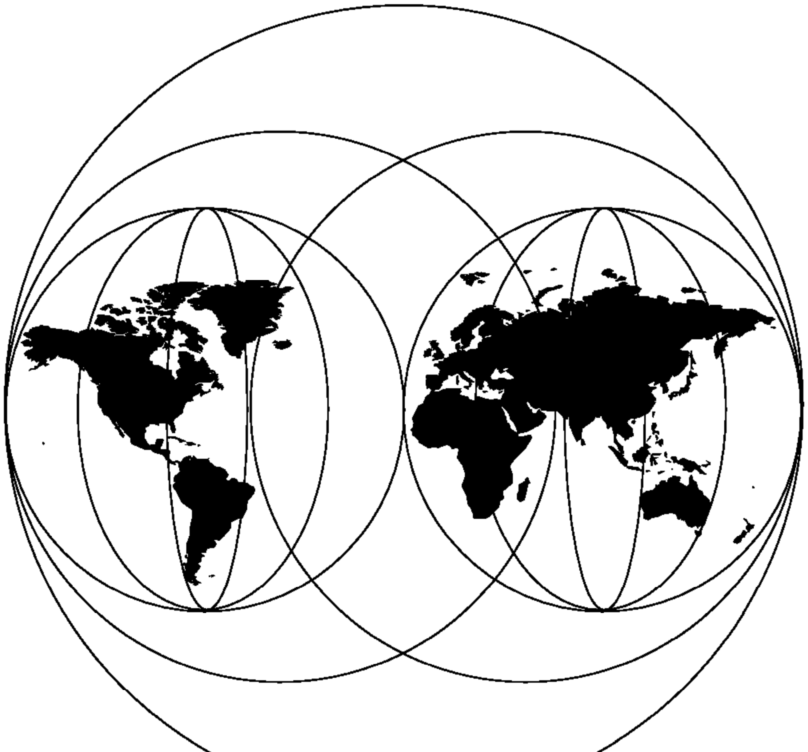
International Technical Support Organization

SG24-4695-00

Backup, Recovery and Availability with DB2 Parallel Edition on RISC/6000 SP

April 1996



International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4695-00

Backup, Recovery and Availability with DB2 Parallel Edition on RISC/6000 SP

April 1996

Take Note!

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

First Edition (April 1996)

This edition applies to Version 1 Release 1 of DB2 Parallel Edition for use with RISC/6000 SP, PSSP Version 2 Release 1.

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

An ITSO Technical Bulletin Evaluation Form for reader′s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 541 Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

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

Abstract

This redbook is unique in its detailed coverage of backup, recovery and high availability with DB2 Parallel Edition on RISC/6000 SP. It focuses on the implementation of backup and restore operations in a parallel environment using ADSTAR Distribution Storage Manager (ADSM) and Client Input Output/Sockets (CLIO/S). It also focuses on data availability vital in most commercial environments utilizing High Availability Cluster Multi-Processing (HACMP). It provides rudimentary information about one of the best storage managers suitable for parallel and distributed environments (ADSM).

This redbook was written for customers and IBM systems engineers who will be faced with the challenge of implementing backup and restore operations and data availability with DB2 Parallel Edition on RISC/6000 SP. Some knowledge of RISC/6000 SP, DB2 Parallel Edition, ADSM and HACMP is assumed.

(134 pages)

Contents

Figures

Special Notices

This redbook is intended to help system engineers and IBM customer who will be faced with the challenges of implementing backup and restore operation and high availability on RISC/6000 SP with DB2 Parallel Edition. The information in this publication is not intended as the specification of any programming interfaces that are provided by PSSP, ADSM, HACMP or DB2 Parallel Edition packages. See the PUBLICATIONS section of the IBM Programming Announcement for PSSP, ADSM, HACMP and DB2 Parallel Edition Packages for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

ADSTAR	AIX
APPN	C Set ++
DATABASE 2	DataHub
DataJoiner	DataPropagator
DB2	DB2 Paralle Edition
Distributed Database Connection Services/2	Distributed Relational Database Architecture
DProp	DRDA
Enterprise Systems Architecture/390	Enterprise Systems Connection Architecture
ES/3090	ESCON XDF
ESCON	Hardware Configuration Definition
IBM	Micro Channel
MVS/ESA	MVS/XA
S/390	Scalable POWERparallel Systems
SP	System/390
VTAM	9076 RISC/6000 SP

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Windows is a trademark of Microsoft Corporation.

HYPERchannel.*	Network Systems Corporation
SunOS, SPARCstation, Network File System, NFS.*	Sun Microsystems, Inc.

Other trademarks are trademarks of their respective companies.

Preface

This redbook is intended for customers and IBM systems engineers who plan to implement DB2 Parallel Edition backup and restore using ADSM and CLIO/S on RISC/6000 SP. It also describes the implementation of High Availability Cluster Multi-Processing (HACMP) on RISC/6000 SP with DB2 Parallel Edition. It contains detailed scenarios involved in implementing database backup, online/offline backup, log backup and DB2 Parallel Edition back from RISC/6000 SP to mainframe tape using CLIO/S. It also contains examples on recovering and restoring the DB2 Parallel Edition database. While this redbook has described a set of scenarios and some parameters that can affect performance using these scenarios, it does not discuss backup and recovery in general. It does not discuss how to select a system configuration that will facilitate backup and recovery activities.

How This Document is Organized

The document is organized as follows:

- Chapter 1, "RISC/6000 SP and DB2 Parallel Edition Overview"

  This chapter provides an overview of the DB2 Parallel Edition, HACMP and RISC/6000 SP. It introduces some of the functions used by DB2 Parallel Edition.

- Chapter 2, "Introduction to ADSM"

  This chapter provides a brief overview of ADSM and the many platforms that it supports. It describes the key components and functions of ADSM such as policy management and scheduling.

- Chapter 3, "Exploiting ADSM Version 2 for Backup and Restore"

  This chapter focuses on the use of ADSM to perform backup and recovery of DB2 Parallel Edition databases. It deals with the rudimentary procedure of installing and configuring ADSM for backup and recovery of DB2 Parallel Edition databases.

- Chapter 4, "Configuring ADSM and DB2 Parallel Edition"

  This chapter details the steps required to fully configure ADSM server and client parameters. It shown how to define the communication protocols and the various database buffer pools.

- Chapter 5, "Database Backup"

  This chapter focuses on the DB2 Parallel Edition database backup. It provides the details required to perform online/offline backup and log backups. It describes the process of initializing the 3490E tape device and how to manage the additional licensing requirement to enable ADSM operation with the tape device.

- Chapter 6, "Recovering a DB2 Parallel Edition Database"

  This chapter deals with the recovery of DB2 Parallel Edition database and the automation of this operation with ADSM. It also describes the implementation of the rollforward recovery with the user exit program.

- Chapter 7, "Using CLIO/S for Backup and Restore to Mainframe Tape"

This chapter provides the detailed scenarios required to perform DB2 Parallel Edition backup and recovery from RISC/6000 SP to mainframe 3490 tape using CLIO/S. It describes some of the CLIO/S commands and the process of using these commands to perform DB2 Parallel Edition backup and restore from the mainframe.

- Chapter 8, "Performance Considerations"

  This chapter examines the performance consideration factors that may be required to enhance your system environment during backup operation. These include, network parameters, ADSM and DB2 Parallel Edition related parameters.

- Chapter 9, "High Availability for DB2 Parallel Edition on RISC/6000 SP"

  This chapter examines the use of High Availability Cluster Multi-Processing in implementing high availability from a DB2 Parallel Edition perspective. It provides the basic overview of HACMP and offers alternative suggestions to some of the issues raised by DB2 Parallel Edition.

- Chapter 10, "HACMP Considerations and Implementation Alternatives for DB2 Parallel Edition"

  This chapter describes the factors to be considered in implementing HACMP with DB2 Parallel Edition on RISC/6000 SP. These factors include the RISC/6000 SP network and various DB2 Parallel Edition possible component failures and recoveries.

- Chapter 11, "Prerequisite for HACMP Installation and Installing HACMP"

  This chapter focuses on the prerequisite factors for High Availability Cluster Multi-Processing installation and configurations. These include the RISC/6000 SP automounter (amd), disk mirroring and cluster configurations.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *DB2 Parallel Edition for AIX Administration Guide and Reference*, SC09-1982
- *ADSTAR Distribution Storage Manager for AIX Administrator's Reference*, SH35-0135
- *ADSTAR Distribution Storage Manager for AIX Installing the Server and Administrative Client Version 2*, SH35-0136
- *HACMP Version 4 Release 1 for AIX Installation Guide* , SC23-2769
- *IBM RISC/6000 Scalable POWERparallel Systems Diagnosis and Messages Guide*, GC23-3899
- *AIX Version 4 Release 1 Problem Solving Guide and Reference*, SC23-2606
- *IBM Client Input Output/Sockets General Information Manual*, GC23-3879
- *IBM Client Input Output/Sockets User's Guide and Reference*, CLIO-user-02 (available with the product)

International Technical Support Organization Publications

- *ADSTAR Distributed Storage Manager/6000 on 9076 SP2*, GG24-4499
- *Using ADSM to Back Up Databases*, GG24-4335
- *ADSM Version 2 Presentation Guide*, SG24-4532
- *Risc/6000 370 Channel Support ESCON and Block Multiplexer* , SG24-4589

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

> *International Technical Support Organization Bibliography of Redbooks,* GG24-3070.

How Customers Can Get Redbooks and Other ITSO Deliverables

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **IBMLINK**

  Registered customers have access to PUBORDER to order hardcopy, to REDBOOKS disk to obtain BookManager BOOKs

- **IBM Bookshop** — send orders to:

  usib6fpl@ibmmail.com (United States)
  bookshop@dk.ibm.com (Outside United States)

- **Telephone orders**

  1-800-879-2755 (United States)
  354-9408 (Australia)
  359-2-731076 (Bulgaria)
  42-2-67106-250 (Czech Republic)
  593-2-5651-00 (Ecuador)
  03-69-78901 (Israel)
  905-627-1163 (Mexico)
  064-4-57659-36 (New Zealand)
  027-011-320-9299 (South Africa)

  0256-478166 (United Kingdom)
  32-2-225-3738 (Belgium)
  1-800-IBM-CALL (Canada)
  45-934545 (Denmark)
  01805-5090 (Germany)
  0462-73-6669 (Japan)
  31-20513-5100 (The Netherlands)
  507-639977 (Panama)

- **Mail Orders** — send orders to:

  IBM Publications
  P.O. Box 9046
  Boulder, CO 80301-9191
  USA

  IBM Direct Services
  Sortemosevej 21,
  3450 Allerod
  Denmark

- **Fax** — send orders to:

  1-800-445-9269 (United States)
  32-2-225-3478 (Belgium)
  905-316-7210 (Canada)
  593-2-5651-45 (Ecuador)
  03-69-59985 (Israel)
  31-20513-3296 (The Netherlands)
  507-693604 (Panama)

  0256-843173 (United Kingdom)
  359-2-730235 (Bulgaria)
  42-2-67106-402 (Czech Republic)
  07032-15-3300 (Germany)
  0462-73-7313 (Japan)
  064-4-57659-16 (New Zealand)
  027-011-320-9113 (South Africa)

- **1-800-IBM-4FAX (United States only)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **Direct Services**

  Send note to softwareshop@vnet.ibm.com

- **Redbooks Home Page on the World Wide Web**

  http://www.redbooks.ibm.com/redbooks

- **E-mail (Internet)**

  Send note to redbook@vnet.ibm.com

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How IBM Employees Can Get Redbooks and ITSO Deliverables

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States

- **GOPHER link to the Internet**

  Type GOPHER
  Select IBM GOPHER SERVERS
  Select ITSO GOPHER SERVER for Redbooks

- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET GG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET GG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks/redbooks.html

- **ITSO4USA category on INEWS**

- **IBM Bookshop** — send orders to:

  USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

- **Internet Listserver**

With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

Acknowledgments

This project was designed and managed by:

Endy Chiakpo
International Technical Support Organization, Poughkeepsie Center

Eric Le Bail
IBM EMEA PSSC Montpellier France

The authors of this document are:

Alain Benhaim
IBM France

Endy Chiakpo
ITSO Poughkeepsie

Alain Fournil
IBM France

Otto Goerlich
IBM Germany

Eric Le Bail
IBM France

Dr. Rodolphe Michel
IBM UK (ISC Hursley)

Jack Woodson
IBM Dallas Systems Center

This publication is the result of a residency conducted at the International Technical Support Organization, Poughkeepsie Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Cindy Berehens
International Technical Support Organization, San Jose Center

Michel Perraud
International Technical Support Organization, Poughkeepsie Center

Quan Cung
International Technical Support Organization, Austin Center

David Thiess
International Technical Support Organization, Austin Center

POWERparallel Systems Poughkeepsie:
Bill Carlson
Brian O′Leary
Iwao Hatanaka

IBM Toronto Laboratory:
Scott Bailey
Kenneth Chen
George Chan
Roger Zheng

IBM TJ Watson Research Center:
Hui-I Hsiao

Chapter 1. RISC/6000 SP and DB2 Parallel Edition Overview

The IBM RISC/6000 SP provides a robust and reliable platform for customers in both the commercial and scientific environments to process their applications. The robustness of the RISC/6000 SP and the high industrial strength that it provides through a high degree of scalability and parallelism makes it very suitable for parallel databases. The focus of this redbook will be on the implementation of backup and restore using ADSM and continuous data availability using High Availability Cluster Multi-Processing (HACMP).

1.1 Parallel Databases

The concept of processing data in a parallel environment is fast becoming the solution some businesses and commercial customers are turning to for today and future data processing challenges. This is due to the fact that the rate of increase in database size and also because the response time requirements have outpaced advancements in processor and mass storage technology. One method of fulfilling the increasing demand for high processing power and input/output bandwidth in complex database applications is to have a number of processors, loosely or tightly coupled, serving parallel database requests simultaneously.

Large scale parallel processing technology has made giant strides in the past decade and there is no doubt that it has established a place for itself. However, the current generation of massively parallel processor systems, in particular, IBM′s RISC/6000 Scalable Parallel class of systems, are much more robust and easy to use.

One of the main enablers for commercial applications is Database Management Systems (DBMS). Thus, a parallel DBMS is a natural step. Several businesses and industries are investing in decision support applications in order to understand various sales and purchase trends. These applications pose complex questions (queries) against large sets of data in order to gain an insight into the trends. Single system (or serial) DBMSs cannot handle the capacity and the complex requirements of these applications. Besides decision support, there are other new application classes such as data mining, electronic libraries, that require either large capacity or the ability to handle complexity. All these applications require parallel DBMS software.

1.2 DB2 Parallel Edition

DB2 Parallel Edition is one of the first IBM answers to the market place, to address the high demand by the customers for a much faster and reliable method of processing data in a parallel environment.

DB2 Parallel Edition is a parallel database software solution that can execute on AIX 3.2.5 or later versions. Its Shared-Nothing (SN) architecture model and Function Shipping execution model provide very important assets, namely scalability and capacity. This system can easily accommodate databases with hundreds of gigabytes of data. Likewise, the system model enables databases to be easily scaled with the addition of more system CPU and disk resources. DB2 Parallel Edition has been designed and implemented to provide the best query processing performance. The query optimization technology considers a variety of parallel execution strategies for different operations and queries and uses the lowest cost in order to choose the best possible execution strategy. The execution time environment is optimized to reduce process overhead, synchronization overhead, and data transfer overhead. Refer to the *DB2 Parallel Edition for AIX Administration Guide and Reference,* SC09-1982 for more details.

1.2.1 Backup and Restore Considerations

The degree of parallelism achieved during backup and restore of a database is determined by the number of backup devices available. The DB2 Parallel Edition backup and restore design allows each node in the system to be backed up independently. Data from several nodes can be backed up simultaneously, if multiple backup devices are available. The backup utility creates a backup image of the entire database partition resident at a given node.

At restore time, it is necessary to ensure that the database partition that is being restored is in a consistent state with respect to the rest of the nodes in the system. This can be achieved using any of the three methods below:

1. By restoring all nodes in the system using backup images that are known to be consistent.
2. By restoring all nodes and rolling forward logs to a point in time where the database state is consistent across all nodes.
3. By restoring a subset of nodes and rollforward to end of log.

