consideration: If you use the GBPCACHE ALL option, it increases the need for coupling facility resources: processing power, storage, and channel utilization. If you cannot afford the additional strain on coupling facility resources, consider using a 3990 Model 6 cache controller that exploits record and track level caching to achieve caching benefits for a read-intensive work load.

Write Operations

With data sharing, DB2 must first write changed data to the group buffer pool before writing that changed data to DASD. This section includes the following topics:

“Writing to the Group Buffer Pool”
“Writing to DASD from the Group Buffer Pool” on page 248

Writing to the Group Buffer Pool

With data sharing, DB2 still performs deferred writes for DB2 table spaces, indexes or partitions. However, when an update is to a page set that has inter-DB2 R/W interest, DB2 forces the updated pages to the group buffer pool when the transaction commits, or before. Updated pages can be written to the group buffer pool before the updating transaction is committed when:

- One of the deferred write thresholds is reached.
- The buffer pool is short of reassignable buffers because writes to the group buffer pool cannot keep up with update activity in the buffer pool. The shortage of buffers can occur when the deferred write thresholds are too high, or if the application is not committing frequently enough— in a data sharing environment, the commits make buffers reassignable.
- An updated page has stayed in the buffer pool for a long period of time since it was last referenced or updated (such as with a long running transaction that doesn't issue frequent commits). In this case, a system checkpoint can clean out the buffer pool before the commit.
- The same page is required for update by another system because there is no conflict on transaction locking (such as page sets that are using row locking, index pages, space map pages, and so on). This write is part of the page P-lock negotiation process.

When a page of data is written to the group buffer pool, all copies of that page cached in other members' buffer pool are invalidated. This means that the next time one of those members needs that page, the page must be refreshed from the group buffer pool (or DASD).

Before an updated page is written to the group buffer pool or DASD, DB2 also ensures that the last update log record for that page is externalized to the active log. This is necessary to ensure that updates can be backed out when necessary.

When committing an updating transaction, pages that were updated but not yet written to the group buffer pool are synchronously written to the group buffer pool. If a group buffer pool is required and unavailable (because of a channel or hardware failure) at the time the transaction commits, DB2 places all the transaction's updated pages on the logical page list (LPL) associated with each page set. After the problem is fixed, use a START DATABASE command with the SPACENAM option to recover the pages on the LPL.

Writing to a duplexed group buffer pool: When a group buffer pool is duplexed, the following events occur:

1. For some fixed number of pages that must be written, do the following for each page:
   a. Write the page to the secondary structure asynchronously
   b. Write the page to the primary structure synchronously
2. After all pages have been written to the primary structure, DB2 checks to see if all pages have been written to the secondary structure. If some pages are still to be written, DB2 forces the completion of those writes.

Writing to DASD from the Group Buffer Pool

The process of writing pages from the group buffer pool to DASD is called *castout*. Because there is no physical connection between the group buffer pool and DASD, the castout process involves reading the page from the group buffer pool into a particular DB2's private buffer (not part of the buffer pool storage) and writing the page from the private buffer to DASD. This DB2 is the owner of the castout process for the page set or partition. The DB2 that is assigned ownership of castout is the DB2 subsystem that had the first update intent (except during restart) on the page set or partition. After the castout ownership is assigned, subsequent updating DB2 subsystems become backup owners. One of the backup owners becomes the castout owner when the original castout owner no longer has read/write interest in the page set.

Other DB2s can write this page to the group buffer pool even as the page is being cast out. Some events explicitly cause pages to be cast out to DASD, such as the STOP DATABASE command.

Castout also occurs when:

- The number of changed pages for a castout class queue exceeds a class threshold value. Castout class thresholds are described in “Group Buffer Pool Class Castout Threshold” on page 252.
- The total number of changed pages for a group buffer pool exceeds a group buffer pool threshold value, described in “Group Buffer Pool Castout Threshold” on page 253.
- The group buffer pool checkpoint is triggered. See “Group Buffer Pool Checkpoint” on page 249 for more information.
- There is no more inter-DB2 R/W interest in the page set.
- The group buffer pool is being rebuilt, but the alternate group buffer pool is not large enough to contain the pages from the group buffer that is being rebuilt.

Pages that are cast out as a result of meeting a threshold remain cached in the group buffer pool, and the buffers are available for stealing. Pages that are cast out because there is no more shared interest in the page set are purged from the group buffer pool.

Casting out from a duplexed group buffer pool: DB2 casts out data to DASD only from the primary structure. After a set of pages has been cast out, the same set of pages is deleted from the secondary structure. See the DELETE NAME LIST counter in the DISPLAY GROUP BUFFERPOOL MDETAIL report for how many times this event occurs. DB2 ensures that any pages that might have been written

to the group buffer pool during castout processing are not deleted from the
secondary structure.

General-use Programming Interface

Determining the Castout Owner: Use the DISPLAY DATABASE command with the LOCKS option to display the current castout owner for a given page set:

```
-DB1G    DISPLAY DATABASE(TESTDB) SPACE(*) LOCKS
```

The castout owner for a particular page set or partition is displayed with (CO) by the member name, as shown in Figure 70.

```
DSNT360I -DB1G ***********************************************************
DSNT361I -DB1G *  DISPLAY DATABASE SUMMARY
               *    GLOBAL LOCKS
DSNT360I -DB1G ***********************************************************
DSNT362I -DB1G    DATABASE = TESTDB  STATUS = RW
                  DBD LENGTH = 4028
DSNT397I -DB1G
NAME     TYPE PART STATUS            CONNID   CORRID       LOCKINFO
-------- ---- ---- ------------------ -------- ------------ ---------
TBS43    TS    01  RW                BATCH    SELEC        H-IS,P,C
    -                   MEMBER NAME DB1G
TBS43    TS    01  RW                                      H-IS,PP,I
    -                   MEMBER NAME DB1G
TBS43    TS    01  RW                BATCH    UPD2         H-IX,P,C
    -                   MEMBER NAME DB2G
TBS43    TS    01  RW                                      H-SIX,PP,I
    -                   MEMBER NAME DB2G     (CO)
TBS43    TS    02  RW                BATCH    SELEC        H-IS,P,C
    -                   MEMBER NAME DB1G
TBS43    TS    02  RW                BATCH    UPD2         H-IX,P,C
    -                   MEMBER NAME DB2G
⋮
```

Figure 70. Partial DISPLAY DATABASE Output Showing Castout Owner for a Partition

End of General-use Programming Interface

Group Buffer Pool Checkpoint

When a group buffer pool is damaged, all changed data that belong to GBP-dependent page sets must be recovered to the page sets from the DB2 logs. The number of log records that need to applied to the page set is determined by the frequency of the group buffer pool checkpoint. *Group buffer pool checkpoint* is
the process of writing all changed pages in the group buffer pool (the primary one
only, if duplexed) to the page set. The purpose of the checkpoint is to reduce the
amount of time needed to recover data in a group buffer pool. At group buffer pool checkpoint, DB2 records in the member BSDSs and SCA the log record sequence number from which group buffer pool recovery would need to take place. Group buffer pool checkpoint does not record anything in the log.

The group buffer pool checkpoint is triggered by the *structure owner*. The structure owner is usually the first DB2 that connects to this group buffer pool, although the ownership can change over time. Message DSNB798I in the DISPLAY GROUPBUFFERPOOL output shows which DB2 is the current structure owner.

Default Checkpoint Frequency: The default checkpoint frequency is 8 minutes. You can change the default checkpoint frequency by using the ALTER GROUPBUFFERPOOL command, described in “Changing the Checkpoint Frequency” on page 266.

Tuning the Group Buffer Pool Checkpoint Interval: At the group buffer pool checkpoint, the structure owner records, in the SCA and in its own BSDS, the LRSN from which group buffer pool recovery should take place, if necessary. This LRSN is displayed in the DISPLAY GROUPBUFFERPOOL output.

DB2 has two possible ways of gathering checkpoint information:

- By issuing many “read directory info” requests

  This method is used when your Sysplex is not at the maintenance level required for the more efficient method described below. These “read directory info” requests are reported as an increase in the SRB time of the *ssnm*DBM1 address space, especially for the structure owner. This is because of the increased number of times it has to read the directory entries to compute the recovery LRSN.

- By issuing one “read castout statistics” request

  This method is used when the group buffer pools is allocated in a coupling facility at CFLEVEL=5 or higher, and when APAR OW28460 is applied to MVS on all members of the data sharing group. (Or when the members are running at OS/390 Release 6, or a subsequent release.)

  When you look at the MDETAIL report from DISPLAY GROUPBUFFERPOOL, as shown in Figure 71 on page 251, you will see significantly fewer “read directory info” requests when you have the proper maintenance applied for this feature.

Recommendation: Because group buffer pool checkpoint consumes processor, coupling facility, and I/O resources and can impact other work in the system, balance the performance impact of frequent group buffer pool checkpoints (the lower the checkpoint interval, the higher the system resource consumption) with the recovery impact of infrequent checkpoints (the lower the checkpoint interval, the faster DB2 can recover from a group buffer pool failure). The default checkpoint interval of 8 minutes is a good balance between the performance and recovery considerations in most cases.

If the resource consumption of the group buffer pool checkpoint is higher than you prefer:

- Apply the proper maintenance and allocate the group buffer pool in a CFLEVEL=5 coupling facility to take advantage of the checkpoint performance enhancement.
- Increase the checkpoint interval to have the checkpoint occur less frequently.

If the checkpoint is not moving the recovery LRSN forward fast enough, decrease the checkpoint interval.

General-use Programming Interface

The following instrumentation helps you more effectively monitor and tune the group buffer pool checkpoint:

- DISPLAY GROUPBUFFERPOOL shows which member is the structure owner and also shows the group buffer pool checkpoint recovery LRSN:

```
DSNB798I -DB1G LAST GROUP BUFFER POOL CHECKPOINT 17:23:21 MAY  9, 1996
               GBP CHECKPOINT RECOVERY LRSN             = ACD74C01EE30
               STRUCTURE OWNER                          = DB1G
```

- The DISPLAY GROUPBUFFERPOOL MDETAIL contains the number of checkpoints that occurred for this group buffer pool. It also includes the number of requests made to read directory entries, which are used to determine the recovery LRSN. The statistics trace also includes this information.

  If you are experiencing surges of coupling facility use, it could be related to group buffer pool checkpointing. Examine the number of read directory info requests. If there are many requests per checkpoint, you can probably benefit from applying the proper maintenance to use the group buffer pool checkpoint performance enhancement.

```
DSNB778I -DB1G CASTOUT THRESHOLDS DETECTED
               FOR CLASSES                             = 3
               FOR GROUP BUFFER POOL                   = 1
             GBP CHECKPOINTS TRIGGERED                 = 1
             PARTICIPATION IN REBUILD                  = 0
DSNB796I -DB1G CASTOUTS
               PAGES CASTOUT                           = 18
               UNLOCK CASTOUT                          = 3
               READ CASTOUT CLASS                      = 5
               READ CASTOUT STATISTICS                 = 6
               READ DIRECTORY INFO                     = 0
DSNB779I -DB1G   ENGINES NOT AVAILABLE
                 FOR CASTOUT                           = 0
                 FOR WRITING                           = 0
```

Figure 71. DISPLAY GROUPBUFFERPOOL MDETAIL report

- IFCID 0261, which gives summary statistics for each group buffer pool checkpoint. You can use this record to estimate the processor cost and to monitor the coupling facility interactions for each group buffer pool checkpoint.
- IFCID 0263, which gives summary statistics for the castouts. You can use this record to monitor the castout activity that is caused by each group buffer pool checkpoint (or castout that's triggered for any other reason).

End of General-use Programming Interface

If the recovery LRSN for group buffer pool checkpoint is not advancing as fast as you want, determine if there have been any DASD or coupling facility connectivity problems that are impairing DB2's ability to cast out.