1.2.2 High Availability

High availability is supported by the use of HACMP software. The HACMP software provides takeover of the disk and communications resources of the failed node. System nodes are clustered together and each cluster has access to shared disks. If one of the processors in a cluster fails, one of the other processors can take over and the system can continue to operate.

To enable use of HACMP software, the database engine has been designed to allow the situation where a single processor executes multiple copies of the database engine. In other words, multiple database or logical nodes are mapped to the same physical node. While this method provides quick takeover of a failed node, there may be an impact on performance due to the increased load on the takeover processor. In many decision support applications, it is not essential to provide instant takeover capability, whereas it is important not to degrade overall system performance. Thus, it may be acceptable to have a particular node become inaccessible for say, tens of minutes, in order to be able to recover from a failure of that node without any subsequent performance penalty. This can be achieved by configuring one or more spare nodes in the system which can take over on behalf of any failed node. When a node fails, its database files are available to the spare node through the shared disks.

Chapter 2. Introduction to ADSM

This chapter provides a brief overview of ADSM, the market′s best storage manager in a distributed environment. This particular chapter is not specific to DB2 Parallel Edition. We look at the many platforms ADSM supports, the main components of ADSM, and descriptions of key functions such as scheduling and policy management. We conclude with a summary of ADSM advantages.

ADSTAR

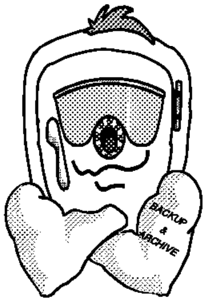
Distributed Storage Manager

The market's best

Storage Manager

in a

Distributed Environment



2.1 What Is ADSM?

ADSM, IBM′s solution to enterprise-wide distributed storage management, is a client/server program product. It provides highly automated, centrally scheduled, network-based backup and archive functions for workstations and LAN file servers. ADSM supports a wide variety of IBM and non-IBM clients and servers, as shown in Figure 1 on page 4, and addresses the need for customer asset protection and data availability for distributed environments.

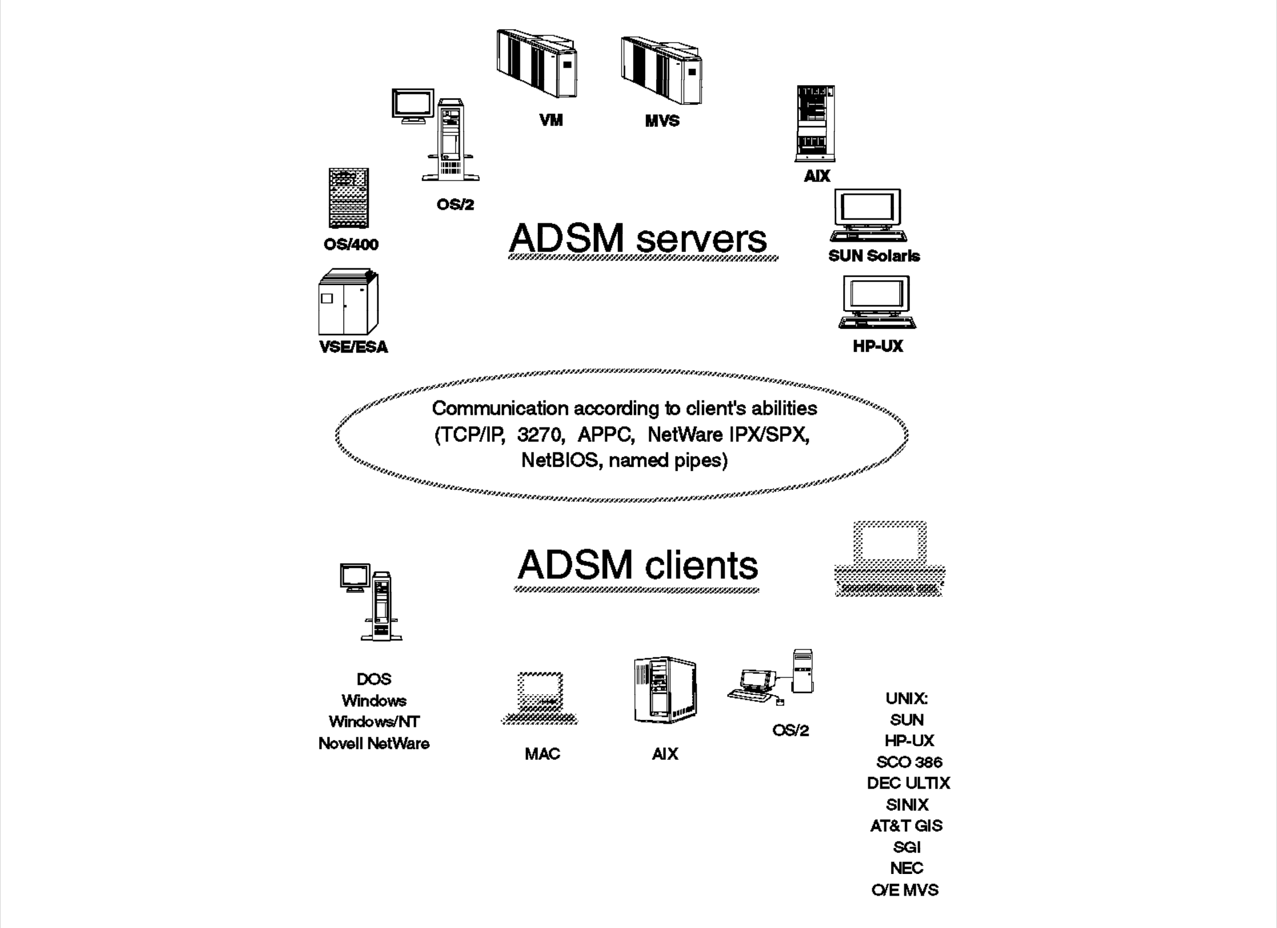


Figure 1. ADSM Platform Support

2.2 Main Components

Let us look at the main components of ADSM—the backup/archive client, the administrative client, and the server, as shown in Figure 2 on page 5—and then briefly review the application client.

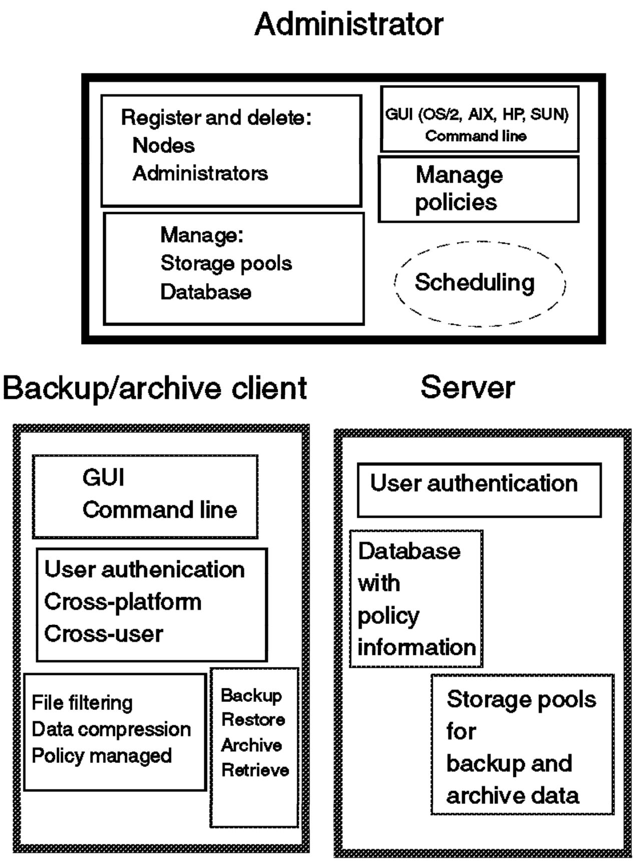


Figure 2. ADSM Storage Management Components

2.2.1 Backup/Archive Client

The backup/archive client runs on the workstation and, depending on the platform, provides both a graphical user interface (GUI) and command line interface (CLI). Although all clients are similar, they each have the look and feel of the platform on which they are running. Thus users can back up or restore files using an interface with which they are familiar.

The main function of ADSM is backup and restore. You can back up all of your files (full), specific files (selective), or only those files that have changed since your last backup (incremental).

The file compression provided on the client platform reduces network traffic and the amount of storage required on the server to store the files.

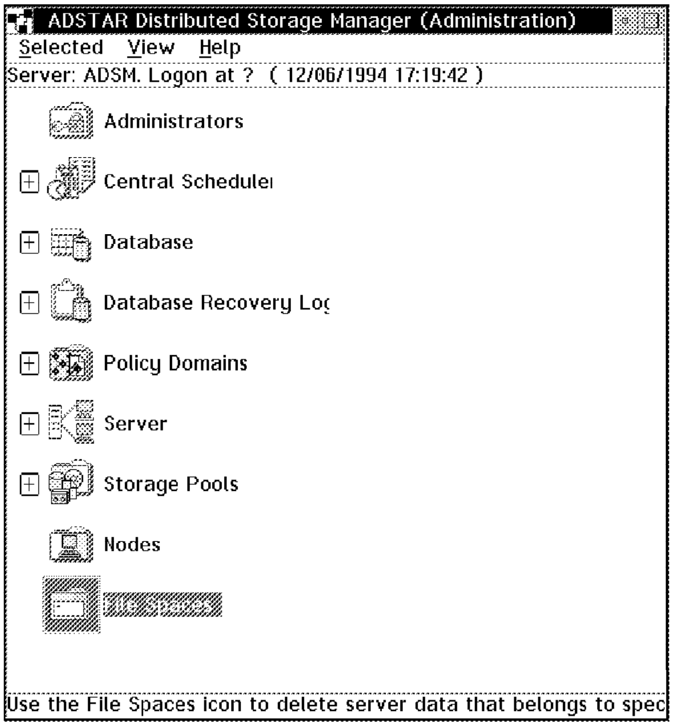
You can specifically include or exclude certain files from being backed up. For example, you might not want everyone to back up their local copies of the OS/2 operating system!

ADSM′s cross-user and cross-platform restore provide you with significant flexibility. Cross-user restore enables you to authorize someone else to restore your files. Cross-platform restore enables you to restore your file on a platform different from the platform on which it was backed up. For example, you could back up your file from a DOS workstation but then restore it to an OS/2 workstation. Cross-platform restore can be extremely useful when you migrate to new workstation platforms, or even if you happen to work at a different office one day that has different workstations. You will still have access to the data you backed up.

A separate archive/retrieve function is also part of ADSM. This function provides a way for you to store files that you may not use but need to retain for long-term storage. Archive is also useful as a way of reducing the disk space on your workstation. You can archive files for long-term storage and erase the original files from your workstation to create room for more active files and applications.

2.2.2 Administrative Client

As shown in Figure 3, an administrator controls or monitors server activity, defines storage management policies for workstation files, and sets up schedules to provide backup and archive services at regular intervals.



ADSTAR Distributed Storage Manager (Administration)

Server: ADSM. Logon at ? (12/06/1994 17:19:42)

Use the File Spaces icon to delete server data that belongs to spec

Figure 3. ADSM Administrative Client

An administrative client is a program that allows administrators to control and monitor the server through administrative commands. The administrative program can be installed on a programmable workstation (PWS), personal computer, or mainframe. An administrative client passes commands through an administrative command line. In some cases, a GUI has been added to the administrative client code.

ADSM provides a hierarchical structure to the authority you can grant an administrator. Thus you can establish as flexible an administration scheme as you would like while still providing control over your system. The ADSM administrator with overall authority is called the system administrator. The other administrators are called policy, storage, operator, or analyst administrators, depending on which part of the system they control. Their administrative tasks are separated into logical categories, such as controlling the management policies, the storage pools and databases, the operation of the server, and the analysis of certain server events.

Dividing up the administrative authority based on logical categories of tasks is not the only way of granting authority. You can also divide up the administrative authority by organization. You can give the logical categories of authority to a department, but only for the data that belongs to that department. For example, you can give a department policy and storage authority for the policy domain and storage pools that it owns.

2.2.3 Server

The server component provides storage resources and services for the backup/archive clients. Users can back up or archive their files onto server storage resources, such as disk, tape, or optical devices that are managed and monitored by ADSM server policy.

Figure 4 shows the two key components of the ADSM server: the storage pools where the client files are actually stored, and the database that serves as an inventory or index to the client files within the storage pools. The database consists of the database space and the recovery log. The recovery log keeps track of all changes made to the database.

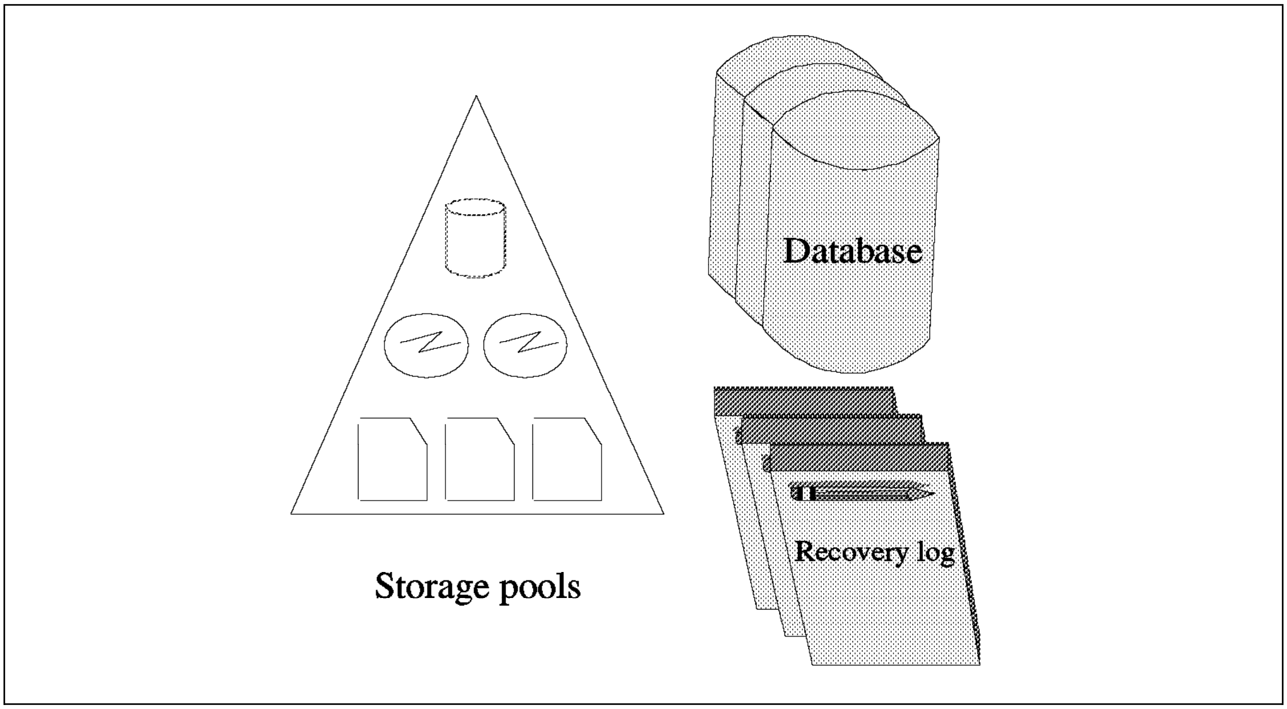


Figure 4. ADSM Server Components

The storage pools contain the client files that have been backed up or archived. A hierarchy of storage media can be used to define the storage pools. The pools can contain disk storage, optical devices, and tape devices. Each ADSM server platform supports a different set of storage media, so please verify the devices that are supported in your environment.

Data can be moved automatically through the storage hierarchy onto less expensive media with ADSM′s migration function. Additional management functions are provided, such as reclamation and collocation for tape management.

The ADSM server is multitasking, so multiple clients can back up data concurrently.

The ADSM database is the heart of the server. The server database is critical to the operation of ADSM because it contains file location information as well as policy and scheduling information. The following information is stored in the database:

- Information about registered client nodes
- Policies assigned to those client nodes
- Schedules and their association with client nodes
- Event records, such as whether a schedule successfully completed
- The activity log that contains the messages generated by the server
- Information about ADSM volumes
- The data storage inventory, that is, the information used to locate files that reside in storage pools

The database has all of the features associated with a database management system. Because the database is critical, many features are built in to ADSM to help maintain the availability, integrity, and performance of the database. Two of these features are the recovery log and mirroring.

A recovery log is used to help maintain the integrity of the database. It keeps track of all changes made to the database, so that if a system outage were to occur, a record of the changes would be available in the log. When a change to the database occurs, the recovery log is updated with some transaction information before the database is updated. Thus uncommitted transactions can be rolled back during recovery so that the database remains consistent.

The administrator can configure the server so that up to three copies of the database and recovery logs are maintained at all times. This *mirroring* capability provides nondisruptive and immediate recovery from physical failures on database and recovery log volumes. Mirroring is the process of writing the same data to multiple storage devices at the same time.

If a mirrored volume encounters a media failure, the server automatically places the failing volume offline and continues database operations using the other mirrored copies. Once the failed disk is replaced and made available to the server, it is automatically synchronized with the intact copies.

The mirroring facility improves database performance. The mirrored copies are treated equally; there is no concept of primary copy and alternate copies. Therefore, the server reads from the database copy that is on the device with the best response time.

Another server function, export/import, creates a self-describing copy of specified server information. Information that can be exported includes:

- Administrator information

- Client node definitions
- Policy information
- Backup and archive data

Export/import is useful for migration and conversion, workload balancing, as well as cloning of information.

ADSM provides extensive ADSM server database and storage pool backup facilities. Incremental backups are provided as well as a mechanism for offsite backups to aid in disaster recovery.

2.2.4 Application Client

The application client is a software application that runs on a workstation and uses the ADSM application programming interface (API) to back up, archive, restore, or retrieve objects from an ADSM server.

As shown in Figure 5, the application client program enables other IBM and non-IBM products to use the storage management services of ADSM. The application client allows applications to back up or archive valuable data in any format that an application programmer specifies.

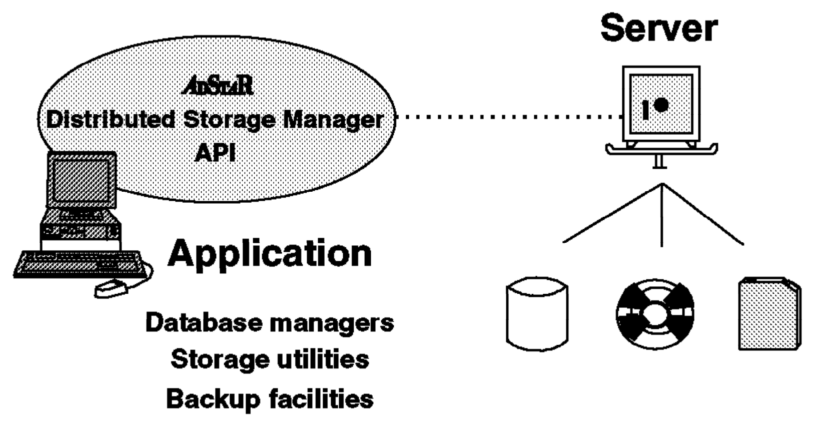


Figure 5. Application Client and ADSM API

The number of ways of using the API is unlimited. You can use it to provide better handling of nonfile data in the enterprise, such as databases or image volumes. You could, for example, provide extensions to the existing ADSM backup and restore functions to meet your user′s needs or write a virtual tape device driver so that other applications can use ADSM transparently.

The API is available for the C programming language.

2.3 Functions

Let us look at the ADSM backup and restore, archive and retrieve, central scheduling, and policy management functions.

2.3.1 Backup and Restore

The backup process creates a copy of a client file on the ADSM server, such as the \myfile.data file shown in Figure 6. The backup process also backs up the directory in which the file resides.

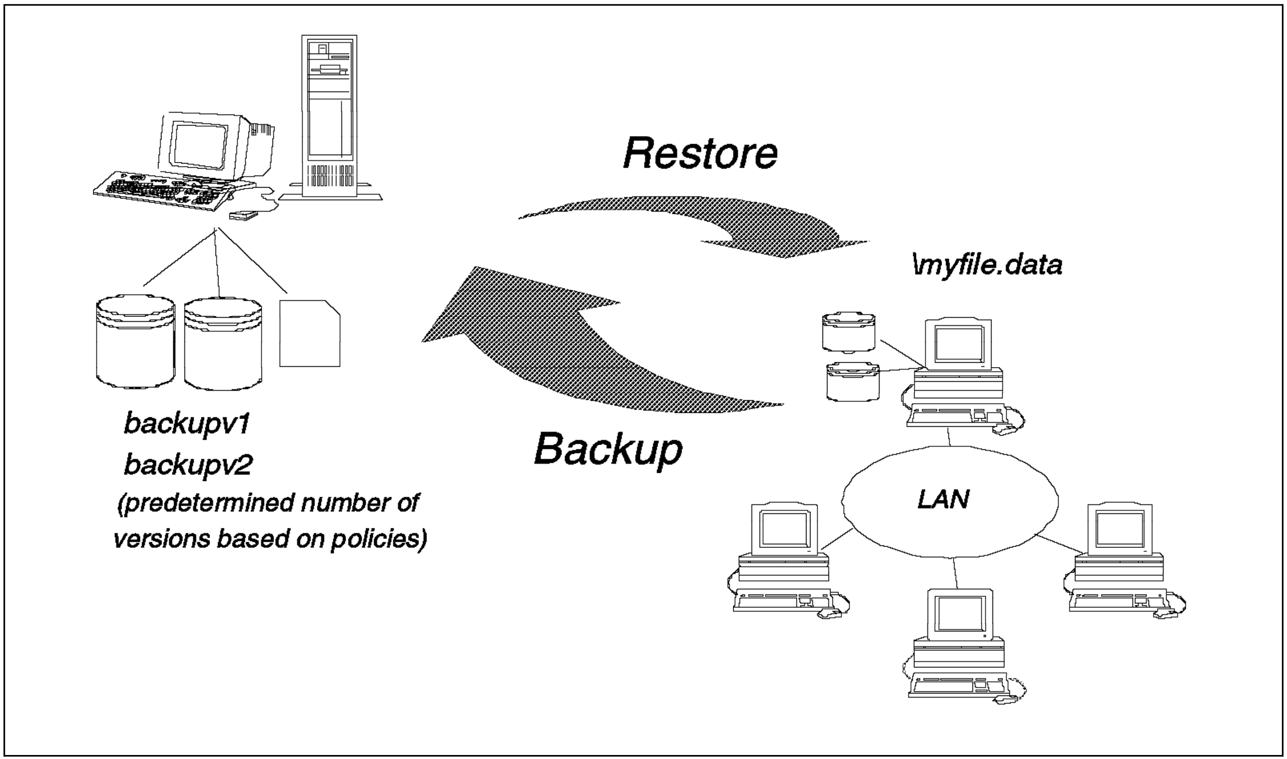


Figure 6. Backup and Restore

Incremental backup sends to the server the files that have changed since the last backup. The first time an incremental is done, all files are sent to the ADSM server. This is a full backup. ADSM determines that a file has changed if any of the following has changed: file size, date and/or time stamp, file owner, file group, file permission, or attribute change time.

Selective backup specifies which files a user wants to back up. A selective backup can consist of a single file, or a user can select a directory or subdirectory tree to back up. Because wildcards (*) are allowed in the specification, there is great flexibility in file selection.

The files are backed up according to policies that the administrator has predefined. The policies define, for example, how many backup versions should be retained in the ADSM storage pools, how long to retain those versions, and whether to back up files that are in use. Figure 6 shows that two versions of the \myfile.data file are saved in the storage pools.

Restore is the process of copying a backup version from the server to the client. This process is system assisted; that is, the system performs the restore for the user; the user does not have to call the ADSM administrator to request restoration of the file.

2.3.2 Archive and Retrieve

The archive process creates a copy of a client file on the ADSM server, such as the \myfile.data file shown in Figure 7.

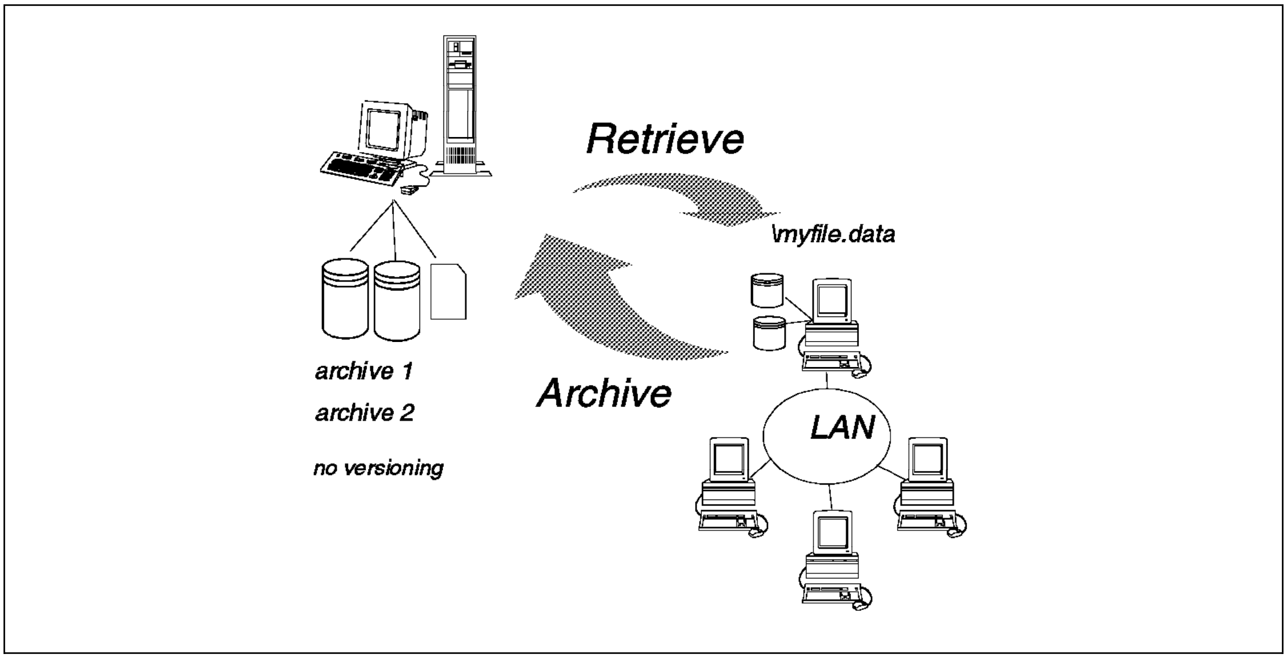


Figure 7. Archive and Retrieve

As with backup, archived files are managed on the basis of policies; however, the archive function does not have a concept of versioning. You can archive multiple versions of a file by invoking the archive function multiple times. In other words, each archived copy is treated as a separate file, not as multiple versions of a single file.

A user can save a description of an archived file so that it will be easier to retrieve the file if multiple files are archived with the same file name.

The key difference between backing up a file and archiving a file is that the user can erase the original file after archiving it. The archived version is expected to be retained for a long time. Erasing the original file does not affect the retention period for the archived file.

2.3.3 Central Scheduling

As shown in Figure 8 on page 12, the ADSM central scheduling facility automates the initiation of client backup, archive, restore, and retrieve, as well as ADSM server administrative operations. It also can schedule any client OS command and ADSM client macros. New clients can be easily associated with schedules in a nondisruptive manner. The central scheduler consists of client and server processes that cooperate to execute the scheduled functions. Thus ADSM requires the client workstations to be communicating with the server. If you want to automate your backups for off-hours or weekends, you must enforce a policy that requires users to leave their workstations powered on.

The administrator is responsible for defining and maintaining the schedules and has the authority to prioritize clients so that clients that contain more important data are given preferential treatment.

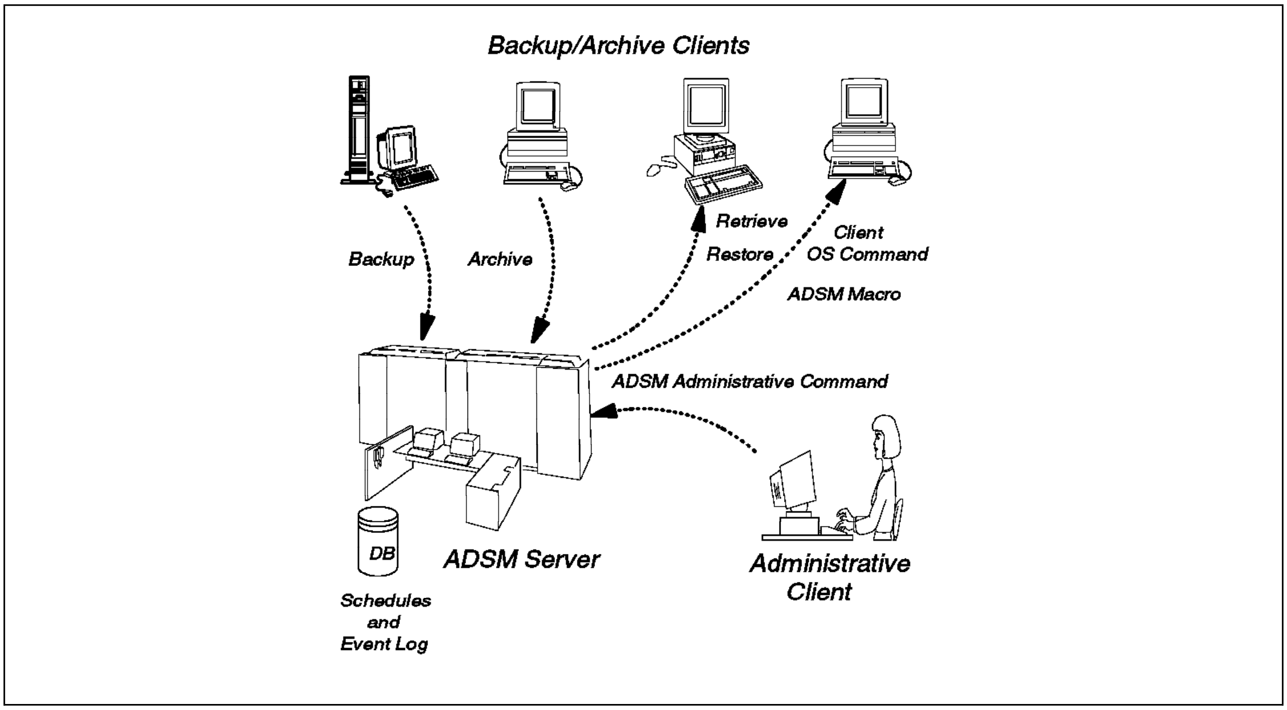


Figure 8. ADSM Central Scheduling

A schedule event log is maintained in the server database. Whenever a schedule process starts or fails, an event record is written to the log. An administrator can query the log to determine whether scheduled events completed successfully or not.

Two types of scheduling modes are supported:

- Client polling

  Client polling is supported for all client workstations and all communication methods. With client polling, a client periodically queries (or polls) the server for a scheduled operation and the date and time the operation is to start. The server sends this information to the client. Then the client waits until it is time to start the scheduled operation and executes the operations. Operations that can be scheduled are backup and archive. Restore and retrieve cannot be scheduled. Before executing the operation, the client notifies the server that a scheduled operation is starting. Upon completing the operation, the client notifies the server that the operation has completed either successfully or unsuccessfully.

  The client initiates client polling by starting the client scheduling program. To start the program the client enters `DSMC SCHEDULE`. The program will continue to query the server and execute schedules until the user explicitly stops the program or the machine is shut down.

- Server-prompted

  Server-prompted scheduling is supported for all client workstations that use TCP/IP to communicate with the server.

  With server-prompted scheduling, the ADSM client registers its TCP/IP address with the server and then waits to be prompted by the server to begin the scheduled operation. The client then starts and executes the operation. The operation can be backup or archive. Restores and retrieves cannot be scheduled. Upon completing the operation, the client notifies the server that the operation completed either successfully or unsuccessfully.