Group Buffer Pool Thresholds

You can control the castout process by changing the two group buffer pool thresholds:

- Group buffer pool castout threshold
- Class castout threshold

As Figure 72 illustrates, the group buffer pool castout threshold is a percentage of changed pages in the group buffer pool. The class castout threshold is the percentage of changed pages in the group buffer pool per *castout queue*.

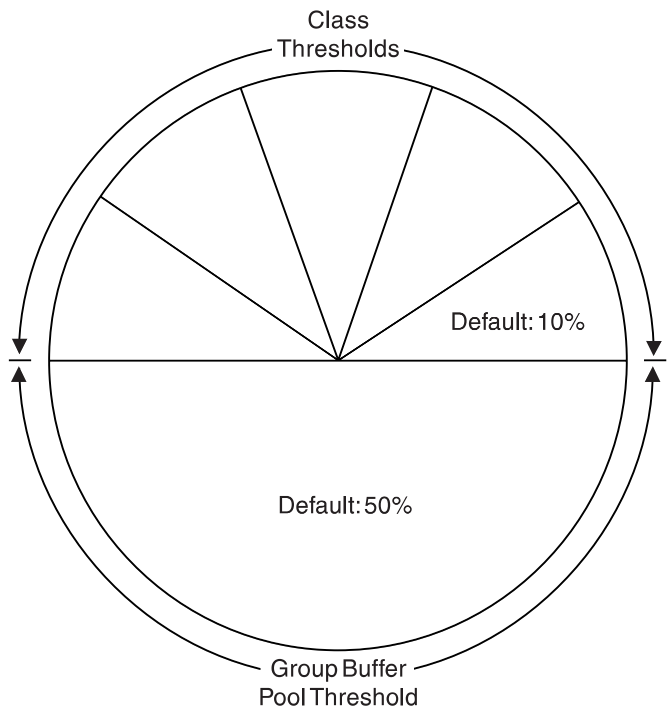

Figure 72. Group Buffer Pool Castout Thresholds. The group buffer pool castout threshold is a percentage of the total number of pages in the group buffer pool. The castout class threshold is a percentage of the total number of changed pages in the group buffer pool for a given castout class.

Group Buffer Pool Class Castout Threshold

In each group buffer pool there is a fixed number of castout class queues. This number is an internal value set by DB2. DB2 internally maps updated pages that belong to the same page sets or partitions to the same castout class queues. Because of a limited number of castout class queues, it is possible that more than one page set or partition gets mapped into the same castout class queue. This internal mapping scheme is the same across all sharing subsystems.

When DB2 writes changed pages to the group buffer pool, it determines how many changed pages are on a particular class castout queue. If the number of changed pages on a specified castout class queue exceeds the threshold, DB2 casts out a number of pages from that queue.

How DB2 determines the castout threshold for a duplexed group buffer pool:
For duplexed group buffer pools, DB2 uses the smaller of the number of data

entries in the primary and secondary structures. For example, if the primary structure contains 5000 data entries and the secondary structure contains 1000 data entries, and CLASST is 10%, then DB2 sets CLASST to 100 pages (10% of 1000 pages).

Default Group Buffer Pool Class Castout Threshold: The default for the class castout is 10, which means that castout is initiated for a particular page set or partition when 10 percent of the group buffer pool contains changed pages for the class.

Group Buffer Pool Castout Threshold

This threshold determines the total number of changed pages that can exist in the group buffer pool before castout occurs. DB2 casts out enough class castout queues to bring the number of changed pages below the threshold. DB2 periodically determines whether the threshold is exceeded.

How DB2 determines the group buffer pool castout threshold for a duplexed group buffer pool: For duplexed group buffer pools, DB2 uses the smaller of the number of data entries in the primary and secondary group buffer pools. For example, if the primary contains 5000 data entries and the secondary contains 1000 data entries, and GBPOOLT is 50%, then DB2 sets GBPOOLT to 500 pages (50% of 1000 pages).

Default Group Buffer Pool Castout Threshold: The default value for the group buffer pool castout threshold is 50, which means that when the group buffer pool is 50 percent full of changed pages, castout is initiated.

Guidelines

In most cases, we suggest you use the default values (10 percent for the class threshold and 50 percent for the group buffer pool threshold). Depending on your work load, these values help reduce DASD contention during castout.

If you find that some writes to the group buffer pool cannot occur because of a lack of storage in the group buffer pool, increase the group buffer pool size, or decrease the group buffer pool castout thresholds. One way to tell if this is happening is to see the detail report of DISPLAY GROUPBUFFERPOOL. An example report is shown in Figure 76 on page 260. The field indicated by **E** is the one to watch for this type of problem.

Tuning the Castout Thresholds: The following can help you more effectively monitor the group buffer pool castout thresholds:

- The DISPLAY GROUPBUFFERPOOL command with the MDETAIL option.
- The DB2 statistics trace.
- IFCID 0262, which gives summary statistics for each time that the GBPOOLT threshold is reached. You can use this record to monitor how efficiently the GBPOOLT threshold is handling the castout work.
- IFCID 0263, which gives summary statistics for the castouts done by the page set and partition castout owners. All castout work for a given page set or partition is done by the castout owner. You can use this record to monitor the efficiency with which the page set or partition castout owners are doing their work.

General-use Programming Interface

Example from MDETAIL Report: Here is partial output from the command DISPLAY GROUPBUFFERPOOL (GBP0) MDETAIL:

```
⋮
DSNB796I -DB1G CASTOUTS
               PAGES CASTOUT                          = 217
               UNLOCK CASTOUT                         = 35
               READ CASTOUT CLASS                     = 47
               READ CASTOUT STATISTICS                = 47
               READ DIRECTORY INFO                    = 290
⋮
```

The UNLOCK CASTOUT counter should always be significantly less than the PAGES CASTOUT counter. If it is not (for example, if "unlock castout" is more than half of "pages cast out"), then the castout write I/O is not being done efficiently (the number of pages written per I/O is normally close to the number you get by dividing PAGES CASTOUT by UNLOCK CASTOUT). This is probably because you have random update patterns on the DB2 data.

End of General-use Programming Interface

Effect of GBPCACHE ALL on Guidelines: If you are using a group buffer pool to cache pages as they are read in from DASD (GBPCACHE ALL page sets), then consider lowering the threshold values to allow more space for caching those clean pages.

Monitoring Group Buffer Pools

This section describes how you can monitor group buffer pool activity:

- “Using the MVS D XCF,STR Command”
- “Using the Coupling Facility Activity Report of RMF” on page 255
- “Using the DISPLAY GROUPBUFFERPOOL Command” on page 256
- “Using DB2 Statistics Trace” on page 258

Using the MVS D XCF,STR Command

You can use MVS command D XCF,STR to get information about coupling facility structures:

- CFRM policy definition
- Preference list
- Coupling facility name
- Connections
- Duplexing status

The following command displays information about GBP1 in group DSNDB0G:

```
D XCF,STR,STRNAME=DSNDB0G_GBP1
```

This particular group buffer pool is duplexed, so you see information about both allocations of the structure (the old structure is the primary structure, and the new structure is the secondary one). Output similar to the following is produced:

```
DISPLAY XCF
STRNAME: DSNDB0G_GBP1
 STATUS: REASON SPECIFIED WITH REBUILD START:
           OPERATOR INITIATED
         DUPLEXING REBUILD
         REBUILD PHASE: DUPLEX ESTABLISHED
  POLICY SIZE    : 204800 K
  POLICY INITSIZE: 102400 K
  REBUILD PERCENT: 1
  DUPLEX         : ALLOWED
  PREFERENCE LIST: CACHE01  LF01
  EXCLUSION LIST IS EMPTY

 DUPLEXING REBUILD NEW STRUCTURE
 -------------------------------
  ALLOCATION TIME: 10/14/1997 17:01:48
  CFNAME         : LF01
  COUPLING FACILITY: ND01...
                   PARTITION: 0   CPCID: 00
  ACTUAL SIZE    : 102400 K
  STORAGE INCREMENT SIZE: 256 K
  VERSION        : AF6935AA 78004403
  DISPOSITION    : DELETE
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 2

 DUPLEXING REBUILD OLD STRUCTURE
 -------------------------------
  ALLOCATION TIME: 10/14/1997 17:00:38
  CFNAME         : CACHE01
  COUPLING FACILITY: ND02...
                   PARTITION: 0   CPCID: 00
  ACTUAL SIZE    : 102400 K
  STORAGE INCREMENT SIZE: 256 K
  VERSION        : AF693567 9B48B802
  ACCESS TIME    : 0
  MAX CONNECTIONS: 32
  # CONNECTIONS  : 2

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 DB2_DB1G         01 00010001 UTEC469  DB1GDBM1 002E ACTIVE NEW,OLD
 DB2_DB2G         02 00020001 UTEC469  DB2GDBM1 0031 ACTIVE NEW,OLD
```

Figure 73. MVS Command D XCF Showing Group Buffer Pool Information

For more information about the D XCF command, see *MVS/ESA System Commands*.

Using the Coupling Facility Activity Report of RMF

The report in Figure 74 on page 256 shows, in the CHNGD field (**B**) that 3.2 percent of all accesses that were supposed to be done synchronously had to be done asynchronously. The NO SCH field (**A**) indicates the amount of time that requests were queued because of a lack of subchannel resources. If the value in **B** is over 10 percent or so, and there is a non-zero value in **A**, it could mean that your configuration does not have enough subchannels to handle the work.

```
COUPLING FACILITY NAME = LF02
------------------------------------------------------------------------------------------------------------
                                          COUPLING  FACILITY  STRUCTURE  ACTIVITY
------------------------------------------------------------------------------------------------------------

STRUCTURE NAME = DSNDB0G_GBP3      TYPE = CACHE
          # REQ    -------------- REQUESTS -------------   -------------- QUEUED REQUESTS --------------
SYSTEM    TOTAL             #     % OF  -SERV TIME(MIC)-          (ARRIVAL  RATE) --- QUEUE TIME(MIC)--
NAME      AVG/SEC          REQ    ALL     AVG   STD_DEV           MIN   AVG   MAX          AVG  STD_DEV

STLABC1     590K   SYNC    571K  96.7%   257.0    119.3    HPRIO     0 335.8   423  A NO SCH 12666   10488
          459.5    ASYNC   465    0.1%  2980.3   2401.4    LPRIO     0   0.0     0
              B CHNGD    19K    3.2%  INCLUDED IN ASYNC  DUMP      0   0.0     0  DUMP     0.0     0.0

⋮
```

Figure 74. Portion of RMF Coupling Facility Structure Activity Report

General-use Programming Interface

Using the DISPLAY GROUPBUFFERPOOL Command

Use the DISPLAY GROUPBUFFERPOOL command to display information about group buffer pools. Assume that you want a summary report about group buffer pool zero, including all connections to that group buffer pool. Enter the following command:

```
-DB1G DISPLAY GROUPBUFFERPOOL(GBP0) CONNLIST(YES)
```

| Here is what the display might look like, assuming that the group buffer pool is
| duplexed:

Here is what the display might look like:

```
DSNB750I -DB1G DISPLAY FOR GROUP BUFFER POOL GBP0 FOLLOWS
DSNB755I -DB1G DB2 GROUP BUFFER POOL STATUS
                 CONNECTED                                  = YES
                 CURRENT DIRECTORY TO DATA RATIO            = 5.4
                 PENDING DIRECTORY TO DATA RATIO            = 6.0
DSNB756I -DB1G   CLASS CASTOUT THRESHOLD                    = 10%
                 GROUP BUFFER POOL CASTOUT THRESHOLD        = 50%
                 GROUP BUFFER POOL CHECKPOINT INTERVAL      = 8 MINUTES
                 RECOVERY STATUS                            = NORMAL
                 AUTOMATIC RECOVERY                         = Y
DSNB757I -DB1G MVS CFRM POLICY STATUS FOR DSNDB0G_GBP0      = NORMAL
                 MAX SIZE INDICATED IN MVS POLICY           = 61440 KB
                 DUPLEX INDICATOR IN POLICY                 = ENABLED
                 CURRENT DUPLEXING MODE                     = DUPLEX
                 ALLOCATED                                  = YES
DSNB758I -DB1G   ALLOCATED SIZE                             = 61440 KB
                 VOLATILITY STATUS                          = NON-VOLATILE
                 REBUILD STATUS                            = DUPLEXED
                 CFNAME                                     = LF01
                 CFLEVEL                                    = 5
DSNB759I -DB1G   NUMBER OF DIRECTORY ENTRIES                = 61394
                 NUMBER OF DATA PAGES                       = 11370
                 NUMBER OF CONNECTIONS                      = 3
DSNB798I -DB1G LAST GROUP BUFFER POOL CHECKPOINT 17:31:23 MAY  9, 1996
                 GBP CHECKPOINT RECOVERY LRSN              = ACD74C77388C
                 STRUCTURE OWNER                           = DB1G
DSNB799I  -DB1G SECONDARY GBP IS ALLOCATED
                 ALLOCATED SIZE                            = 61440 KB
                 VOLATILITY STATUS                         = NON-VOLATILE
                 CFNAME, CFLEVEL                           = LF01, 5
                 NUMBER OF DIRECTORY ENTRIES               = 61394
                 NUMBER OF DATA PAGES                      = 11370
DSNB766I -DB1G THE CONNLIST REPORT FOLLOWS
DSNB767I -DB1G CONNECTION NAME = DB2_DB1G     , CONNECTION STATUS = A
                   CONNECTOR'S RELEASE                    =     6100
DSNB767I -DB1G CONNECTION NAME = DB2_DB2G     , CONNECTION STATUS = A
                   CONNECTOR'S RELEASE                    =     6100
DSNB767I -DB1G CONNECTION NAME = DB2_DB3G     , CONNECTION STATUS = F
                   CONNECTOR'S RELEASE                    =     6100
DSNB769I -DB1G THE CONNLIST REPORT IS COMPLETE
DSNB790I -DB1G DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE
DSN9022I -DB1G DSNB1CMD 'DISPLAY GROUPBUFFERPOOL' NORMAL COMPLETION
```

\# *Figure 75. Summary Report with Connection List Included*

See Chapter 2 of *Command Reference* for more information about the syntax of the command.

Detailed statistics can be displayed using the GDETAIL or MDETAIL keywords. See Figure 76 on page 260 to see what statistical information looks like.

Hint: For an easy way to collect interval statistics for performance analysis, create a batch job that issues the following command periodically:

```
-DB1G DISPLAY GROUPBUFFERPOOL(*) GDETAIL(INTERVAL)
```

The first time you run the batch job is the base which purges existing statistics and resets the interval. If you run the job the second time 5 minutes after the first, you can continue running the job every 5 minutes to gather meaningful statistical data on group buffer pool activity.

End of General-use Programming Interface

Using DB2 Statistics Trace

Use DB2 statistics class 1 to do high-level monitoring of DB2 subsystem activity. A new data section, mapped by DSNDQBGL, records statistics for a DB2 member's use of group buffer pools. The counters are cumulative since the time the member connected to a particular group buffer pool.

Statistics reporting intervals are not synchronized across the members of the data sharing group. For counters that are pertinent to the entire data sharing group, like group buffer statistics, DB2 PM group-scope statistics reports combine the data of the individual members and present it to for the entire group. The member data is apportioned to the same user-specified interval. DB2 PM presents the synchronized statistics intervals for each member, adds up the counters across all members and presents them as statistics on a per-group basis.

Consider also using DB2 PM to do group-scope exception reporting when a particular counter exceeds a user-specified value.

For more information about using the statistics report, see "What to Look for in a DB2 PM Statistics Report" on page 262.

Making Sure Group Buffer Pools are the Right Size and Ratio

One of the critical tuning factors in a DB2 data sharing configuration is the size of the group buffer pools. There are three aspects of group buffer pool (cache structure) size that need to be considered:

- Total structure size

  As described in "Storage for Coupling Facility Structures" on page 67, the total structure size of a group buffer pool is specified in the coupling facility policy definition for the cache structure.

- Number of directory entries

  A directory entry contains control information for one database page, no matter in how many places that page is cached. For example, if page P1 is cached in the group buffer pool and in the virtual buffer pools of three members, that page still has only one directory entry.

  Each directory entry is 208 bytes for 4KB pages and 264 bytes for 32K pages, assuming CFLEVEL=0. See *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide* for information about sizes for other levels. A directory entry is used by the coupling facility to determine where to send cross-invalidation signals when a page of data is changed or when that directory entry must be reused.

- Number of data entries

  Data entries are the actual places where the data page resides. These are 4KB or 32KB in size (the same size as the data page).

The number of directory entries and data entries in the coupling facility structure is determined by the size specified in the coupling facility policy and the ratio of directory entries to data pages. The ratio is automatically defined for each group buffer pool at the time the first member of the DB2 group is installed. The default value used is 5 directory entries per data page.

When choosing a value for the ratio, consider the following:

- The higher the write-to-read ratio, the more directory entries are needed.
- The greater the number of distinct pages that are cached, the more directory entries are needed. Each distinct page requires a single directory entry, no matter how many DB2s are using that page. If there are “hot” pages, distinct pages that are cached in many members' buffer pools, fewer directory entries are needed than if many different pages are cached.

After installation, you can change the ratio with the ALTER GROUPBUFFERPOOL command. However, the change does not take effect until the next time the group buffer pool is allocated.

The following sections describe the symptoms of values that are not ideal for best performance and how you can fix the problems.

Group Buffer Pool Size is Too Small

When the group buffer pool is too small, the following problems can occur:

- The thresholds for changed pages is reached more frequently, causing data to be cast out to DASD more often.

  If castout cannot keep up with the writes to the group buffer pool, a more serious problem occurs: pages are instead written to the logical page list and are unavailable until they are recovered. See “Monitor Storage of the Group Buffer Pool” on page 261 for a hint about avoiding this problem. See “Problem: Storage Shortage in the Group Buffer Pool” on page 179 for recovery actions should the problem occur.

- Possibly many cross-invalidations caused by reusing existing directory entries, which might require refreshing a page from DASD later when the page is referenced again.

General-use Programming Interface

In any event, pages in the group buffer pool have to be refreshed from DASD more often because they are not in the group buffer pool. You can use the GDETAIL option of the DISPLAY GROUPBUFFERPOOL command to gather detailed statistical information about how often data is returned on a read request to the group buffer pool:

```
-DB1G DISPLAY GROUPBUFFERPOOL(GBP0) GDETAIL(*)
```

Here is what the detail portion of the report output looks like:

```
DSNB783I -DB1G CUMULATIVE GROUP DETAIL STATISTICS SINCE 15:35:23 Mar 17, 1994
DSNB784I -DB1G GROUP DETAIL STATISTICS
                 READS
                   DATA RETURNED A                    = 3845
DSNB785I -DB1G     DATA NOT RETURNED
                     DIRECTORY ENTRY EXISTED B        = 27
                     DIRECTORY ENTRY CREATED C        = 28336
                     DIRECTORY ENTRY NOT CREATED      = 332, 0
                                                         D        .
DSNB786I -DB1G   WRITES
                   CHANGED PAGES                      = 20909
                   CLEAN PAGES                        = 0
                   FAILED DUE TO LACK OF STORAGE E    = 8
                 CHANGED PAGES SNAPSHOT VALUE         = 974
DSNB787I -DB1G   RECLAIMS
                   FOR DIRECTORY ENTRIES F            = 18281
                   FOR DATA ENTRIES                   = 47
                 CASTOUTS                             = 16073
DSNB788I -DB1G   CROSS INVALIDATIONS
                   DUE TO DIRECTORY RECLAIMS G        = 4489
                   DUE TO WRITES                      = 3624
DSNB762I -DB1G   DUPLEXING STATISTICS FOR GBP0-SEC
                    WRITES
                     CHANGED PAGES                    = 20909
                     FAILED DUE TO LACK OF STORAGE    = 8
                    CHANGED PAGES SNAPSHOT VALUE      = 974
DSNB790I -DB1G DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE
DSN9022I -DB1G DSNB1CMD 'DISPLAY GROUPBUFFERPOOL' NORMAL COMPLETION
```

Figure 76. Example Output of Group Detail Statistics

What you need to determine is the *read hit* percentage. To calculate this value, you need to determine how many of the total number of reads were successful in returning data. Use the following formula:

(**A** / (**A** + **B** + **C** + **D** (first number)))× 100

In our example, the calculation is:

(3845 / 32540) × 100 = 11.81%

Data was returned in approximately 12 percent of the read requests to the group buffer pool. This low percentage of “read hits” might indicate that the average residency time for a cached page in group buffer pool is too short. You might benefit from altering the group buffer pool to increase the total size, as described in “Changing the Size of the Group Buffer Pool” on page 266.

However, a low percentage of read hits could be caused by other factors:

- A high read-to-write ratio.

  If you are caching only changed pages, it is to be expected that not many pages you need would be resident in the group buffer pool.

- Random reference patterns.

  Pages that are frequently referenced are most likely to be resident in the group buffer pool. If the application keeps requesting new pages, it is unlikely to find any given page in the group buffer pool.

To determine if the low read hit percentage is a problem, see the field indicated by **B** in the statistics report, shown in Figure 78 on page 263. (The same counter also exists in the accounting report.) Ideally, that field contains 0. A non-zero value there in conjunction with a low read hit percentage can indicate that your group buffer pool is too small.

"Too Few Directory Entries" and "Too Few Data Entries" on page 262 describe how to determine if the problem is caused by a suboptimal ratio of directory entries to data entries.

End of General-use Programming Interface

General-use Programming Interface

Monitor Storage of the Group Buffer Pool: By monitoring the storage use of the group buffer pool, you can avoid data outages caused by a serious lack of storage in the group buffer pool. We recommend issuing periodic DISPLAY GROUPBUFFERPOOL commands with the GDETAIL option. The GDETAIL statistics show a "snapshot" value of the number of changed pages in the group buffer pool. Make sure this snapshot value does not rise significantly above the group buffer pool castout threshold. Figure 77 highlights the key fields from the report.

```
DSNB756I -DB1G   CLASS CASTOUT THRESHOLD                    = 10%
               A GROUP BUFFER POOL CASTOUT THRESHOLD        = 50%
                 GROUP BUFFER POOL CHECKPOINT INTERVAL      = 8 MINUTES
                 RECOVERY STATUS                            = NORMAL
DSNB759I -DB1G   NUMBER OF DIRECTORY ENTRIES                = 61394
               B NUMBER OF DATA PAGES                       = 11370
                 NUMBER OF CONNECTIONS                      = 3
DSNB783I -DB1G CUMULATIVE GROUP DETAIL STATISTICS SINCE 15:35:23 Mar 17, 1994
DSNB784I -DB1G GROUP DETAIL STATISTICS
⋮
DSNB786I -DB1G   WRITES
                   CHANGED PAGES                      = 1576
                   CLEAN PAGES                        = 0
                   FAILED DUE TO LACK OF STORAGE      = 0
                 C CHANGED PAGES SNAPSHOT VALUE       = 311
⋮
```

*Figure 77. Partial Output of DISPLAY GROUPBUFFERPOOL Command. Make sure the SNAPSHOT value (**C**) does not rise significantly above (**B** × **A**).*

End of General-use Programming Interface

General-use Programming Interface

Too Few Directory Entries

When existing directory entries are being reclaimed to handle new work, cross-invalidation must occur for all the DB2 subsystems that have the particular data pages in their buffer pools, even when the data hasn't actually changed.