Server-prompted scheduling allows the server to control when clients are contacted to perform a scheduled operation. Server-prompted scheduling maximizes the use of scheduled sessions.

To initiate a schedule, the client must start the client scheduling program by issuing the DSMC SCHEDULE command.

2.3.4 Policy Management

As shown in Figure 9, ADSM allows you to manage the backup and archive process based on policies you establish for your enterprise. The granularity of control that you have is down to the file level. You can decide on how granular you want your policies to be. You can establish an overall system policy, policies by department or organizational structure, or policies by user or file name. Policy management makes ADSM a true system-managed storage implementation. The elements of policy management are discussed in the following sections.

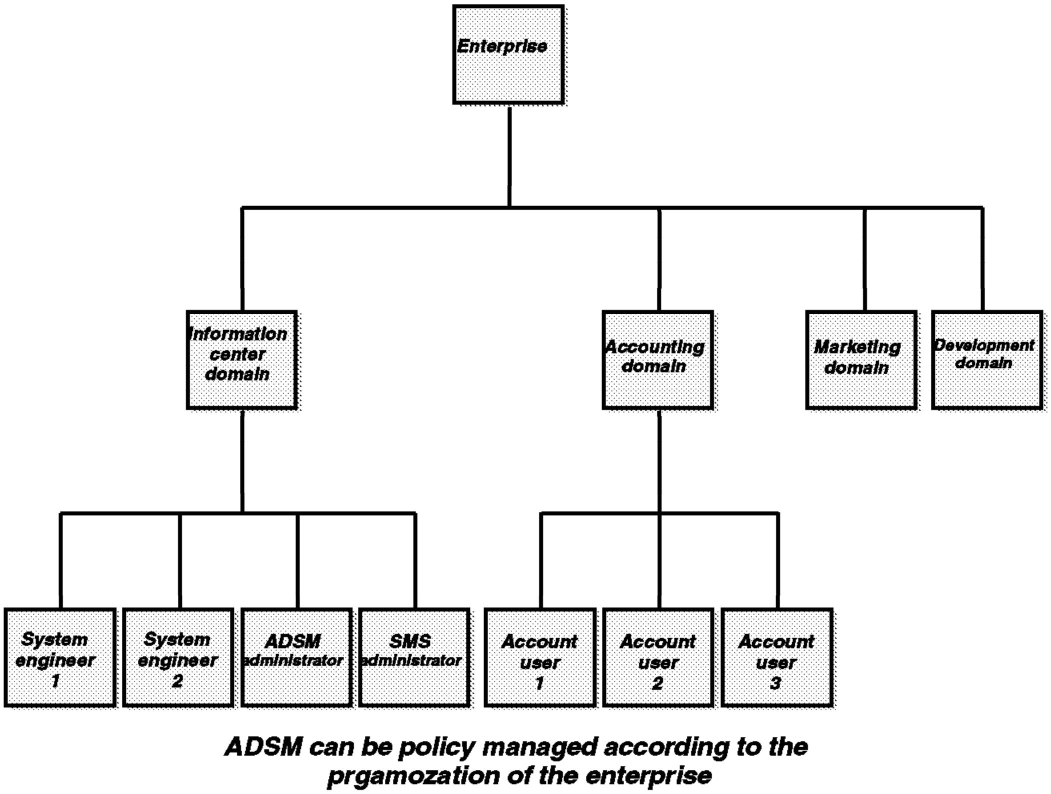


Figure 9. Policy Management

2.3.4.1 Policy Domain

A policy domain is a group of clients who are working according to the same set of policy needs. A policy domain provides a logical way of managing backup and archive policies for a group of client nodes. There is no limit to the number of policy domains that can be defined on an ADSM server. Policy domains can be used to provide standard storage management policies to most users, group together clients that have similar storage management requirements, limit the number of clients to be managed by a single policy administrator, or restrict the number of management classes to which users have access. Figure 10 on page 15 shows that clients ACCTREP1, ACCTREP2, and ACCTREP3 belong to the SALES policy domain. Note that schedules also belong to a particular policy domain.

2.3.4.2 Policy Set

Each policy domain can contain one or more policy sets. A policy set contains one or more management classes. A policy domain can have more than one policy set, but only one policy set can be activated at any point in time. Each policy set contains a default management class and can contain any number of additional management classes. Policy domain and policy set information is stored in the server database. Figure 10 on page 15 shows three policy sets with an active policy set that contains two management classes.

2.3.4.3 Management Class

Policy sets contain one or more management classes. Management classes contain a backup copy group and/or an archive copy group or no copy group. You can think of management classes as a Service Level Agreement you have with your clients on how their backup and archive data will be handled. There is a concept of binding the management class to the file when it is backed up or archived. Thus the management class is associated with that file. You can rebind a file with a new management class. Users can use the default management class or explicitly select a management class that is within the active policy set to which they have access. Figure 10 on page 15 shows two management classes, MC1 and MC2. Management class MC1 contains both a backup and archive copy group; MC2 contains only an archive copy group.

2.3.4.4 Copy Group

Copy groups are where you specify the parameters that will control the generation and expiration of backup and archive data. There is a separate copy group for backup and one for archive. In the current ADSM product, all copy groups are named STANDARD. Again, copy group information is stored in the ADSM server database. Let us look at the parameters that you can use to control your backup and archive data. Remember that the span of control is at a file level.

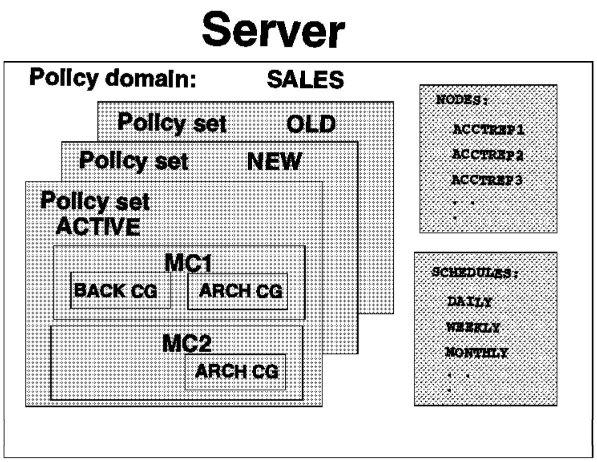


Figure 10. Policy Management Elements

Both backup and archive copy groups have similar parameters except that there is no concept of versioning with archived files. Let us look at each parameter shown in Figure 11.

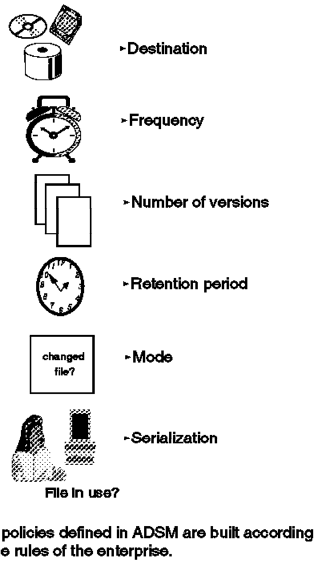


Figure 11. Copy Group Parameters

Destination specifies the name of the storage pool where the server stores the backed up or archived files.

Frequency for a backup file specifies the minimum number of days that must elapse between incremental backups. This parameter is not used for selective backups. Frequency for an archive file is always command (CMD). A file is only archived when a client issues an archive command or chooses archive from the GUI.

The concept of **versioning** applies only to backup files. You can specify two different parameters to tell ADSM how many versions of a backup file you want it to maintain. The Version-data-exists parameter specifies the maximum number of different backup versions the server retains for files and directories that currently exist on the client workstation. The most current backup version is called the active version. All other versions are called the inactive versions. When the maximum number of versions is exceeded, the server rolls off the oldest version. The Version-data-deleted parameter specifies the maximum number of different backup versions the server retains for files and directories that have been erased from the client workstation.

The **retention period** parameter specifies how long to retain the backed up and archived files. There are two retention parameters for backed up files that correspond to the two types of versioning, and there is one retention parameter for the archived files. Retain extra versions specifies how many days the server retains the inactive backup versions when the original file no longer exists on the client′s workstation. Retain only version specifies how many days the server retains the backup versions it has of a file when the original file has been deleted from the workstation. Retain version specifies the number of days an archived copy remains in data storage.

With the **mode** parameter you can specify file backup depending on whether the file has changed since the last backup. This parameter applies to incremental backups, not selective backups. The options for mode are modified and absolute. Modified means that you want to back up the file only if it has changed. Absolute means that you want to back up the file regardless of whether it has changed. For archive files, the mode is always absolute.

Serialization specifies how files or directories are handled if they are modified during the backup or archive process. The serialization parameter has four options: static, shared static, shared dynamic, and dynamic.

- **Static** specifies that if a file or directory is modified during the backup or archive process, ADSM will not back up or archive the file. The static mode is not supported on the DOS platform.
- **Shared static** specifies that ADSM will retry the backup operation as many times as specified in the client′s option file. The default is four retries. If the file or directory is modified during each backup or archive attempt, ADSM will not back up or archive the file.
- **Shared dynamic** specifies that if a file is modified during a backup or archive attempt, ADSM will only back up or archive the file on its last retry.
- **Dynamic** specifies that even if the file is modified during the backup or archive attempt, ADSM will back up or archive the file anyway. No retries are required.

2.4 Advantages

IBM′s commitment to provide storage management functions for remote systems and multiple platforms will enable the enterprise to gain substantial benefits. Let us conclude this section of the book by summarizing the many advantages of ADSM. We have grouped them into three categories: cost reductions, increased productivity, and increased security of corporate assets.

2.4.1 Cost Reductions

As proven in the large systems arena, fewer people will need to be involved in the process of storage management. Because ADSM provides a single approach to storage management for your entire enterprise, perhaps all enterprise storage can be managed by one central group.

Available storage is managed more effectively because inactive data can be moved to other media, thus increasing the amount of fast access disk space required for active data.

Distributed data growth can be easily monitored and controlled because less duplication of data is likely to occur as users become confident that their data is secured. ADSM can also help you avoid backing up multiple copies of system software.

2.4.2 Increased Productivity

With storage management across an enterprise, any issues regarding data integrity can be restored by the system more quickly and accurately while adhering to enterprise-defined standards.

Many of today′s tasks that many people do manually can now be automatically managed by fewer people. End users or system administrators from different systems and platforms will spend less time on availability and space management. They will experience fewer losses of critical data and fewer out-of-space incidents. They will be freed from the tedious manual work of managing the disks they own; data management can be done transparently with little impact on their usual work.

Using a single storage management strategy across your heterogeneous platforms encourages networkwide standards. This might be at a high level, where standard policies of how to manage the data are established, or at a lower level, where, for example, file naming conventions become more standardized to improve restorability to other platforms.

2.4.3 Increased Security of Corporate Assets

With a structured process for backing up your data in your enterprise, you can be ensured that your data will be available when you need it. The risk of losing valuable data, which is a valuable corporate asset, has been substantially reduced.

You may also be taking advantage of more reliable server storage than that provided on your local workstation. The system on which the server storage resides may also have a higher level of security enforcement and control. The server storage might provide a higher level of security simply by being in a central data center (glass house) that requires a badge lock to enter.

Now that you know what ADSM is all about, let us look at how you can use ADSM to back up and manage your DB2 Parallel Edition data including databases and system configuration information.

Chapter 3. Exploiting ADSM Version 2 for Backup and Restore

This chapter focuses on DB2 Parallel Edition database backups and restores using ADSM. It also deals with basic ADSM installation and configuration so that DB2 Parallel Edition database backup can use ADSM services. It is mainly concerned with the choices that have to be made and the impact they will have when using ADSM to backup DB2 Parallel Edition databases on the RISC/6000 SP. A more complete overview of ADSM can be found in the ADSM related documentation whose references can be found in the *Bibliography of Redbooks*.

Figure 12 on page 20 illustrates the kind of flexibility possible with attaching different kinds of external devices to the RISC/6000 SP. The figure also shows the system environment used in implementing the scenarios presented in this document.

3.1 Planning for Installation and Configuration

There are some basic requirement necessary to successfully install, customize and configure ADSM in your system. These requirements includes the following:

- Memory Size
- Disk Space

3.1.1 Memory Requirements

ADSM memory requirements largely depend on how much backing up activity will be undertaken.

For further details, see *ADSM Administrator's Reference Version 2,* SH35-0135-00.

3.1.2 Disk Space Requirements

The disk requirements could be subdivided into two parts, namely; server and client requirement considerations. Each one of them is discussed below.

3.1.2.1 Server Requirements

The following file sets, at least, have to be installed:

- adsmserv.base.obj (ADSM Common Files),approximately 13 MB
- adsm.base.obj (ADSM Backup/Archive Client), approximately 4 MB
- adsm.common.obj, approximately 750 KB
- adsm.admin.obj (ADSM Administrator GUI Client), approximately 6 MB

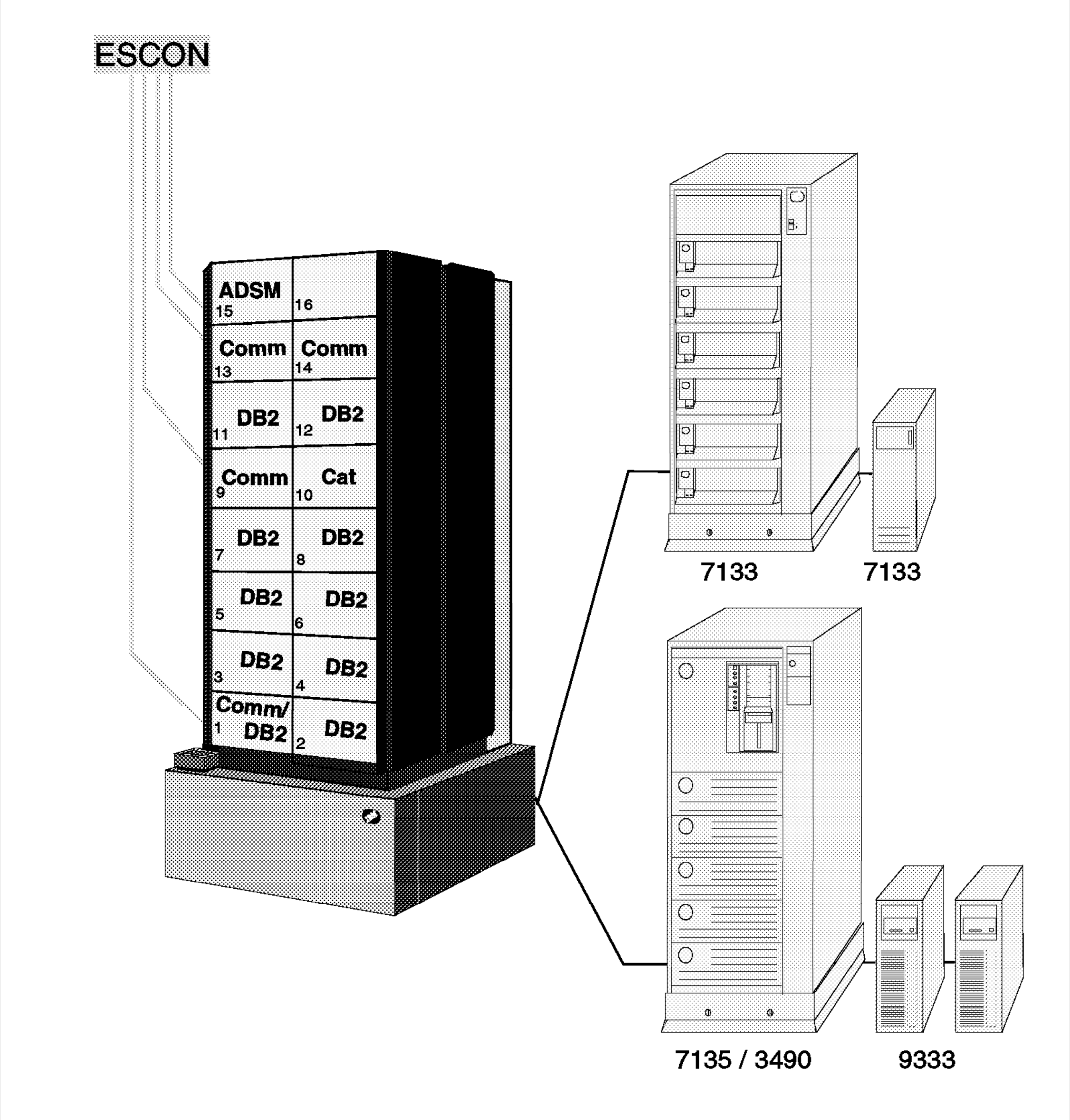


Figure 12. RISC/6000 SP with Thin Nodes, External Attached Disks and Tape Units

The following default ADSM volumes are created in the /var file system (if space is available) or in /usr/lpp/adsmserv/bin.

- db.dsm, 5 MB
- log.dsm, 9 MB
- archive.dsm, 8 MB
- backup.dsm, 8 MB
- spcmgmt.dsm, 8 MB

Note: Those volumes are created by default by ADSM. Any additional ADSM database, log or storage volumes that the administrator may require to create are not taken into account.

The following additional file sets can also be installed:

- adsm.api.obj (ADSM API), approximately 750 KB
- adsm.spcmgmt.obj (Hierarchical Storage Manager)

  Note: We will not deal here with the *Hierarchical Storage Manager* services provided by ADSM as the space manager is not yet available on AIX 4.1.3 at the time of the writing of this redbook.

  Note: Copy /etc/vfs to /etc/vfs.ref before attempting any HSM installation. When attempting to install the HSM code, /etc/vfs may be set to size 0. No NFS mounts can take place until /etc/vfs has been restored.

3.1.2.2 Client Requirements

The following file sets, at least, have to be installed:

- adsm.base.obj (ADSM Backup/Archive Client), approximately 10 MB
- adsm.common.obj, approximately 750 KB

The following additional file sets can also be installed:

- adsm.api.obj (ADSM API), approximately 750 KB
- adsm.admin.obj (ADSM Administrator GUI Client), approximately 6 MB

3.1.3 Graphical User Interface

The ADSM graphical interface has the following prerequisites:

1. X11, including the X11.base.rte file set
2. Motif 1.2

   Note: A great number of ADSM actions can be passed using the ADSM GUI. There are still actions requiring the user to issue commands from a command line. For consistency reasons we decided to always use the command line and not the GUI.

   Further information on using the ADSM GUI can be found in *ADSM Version 2 Presentation Guide SG24-4532*.

3.2 Installing ADSM

ADSM consists of 2 components:

- The ADSM server provides administrative services and server resources to ADSM clients.
- The ADSM clients (nodes) use the backup/restore services from the server.

Though there is strong interaction between those two components, they need to be considered independently.

Note: We will not deal here with the archive/retrieve services provided by ADSM.

3.2.1 Installing the ADSM Server

ADSM allows multiple servers to be defined. This section will consider the pros and cons of multiple ADSM servers before actually dealing with the install itself.

3.2.1.1 Single or Multiple Servers

If a single ADSM server is defined, all the client nodes will connect to the same server. There can be no load balancing. As large scale databases will probably be run on an RISC/6000 SP with a great number of nodes, that is, a great number of ADSM clients, it is probably useful to envisage installing more than one ADSM server.

Multiple ADSM servers can be defined on the same RISC/6000 SP node or across more than one RISC/6000 SP node. Running multiple ADSM servers on the same node allows different ADSM versions to coexist (for migration or development purposes). It also enables data isolation, when necessary, but it does not improve load balancing or improve ADSM server availability.

Running multiple ADSM servers on different nodes ensures better load balancing across the RISC/6000 SP. It enables ADSM servers to back each other up (more information on that topic can be found in *ADSTAR Distributed Storage Manager/6000 on 9076 SP2*, GG24-4499). On the other hand, it probably implies that more than one RISC/6000 SP node will have to be devoted to backups.

At first, consider only installing ADSM on one node as transferring ADSM clients from one ADSM server to another can be easily achieved in a second step.

3.2.1.2 Choosing ADSM Server Location

The following criteria should be taken into account when choosing which node(s) to use as ADSM server(s):

- The ADSM server should run on a dedicated RISC/6000 SP node especially on this environment, there will probably be enough ADSM clients to fully utilize the server node′s resources.
- The ADSM server should run on a RISC/6000 SP node with enough backup resources. These include:

  – DASDs in sufficient number to ensure good I/O throughput and enough storage space

  and/or

  – One or more tape drives according to need (more than one is recommended)

  and/or

  – One or more ESCON mainframe connections

3.2.1.3 Installing

1. As root, logon to the node selected as the ADSM server node.
2. Use `smitty installp` to select the appropriate file sets.

3.2.2 Installing the ADSM Clients

Before the installation of the ADSM Clients package, you have to make a decision on which nodes in your system will best serve as the ADSM clients.

The ADSM client should be installed on all the DB2 Parallel Edition nodes, including the catalog node or nodes.

3.2.2.1 Installing

There might be a large number of nodes where you need to install the ADSM clients and in such situation we recommend that you consider automating the installation process by referring to the procedure in the redbook captioned *Migrating and Managing data on RISC/6000 SP with DB2 Parallel Edition* SG24-4658-00 the chapter is titled *Implementing DB2 Parallel Edition on the RISC/6000 SP* in the section labelled *Installing DB2 Parallel Edition code*.

Chapter 4. Configuring ADSM and DB2 Parallel Edition

This chapter will detail the rudimentary steps necessary to configure ADSM and DB2 Parallel Edition. It is of vital importance to exercise care in carrying out these steps and in the definition of the various parameters.

4.1 Configuring an ADSM Server (dsmserv.opt)

The *dsmserv.opt* file is where all the options applied to the server should be defined.

Note: The options included in the *dsmserv.opt* file are not case sensitive.

4.1.1 Defining Communication Protocol and Parameters

The ADSM server option file (/usr/lpp/adsmserv/bin/dsmserv.opt) must define the protocol used by the ADSM server and clients.

Since we intend to use the High Performance Switch (the switch), the communication protocol will be TCPIP.

Note: Other communication protocols can be used on Ethernet or token ring (SNA, IPX, ...). More than one protocol can be defined in the dsmserv.opt file. We included the following lines to our dsmserv.opt:

```
COMMmethod TCPIP
TCPPort 32767
```

Note: The TCP port that is defined in dsmserv.opt must not be already in use or defined in /etc/services.

The default TCP port used by the ADSM server is 1500.

```
TCPWindowsize 0
```

The possible values range from 0 to 640.

Setting this parameter to 0 allows the server to adapt to the network's *Maximum Transfer Unit*.

4.1.2 Defining Database Buffer Pool

The following parameter defines ADSM's database buffer pool size, not DB2 Parallel Edition's. Its default value is set to 512 KB, which is also the minimum database buffer pool size.

```
BUFFPOOLsize 512
```

As backing up a DB2 Parallel Edition database implies backing up a small number of large files, this setting should be sufficient.

More information on database buffer pool size can be found in *Monitoring the Database Buffer Pool* in *ADSM Administrator's Reference Version 2 SH35-0135-00*.

4.1.3 Defining Recovery Log Buffer Pool

The default value for this ADSM parameter is set to 128 KB, which also is the minimum log recovery buffer size.

```
LOGPOOLsize 128
```

More information on recovery log buffer pool size can be found in *Monitoring the Recovery Log Buffer Pool* in *ADSM Administrator's Guide Version 2 SH35-0135-00*.

4.1.4 Defining the Maximum Number of Concurrent Sessions

The value given the MAXSESSIONS parameter should, at least, equal the number of client nodes to back up in parallel on a server so as not to prevent any of the client nodes from connecting to the server.

```
MAXSESSIONS     30
```

Note: A maximum number of scheduled sessions can also be defined. The MAXSESSIONS value includes the scheduled sessions (see *ADSM Installing the Server and Administrative Client*).

Note: The ADSM scheduling function is not supported with the current version of DB2 Parallel Edition at this time. More information on scheduling can be found in *ADSM Administrator's Guide Version 2,* SH35-0135-00.

4.1.5 Defining a Device Configuration File

Unless you define a device configuration file, you will need to redefine your devices every time you start a new session.

To keep your drive definitions from one session to the other, add the following line to the dsmserv.opt file:

```
DEVCONFig     /mydir/mydevconfig
```

Note: The /mydir/mydevconfig should be replaced with the complete pathname to your device configuration file name.

The full set of ADSM server options can be found in *ADSM Installing the Server and Administrative Client,* SH35-0136-00.

4.2 Starting the ADSM Server

The `installp` process defines the ADSM server to be automatically started at boot time by adding the following 2 entries to /etc/inittab:

```
adsm:2:once:/usr/lpp/adsmserv/bin/loadpkx -f
/usr/lpp/adsmserv/bin/pkmonx >/dev/console 2>&1 # Load ADSM kernel extension
autosrvr:2:once:/usr/lpp/adsmserv/bin/rc.adsmserv \
>/dev/console 2>&1 # Start the ADSM server
```

The installation process then loads the ADSM kernel extensions.

We advise not setting the automatic starting up of the ADSM server until the server has been fully configured.

To remove this entry from /etc/inittab, issue the following command:

```
sp2n15-root / -> /usr/lpp/adsmserv/bin/dsm_rmv_itab autostart
```

When the server is fully configured, issue the following command:

```
sp2n15-root / -> usr/lpp/adsmserv/bin/dsm_update_itab autostart
```

4.2.1 Starting the Server

1. Logon as root on the ADSM server node
2. Start the server from */usr/lpp/adsmserv/bin* directory:

```
sp2n15-root / -> cd /usr/lpp/adsmserv/bin
sp2n15-root /usr/lpp/adsmserv/bin -> dsmserv -F
ANR7800I DSMSERV generated at 09:25:06 on Aug 23 1995.

ADSTAR Distributed Storage Manager for AIX-RS/6000
Version 2, Release 1, Level 0.1/0.1

Licensed Materials - Property of IBM

5765-564 (C) Copyright IBM Corporation 1990, 1995. All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR7801I Subsystem (master) PID is 16510.
ANR0900I Processing options file dsmserv.opt.
ANR0990I ADSM server restart-recovery in progress.
ANR0200I Recovery log assigned capacity is 8 megabytes.
ANR0201I Database assigned capacity is 4 megabytes.
 .
 .
 . . .
ANR2803I License manager started.
ANR2835I Server is licensed for 1 clients.
ANR8200I TCP/IP driver ready for connection with clients on port 32767.
ANR2560I Schedule manager started.
ANR0993I ADSM server initialization complete.

adsm>
```

Note: The *-F* option allows overwriting of any ADSM shared memory segments from previously started and shutdown servers.

Note: To start the server from another directory, you must ensure that a dsmserv.dsk file exists in that directory. The dsmserv.dsk file is created during the server′s database and recovery log initialization.

- Only one server has been defined (default):

  The `installp` process issued the `dsmserv install` command to initialize the server′s database and recovery log. The resulting dsmserv.dsk file is put in the /usr/lpp/adsmserv/bin directory. The server must be started from there.

- More than one server has been defined:

  Each server uses its own dsmserv.dsk file created during each of the server′s database and recovery log initialization. Each dsmserv.dsk file is written by the `dsmserv install` command in the directory where it is issued from. A server can only be started from the directory where its dsmserv.dsk file resides.

The ADSM server is now up and running and using the previously defined TCP/IP port. ADSM commands can now be entered from the ADSM server console command line.

4.2.2 Stopping the ADSM Server

The server can be shut down from the ADSM command line with the HALT command.

```
adsm> halt
ANR7835I Process 18074 terminated in response to server shutdown.
ANR7835I Process 15772 terminated in response to server shutdown.
 .
 .
 .
ANR7835I Process 18902 terminated in response to server shutdown.
ANR0991I ADSM server shutdown complete.
sp2n15-root /usr/lpp/adsmserv/bin ->
```

4.3 Configuring ADSM Clients

A dsm.sys ADSM system file and a dsm.opt client option file have to be defined on each of the client nodes.

Note: The options included in the dsm.sys and dsm.opt files are not case sensitive.

4.3.1 Password Definition

An ADSM client always requires a password in order to connect to its associated server, unless the server has been configured to bypass password authentication. As can be easily understood, bypassing password authentication is not recommended.

4.3.2 ADSM Password Setting

There are some options that can be used in setting the password namely;

- Generate
- Prompt

When the generate option is used, the user does not have to remember the password but the user must execute the `dsmapipw` once.

If PASSWORDACCESS is set to prompt, you must update the password in the database configuration every time the password is changed. This implies that the user must store the password in the database configuration. Each database partition has its own configuration file and if multiple databases are used, the password should be stored on each database partition.

Add the following line to the dsm.sys file so that authentication will be automatically done on the server side:

```
passwordaccess     generate
```

Note: This option prevents ADSM clients from restoring to other ADSM clients.

When DB2 Parallel Edition uses ADSM for its backups, the client must be authenticated. The *dsmapipw* executable provided by DB2 Parallel Edition allows client authentication to be done on the client side without requiring any operator intervention thus enabling batch processing.

The *dsmapipw* executable must be run before performing any DB2 Parallel Edition backup and it is required if PASSWORDACCESS is set to generate.

The dsmapipw executable requires access to the ADSM API library (libApiDS.a). Ensure that the dsmapipw executable is calling the appropriate library as explained in 4.7, "Using the ADSM API" on page 35.

Passwords must be set on every ADSM client node (the nodes must have been registered on the ADSM server).

4.3.3 Running the Dsmapipw Executable

We can automate this process (a little) by creating a file (XYZ123) with a list of ADSM client nodes and issuing the following script as root:

```
for i in `cat XYZ123`
do
echo ″ Accessing node $i″
echo ″ Enter root′s password,″
echo ″ type : /usr/lpp/db2pe_01_01/adsm/dsmapipw ″
echo ″ Ctrl-D to exit ″
rlogin $i
done
```

A typical password definition on an ADSM node is as follows:

```
sp2n01-root / -> /usr/lpp/db2pe_01_01/adsm/dsmapipw
*************************************************************
* ADSTAR Distributed Storage Manager                        *
* API Version = 2.1.1                                       *
*************************************************************
Enter your node name (or just ENTER to use default):
Enter your current password: *****
Enter your new password: *****
Enter your new password again: *****

Your new password has been accepted and updated.
```

Note: The ADSM server must be up and running. DB2 Parallel Edition need not be up and running.

4.3.4 The ADSM System Configuration File (dsm.sys)

A sample dsm.sys.smp is provided by ADSM in /usr/lpp/adsm/bin. A sample dsm.sys.smp is also provided by DB2 Parallel Edition in $HOME/sqllib/adsm. As the dsm.sys enables multiple server definition, nothing prevents sharing this file between client nodes accessing the same or different servers. Sharing enables easier system configuration file updates.

To use the same dsm.sys file across all ADSM client nodes, you can:

1. Use the supper command to propagate the dsm.sys file.

Use the following steps:

a. Copy one of the dsm.sys.smp samples as dsm.sys.

b. Copy dsm.sys to /usr/lpp/adsm/bin/dsm.sys on the control workstation.

c. Include /usr/lpp/adsm/bin/dsm.sys to the /var/sysman/sup/lists/user.admin file collection.

d. Use the supper command to propagate File collections. Propagation can be forced by issuing the following command from the control workstation:

```
dsh -a /var/sysman/supper update user.admin
```

2. Use NFS filesystem as follows:

a. Copy one of the dsm.sys.smp samples as dsm.sys to an NFS filesystem shared by all the ADSM client nodes.

Note: The default will prevent root from writing to an NFS filesystem on any node but that where the filesystem physically resides.

b. Logon to the appropriate node.

From the control workstation, as root:

- Start SMIT (or smitty).
- Select **9076 SP Configuration Database Management**.
- Select **9076 SP Users**.
- Select **Change/Show Characteristics of a User**.
- Enter the user′s name.
- The node where his home directory was physically created in mentioned in the HOME directory field.

c. Move the *dsm.sys* to an NFS mounted filesystem such as the home directory of the instance owner of the DB2 Parallel Edition database using the ADSM services.