For example, in Figure 76 on page 260, **F** indicates that there have been 18281 directory reclaims. **G** shows that, because of those reclaims, 4489 cross-invalidations occurred. The pages in those members' buffer pools need to be refreshed when next needed, probably from DASD, which can degrade performance of the system.

If there is a high value in **G**, check the group buffer pool hit percentage (described in “Group Buffer Pool Size is Too Small” on page 259) to see if the lack of directory entries might be causing an excessive number of reads from DASD.

You can also check the “SYNCHRONOUS READS DUE TO BUFFER INVALIDATION” counters shown in the member detail report.

To increase the number of directory entries in the group buffer pool, you can do one of the following:

- Increase the total size of the group buffer pool, as described in “Changing the Size of the Group Buffer Pool” on page 266.
- Use the ALTER GROUPBUFFERPOOL command to adjust the ratio in favor of directory entries, as described in “Changing the Ratio of Directory to Data Entries” on page 268.

End of General-use Programming Interface

General-use Programming Interface

Too Few Data Entries

If a group buffer pool does not have enough data entries, then castout to DASD occurs more frequently. You can see the number of pages cast out by using the GDETAIL option of the DISPLAY GROUPBUFFERPOOL command.

A more serious data entry shortage is indicated by the field denoted by **E** in the DISPLAY GROUPBUFFERPOOL GDETAIL report shown in Figure 76 on page 260. A value in this field indicates that the data page resources of the coupling facility are being consumed faster than the DB2 castout processes can free them.

To increase the number of data entries in the group buffer pool, you can do one of the following:

- Increase the total size of the group buffer pool, as described in “Changing the Size of the Group Buffer Pool” on page 266.
- Use the ALTER GROUPBUFFERPOOL command to adjust the ratio in favor of data entries, as described in “Changing the Ratio of Directory to Data Entries” on page 268.

End of General-use Programming Interface

What to Look for in a DB2 PM Statistics Report

See the DB2 PM statistics detail report shown in Figure 78 on page 263. We'll use fields from that report to explain some of the activity that takes place for cross-invalidation and refresh of buffers. You'll be looking at much of the same type of information described in the DISPLAY GROUPBUFFERPOOL output described earlier in this chapter.

```
GROUP BP4                      QUANTITY  /MINUTE  /THREAD  /COMMIT
---------------------------    --------  -------  -------  -------
SYN.READS(XI)-DATA RETURNED A 3032.00   302.92   606.40     0.21
SYN.READS(XI)-NO DATA RETURN  B 0.00      0.00     0.00     0.00
SYN.READS(NF)-DATA RETURNED C  25.00      2.50     5.00     0.00
 D
SYN.READS(NF)-NO DATA RETURN    19.00      1.90     3.80     0.00
UNREGISTER PAGE  T              36.00      3.60     7.20     0.00
 E

 F
CLEAN PAGES SYNC.WRITTEN         0.00      0.00     0.00     0.00
CHANGED PAGES SYNC.WRITTEN    5868.00    586.26  1173.60     0.40
 G
CLEAN PAGES ASYNC.WRITTEN        0.00      0.00     0.00     0.00
CHANGED PAGES ASYNC.WRITTEN      0.00      0.00     0.00     0.00

ASYNC.READS-DATA RETURNED        0.00      0.00     0.00     0.00
ASYNC.READS-NO DATA RETURNED     0.00      0.00     0.00     0.00
 H
REG.PAGE LIST (RPL) REQUEST     25.00      2.50     5.00     0.00
CLEAN PAGES READ AFTER RPL      37.00      3.70     7.40     0.00
CHANGED PGS READ AFTER RPL      59.00      5.90    11.80     0.00

PAGES CASTOUT  I              1038.00    103.70   207.60     0.07
UNLOCK CASTOUT     J           259.50     25.95    51.90     0.02
READ CASTOUT CLASS  K           42.00      4.20     8.40     0.00
READ CASTOUT STATISTICS  L       0.00      0.00     0.00     0.00

READ DIRECTORY INFO  M           2.00      0.20     0.40     0.00
  N
READ STORAGE STATISTICS  V       1.00      0.10     0.20     0.00
REGISTER PAGE  S                80.00      8.00    16.00     0.01
DELETE NAME  U                   3.00      0.30     0.60     0.00

CASTOUT CLASS THRESHOLD         10.00      1.00     2.00     0.00
GROUP BP CASTOUT THRESHOLD       0.00      0.00     0.00     0.00
GBP CHECKPOINTS TRIGGERED O      1.00      0.10     0.20     0.00
PARTICIPATION IN GBP REBUILD P   0.00      0.00     0.00     0.00
 Q

CASTOUT ENGINE NOT AVAIL.        0.00      0.00     0.00     0.00
WRITE ENGINE NOT AVAILABLE       0.00      0.00     0.00     0.00
 R
READ FAILED-NO STORAGE           0.00      0.00     0.00     0.00
WRITE FAILED-NO STORAGE          0.00      0.00     0.00     0.00
WRITE TO SGBP                    0.00      0.00     0.00     0.00 WRITE TO
SGBP FAILED  W          0.00     0.00      0.00     0.00
DELETE NAME LIST SGBP            0.00      0.00     0.00     0.00
DELETE NAME FROM SGBP  X         0.00      0.00     0.00     0.00
READ CASTOUT STATS SGBP          0.00      0.00     0.00     0.00
```

Figure 78. Portion of DB2 PM Statistics Detail Report Showing GBP Activity

Explanation of Fields

A The number of reads to the group buffer pool that were required because the page was invalidated in the member's buffer pool. The member did find the needed data in the group buffer pool.

B The number of reads to the group buffer pool that were required because the page was invalidated in the member's buffer pool. The member did not find the needed data in the group buffer pool and had to go to DASD to retrieve the page.

C The number of reads to the group buffer pool that were required because the page was not in the member's buffer pool. The member did find the needed data in the group buffer pool.

D The number of reads to the group buffer pool that were required because the page was not in the member's buffer pool. The member did not find the needed data in the group buffer pool and had to go to DASD to find the page.

E This section of counters show asynchronous reads from the group buffer pool. DB2 uses the term *asynchronous* to mean that the request was done under a system execution unit, asynchronous to the allied work unit.

Asynchronous reads can occur for prefetch processing, and for those cases when P-lock negotiation necessitates registering pages cached in the member's buffer pool.

F These counters indicate the number of changed and clean pages that were synchronously written to the group buffer pool from the virtual pool.

G These counters indicate the number of changed and clean pages that were asynchronously written to the group buffer pool from the virtual pool. Pages can be forced out before the application commits if a buffer pool threshold is reached, or when P-lock negotiation forces the pages on the vertical deferred write queue to be written to the group buffer pool.

H This section of counters reflect the DB2's use of the ability to register a list of pages for prefetch activity. See “Prefetch Processing” on page 245 for more information about this.

I The number of data pages that were cast out of the group buffer pool.

J The number of unlock-for-castout requests was issued for castout I/Os that have completed. When pages are in the process of being cast out to DASD, they are “locked for castout” in the coupling facility. This castout lock is not an IRLM lock; its purpose is to enforce that only one system at a time is casting out a particular page.

DB2 usually includes more than one page in the request to write pages to DASD. Therefore, this counter should be less-than or equal-to the value in **I**.

K The number of requests made to the group buffer pool to determine which pages that belong to a particular page set or partition must be cast out because they are cached as changed pages. This request is issued by the page set or partition castout owner and, when the group buffer pool castout threshold is reached, by the group buffer pool structure owner.

L The number of requests issued by the group buffer pool structure owner to determine which castout classes have pages that are changed. This request is issued by the group buffer pool structure owner when the group buffer pool threshold is reached. The request is generally issued only once or twice for each occurrence of the group buffer pool threshold.

This request is also issued for group buffer pool checkpoint when your Sysplex is enabled to take advantage of the group buffer pool checkpoint enhancement.

M The number of requests issued by the group buffer pool structure owner to read the directory entries of all changed pages in the group buffer pool. This request is issued at group buffer pool checkpoints. The purpose of the request is to record the oldest recovery LRSN, which is used as a basis for recovery if the group buffer pool fails. See "Tuning the Group Buffer Pool Checkpoint Interval" on page 250 for more information.

N These counters indicate how many times the castout class and group buffer pool castout thresholds have been reached.

O The number of group buffer pool checkpoints that occurred.

P The number of times this member participated in rebuilding this group buffer pool.

Q These counters indicate a problem obtaining the capability to write to the group buffer pool or to cast out to DASD. These numbers should be very low, ideally 0.

R These counters indicate storage problems in the coupling facility. You might need to increase the size of the group buffer pool if these numbers are large and are not reflecting just a momentary surge in activity.

S The number of times DB2 registered interest in a single page. These are "register-only" requests, which means that DB2 is not requesting any data back from the request. This request is made only to create a directory entry for the page to be used for cross-invalidation when the page set or partition P-lock is downgraded from S to IS mode, or from SIX to IX mode.

T The number of times DB2 unregistered interest for a single page. This happens when DB2 steals pages from the local buffer pool that belong to GBP-dependent page sets or partitions.

U The number of times DB2 issued a request to delete directory and data entries associated with a particular page set or partition. DB2 issues this request when it converts a page set or partition from GBP-dependent to non-GBP-dependent. DB2 also issues this request for objects that are defined with GBPCACHE ALL when those objects are first opened.

V The number of times DB2 requested statistics information from the group buffer pool. It is issued on a timer interval basis by the group buffer pool structure owner to detect when the GBPOOLT castout threshold is reached. It is also issued whenever the DISPLAY GROUPBUFFERPOOL GDETAIL command is issued or when IFCID 0254 is recorded.

W The number of times a write to the secondary group buffer pool failed because of lack of storage.

X The number of times a "delete name list" request was issued for the secondary group buffer pool. The delete name list request is used to clean out the secondary group buffer pool after the primary group buffer pool has been cast out.

Changing Group Buffer Pools

The information under this heading, up to "Access Path Selection in a Data Sharing Group" on page 268, is General-use Programming Interface and Associated Guidance Information as defined in "Notices" on page v.

This section describes how you can change attributes of the group buffer pool. The following tasks are described:

- "Changing the Castout Threshold Values" on page 266
- "Changing the Checkpoint Frequency" on page 266
- "Changing the Size of the Group Buffer Pool" on page 266
- "Changing the Ratio of Directory to Data Entries" on page 268

If you want to start or stop duplexing for a group buffer pool, see “Starting and stopping duplexing for a group buffer pool” on page 189. If you want to make hardware changes to the coupling facility or move a group buffer pool from one coupling facility to another, see “Shutting down the Coupling Facility” on page 190.

Changing the Castout Threshold Values

Use ALTER GROUPBUFFERPOOL to change the group buffer pool castout thresholds. For example, the following command:

```
-DB1G ALTER GROUPBUFFERPOOL(GBP1) CLASST(15) GBPOOLT(55)
```

changes the class castout threshold to 15 percent and the group buffer pool threshold to 55 percent. These changes take effect immediately.

Changing the Checkpoint Frequency

Use ALTER GROUPBUFFERPOOL to change the group buffer pool castout thresholds. For example, to indicate that you want group buffer pool checkpoints to occur every 3 minutes, enter the following command:

```
-DB1G ALTER GROUPBUFFERPOOL(GBP1) GBPCHKPT(3)
```

This change takes effect immediately.

Changing the Size of the Group Buffer Pool

You can use two methods to change the size of the group buffer pool. The method you choose depends on the level of MVS that you have and whether the group buffer pool is already allocated at the maximum size. For a duplexed group buffer pool, you are changing the size of both the primary and secondary structure with a single command.

Dynamic Method: If all of the following conditions are true:

- All members of the group are running with MVS Version 5 Release 2 or subsequent releases.
- The group buffer pool is allocated in a coupling facility with CFLEVEL greater than zero.
- The currently allocated size of the structure is less than the maximum size as defined in the SIZE parameter of the CFRM policy.

Then you can enter the following command (this example assumes the group name is DSNDB0G):

```
SETXCF START,ALTER,STRNAME=DSNDB0G_GBPn,SIZE=newsize
```

This example assumes that *newsize* is less than or equal to the maximum size defined the CFRM policy for the group buffer pool.

If the maximum size (SIZE in the CFRM policy) is still not big enough, you must use the method described in “Static Method” on page 267.

Assume a DISPLAY GROUPBUFFERPOOL command shows the following:

```
:
DSNB757I -DB1G MVS CFRM POLICY STATUS FOR DSNDB0G_GBP1 = NORMAL
             MAX SIZE INDICATED IN POLICY            = 4096 KB
             ALLOCATED                               = YES
DSNB758I -DB1G ALLOCATED SIZE                        = 1024 KB
               VOLATILITY STATUS                     = VOLATILE
               REBUILD STATUS                        = NONE
               DUPLEXING STATUS                      = SIMPLEXED
               CFNAME, CFLEVEL                       = LF01, 5
DSNB759I -DB1G NUMBER OF DIRECTORY ENTRIES           = 924
               NUMBER OF DATA PAGES                  = 180
               NUMBER OF CONNECTIONS                 = 2
:
```

And then you enter the following MVS command to increase the size:

```
SETXCF START,ALTER,STRNM=DSNDB0G_GBP1,SIZE=1536
```

Here's what the DISPLAY GROUPBUFFERPOOL command output might look like after you alter the size:

```
:
DSNB757I -DB1G MVS CFRM POLICY STATUS FOR DSNDB0G_GBP1 = NORMAL
             MAX SIZE INDICATED IN POLICY            = 4096 KB
             ALLOCATED                               = YES
DSNB758I -DB1G ALLOCATED SIZE                        = 1536 KB
               VOLATILITY STATUS                     = VOLATILE
               REBUILD STATUS                        = NONE
               DUPLEXING STATUS                      = SIMPLEXED
               CFNAME, CFLEVEL                       = LF01, 5
DSNB759I -DB1G NUMBER OF DIRECTORY ENTRIES           = 1426
               NUMBER OF DATA PAGES                  = 284
               NUMBER OF CONNECTIONS                 = 2
:
```

Notice that the allocated size has increased and the numbers of directory entries and data pages have increased as well. The existing ratio is maintained.

Static Method: If *any* of the following conditions are true:

- A member of the group does not run with MVS Version 5 Release 2 or a subsequent release.
- The group buffer pool is allocated in a coupling facility at CFLEVEL=0.
- The allocated size of the structure is already at the maximum size defined by the SIZE parameter of the CFRM policy.

Then you must use the following procedure. Because the group buffer pool must be rebuilt, use this procedure when there is less activity in the group.

1. Increase the storage for the group buffer pool in the CFRM policy.
2. Use the following MVS command to start the updated policy:

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

3. Use the following command to rebuild the group buffer pool:

   `SETXCF START,REBUILD,STRNAME=`*`strname`*

Changing the Ratio of Directory to Data Entries

To change the ratio of directory to data entries, you must use the ALTER GROUPBUFFERPOOL command. For example, if the current ratio is 5 (that is, there are 5 directory entries to every one data page), you can use the ALTER GROUPBUFFERPOOL command shown here to increase the ratio to 7:

```
-DB1G ALTER GROUPBUFFERPOOL (GBP0) RATIO (7)
```

For the change to take effect, you must rebuild the group buffer pool by using the SETXCF START, REBUILD command.

For duplexed group buffer pools, you must stop duplexing before you can rebuild the group buffer pool. The procedure, then, is this:

1. Stop duplexing, as described in “Stopping duplexing” on page 190 to revert the group buffer pool to simplex mode.
2. Alter the group buffer pool ratio and issue the SETXCF START,REBUILD command.
3. Start duplexing, as described in “Starting duplexing” on page 189.

Access Path Selection in a Data Sharing Group

This section describes the following:

Effect of Member Configuration on Access Path Selection

Because plans and packages are bound on individual members in the group, the way a member is configured influences the access path chosen for statements in that plan or package. For example, it is possible to have different buffer pool sizes and different RID (record identifier) pool sizes on each member. It is also possible that members are on different CPC models.

When you bind your application from one of the members, DB2 chooses the best access path, given the catalog statistics, CPC model, buffer pool sizes, among other things. Suppose, though, that the selected access path is optimal for the one member, but is a relatively poor choice for a different member in the same group. Because the group shares the catalog and directory, the same plan (and hence the same access paths) are used regardless of member, after the application is bound.

Where to Bind in a Mixed Data Sharing Configuration: If your data sharing group consists of mixed CPC models, be aware that the speed of a central processor (CP) might change your access path. This effect is more likely with long-running queries than with fast-running transactions.

Automatic Rebind: The access path can change if automatic rebind occurs while the application is executing on a different member than the one the original bind occurred on.

Using EXPLAIN in a Data Sharing Group

Product-sensitive Programming Interface

EXPLAIN informs you about access paths DB2 chooses. Because EXPLAIN can be run on one DB2 subsystem and a plan can be bound and executed on other DB2 subsystems in a data sharing group, it is important to know which member performed the EXPLAIN. The PLAN_TABLE column GROUP_MEMBER contains the member name of the DB2 that performed the EXPLAIN. The column is blank if the DB2 subsystem was not in a data sharing environment when the EXPLAIN was performed.

End of Product-sensitive Programming Interface

Appendix A. DB2 and IRLM Names

This appendix includes information about name formats for the licensed program DB2 for OS/390 and its IRLM.

DB2 Group Names

Table 36. Group Names. There is one set of names per group.

Name	Length	Example Format	Comments
Catalog alias	8	*catalias*	You must place DB2's catalog alias in the MVS master catalog.
Catalog and directory database names		*catalias*.DSNDB01 *catalias*.DSNDB06	All DB2s in the group share the same catalog and directory.
DSNHDECP	8	DSNHDECP	This resides in SDSNEXIT. There is a single load module for the group.
Generic LU name	8	*xxxxxxxx*	One name per group. Requesters use this name to configure their communications control information.
Group attachment name	4	*gssn*	This name can be used by TSO/batch, CAF, RRSAF, and utilities as a generic attachment name.
Group buffer pools (coupling facility cache structures)	16	*groupname*_GBP*xxxx*	You must enter this name on the CFRM policy.
Group name	8	*xxxxxxxx*	This name must be unique within the Sysplex.
location name	16	*xxxxxxxxxxxxxxxx*	One name per group. Requesters use this name in their SQL applications.
lock structure	16	*groupname*_LOCK1	You must enter this name on the CFRM policy.
shared communications area (coupling facility list structure)	16	*groupname*_SCA	You must enter this name on the CFRM policy.
Target libraries		*catalias*.SDSNCLST *catalias*.SDSNLINK *catalias*.SDSNLOAD *catalias*.SDSNEXIT	Target libraries can be shared among DB2 group members or can be replicated.

DB2 Member Names

Member names include the individual DB2 member (subsystem) name and its associated MVS subsystem name, procedure names, and BSDS and log names.

Table 37. Member Names. There is One Set of Names per Member.

Name	Length	Example Format	Comments
Active log prefixes	30	*membname*.LOGCOPY1 *membname*.LOGCOPY2	The catalog alias can be added as first qualifier.
Archive log prefixes	35 or 19	*membname*.ARCLG1 *membname*.ARCLG2	The catalog alias can be added as the first qualifier. There are only 19 characters for the prefix if the name is timestamped.
BSDS	33	*membname*.BSDS01 *membname*.BSDS02	The catalog alias can be added as the first qualifier.
SCA and group buffer pool connection names	16	DB2_*membername*	The connection name is generated by DB2. You see it only in certain commands (MVS D XCF,STRUCTURE and the DB2 connection list display of DISPLAY GROUPBUFFERPOOL).
Command prefix	8	*xmembcpf*	The first character must be a special character.
LU name	8	*luname*	This name must be unique within the data sharing group and the network.
Member name	8	*membname*	This name must be unique within the data sharing group. If the member name is also used as the high level qualifier for the member's data sets (BSDS, logs, and so on), then member names must be unique within the MVS Sysplex. This is because the MVS Sysplex can have a shared master catalog.
Procedure names	8	*mssn*MSTR *mssn*DBM1 *mssn*DIST *mssn*SPAS	These names are generated from the member subsystem name at installation time.
Subsystem name	4	*mssn*	This name is used by all the attachment interfaces and must be unique within the MVS Sysplex.
Subsystem parameters load module	8	DSNZP*xxx*	Resides in SDSNEXIT. The name is specified as a parameter on *mssn*MSTR procedure.
XCF member name	16	*membname*	This is the same as DB2's member name; thus, a maximum of 8 characters is used.
Work file database	8	*mworkdb*	Work file data sets have names of the format *catalias*.DSNDBC.*mworkdb*.DSN*kknn*.I0001.A001

IRLM Names

Each DB2 subsystem in the data sharing group has an associated IRLM. Each member's IRLM and its associated procedures must be named. The IRLM group name, subsystem name, and member ID are parameters on the IRLM proc. This requires a separate IRLM proc for every IRLM in the group.

Table 38. IRLM Names. There is one set of names per DB2 member.

Name	Length	Example Format	Comments
IRLM group name	8	*xxxxxxxx*	This is a parameter on *issn*IRLM proc. It must be unique across the Sysplex, to avoid name conflicts.
IRLM subsystem name	4	*issn*	This is a parameter on the *issn* IRLM procedure. This subsystem name must be unique within the data sharing group to avoid the problem of DB2 connecting to the wrong IRLM.
IRLM procedure name	8	*mssn*IRLM	Each DB2 member knows its IRLM by the procedure and subsystem name saved in that member's subsystem parameter load module.
IRLM member ID	3	This is a number between 1 and 255 (inclusive).	This ID uniquely names an IRLM within a group. It is a parameter on *issn*IRLM procedure and must be unique within the data sharing group.
IRLM member XCF name	16	*xxxxxxxx*$*issnNNN*	*xxxxxxxx* is the IRLM group name, *issn* is the IRLM subsystem ID, and *NNN* is the IRLM member ID. Dollar signs ($) are used as padding. This name is generated at startup time.
lock structure connection name	16	*xxxxxxxx*$*issnNNN*	The connection name is the same as the IRLM XCF member name. It is generated by IRLM. You see it only as the output of certain commands (such as MVS D XCF,STRUCTURE). In some cases, the connection name can be of the format *xxxxxxxx#issnNNN*.

Glossary

The following terms and abbreviations are defined as they are used in the DB2 library. If you do not find the term you are looking for, refer to the index or to *Dictionary of Computing*.

A

active log. The portion of the DB2 log to which log records are written as they are generated. The active log always contains the most recent log records, whereas the archive log holds those records that are older and no longer will fit on the active log.

active member state. A state of a member of a data sharing group. An active member is identified with a group by XCF, which associates the member with a particular task, address space, and MVS system. A member that is not active is failed or quiesced.

archive log. The portion of the DB2 log that contains log records that have been copied from the active log.

C

cache structure. A coupling facilitystructure that stores data that can be available to all members of a Sysplex. A DB2 data sharing group uses cache structures as group buffer pools.

castout. The DB2 process of writing changed pages from a group buffer pool to DASD.

castout owner. The DB2 member that is responsible for casting out a particular page set or partition.

CEC. See *central processor complex*.

central electronic complex (CEC). See *central processor complex*.

central processor complex (CPC). A physical collection of hardware (such as an ES/3090) that consists of main storage, one or more central processors, timers, and channels.