Note: If you are using more than one DB2 Parallel Edition instance, make sure that they all can access the chosen directory and that they have read access to the *dsm.sys* file.

d. Link the dsm.sys file to /usr/lpp/adsm/bin/dsm.sys on all the client nodes. Some ADSM programs will look for the dsm.sys file in /usr/lpp/adsm/bin and nowhere else. A link is therefore necessary.

```
sp2n15-root / > dsh ″ln -fs /u/dbusr/dsm.sys /usr/lpp/adsm/bin/dsm.sys″
```

Note: The Working Collectives (WCOLL) environment variable should have been set to include all nodes on which ADSM clients are defined. The WCOLL is used to specify the filename of the file containing the nodes you wish to work on. It is mostly used with the Distributed Shell (dsh) command.

e. The root user should be the only user to have write access to the dsm.sys file. Check that it is the case and eventually change the file′s access authorizations from the node where the DB2 Parallel Edition instance owner′s $HOME has been physically created.

```
sp2n15-root /u/dbusr -> chown root.system dsm.sys
sp2n15-root /u/dbusr -> chmod 644 dsm.sys
```

The following options, at least, have to be set:

1. SErvername

   This definition, apart from defining the servername, also indicates the beginning of a server definition stanza. The stanza will end with a new SErvername definition or the end of the dsm.sys file.

   ```
   SERvername db2pe
   ```

2. COMMmethod

   It is possible to define communication protocols other than TCP/IP. A complete list of supported protocols, including their related parameters can be found in *ADSM Getting Started with ADSM AIX Clients,* GG24-4243.

   ```
   COMMmethod TCPIP
   ```

3. TCPPort

   This parameter only applies to the TCP/IP protocol. An unused port should be defined.

   ```
   TCPPort 32767
   ```

   Note: The port number value should be kept within 32767.

4. TCPServeraddress

   This parameter associates a server name, as defined in the SERVername definition of the current stanza, to a server′s TCP/IP address. Defining the server node′s switch TCP/IP address will ensure that backups and restores use the switch.

   ```
   TCPSErveraddress sp2sw15.itsc.pok.ibm.com
   ```

   Note: All valid TCP/IP address formats can be used (9.12.0.52, sp2sw15, ...).

Note: More options can be set. Some performance related ones are described in 8.1.2, “ADSM Performance Parameters” on page 78. The complete list can be found in *ADSM Getting Started with ADSM AIX Clients*.

4.4 Licensing

Out-of-the-box ADSM is licensed for one client.

4.4.1 Checking Your System′s ADSM Licensing

Ensure that your system is properly licensed.

ADSM defines the following set of licensing options:

Single client
5 clients
10 clients
50 clients
Environment Support (UNIX)
Environment Support (Desktop)
Space Management Support
Device Support Module 1

Device Support Module 2
Device Support Module 3
Device Support Module 4
Device Support Module upgrade: Module 1 to 2
Device Support Module upgrade: Module 2 to 3
Device Support Module upgrade: Module 3 to 4

In the following example, licensing action is required:

```
adsm> audit license
ANR2817I AUDIT LICENSES: License audit started as process 16.
ANR2825I License audit process 16 completed successfully - 3 nodes audited.
ANR2841W Server is NOT IN COMPLIANCE with license terms.
```

In the following example, no licensing action is required:

```
adsm> audit license
ANR2817I AUDIT LICENSES: License audit started as process 17.
ANR2825I License audit process 17 completed successfully - 3 nodes audited.
ANR2811I Audit License completed - Server is in compliance with license terms.
```

4.4.2 Modifying Your System′s ADSM Licensing

Should you want to use 6 clients, you will have to install the *5 clients* option to the initial ADSM client.

From the ADSM command line:

```
adsm > register license d345143773124ce06c9f734bc1d7fdea432b
ANR2852I Current license information:
ANR2835I Server is licensed for 1 clients.
ANR2853I New license information:
ANR2835I Server is licensed for 6 clients.
```

Note: The license registration code shown above is only an example. It is not a valid license.

4.5 Registering ADSM Clients

Client nodes must be registered with the server before they can actually use the server.

ADSM defines 2 types of registration policies that are defined at ADSM server level:

- Closed registration

  All client nodes must be registered by the server before attempting to connect to the server. An error message will be issued by the server if an unregistered client attempts connection.

  Issue the following ADSM command to register a client node from the server:

  ```
  adsm> Register node sp2sw05 sp2
  ANR2060I Node SP2SW05 registered in policy domain STANDARD.
  ```

sp2sw05 represents the client node name, sp2 is the client node′s password.

- Open registration

  Unregistered clients will be prompted for registration by the server on their first connection attempt. No registration command need to be issued from the server side.

Note: One way or the other, ADSM clients must already be registered before you can successfully issue the use adsm parameter of the DB2 Parallel Edition `backup` command.

Passwords are defined during registration but DB2 Parallel Edition backups or restore will require a further level of password setting.

4.5.1 ADSM Client Option File (dsm.opt)

A sample dsm.opt.smp is provided by ADSM in /usr/lpp/adsm/bin. A sample dsm.opt.smp is also provided by DB2 Parallel Edition in $HOME/sqllib/adsm.

Copy one of the dsm.opt.smp samples and edit the copy as dsm.opt.

The dsm.opt file should only be modified by the root user or by members of the DB2 Parallel Edition group. Check that it is the case and eventually change the file′s access authorizations.

```
sp2n15-root /u/dbusr -> chown root.dbgrp dsm.opt
sp2n15-root /u/dbusr -> chmod 664 dsm.opt
```

The following options, at least, have to be set:

1. SErvername

   The server name defined here should correspond to one of the server names defined in the server′s option file (dsmserv.opt)

   ```
   SErvername db2pe
   ```

   Note: More options can be set. Some performance related ones are described in 8.1.2, “ADSM Performance Parameters” on page 78. The complete list can be found in *ADSM Getting Started with ADSM AIX Clients*.

4.5.1.1 Single or Multiple Client Option Files

If more than one ADSM server has been defined, you will probably want to distribute the clients over the various servers. This means that not all clients should access the same dsm.opt file.

By default, the path and file name used by ADSM as the client option file is specified in the DSM_CONFIG environment variable. It can be any filename.

- If you want to group clients according to the ADSM server they will connect to, all the clients in that group should use the same client option file.

  1. Choose a unique identifier for each group
  2. Create a client option file called dsm.opt.gr_id where gr_id is the chosen group identifier
  3. Set DSM_CONFIG to the full pathname of this new client option file f or all the clients in a group.
  4. Repeat the two previous steps for each group

For example, this can be achieved by including in the *.profile* the following lines:

```
DSM_CONFIG=/mydir/dsm.opt
 case `hostname` in
(client1 | client2 | client3)
       GROUP=group1 ;;
(client4 | client5 | client6)
       GROUP=group2 ;;
(client7 | client8 | client9)
       GROUP=group3 ;;
 esac
export DSM_CONFIG=${DSM_CONFIG}.$GROUP
```

Replace client1, client2, ... with the different hostnames of the ADSM clients, /mydir with the appropriate pathname, and group1, group2, ... with your group identifiers.

- Move your dsm.opt files so that they correspond to your ADSM client′s DSM_CONFIG.

4.6 Configuring DB2 Parallel Edition for ADSM

DB2 Parallel Edition users will need to define a set of environment variables and set the necessary passwords in order to use ADSM.

4.6.1 Editing db2profile

Edit the following environment variables defined in DB2 Parallel Edition′s db2profile:

- DSMI_CONFIG

  This variable indicates the full path to the client′s option file.

  The default value is:
  DSMI_CONFIG=$HOME/sqllib/adsm/dsm.opt where $HOME is the DB2 Parallel Edition instance owner′s home directory.

  Ensure that your DSMI_CONFIG is pointing to an existing dsm.opt file (see 4.5.1.1, "Single or Multiple Client Option Files" on page 33).

- DSMI_DIR

  This variable indicates where the messages file (dsiameng.txt) is located. *dscameng.txt* replaces the *dsiameng.txt* that existed in version 1.

  The default value is:
  DSMI_DIR=$HOME/sqllib/adsm

- DSMI_LOG

  This variable indicates where the error log (dsierror.log) is created.

  The default value is:
  DSMI_LOG=$HOME

  You may want to change this setting to an instance independent directory. This directory must exist and be accessible by the DB2 Parallel Edition instance(s) owner.

Note: Ensure that dsmapitca is placed in the */usr/lpp/adsm/bin* directory (dsmapitca replaces the dsmtca that existed in version 1). It should also have root previlage.

Note: When db2profile is updated the `db2stop` and `db2start` commands should be executed to use the values.

4.6.2 Single or Multiple ADSM Server Configurations

If you want groups of DB2 Parallel Edition nodes to access different ADSM servers, refer to 4.5.1.1, “Single or Multiple Client Option Files” on page 33. Note that the environment variables used by all ADSM API and DB2 Parallel Edition to access ADSM services start with DSMI_ whereas those used to directly access ADSM services start with DSM_. Ensure that the values in the .profile file are reflected in the db2profile.

4.7 Using the ADSM API

Some utility programs, adsmqry (see 5.1.1, “Online DB2 Parallel Edition Backup” on page 43), dsmapipw (see 4.3.1, “Password Definition” on page 28), db2uexit (see 5.2.1.2, “User Exit Program” on page 48) are calling some of the ADSM APIs contained in the ADSM libApiDS.a library. These programs excluding the *dsmapipw* need to be compiled/re-compiled using the following ADSM ″include″ files:

- dsmapitd.h: definitions for ADSM dsm external constants.
- dsmapifp.h: ADSM dsm API function definitions.
- dsmrc.h: return codes from ADSM dsm APIs.

You should ensure that these programs are using the ADSM APIs that can be found in /usr/lpp/adsm/api/bin/libApiDS.a.

The libApiDS.a library as well as the ADSM include files are provided both by ADSM and DB2 Parallel Edition products:

- in /usr/lpp/db2pe_01_01/adsm for DB2 Parallel Edition
- in /usr/lpp/adsm/api/bin for ADSM

The latest level should be used.

To ensure consistency in the API level used by ADSM and DB2 Parallel Edition we suggest running the following commands.

4.7.1 Library Version Level

Compare the version/release/level of the 2 libraries by running:

```
grep -i DSM_API /usr/lpp/db2pe_01_01/adsm/dsmapitd.h

#define DSM_API_VERSION     1
#define DSM_API_RELEASE     2
#define DSM_API_LEVEL        5

grep -i DSM_API /usr/lpp/adsm/api/bin/dsmapitd.h

#define DSM_API_VERSION     2
#define DSM_API_RELEASE    1
#define DSM_API_LEVEL       1
```

Our example shows that the APIs provided by ADSM are at the latest level, therefore we should be using them.

4.7.2 Ensuring API Consistency

Once you have determine which ADSM APIs you will be using (either the ones provided by ADSM product or the ones provided by DB2 Parallel Edition product) make sure the APIs provided by DB2 Parallel Edition are pointing to the latest level. In our example, we did so by issuing the following link commands:

1. libApiDS.a

```
dsh ″ ln -fs /usr/lpp/adsm/api/bin/libApiDS.a /usr/lpp/db2pe_01_01/adsm/libApiDS.a″
dsh ″ ln -fs /usr/lpp/adsm/api/bin/libApiDS.a /usr/lib/libApiDS.a″
```

2. dsmapitca

```
dsh ″ ln -fs /usr/lpp/adsm/bin/dsmapitca /usr/lpp/db2pe_01_01/adsm/dsmapitca″
```

3. dsmapitd.h,dsmapifp.h, dsmrc.h

```
dsh ″ ln -fs /usr/lpp/adsm/api/bin/dsmapitd.h /usr/lpp/db2pe_01_01/adsm/dsmapitd.h″
dsh ″ ln -fs /usr/lpp/adsm/api/bin/dsmapifp.h /usr/lpp/db2pe_01_01/adsm/dsmapifp.h″
dsh ″ ln -fs /usr/lpp/adsm/api/bin/dsmrc.h /usr/lpp/db2pe_01_01/adsm/dsmrc.h″
```

4. dscameng.txt

```
dsh ″ln -fs /usr/lpp/adsm/bin/dscameng.txt /usr/lpp/db2pe_01_01/adsm/dscameng.txt″
```

In order to verify consistency, we can issue the following command:

```
dsh ″ cd /usr/lpp/db2pe_01*/adsm ; ls -al ″
```

4.8 Disk Storage Pools

Storage pools are logical containers client nodes back up to. Defining a storage pool per client ensures that each client writes to its own set of files. It also facilitates I/O optimization on the server node.

1. Issue the following def stgpool ADSM command to define a disk storage pool:

```
adsm> def STGpool nod1stg DISK
ANR2200I Storage pool NOD1STG defined (device class DISK).
```

2. Issue the q stgpool ADSM command to find out which disk storage pools have been defined and how much space they have.

4.8.1 Disk Storage Pool Volumes

Disk storage pool volumes are special files to be grouped in storage pools.

Before a pool volume can be added to a defined storage pool, you must:

- Decide on the physical location of the volume to be added
- Format the pool volume

4.8.1.1 Choosing Disk Location for New Pool Volumes

As ADSM only writes to one of the pool volumes defined in a storage pool at a time, there is no point in spreading pool volumes across disks.

On the other hand, you might want to use different disks for pool volumes belonging to different storage pools. Careful planning and designing will optimize I/O throughput during concurrent backups to different storage pools (see 5.1.3, “Scenario Using ADSM and DB2 Parallel Edition” on page 45).

4.8.1.2 Formatting Pool Volumes

All disk pool volumes must be formatted beforehand.

Note: Beforehand formatting applies to disk storage pool volumes but also to database or recovery log volumes.

Formatting is done through the dsmfmt command issued from the AIX command line. dsmfmt supports sequential multiple storage pool volume formatting. Multiple parallel volume formatting can be achieved by issuing more than one single volume formatting commands at the same time but this is really interesting only if the different volumes are defined on different disks.

To format a 4 MB storage pool volume called stg1.vol in /db/vol_client1:

```
sp2n15-root / -> dsmfmt -m -data /db/vol_client1/stg1.vol 4
ADSTAR Distributed Storage Manager/6000
AIX ADSM Server DSMFMT Extent/Volume Formatting Program

Licensed Materials - Property of IBM

5765-564 (C) Copyright IBM Corporation 1990, 1995.  All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

Allocated space for /db/vol_client1/stg1.vol: 4194304 bytes
```

Note: Replacing the -data option with -db will format a database pool volume. Using the -log option will format a recovery log pool volume. Database or log volumes have minimum respective sizes of 5 MB and 9 MB. They are allocated in multiples of 4 MB, plus an additional 1 MB.

Note: The dsmfmt command will not format files bigger than 2 GB even if the files are to be written to a type of file system not limited to 2 GB files (PIOFS, for example).

4.8.1.3 Allocating New Storage Pool Volumes

1. To add a (formatted) storage pool volume to a defined storage pool:

   ```
   adsm> define volume NOD1POOL /db/vol_client1/stg1.vol acc=READWrite
   ANR2206I Volume /db/vol_client1/stg1.vol defined in storage pool
   NOD1POOL (device
   class
   DISK).
   ANR1305I Disk volume /db/vol_client1/stg1.vol varied online.
   ```

   Note: There is no limit to the total size of the volumes defined in a storage pool. It may exceed 2 GB, thus allowing disk backups of DB2 Parallel Edition nodes larger than 2 GB.

2. Issue the `q volume` ADSM command to find out what volume pools are and to which storage pool they belong.

4.8.2 Associating a Disk Backup Storage Pool to a Client Node

Rather than starting from scratch we are going to copy and modify some already existing and configured ADSM objects. For more information on ADSM object definition, refer to *ADSM Administrator′s Reference Version 2 SH35-0135*.

1. Domains establish connection possibilities between ADSM clients and storage pools (and thus with storage pool volumes). Copy the STANDARD domain and give it a new domain name:

```
adsm> copy domain standard nod1disk
ANR1503I Policy domain STANDARD copied to domain NOD1DISK.
```

Copying the fully operational STANDARD domain also copies all the other ADSM objects that are associated with the defined STANDARD domain. We may now only update the copied objects and change some of their attributes instead of having to create them from scratch.

2. Associate a storage pool to the previously defined domain:

```
adsm> update copygroup nod1disk standard standard dest=nod1stg
ANR1532I Backup copy group STANDARD updated in policy domain NOD1DISK, set
STANDARD, management class STANDARD.
```

Note: The dest parameter must refer to the storage pool defined in step 4.8, “Disk Storage Pools” on page 36.