CFRM policy. A declaration by an MVS administrator regarding the allocation rules for a coupling facility structure.

command prefix. A one- to eight-character command identifier. The command prefix distinguishes the command as belonging to an application or subsystem rather than MVS.

command scope. The scope of command operation in a data sharing group. If a command has *member scope*, the command displays information from the one member only or affects only non-shared resources owned locally by that member. If a command has *group scope*, the command displays information from all members, affects non-shared resources owned locally by all members, displays information on sharable resources, or affects sharable resources.

coupling facility. A special PR/SM LPAR logical partition that runs the coupling facility control program and provides high-speed caching, list processing, and locking functions in a Sysplex.

CPC. See *central processor complex*.

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

cross-system extended services (XES). A set of MVS services that allow multiple instances of an application or subsystem, running on different systems in a Sysplex, to implement high-performance, high-availability data sharing by using a coupling facility.

D

database exception status. An indication that something is wrong with a database. The exceptional status of databases must be known and shared by all members of a data sharing group.

data sharing. The ability of two or more DB2 subsystems to directly access and change a single set of data.

data sharing group. A collection of one or more DB2 subsystems that directly access and change the same data while maintaining data integrity.

data sharing member. A DB2 subsystem assigned by XCF services to a data sharing group.

degree of parallelism. The number of concurrently executed operations that are initiated to process a query.

domain name. The name used by TCP/IP applications to refer to a TCP/IP host within a TCP/IP network.

domain name server (DNS). A special TCP/IP network server that manages a distributed directory that is used to map TCP/IP host names to IP addresses.

E

exclusive lock. A lock that prevents concurrently executing application processes from reading or changing data. Contrast with *shared lock*.

explicit hierarchical locking. Locking used to make the parent/child relationship between resources known to IRLM. This is done to avoid global locking overhead when no inter-DB2 interest exists on a resource.

F

failed member state. A state of a member of a data sharing group. A failed member has permanent status recording with XCF, and its task, address space, or MVS system has terminated before the state changed from active to quiesced.

false global lock contention. A contention indication from the coupling facility when multiple lock names are hashed to the same indicator and when there is no real contention.

G

GBP-dependent. A page set or page set partition status when it is dependent upon the group bufferpool. There is either inter-DB2 read/write interest active for this page set or the page set has changed pages in the group buffer pool that have not yet been castout to DASD.

generic resource name. A name used by VTAM that represents several application programs that provide the same function in order to handle session distribution and balancing in a Sysplex.

global lock. A lock that provides both intra-DB2 concurrency control and inter-DB2 concurrency control, that is, the scope of the lock is across all the DB2s of a data sharing group.

global lock contention. Conflicts on locking requests between different DB2 members of a data sharing group regarding attempts to serialize shared resources.

gross lock. The *shared*, *update*, or *exclusive* mode locks on a table, partition, or table space.

group buffer pool. A coupling facility cache structure used by a data sharing group to cache data and to ensure that the data is consistent for all members.

group buffer pool duplexing. The ability to write data to two instances of a group buffer pool structure; a *primary group buffer pool* and a *secondary group buffer pool*. OS/390 publications refer to these instances as the 'old' (for primary) and 'new' (for secondary) structures.

group name. The MVS XCF identifier for a data sharing group.

group restart. A restart of at least one member of a data sharing group after either locks or the shared communications area have been lost.

I

index partition. A VSAM data set that is contained within a partitioned index space.

inter-DB2 R/W interest. A property of data in a table space, index, or partition that has been opened by more than one member of a data sharing group and that has been opened for writing by at least one of those members.

IP address. A 4-byte value that uniquely identifies a TCP/IP host.

L

list structure. A coupling facility structure allowing data to be shared and manipulated as elements of a queue.

L-lock. See *logical lock*.

local lock. A lock that provides intra-DB2 concurrency control, but does not provide inter-DB2 concurrency control; that is, its scope is a single DB2.

lock. A means of controlling concurrent events or access to data. DB2 locking is performed by the IRLM.

lock duration. The interval over which a DB2 lock is held.

lock escalation. The promotion of a lock from a row or page lock to a table space lock because the number of page locks concurrently held on a given resource exceeds a preset limit.

locking. The process by which the integrity of data is ensured. Locking prevents concurrent users from accessing inconsistent data.

lock mode. A representation for the type of access concurrently running programs can have to a resource held by a DB2 lock.

lock object. The resource that is controlled by a DB2 lock.

lock parent. For explicit hierarchical locking, a lock held on a resource that has child locks that are lower in the hierarchy; usually the table space or partition intent locks are the parent locks.

lock promotion. The process of changing the size or mode of a DB2 lock to a higher level.

lock size. The amount of data controlled by a DB2 lock on table data; the value can be a row, a page, a table, or a table space.

lock structure. A coupling facility data structure composed of a series of lock entries to support shared and exclusive locking for logical resources.

log. A collection of records that describe the events that occur during DB2 execution and their sequence. The information thus recorded is used for recovery in the event of a failure during DB2 execution.

logical index partition. The set of all keys that reference the same data partition.

logical lock. The lock type used by transactions to control intra- and inter-DB2 data concurrency between transactions.

logically complete. The concurrent copy process is finished with the initialization of the target objects being copied. The target objects are available for update.

logical page list (LPL). A list of pages in error that cannot be referenced by applications until the pages are recovered. The page is in 'logical error' because there may be nothing wrong with the media (coupling facility or DASD) itself. Usually a connection to the media has been lost.

log initialization. The first phase of restart processing during which DB2 attempts to locate the current end of the log.

log record sequence number (LRSN). A number DB2 generates and associates with each log record. DB2 also uses the LRSN for page versioning. The LRSNs generated by a given DB2 data sharing group form a strictly increasing sequence for each DB2 log and a strictly increasing sequence for each page across the DB2 group.

log truncation. A process by which an explicit starting RBA is established. This RBA is the point at which the next byte of log data will be written.

LPL. See *logical page list*.

LRSN. See *log record sequence number*.

M

member name. The MVS XCF identifier for a particular DB2 subsystem in a data sharing group.

modify locks. An L-lock or P-lock that has been specifically requested as having the MODIFY attribute. A list of these active locks are kept at all times in the coupling facilitylock structure. If the requesting DB2 fails, that DB2's modify locks are converted to *retained locks*.

N

negotiable lock. A lock whose mode can be downgraded, by agreement among contending users, to be compatible to all. A physical lock is an example of a negotiable lock.

nonpartitioned index. Any index that is not the partitioned index of a partitioned table space.

P

page version number. A 6-byte field in a page header that is strictly increasing.

parallelism assistant. In Sysplex query parallelism, a DB2 that is helping to process parts of a parallel query that originated on another DB2 in the data sharing group.

parallelism coordinator. In Sysplex query parallelism, the DB2 from which the parallel query originated.

physical lock contention. Conflicting states of the requesters for a physical lock. See *negotiable lock*.

physical lock (P-lock). A lock type used only by data sharing that is acquired by DB2 to provide consistency of data cached in different DB2 subsystems.

physically complete. The concurrent copy process is completed and the output data set has been created.

P-lock. See *physical lock*.

policy. See *CFRM policy*.

primary group buffer pool. For a duplexed group buffer pool, the structure used to maintain the coherency of cached data; that is, the structure used for page registration and cross-invalidation. The OS/390 equivalent is 'old' structure. Compare with *secondary group buffer pool*.

Q

quiesced member state. A state of a member of a data sharing group. An active member becomes quiesced when a STOP DB2 command takes effect without a failure. If the member's task, address space, or MVS system fails before the command takes effect, the member state is failed.

R

retained lock. A MODIFY lock that was held by a DB2 when that DB2 failed. The lock is retained in the coupling facility lock structure across a DB2 failure.

S

SCA. See *shared communications area*.

secondary group buffer pool. For a duplexed group buffer pool, the structure used to back up changed pages that are written to the primary group buffer pool. No page registration or cross-invalidation occurs using the secondary group buffer pool. The OS/390 equivalent is 'new' structure.

shared communications area (SCA). A coupling facility list structure used by a DB2 data sharing group for inter-DB2 communication.

shared lock. A lock that prevents concurrently executing application processes from changing data, but not from reading data.

structure. A construct used by MVS to map and manage storage on a coupling facility. See *cache structure*, *list structure*, or *lock structure*.

structure owner. In relation to group buffer pools, this is the DB2 member that is responsible for the following activities:

- Coordinating rebuild, checkpoint, and damage assessment processing
- Monitoring the group buffer pool threshold and notifying castout owners when the threshold has been reached.

Sysplex. A set of MVS systems that communicate and cooperate with each other through certain multisystem hardware components and software services to process customer workloads.

T

TCP/IP. A network communication protocol used by computer systems to exchange information across telecommunication links.

TCP/IP port. A 2-byte value that identifies an end user or a TCP/IP network application within a TCP/IP host.

transaction lock. A lock used to control concurrent execution of SQL statements.

type 1 indexes. Indexes that were created by a release of DB2 before DB2 Version 4 or that are specified as type 1 indexes in Version 4. Contrast with *type 2 indexes*.

type 2 indexes. A new type of indexes available in Version 4. They differ from *type 1 indexes* in several respects; for example, they are the only indexes allowed on a table space that uses *row locks*.

X

XCF. See *cross-system coupling facility*.

XES. See *cross-system extended services*.

Bibliography

DB2 for OS/390 Version 5

- *Administration Guide, SC26-8957*
- *Application Programming and SQL Guide, SC26-8958*
- *Call Level Interface Guide and Reference, SC26-8959*
- *Command Reference, SC26-8960*
- *Data Sharing: Planning and Administration, SC26-8961*
- *Data Sharing Quick Reference Card, SX26-3841*
- *Diagnosis Guide and Reference, LY27-9659*
- *Diagnostic Quick Reference Card, LY27-9660*
- *Installation Guide, GC26-8970*
- *Application Programming Guide and Reference for Java™, SC26-9547*
- *Licensed Program Specifications, GC26-8969*
- *Messages and Codes, GC26-8979*
- *Reference for Remote DRDA Requesters and Servers, SC26-8964*
- *Reference Summary, SX26-3842*
- *Release Guide, SC26-8965*
- *SQL Reference, SC26-8966*
- *Utility Guide and Reference, SC26-8967*
- *What's New?, GC26-8971*
- *Program Directory*

DB2 PM for OS/390 Version 5

- *Batch User's Guide, SC26-8991*
- *Command Reference, SC26-8987*
- *General Information, GC26-8982*
- *Getting Started on the Workstation, SC26-8989*
- *Master Index, SC26-8984*
- *Messages Manual, SC26-8988*
- *Online Monitor User's Guide, SC26-8990*
- *Report Reference Volume 1, SC26-8985*
- *Report Reference Volume 2, SC26-8986*
- *Program Directory*

Ada/370

- *IBM Ada/370 Language Reference, SC09-1297*
- *IBM Ada/370 Programmer's Guide, SC09-1414*
- *IBM Ada/370 SQL Module Processor for DB2 Database Manager User's Guide, SC09-1450*

APL2

- *APL2 Programming Guide, SH21-1072*
- *APL2 Programming: Language Reference, SH21-1061*
- *APL2 Programming: Using Structured Query Language (SQL), SH21-1057*

AS/400

- *DB2 for OS/400 SQL Programming, SC41-4611*
- *DB2 for OS/400 SQL Reference, SC41-4612*

BASIC

- *IBM BASIC/MVS Language Reference, GC26-4026*
- *IBM BASIC/MVS Programming Guide, SC26-4027*

C/370

- *IBM SAA AD/Cycle C/370 Programming Guide, SC09-1356*
- *IBM SAA AD/Cycle C/370 Programming Guide for Language Environment/370, SC09-1840*
- *IBM SAA AD/Cycle C/370 User's Guide, SC09-1763*
- *SAA CPI C Reference, SC09-1308*

Character Data Representation Architecture

- *Character Data Representation Architecture*
Overview, GC09-2207
- *Character Data Representation Architecture*
Reference, SC09-2190

CICS/ESA

- *CICS/ESA Application Programming Guide, SC33-1169*
- *CICS/ESA Application Programming Reference, SC33-1170*
- *CICS/ESA CICS - RACF Security Guide, SC33-1185*
- *CICS/ESA CICS-Supplied Transactions, SC33-1168*
- *CICS/ESA Customization Guide, SC33-1165*
- *CICS/ESA Data Areas, LY33-6083*
- *CICS/ESA Installation Guide, SC33-1163*
- *CICS/ESA Intercommunication Guide, SC33-1181*
- *CICS/ESA Messages and Codes, SC33-1177*
- *CICS/ESA Operations and Utilities Guide, SC33-1167*
- *CICS/ESA Performance Guide, SC33-1183*
- *CICS/ESA Problem Determination Guide, SC33-1176*
- *CICS/ESA Resource Definition Guide, SC33-1166*
- *CICS/ESA System Definition Guide, SC33-1164*
- *CICS/ESA System Programming Reference, GC33-1171*

CICS/MVS

- *CICS/MVS Application Programming Primer, SC33-0139*
- *CICS/MVS Application Programmer's Reference, SC33-0512*
- *CICS/MVS Facilities and Planning Guide, SC33-0504*
- *CICS/MVS Installation Guide, SC33-0506*
- *CICS/MVS Operations Guide, SC33-0510*
- *CICS/MVS Problem Determination Guide, SC33-0516*
- *CICS/MVS Resource Definition (Macro), SC33-0509*
- *CICS/MVS Resource Definition (Online), SC33-0508*

IBM C/C++ for MVS/ESA or OS/390

- *IBM C/C++ for MVS/ESA Library Reference, SC09-1995*
- *IBM C/C++ for MVS/ESA Programming Guide, SC09-1994*
- *IBM C/C++ for OS/390 User's Guide, SC09-2361*

IBM COBOL for MVS & VM

- *IBM COBOL for MVS & VM Language Reference, SC26-4769*
- *IBM COBOL for MVS & VM Programming Guide, SC26-4767*

Conversion Guides

- *DBMS Conversion Guide: DATACOM/DB to DB2, GH20-7564*
- *DBMS Conversion Guide: IDMS to DB2, GH20-7562*
- *DBMS Conversion Guide: Model 204 to DB2 or SQL/DS, GH20-7565*
- *DBMS Conversion Guide: VSAM to DB2, GH20-7566*
- *IMS-DB and DB2 Migration and Coexistence Guide, GH21-1083*

Cooperative Development Environment

- *CoOperative Development Environment/370: Debug Tool, SC09-1623*

DATABASE 2 for Common Servers

- *DATABASE 2 Administration Guide for common servers, S20H-4580*
- *DATABASE 2 Application Programming Guide for common servers, S20H-4643*
- *DATABASE 2 Software Developer's Kit for AIX: Building Your Applications, S20H-4780*
- *DATABASE 2 Software Developer's Kit for OS/2: Building Your Applications, S20H-4787*
- *DATABASE 2 SQL Reference for common servers, S20H-4665*
- *DATABASE 2 Call Level Interface Guide and Reference for common servers, S20H-4644*

Data Extract (DXT)

- *Data Extract Version 2: General Information, GC26-4666*
- *Data Extract Version 2: Planning and Administration Guide, SC26-4631*

DataPropagator NonRelational

- *DataPropagator NonRelational MVS/ESA Administration Guide, SH19-5036*
- *DataPropagator NonRelational MVS/ESA Reference, SH19-5039*

DataPropagator Relational

- *DataPropagator Relational User's Guide, SC26-3399*
- *IBM An Introduction to DataPropagator Relational, GC26-3398*

Data Facility Data Set Services

- *Data Facility Data Set Services: User's Guide and Reference, SC26-4125*

Database Design

- *DB2 Database Design and Implementation Using DB2, SH24-6101*
- *DB2 Design and Development Guide, Gabrielle Wiorkowski and David Kull, Addison Wesley*
- *Handbook of Relational Database Design, C. Fleming and B Von Halle, Addison Wesley*
- *Principles of Database Systems, Jeffrey D. Ullman, Computer Science Press*

DataHub

- *IBM DataHub General Information, GC26-4874*

DB2 Universal Database

- *DB2 Universal Database Administration Guide, S10J-8157*
- *DB2 Universal Database API Reference, S10J-8167*
- *DB2 Universal Database Building Applications for UNIX Environments, S10J-8161*
- *DB2 Universal Database Building Applications for Windows and OS/2 Environments, S10J-8160*
- *DB2 Universal Database CLI Guide and Reference, S10J-8159*
- *DB2 Universal Database SQL Reference, S10J-8165*

Device Support Facilities

- *Device Support Facilities User's Guide and Reference, GC35-0033*

DFSMS/MVS

- *DFSMS/MVS: Access Method Services for the Integrated Catalog, SC26-4906*

- *DFSMS/MVS: Access Method Services for VSAM Catalogs, SC26-4905*
- *DFSMS/MVS: Administration Reference for DFSMSdss, SC26-4929*
- *DFSMS/MVS: DFSMShsm Managing Your Own Data, SH21-1077*
- *DFSMS/MVS: Diagnosis Reference for DFSMSdfp, LY27-9606*
- *DFSMS/MVS: Macro Instructions for Data Sets, SC26-4913*
- *DFSMS/MVS: Managing Catalogs, SC26-4914*
- *DFSMS/MVS: Program Management, SC26-4916*
- *DFSMS/MVS: Storage Administration Reference for DFSMSdfp, SC26-4920*
- *DFSMS/MVS: Using Advanced Services for Data Sets, SC26-4921*
- *DFSMS/MVS: Utilities, SC26-4926*
- *MVS/DFP: Managing Non-VSAM Data Sets, SC26-4557*

DFSORT

- *DFSORT Application Programming: Guide, SC33-4035*

Distributed Relational Database

- *Data Stream and OPA Reference, SC31-6806*
- *Distributed Relational Database Architecture: Application Programming Guide, SC26-4773*
- *Distributed Relational Database Architecture: Connectivity Guide, SC26-4783*
- *Distributed Relational Database Architecture: Evaluation and Planning Guide, SC26-4650*
- *Distributed Relational Database Architecture: Problem Determination Guide, SC26-4782*
- *Distributed Relational Database: Every Manager's Guide, GC26-3195*
- *IBM SQL Reference, SC26-8416*
- *Open Group Technical Standard (the Open Group presently makes the following books available through their website at www.opengroup.org):*
  - *DRDA Volume 1: Distributed Relational Database Architecture (DRDA), ISBN 1-85912-295-7*
  - *DRDA Volume 3: Distributed Database Management (DDM) Architecture, ISBN 1-85912-206-X*

Education

- *Dictionary of Computing, SC20-1699*
- *IBM Enterprise Systems Training Solutions Catalog, GR28-5467*

Enterprise System/9000 and Enterprise System/3090

- *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide, GA22-7123*

FORTRAN

- *VS FORTRAN Version 2: Language and Library Reference, SC26-4221*
- *VS FORTRAN Version 2: Programming Guide for CMS and MVS, SC26-4222*

High Level Assembler

- *High Level Assembler/MVS and VM and VSE Language Reference, SC26-4940*
- *High Level Assembler/MVS and VM and VSE Programmer's Guide, SC26-4941*

Parallel Sysplex Library

- *System/390 MVS Sysplex Application Migration, GC28-1211*
- *System/390 MVS Sysplex Hardware and Software Migration, GC28-1210*
- *System/390 MVS Sysplex Overview: An Introduction to Data Sharing and Parallelism, GC28-1208*
- *System/390 MVS Sysplex Systems Management, GC28-1209*
- *System/390 MVS 9672/9674 System Overview, GA22-7148*

ICSF/MVS

- *ICSF/MVS General Information, GC23-0093*

IMS/ESA

- *IMS Batch Terminal Simulator General Information, GH20-5522*
- *IMS/ESA Administration Guide: System, SC26-8013*
- *IMS/ESA Application Programming: Database Manager, SC26-8727*
- *IMS/ESA Application Programming: Design Guide, SC26-8016*
- *IMS/ESA Application Programming: Transaction Manager, SC26-8729*
- *IMS/ESA Customization Guide, SC26-8020*
- *IMS/ESA Installation Volume 1: Installation and Verification, SC26-8023*
- *IMS/ESA Installation Volume 2: System Definition and Tailoring, SC26-8024*
- *IMS/ESA Messages and Codes, SC26-8028*
- *IMS/ESA Operator's Reference, SC26-8030*
- *IMS/ESA Utilities Reference: System, SC26-8035*

ISPF

- *ISPF Version 4 Messages and Codes, SC34-4450*
- *ISPF Version 4 for MVS Dialog Management Guide, SC34-4213*
- *ISPF/PDF Version 4 for MVS Guide and Reference, SC34-4258*
- *ISPF and ISPF/PDF Version 4 for MVS Planning and Customization, SC34-4134*

Language Environment for MVS & VM

- *Language Environment for MVS & VM Concepts Guide, GC26-4786*
- *Language Environment for MVS & VM Debugging and Run-Time Messages Guide, SC26-4829*
- *Language Environment for MVS & VM Installation and Customization, SC26-4817*
- *Language Environment for MVS & VM Programming Guide, SC26-4818*
- *Language Environment for MVS & VM Programming Reference, SC26-3312*

MVS/ESA

- *MVS/ESA Analyzing Resource Measurement Facility Monitor I and Monitor II Reference and User's Guide, LY28-1007*
- *MVS/ESA Analyzing Resource Measurement Facility Monitor III Reference and User's Guide, LY28-1008*
- *MVS/ESA Application Development Reference: Assembler Callable Services for OpenEdition MVS, SC23-3020*
- *MVS/ESA Data Administration: Utilities, SC26-4516*
- *MVS/ESA Diagnosis: Procedures, LY28-1844*
- *MVS/ESA Diagnosis: Tools and Service Aids, LY28-1845*
- *MVS/ESA Initialization and Tuning Guide, SC28-1451*
- *MVS/ESA Initialization and Tuning Reference, SC28-1452*
- *MVS/ESA Installation Exits, SC28-1459*
- *MVS/ESA JCL Reference, GC28-1479*
- *MVS/ESA JCL User's Guide, GC28-1473*
- *MVS/ESA JES2 Initialization and Tuning Guide, SC28-1453*
- *MVS/ESA MVS Configuration Program, GC28-1615*
- *MVS/ESA Planning: Global Resource Serialization, GC28-1450*
- *MVS/ESA Planning: Operations, GC28-1441*
- *MVS/ESA Planning: Workload Management, GC28-1493*
- *MVS/ESA Programming: Assembler Services Guide, GC28-1466*
- *MVS/ESA Programming: Assembler Services Reference, GC28-1474*
- *MVS/ESA Programming: Authorized Assembler Services Guide, GC28-1467*
- *MVS/ESA Programming: Authorized Assembler Services Reference, Volumes 1-4, GC28-1475, GC28-1476, GC28-1477, GC28-1478*
- *MVS/ESA Programming: Extended Addressability Guide, GC28-1468*
- *MVS/ESA Programming: Sysplex Services Guide, GC28-1495*
- *MVS/ESA Programming: Sysplex Services Reference, GC28-1496*
- *MVS/ESA Programming: Workload Management Services, GC28-1494*
- *MVS/ESA Routing and Descriptor Codes, GC28-1487*
- *MVS/ESA Setting Up a Sysplex, GC28-1449*
- *MVS/ESA SPL: Application Development Guide, GC28-1852*
- *MVS/ESA System Codes, GC28-1486*
- *MVS/ESA System Commands, GC28-1442*
- *MVS/ESA System Management Facilities (SMF), GC28-1457*
- *MVS/ESA System Messages Volume 1, GC28-1480*
- *MVS/ESA System Messages Volume 2, GC28-1481*
- *MVS/ESA System Messages Volume 3, GC28-1482*
- *MVS/ESA Using the Subsystem Interface, SC28-1502*