Activate the new definition:

```
adsm> activate policyset nod1disk standard
ANR1514I Policy set STANDARD activated in policy domain NOD1DISK.
```

Associate the domain to a ADSM client node:

- If the client node is not registered:

```
adsm> register node nod1client xxxx domain=nod1disk
ANR2060I Node NOD1CLIENT registered in policy domain NOD1DISK.
```

where nod1client is the ADSM client′s hostname, *xxxx* . represents the client′s password, and nod1disk is the previously defined domain name.

- If the client node is already registered:

```
adsm> update node nod1client domain=nod1disk
ANR2063I Node NOD1CLIENT updated.
```

where nod1client is the ADSM client′s hostname, and nod1disk is the previously defined domain name.

An ADSM backup requested by the client whose hostname is nod1client will use the formatted files defined as part of the nod1stg storage pool.

4.9 Backing Up Data to Devices

In this section, we will exploit the use of external devices to perform database backup. One of such device is the IBM magnetic tape device the 3490E which is Small Computer Standard Interface (SCSI) attached to our RISC/6000 SP system on our ADSM server node (sp2n15).

4.9.1 Installing the IBM 3490E Device Driver

1. Check that the IBM AIX Enhanced Tape and Medium Changer (Atape) device driver has been installed on the ADSM node:

   ```
   sp2n15-root / -> lslpp -h | grep Atape
     Atape.driver
   ```

2. If needed, install the Atape driver.

   The Atape device driver is provided with your IBM 3490E Tape Drive. The standard AIX 3490 Tape Driver is not supported by ADSM.

Once the Atape device driver is installed, you can define and use your tape drive as you would any other tape drive on your system.

Parameters specific to the 3490E Tape Storage Subsystem using the Atape device driver can be found in *ADSTAR Distributed Storage Manager/6000 on 9076 SP2 GG24-4499-00*.

4.9.1.1 Licensing

The IBM 3490E Tape Storage Subsystem requires additional ADSM licensing.

From the ADSM command line:

```
adsm> register license xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
ANR2852I Current license information:
ANR2835I Server is licensed for 1 clients.
ANR2853I New license information:
ANR2835I Server is licensed for 1 clients.
ANR2854I Server is licensed for device support module 2.
```

The xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx is not a valid code. You must use a valid license code for the Device Support Module 2 or the Device Support Module Upgrade: Module 1 to 2 (if Device Support Module 1 is already installed).

Refer to 4.4, "Licensing" on page 31 for further licensing information.

4.9.2 Defining a Tape Storage Pool

Below are the steps you should follow to define the IBM 3490E tape drive to ADSM in order to use it as an ADSM backup storage device.

1. Before you can use the IBM 3490E tape drive you must first define a library to which the drive will belong.

   The IBM 3490E Model E11, not being an automatic tape library, you must define the libtype as manual.

   ```
   adsm> define library 3490lib libtype=manual
   ANR8400I Library 3490LIB defined.
   ```

Note: A different name can be used as library name.

2. Issue the q library ADSM command to verify your library definition.

3. Check for the 3490E entry in the output of the lsdev AIX command:

```
sp2n15-root / -> lsdev -Cc tape
rmt0 Available 00-03-00-0,0 IBM SSD 3490E Tape Drive
```

4. Associate that device to the previously defined library.

```
adsm> def drive 3490lib 3490drv device=/dev/rmt0
ANR8404I Drive 3490DRV defined in library 3490LIB.
```

5. Define a device class to associate a type of storage medium (disk, cartridge, 8mm, ...) to the previously defined library.

```
adsm> def devclass 3490class devtype=cartridge format=3490c library=3490lib
ANR2203I Device class 3490CLASS defined.
```

6. Issue the q devclass ADSM command to verify your device definition.

7. A storage pool must now be associated to the previously defined device class.

```
adsm> def stgpool 3490pool 3490class maxscratch=200
ANR2200I Storage pool 3490POOL defined (device class 3490CLASS).
```

   The maxscratch parameter defines the maximum number of labeled tapes that can be dynamically allocated to a storage pool.

8. Repeat steps described in 4.8.1, “Disk Storage Pool Volumes” on page 36.

4.9.3 Defining Additional Storage Pool Volumes

4.9.3.1 Labeling Tapes

Each tape must be labelled *beforehand.*

No two tapes must have the same label.

dsmlabel: Tapes must be labelled using the dsmlabel command found in /usr/lpp/adsmserv/bin.

To give a tape the label MINE1, issue the following command on the AIX command line:

```
sp2n15-root /usr/lpp/adsmserv/bin -> dsmlabel -drive=/dev/rmt0 -overwrite

ADSTAR Distributed Storage Manager for AIX-RS/6000
Volume Label Utility Program
Version 2, Release 1, Level 0.1/0.1

Licensed Materials - Property of IBM

5765-203 (C) Copyright IBM Corporation 1994, 1995.  All rights reserved.
U.S. Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corporation.

ANR9722I Insert a new volume in drive ’/dev/rmt0’, then enter the name (1-6 char
acters) to be used for its label; or just press ENTER to quit this program:
MINE1
ANR9743I Attempting to label volume ’MINE1’ using drive ’/dev/rmt0’...
ANR9720I Volume ’MINE1’ was labeled successfully using drive ’/dev/rmt0’.

ANR9722I Insert a new volume in drive ’/dev/rmt0’, then enter the name (1-6 char
acters) to be used for its label; or just press ENTER to quit this program:
```

Note: The -overwrite parameter allows overwriting of a previously defined label. Use with care.

You must have prepared enough labeled tapes before starting a backup as you cannot easily label additional tapes once the backup has started.

Scratch or Private

1. Private tapes

   A private tape is a labelled tape which has been defined as belonging to a storage pool volume. Private tapes can only be used by clients defined as using that storage pool for backups or archive.

   To define a R/W private tape labelled AAA in the tape storage pool we have already defined:

   ```
   adsm> define volume 3490pool aaa acc=readwrite
   ANR2206I Volume AAA defined in storage pool 3490POOL (device class 3490CLASS).
   ```

2. Scratch tapes

   A scratch tape is a labelled tape which has not been defined in any storage pool. Any client needing it can write to it, provided the storage pool the client is using has not yet used a number of scratch tapes equal to its maxscratch value.

If the maxscratch parameter of the def stgpool has been given value 0, that storage pool will only be able to use private tapes. The tape storage pool volumes must then be defined in that tape storage pool before it can be used to backup to.

4.9.4 Preventing AIX Users from Writing over ADSM Tapes

Note: Nothing prevents a user from using the 3490E tape drive. A user, issuing a `tar`, `cpio` or `dd` command will overwrite backups written by ADSM.

4.9.5 Configuring the Client′s Option File for Tape Backups

When using DB2 Parallel Edition and ADSM, it is necessary to modify a client′s dsm.opt file for that client to be able to backup to tape.

1. Add the following line to the client′s dsm.opt file

   ```
   tapeprompt        no
   ```

   Note: You will get an error message if the previous line has not been included to the ADSM client′s dsm.opt file.

Chapter 5. Database Backup

When considering doing a database backup, there are some factors that could influence the type of solution that will be utilized to implement the backup operation. In our environment consisting of RISC/6000 SP and DB2 Parallel Edition, we could not afford to attach a tape drive to each node on our system and this could be true for other commercial environments. It become apparent that one solution among others that could be used for this operation is ADSM and other kinds of solutions that include Client Input Output/Sockets (CLIO/S). With the use of ADSM or CLIO/S we can back up very large files that are greater than 2 GB, which AIX does not currently support. In this chapter we will focus on using ADSM for performing DB2 Parallel Edition database backup. For details on utilizing CLIO/S refer to Chapter 7, “Using CLIO/S for Backup and Restore to Mainframe Tape” on page 67.

5.1 DB2 Parallel Edition Backup Utilities

This section will not deal with defining the different types or advising on backup policies as this is already covered in *DB2 Parallel Edition for AIX Administration Guide and Reference*, SC09-1982.

Incremental backups are also not covered in this section as DB2 Parallel Edition does not yet allow them. DB2 Parallel Edition only supports backups at database level.

Whenever possible, try not to include any database user data on that same database′s catalog node. This will reduce database unavailability during backups (and restores).

5.1.1 Online DB2 Parallel Edition Backup

DB2 Parallel Edition supports online and offline backups of the database. Offline backups require an exclusive lock to be placed on each node to be backed up and can only be performed if nobody is using the database. Online backups allow applications to access the database while the backup is taken. For this reason, there is a possibility that the restore image produced by an online backup may be inconsistent. Logs will be required to rollforward from this online backup to a point in time or to completion.

In order to be able to do an online backup, the LOGRETAIN and/or USEREXIT parameters must be set on.

An offline backup is needed before first database connection after the *LOGRETAIN* and/or *USEREXIT* parameters have been set on.

5.1.2 Scenario Using DB2 Parallel Edition Utilities

DB2 Parallel Edition backup is done at node level. Each node writes its backup file locally.

- If you are backing up to disk, this means that the necessary disk space must be available locally. As each database node backup is written to a single file, the AIX 2 GB file size limitation must be kept in mind.

By default, DB2 Parallel Edition will write its backup file to the directory where the DB2 Parallel Edition backup command is issued from. An explicit backup directory location must be added in order to back up to a different location.

```
BACKUP DATABASE sample TO /mydir
```

Backup files can be written to an NFS directory. Backup files follow a well-defined format that includes the node number (as defined in the *db2nodes.cfg* file). The backup files from two different nodes will then, at least, differ by the node number.

Note: *ALAIN.dbusr.N0C1.19951126194118.001* is a valid backup file name where:

- *ALAIN* is the database alias
- *dbusr* is the instance name
- *N0C1* indicates that the backup was done on node 0 (as defined in *db2nodes.cfg*) and that the database′s catalog node is node 1 (also as defined in *db2nodes.cfg*)
- *19951126194118* is the time stamp following the *yyyymmddhhmmss* format
- *001* is the sequence number
- If you are backing up to tape, it means that the tape drive must be accessible locally. This requires a tape drive to be attached to each of the DB2 Parallel Edition nodes (including the catalog node).

  Explicit backup directory location is required:

  ```
  BACKUP DATABASE sample TO /dev/rmt0
  ```

  Nothing will prevent you from overwriting any data previously written to that tape.

  There is no automatic backup management. You must keep track of node backup locations. Needless to say that this can be a demanding and tiresome task in a complex environment.

- Use the following steps:

  1. Determine which node has been defined as your database catalog node. Issue the following command from one of the nodes on which your database has been defined:

     ```
     sp2n02-dbusr /u/dbusr -> db2 list database directory
      System Database Directory

      Number of entries in the directory = 1

     Database 1 entry:

      Database alias                 = SAMPLE
      Database name                  = SAMPLE
      Local database directory       = /db
      Database directory             =
      Node name                      =
      Database release level         = 7.00
      Comment                        =
      Directory entry type           = Indirect
      Authentication                 = SERVER
      Catalog node number            = 10
     ```

The output of the list database directory may include more than one database entry. Identify the catalog node number for the database you want to back up.

2. Backup the catalog node.

   ```
   export RAHCHECKBUF=no
   db2_all ″<<+10< db2 BACKUP DATABASE sample TO device″
   ```

   Replace device with the appropriate path and file name or device name according to the type of media you are backing up to.

   The <<+x< option is used to specify that a command is to be sent to only one node. The *x* must be replaced with the node′s DB2 Parallel Edition number (10, in our example).

   DB2 Parallel Edition sets an exclusive connection on the catalog node. As connection to a node other than the catalog node also requires a connection to the catalog node, no other node backup can take place while the catalog node is being backed up. The database is unavailable on all nodes.

   Keeping the catalog node dataless will ensure a minimal database unavailability during database catalog node backup.

   RAHCHECKBUF=no prevents buffer allocation verification to be checked on all defined hosts before the command is actually sent to the nodes.

   Note: More information on *db2_all* can be found in *$HOME/sqllib/rahREADME*.

   You must wait until the catalog node back up is over before you can backup (or even connect to) the database on any other node. An error message will result from attempting concurrent DB2 Parallel Edition catalog and non-catalog nodes.

3. Backup all other nodes in parallel. Each node backup sets an exclusive connection to the node it is running on and a shared connection to the catalog node.

   ```
   export RAHCHECKBUF=no
   db2_all ″;<<-10< db2 BACKUP DATABASE sample TO device″
   ```

   The *< < - x <* option is used to specify that a command is not to be sent to a particular node. The *x* must be replaced with the node′s DB2 Parallel Edition number (10, in our example).

   The *;* option is used to specify that a command must be sent to all concerned nodes in parallel.

5.1.3 Scenario Using ADSM and DB2 Parallel Edition

The main difference with the previous scenario is that DB2 Parallel Edition backups on the nodes no longer write a backup file on the local node. The file is sent through the ADSM client to the ADSM server on the ADSM server node.

1. Back up the catalog node. The database is unavailable on all nodes.

   ```
   export RAHCHECKBUF=no
   db2_all ″<<+10< db2 BACKUP DATABASE sample USE ADSM″
   ```

2. Back up all other nodes in parallel.

   All database non-catalog nodes may potentially be backed up concurrently but the gain obtained from parallelization must be properly assessed.

   a. If you are backing up to tape, you should not start a greater number of concurrent non-catalog node backups than the number of tape drives that are attached to the ADSM server node. Furthermore, no two concurrent backups should write to the same tape drive.

      Not following that advice would probably reduce your database availability.

      For example, if you back up in parallel nodes 1 to 5 to the same tape drive, nodes 1 to 5 will connect to the database, each putting an exclusive lock on its part of the database. Node 1 will start writing to the tape but node 5 will have to wait until all the other nodes are done before actually starting to write to the tape. It will maintain its lock 5 times longer than necessary, thus possibly preventing user access to tables defined in nodegroups including node 5.

   b. If you are backing up to disk, you should not start a greater number of concurrent non-catalog node backups than the number of physical disks present on your ADSM server. No two concurrent non-catalog node backups should write to storage pool volumes writing to the same physical disk.

      Not following that advice would probably reduce your database availability, as explained in the previous example.

```
export RAHCHECKBUF=no
db2_all ";<<-10< db2 BACKUP DATABASE sample USE ADSM"
```

5.1.4 Scenario Using Database File Back Up without DB2 Parallel Edition Utilities

Though it is not recommended to use commands such as *tar*, *cpio* or *dd* to backup offline DB2 Parallel Edition databases, it is still a possibility that can be used in case of absolute necessity.

Ensure that all users are disconnected and verify that the database is in a consistent state by issuing the following command as a db2 user on each database node;

```
db2 get database configuration for XXXX
```

where XXXX is the name of the database.

It is required that DB2 Parallel Edition be stopped as nothing would otherwise prevent users from connecting to the database or to another database in the same instance.

Back up the instance owner′s home directory and the directory in which your database has been created. If for example, your database has been defined in the dbusr instance under /db, you should back up /db/dbusr.

Do not forget to also include the directory where your log files are being written, if you are not using the default log directory.

Needless to say that no backup management will be done. It is your responsibility to make sure that consistent backup files are restored to the appropriate nodes.

Note: The same restrictions apply to database volume group or logical volume level backups as in file level backups.

5.2 DB2 Parallel Edition Log Achieving

Recoverable databases have either the logretain or userexit (or both) database parameter turned on. This allows for active and archived logs to be kept and results in the ability for the database to have roll-forward recovery. This policy of keeping the logs is called log-retention logging as opposed to circular logging, for which logs are being reused in a circular manner. Circular logging is always the default policy being used when a database is created. Once the database has been created, the logging policy for the database can be changed by modifying some of the database configuration parameters as explained below (see 5.2.1, “DB2 Parallel Edition Logretain Parameter”). In this chapter we will only focus on log-retention logging, further details on circular logging can be found in *DB2 Parallel Edition for AIX, Administration Guide and Reference*, SC09-1982.

The following database configuration parameters allow you to set the type of policy logging being used for the considered database.