Net.Data for OS/390

- *Net.Data Language Environment Guide,*
http://www.ibm.com/software/net.data/docs
- *Net.Data Programming Guide,*
http://www.ibm.com/software/net.data/docs
- *Net.Data Reference Guide,*
http://www.ibm.com/software/net.data/docs

NetView

- *NetView Installation and Administration Guide, SC31-8043*
- *NetView User's Guide, SC31-8056*

ODBC

- *ODBC 2.0 Programmer's Reference and SDK Guide, ISBN 1-55615-658-8*
- *Inside ODBC, ISBN 1-55615-815-7*

OS/390

- *OS/390 C/C++ Programming Guide, SC09-2362*
- *OS/390 C/C++ Run-Time Library Reference, SC28-1663*
- *OS/390 Information Roadmap, GC28-1727*
- *OS/390 Introduction and Release Guide, GC28-1725*
- *OS/390 JES2 Initialization and Tuning Guide, SC28-1791*
- *OS/390 JES3 Initialization and Tuning Guide, SC28-1802*
- *OS/390 Language Environment for OS/390 & VM Concepts Guide, GC28-1945*
- *OS/390 Language Environment for OS/390 & VM Customization, SC28-1941*
- *OS/390 Language Environment for OS/390 & VM Debugging Guide, SC28-1942*
- *OS/390 Language Environment for OS/390 & VM Programming Guide, SC28-1939*
- *OS/390 Language Environment for OS/390 & VM Programming Reference, SC28-1940*
- *OS/390 MVS Diagnosis: Procedures, LY28-1082*
- *OS/390 MVS Diagnosis: Tools and Service Aids, LY28-1085*

- *OS/390 MVS Initialization and Tuning Guide, SC28-1751*
- *OS/390 MVS Initialization and Tuning Reference, SC28-1752*
- *OS/390 MVS Installation Exits, SC28-1753*
- *OS/390 MVS JCL Reference, GC28-1757*
- *OS/390 MVS JCL User's Guide, GC28-1758*
- *OS/390 MVS Planning: Global Resource Serialization, GC28-1759*
- *OS/390 MVS Planning: Operations, GC28-1760*
- *OS/390 MVS Planning: Workload Management, GC28-1761*
- *OS/390 MVS Programming: Assembler Services Guide, GC28-1762*
- *OS/390 MVS Programming: Assembler Services Reference, GC28-1910*
- *OS/390 MVS Programming: Authorized Assembler Services Guide, GC28-1763*
- *OS/390 MVS Programming: Authorized Assembler Services Reference, Volumes 1-4, GC28-1764, GC28-1765, GC28-1766, GC28-1767*
- *OS/390 MVS Programming: Callable Services for High-Level Languages, GC28-1768*
- *OS/390 MVS Programming: Extended Addressability Guide, GC28-1769*
- *OS/390 MVS Programming: Sysplex Services Guide, GC28-1771*
- *OS/390 MVS Programming: Sysplex Services Reference, GC28-1772*
- *OS/390 MVS Programming: Workload Management Services, GC28-1773*
- *OS/390 MVS Routing and Descriptor Codes, GC28-1778*
- *OS/390 MVS Setting Up a Sysplex, GC28-1779*
- *OS/390 MVS System Codes, GC28-1780*
- *OS/390 MVS System Commands, GC28-1781*
- *OS/390 MVS System Messages Volume 1, GC28-1784*
- *OS/390 MVS System Messages Volume 2, GC28-1785*
- *OS/390 MVS System Messages Volume 3, GC28-1786*
- *OS/390 MVS System Messages Volume 4, GC28-1787*
- *OS/390 MVS System Messages Volume 5, GC28-1788*
- *OS/390 Security Server (RACF) Auditor's Guide, SC28-1916*
- *OS/390 Security Server (RACF) Command Language Reference, SC28-1919*
- *OS/390 Security Server (RACF) General User's Guide, SC28-1917*
- *OS/390 Security Server (RACF) Security Administrator's Guide, SC28-1915*
- *OS/390 Security Server (RACF) System Programmer's Guide, SC28-1913*
- *OS/390 SMP/E Reference, SC28-1806*
- *OS/390 SMP/E User's Guide, SC28-1740*
- *OS/390 RMF User's Guide, SC28-1949*
- *OS/390 TSO/E CLISTS, SC28-1973*
- *OS/390 TSO/E Command Reference, SC28-1969*
- *OS/390 TSO/E Customization, SC28-1965*
- *OS/390 TSO/E Messages, GC28-1978*
- *OS/390 TSO/E Programming Guide, SC28-1970*
- *OS/390 TSO/E Programming Services, SC28-1971*
- *OS/390 TSO/E REXX Reference, SC28-1975*
- *OS/390 TSO/E User's Guide, SC28-1968*

OS/390 OpenEdition

- *OS/390 OpenEdition DCE Administration Guide, SC28-1584*
- *OS/390 OpenEdition DCE Introduction, GC28-1581*
- *OS/390 R1 OE DCE Messages and Codes, ST01-0920*
- *OS/390 OpenEdition Command Reference, SC28-1892*
- *OS/390 OpenEdition Messages and Codes, SC28-1908*
- *OS/390 OpenEdition Planning, SC28-1890*
- *OS/390 OpenEdition User's Guide, SC28-1891*

PL/I for MVS & VM

- *IBM PL/I MVS & VM Language Reference, SC26-3114*
- *IBM PL/I MVS & VM Programming Guide, SC26-3113*

OS PL/I

- *OS PL/I Programming Language Reference, SC26-4308*
- *OS PL/I Programming Guide, SC26-4307*

PROLOG

- *IBM SAA AD/Cycle Prolog/MVS & VM Programmer's Guide, SH19-6892*

Query Management Facility

- *Query Management Facility: Managing QMF for MVS, SC26-8218*
- *Query Management Facility: Reference, SC26-4716*
- *Query Management Facility: Using QMF, SC26-8078*

Remote Recovery Data Facility

- *Remote Recovery Data Facility Program Description and Operations, LY37-3710*

Resource Access Control Facility (RACF)

- *External Security Interface (RACROUTE) Macro Reference for MVS and VM, GC28-1366*
- *Resource Access Control Facility (RACF) Auditor's Guide, SC28-1342*
- *Resource Access Control Facility (RACF) Command Language Reference, SC28-0733*
- *Resource Access Control Facility (RACF) General Information Manual, GC28-0722*

- *Resource Access Control Facility (RACF) General User's Guide, SC28-1341*
- *Resource Access Control Facility (RACF) Security Administrator's Guide, SC28-1340*
- *Recource Access Control Facility (RACF) System Programmer's Guide, SC28-1343*

Storage Management

- *MVS/ESA Storage Management Library: Implementing System-Managed Storage, SC26-3123*
- *MVS/ESA Storage Management Library: Leading an Effective Storage Administration Group, SC26-3126*
- *MVS/ESA Storage Management Library: Managing Data, SC26-3124*
- *MVS/ESA Storage Management Library: Managing Storage Groups, SC26-3125*
- *MVS Storage Management Library: Storage Management Subsystem Migration Planning Guide, SC26-4659*

System/370 and System/390

- *IBM System/370 ESA Principles of Operation, SA22-7200*
- *IBM System/390 ESA Principles of Operation, SA22-7205*
- *System/390 MVS Sysplex Hardware and Software Migration, GC28-1210*

System Modification Program Extended (SMP/E)

- *System Modification Program Extended (SMP/E) Reference, SC28-1107*
- *System Modification Program Extended (SMP/E) User's Guide, SC28-1302*

System Network Architecture (SNA)

- *SNA Formats, GA27-3136*
- *SNA LU 6.2 Peer Protocols Reference, SC31-6808*
- *SNA Transaction Programmer's Reference Manual for LU Type 6.2, GC30-3084*
- *SNA/Management Services Alert Implementation Guide, GC31-6809*

TCP/IP

- *IBM TCP/IP for MVS: Customization & Administration Guide, SC31-7134*
- *IBM TCP/IP for MVS: Diagnosis Guide, LY43-0105*
- *IBM TCP/IP for MVS: Messages and Codes, SC31-7132*
- *IBM TCP/IP for MVS: Planning and Migration Guide, SC31-7189*

TSO Extensions

- *TSO/E CLISTS, SC28-1876*
- *TSO/E Command Reference, SC28-1881*
- *TSO/E Customization, SC28-1872*
- *TSO/E Messages, GC28-1885*
- *TSO/E Programming Guide, SC28-1874*
- *TSO/E Programming Services, SC28-1875*
- *TSO/E User's Guide, SC28-1880*

VS COBOL II

- *VS COBOL II Application Programming Guide for MVS and CMS, SC26-4045*
- *VS COBOL II Application Programming: Language Reference, SC26-4047*
- *VS COBOL II Installation and Customization for MVS, SC26-4048*

VTAM

- *Planning for NetView, NCP, and VTAM, SC31-8063*
- *VTAM for MVS/ESA Diagnosis, LY43-0069*
- *VTAM for MVS/ESA Messages and Codes, SC31-6546*
- *VTAM for MVS/ESA Network Implementation Guide, SC31-6548*
- *VTAM for MVS/ESA Operation, SC31-6549*
- *VTAM for MVS/ESA Programming, SC31-6550*
- *VTAM for MVS/ESA Programming for LU 6.2, SC31-6551*
- *VTAM for MVS/ESA Resource Definition Reference, SC31-6552*

Index

D

E

F

G

H

I

N

O

P

Q

R

S

T

U

V

We'd Like to Hear from You

DB2 for OS/390
Version 5
Data Sharing:
Planning and Administration

Publication No. SC26-8961-02

Please use one of the following ways to send us your comments about this book:

- Mail—Use the Readers' Comments form on the next page. If you are sending the form from a country other than the United States, give it to your local IBM branch office or IBM representative for mailing.
- Fax—Use the Readers' Comments form on the next page and fax it to this U.S. number: 800-426-7773 or (408) 463-4393.
- Electronic mail—Use one of the following network IDs:
  - IBMMail: USIBMXFC @ IBMMAIL
  - IBMLink: DB2PUBS @ STLVM27
  - Internet: DB2PUBS@VNET.IBM.COM

  Be sure to include the following with your comments:

  - Title and publication number of this book
  - Your name, address, and telephone number or your name and electronic address if you would like a reply

Your comments should pertain only to the information in this book and the way the information is presented. To request additional publications, or to comment on other IBM information or the function of IBM products, please give your comments to your IBM representative or to your IBM authorized remarketer.

IBM may use or distribute your comments without obligation.

Readers' Comments

DB2 for OS/390
Version 5
Data Sharing:
Planning and Administration

Publication No. SC26-8961-02

How satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Technically accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐
Grammatically correct and consistent	☐	☐	☐	☐	☐
Graphically well designed	☐	☐	☐	☐	☐
Overall satisfaction	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

May we contact you to discuss your comments? ☐ Yes ☐ No

Name

Address

Company or Organization

Phone No.

Readers' Comments
SC26-8961-02

IBM®

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department BWE/H3
PO Box 49023
San Jose, CA 95161-9945

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5655-DB2

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

DB2 for OS/390
Version 5

SC26-8957	Administration Guide
SC26-8958	Application Programming and SQL Guide
SC26-8959	Call Level Interface Guide and Reference
SC26-8960	Command Reference
SC26-8961	Data Sharing: Planning and Administration
SX26-3841	Data Sharing Quick Reference
LY27-9659	Diagnosis Guide and Reference
LY27-9660	Diagnosis Quick Reference
GC26-8970	Installation Guide
GC26-8979	Master Index
SC26-8962	Messages and Codes
SC26-8964	Reference for Remote DRDA Requesters and Servers
SX26-3842	Reference Summary
SC26-8965	Release Guide
SC26-8966	SQL Reference
SC26-8967	Utility Guide and Reference
GC26-8971	What's New?

SC26-8961-02