5.2.1 DB2 Parallel Edition Logretain Parameter

If this parameter is enabled, log-retention logging is performed. As new logs get allocated the old logs are kept in the current directory used for logging. Since the old logs are not moved away, the disk space they occupy is never reused, and if not carefully planned a disk-full condition may occur. If this happens, database processing will stop.

You can enable the logretain parameter by issuing the following command from the DB2 Parallel Edition command line processor (CLP):

```
db2 ″update database configuration for my-database using LOGRETAIN ON″
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  All applications must disconnect from this database before the
changes become effective.
```

This can also be combined with the db2_allcommand for all the nodes.

5.2.1.1 DB2 Parallel Edition Userexit Parameter

If this parameter is enabled, log retention logging is performed regardless of how the logretain parameter is set. This parameter also indicates that a user exit program should be used to archive and retrieve the log files. Log files are archived when the database manager closes the log file. You can enable the userexit parameter by issuing the following command from the DB2 Parallel Edition command line processor (CLP):

```
db2 ″update database configuration for my-database using USEREXIT ON″
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  All applications must disconnect from this database before the
changes become effective.
```

This can also be combined with the db2_allcommand for all the nodes.

5.2.1.2 User Exit Program

Before enabling the userexit database parameter, you should provide the database manager with a user exit program.

User Exit Setup: The user exit sample programs for UNIX-based environments are installed in the $HOME/sqllib/samples/c directory of the instance owner. These samples consist of two sections: archive and retrieve. While the samples are coded in the C language, your user exit program can be written in a different programming language. These samples are self-documented, and can be used as such or as a model to build a user exit program.

The following rules have to be followed when using a user exit program:

- Only one user exit program can be invoked within the database manager instance. Therefore, each user exit program must have sections for all the actions it may need to perform, including archive and retrieve. One of the parameters passed to the user exit program indicates which of these actions is requested.
- The user exit program must be an executable file whose name is *db2uexit*.
- The user exit executable must be put into the $HOME/sqllib/adm directory of the instance owner to be found when called by the database manager.

In the RISC/6000 SP environment, make sure you put the user exit executable on the node you have decided to use for the physical creation of the home directory of the instance owner (9076 user), for example:

```
NODE:/home/instance_owner/sqllib/adm

    This is equivalent to the $HOME/sqllib/adm
```

In the following sections, we have chosen to use a user exit utilizing ADSM API′s to archive and retrieve the DB2 Parallel Edition log files.

Follow the steps below to setup your user exit program:

1. Copy db2uexit.cadsm to db2uexit.c and place this file into a working directory.
2. Modify the Installation Defined Variables to suit your environment. The following variables can be changed:
   - AUDIT_ACTIVE: you may want an audit trail of the user exit calls. To do so, you must enable audit logging by setting AUDIT_ACTIVE to 1 as follows:

     ```
     #define AUDIT_ACTIVE              1
     ```

     You can disable audit logging by setting AUDIT_ACTIVE to 0.

     Then the archive requests will be traced in a file named ARCHIVE.LOG.NODE*nnnnn* located in the audit and error file path, where *nnnnn* is five digits representing the node number.

The retrieve requests will be traced in a file named RETRIEVE.LOG.NODE*nnnnn* located in the audit and error file path, where *nnnnn* is five digits representing the node number.

- ERROR_ACTIVE: you may want an error trail of the user exit calls. To do so, you must enable error logging by setting ERROR_ACTIVE to 1:

  ```
  #define ERROR_ACTIVE              1
  ```

  You can disable error logging by setting ERROR_ACTIVE to 0.

  The errors will be traced in a file named USEREXIT.ERR located in the audit and error file path.
- AUDIT_ERROR_PATH: Path where audit and error logs will reside.

  ```
  #define AUDIT_ERROR_PATH       ″/u/dblogusr/″
  ```

3. Compile and link ″db2uexit.c″ with the following command (substitute ″instance″ with the userid of the instance owner):

   ```
   cc -o db2uexit db2uexit.c -I/u/instance/sqllib/adsm \
      /u/instance/sqllib/adsm/libApiDS.a              \
      -L/usr/lpp/adsm/api/bin:/usr/lpp/db2pe_01_01/adsm
   ```

4. Place the resultant ″db2uexit″ named executable into the sqllib/adm directory.

Enable User Exit: In order to enable *log retain* policy and the use of a *user exit program* to automatically archive the logs when they become full, the *userexit* database parameter must be turned on. With the user exit enabled, roll-forward recovery is allowed for the database.

```
db2 update database configuration for michel using USEREXIT ON
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  All applications must disconnect from this database before the
changes become effective.
```

After *logretain*, or *userexit*, or both of these parameters are enabled, a full backup of the database must be performed before the database can be accessed again. This state is indicated by the *backup_pending* informational parameter displayed by the *get database configuration for my-database* command.

```
  ...
  ...
 Database State                                       = 33
   Database is consistent                             = YES
   Backup pending                                     = YES
   Rollforward pending                                = NO
   Log retain for recovery status                     = OFF
   User exit for logging status                       = ON
  ...
  ...
```

If a database is in a *backup_pending* state, any attempt to connect to the database will fail with the following SQL error message:

```
db2 connect to michel

SQL1116N  A connection to database ″MICHEL″ cannot be made because of BACKUP
PENDING.  SQLSTATE=57019
```

```
db2 backup database michel

Backup successful. The timestamp for this backup image is : 19951224133011
```

After the full backup of the database, the database is no longer in a *backup_pending* state, and connections to the database may resume.

```
  ...
  ...
 Database State                                        = 33
   Database is consistent                              = YES
   Backup pending                                      = NO
   Rollforward pending                                 = NO
   Log retain for recovery status                      = OFF
   User exit for logging status                        = ON
  ...
  ...
```

At the end of the backup of the database, the current active log is truncated and archived. The log files that are currently active are designated by the following two database configuration parameters (use the `get database configuration` command to display their values):

- *First active log file*: this parameter points to the head of the active log. Logs created prior to this one (that is with a log sequence number smaller) are no longer active and can be moved away from the current log path directory.
- *Next active log file*: this parameter contains the name of the log file that will be used next for logging. When log retention is being used, all logs with a sequence number greater than this file′s sequence number are not used, although they are preallocated to enhance performance and ensure the space is available when required.

In our example the only active log is the log file S0000001.LOG.

```
db2 get database configuration for michel
  ...
  ...
 First active log file                                 = S0000001.LOG
 Next active log file                                  = S0000001.LOG
  ...
  ...
```

The SQLOGDIR directory shows the following:

```
ls -l /db/dblogusr/NODE00010/SQL00001/SQLOGDIR

total 8976
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:25 S0000000.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000001.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000002.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000003.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000004.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000005.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000006.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000007.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:23 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:25 S0000010.LOG
```

At the first connection to the database, the disk space occupied by the archived log file S0000000.LOG is reclaimed by the database manager by deleting the file.

```
ls -l /db/dblogusr/NODE00010/SQL00001/SQLOGDIR

total 8160
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000001.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000002.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000003.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000004.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000005.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000006.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000007.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000010.LOG
```

The ADSM tool adsmqry allows you to list the backup activities which have been performed for the database by the ADSM server. Running the adsmqry command after the backup shows that log file S0000000.LOG has been archived so far by the ADSM server. It shows no DB2 Parallel Edition backup images yet, since the first database backup was made to the disk and not using the ADSM server.

```
adsmqry

Query for database MICHEL
   No DB2 Parallel Edition backup images found
   Log file for NODE00010: S0000000.LOG
```

5.2.1.3 Scenario Using User Exit to Archive the Logs

At this point we ran a scenario in which we inserted (using an SQL INSERT statement in order to generate logging activities) into a table a certain number of rows that generated seven log files.

When running the INSERT statement, a snapshot of the contents of the database log path directory showed the following:

```
ls -l /db/dblogusr/NODE00010/SQL00001/SQLOGDIR

total 8976
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:44 S0000001.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000002.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000003.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000004.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000005.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000006.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000007.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:32 S0000010.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:44 S0000011.LOG
```

After issuing a DB2 `terminate` command, the database manager reclaimed the disk space occupied by the following log files:

`S0000001.LOG`, `S0000002.LOG`, `S0000003.LOG`, `S0000004.LOG`, `S0000005.LOG`, `S0000006.LOG`, `S0000007.LOG`

These log files have been archived (as shown by the contents of the ARCHIVE.LOG.NODE*nnnnn* file) and are no longer part of the active logs (as shown by the value of the *First active log file* and the *Next active log file* database configuration parameters). The current active log in our example is S0000008.LOG.

```
db2 terminate

ls -l /db/dblogusr/NODE00010/SQL00001/SQLOGDIR

total 3264
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:48 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:48 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:48 S0000010.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:48 S0000011.LOG

get database configuration for michel

       Database Configuration for Database michel
 ...
 ...
 No. of secondary log files                 (LOGSECOND) = 2
 Path to log files                          = /db/dblogusr/NODE00010/SQL00001/SQ
LOGDIR/
 Changed path to log files                 (NEWLOGPATH) =
 First active log file                                  = S0000008.LOG
 Next active log file                                   = S0000008.LOG
 ...
 ...
```

Using the ADSM query tool `adsmqry` we can get the list of the log files that have been archived during the execution of the INSERT statement:

```
adsmqry

Query for database MICHEL
   No DB2 Parallel Edition backup images found
   Log file for NODE00010: S0000000.LOG
   Log file for NODE00010: S0000001.LOG
   Log file for NODE00010: S0000002.LOG
   Log file for NODE00010: S0000003.LOG
   Log file for NODE00010: S0000004.LOG
   Log file for NODE00010: S0000005.LOG
   Log file for NODE00010: S0000006.LOG
   Log file for NODE00010: S0000007.LOG
```

The contents of the ARCHIVE.LOG.NODE00010 report file shows the sequence of calls to the db2uexit program that the database manager has made during the execution of the INSERT statement. Note the value of the *User Exit RC* return parameter (always equal to 0 in our tests), which indicates Successful for archiving (or retrieving) the log file using ADSM.

```
********************************************************************************
Time Started:      Sun Dec 24 13:45:04 1995

Parameters Passed: ARCHIVE MICHEL 10 /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/ S
0000001.LOG
System Action:     ARCHIVE /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/S0000001.LOG

Media Type:        ADSM
User Exit RC:      0
Time Completed:    Sun Dec 24 13:45:09 1995

********************************************************************************

  ...
  ...
  ...

********************************************************************************
Time Started:      Sun Dec 24 13:45:37 1995

Parameters Passed: ARCHIVE MICHEL 10 /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/ S
0000006.LOG
System Action:     ARCHIVE /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/S0000006.LOG

Media Type:        ADSM
User Exit RC:      0
Time Completed:    Sun Dec 24 13:45:38 1995

********************************************************************************
Time Started:      Sun Dec 24 13:48:11 1995

Parameters Passed: ARCHIVE MICHEL 10 /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/ S
0000007.LOG
System Action:     ARCHIVE /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/S0000007.LOG

Media Type:        ADSM
User Exit RC:      0
Time Completed:    Sun Dec 24 13:48:15 1995

********************************************************************************
```

Enabling audit logging by setting the AUDIT_ACTIVE flag in the db2uexit sample allows checking for the successful archiving of log files. In case a non-successful user exit return code (that is RC <> 0) is returned, or the database manager tries to call the user exit program but cannot find it (although the userexit database configuration parameter is enabled), no error code is returned to the user. The database manager continues to apply a log retain policy for the database but no log files get moved away from the database log path directory. This case is similar to the following setting of the database configuration parameters:

```
LOGRETAIN   ON
USEREXIT    OFF
```

At this point, we took a full backup of the database using the ADSM server as the backup server. The backup command parameter *USE ADSM* indicates that the backup is being done using the ADSM server.

```
db2 backup database michel USE ADSM
```

We again inserted into the same table a set of rows (using the SQL INSERT statement) which generated seven log files. Different snapshots of the contents of the database log path directory taken during the execution of the INSERT statement show the allocation of more file logs.

```
total 3264
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000010.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000011.LOG

total 3264
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000010.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000011.LOG

total 3408
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000010.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000011.LOG
-rw-------   1 dblogusr db2pegrp   73728 Dec 24 13:55 S0000012.LOG
 ...
 ...
 ...

total 7312
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000010.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000011.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000012.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000013.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000014.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000015.LOG
-rw-------   1 dblogusr db2pegrp  401408 Dec 24 13:55 S0000016.LOG
total 8304
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000009.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000010.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:53 S0000011.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000012.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000013.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000014.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000015.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000016.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:55 S0000017.LOG
-rw-------   1 dblogusr db2pegrp   73728 Dec 24 13:55 S0000018.LOG
```

After issuing a DB2 `terminate` command, the database manager reclaimed the disk space occupied by the following files:

```
S0000008.LOG S0000009.LOG S0000010.LOG S0000011.LOG S0000012.LOG
S0000014.LOG
```

Similar to our previous test, these file logs have been archived and are no longer part of the active logs.

```
db2 terminate

ls -l /db/dblogusr/NODE00010/SQL00001/SQLOGDIR

total 3264
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:59 S0000015.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:59 S0000016.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:59 S0000017.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 13:59 S0000018.LOG

get database configuration for michel

       Database Configuration for Database michel
 ...
 ...
 First active log file                                  = S0000015.LOG
 Next active log file                                   = S0000015.LOG
 ...
 ...
```

This is confirmed by querying the ADSM server when issuing the ADSM query command adsmqry:

```
adsmqry

Query for database MICHEL
   Backup of NODE00010 taken at TIMESTAMP 19951224135302
    oldestLogFile is S0000008.LOG, catalog node number is 6
   Log file for NODE00010: S0000000.LOG
   Log file for NODE00010: S0000001.LOG
   Log file for NODE00010: S0000002.LOG
   Log file for NODE00010: S0000003.LOG
   Log file for NODE00010: S0000004.LOG
   Log file for NODE00010: S0000005.LOG
   Log file for NODE00010: S0000006.LOG
   Log file for NODE00010: S0000007.LOG
   Log file for NODE00010: S0000008.LOG
   Log file for NODE00010: S0000009.LOG
   Log file for NODE00010: S0000010.LOG
   Log file for NODE00010: S0000011.LOG
   Log file for NODE00010: S0000012.LOG
   Log file for NODE00010: S0000013.LOG
   Log file for NODE00010: S0000014.LOG
```

Note this time that the adsmqry command shows also the previous backup of the database taken using the ADSM server.

5.2.1.4 Managing the Logs

Because the database logs can occupy a large amount of storage if you plan on using the roll-forward recovery method, you must decide how to manage the archived logs before a disk-full condition occurs. To do so you may either:

1. Dedicate enough disk space in the database log path directory to retain the logs
2. Manually copy/retrieve the archived logs to or from a storage device or directory other than the database log path directory
3. Provide the database manager with a *user exit program* to copy/retrieve the logs to or from another storage device (including using an ADSM server)

Note: When LOGRETAIN is set to ON, the secondary log files (LOGSECOND) are not used.

Log File Allocation: When creating a new database, three logs (which is the default value for the logprimary database configuration parameter) are allocated for the newly created database. The default value for the log size (defined by the logfilsiz database configuration parameter) is 1000 pages (each page is 4KB in size). By default, database logs are stored in the SQLOGDIR subdirectory of the database directory, and their paths are unique because the path for the database directory contains the node name (NODE*nnnnn*). For example:

```
ls -l /db/dblogusr/NODE00010/SQL00001/SQLOGDIR

total 24048
-rw-------   1 dblogusr db2pegrp 4104192 Dec 23 16:42 S0000000.LOG
-rw-------   1 dblogusr db2pegrp 4104192 Dec 23 16:42 S0000001.LOG
-rw-------   1 dblogusr db2pegrp 4104192 Dec 23 16:42 S0000002.LOG
```

Once the database has been created, you can modify any of the database configuration parameters. If you change the value either of the logprimary database parameter or of the logfilsiz database parameter, the database logs defined with the new values will be allocated when the first application connects to the database.

Assuming you have just created the database whose alias name is michel, the `get database configuration for michel` command will display the following values. Note the default values for the LOGPRIMARY (3) and the LOGFILSIZ (1000) parameters.

```
      Database Configuration for Database michel

Max no. of active applications              (MAXAPPLS) = 20
Max DB files open per appl.                 (MAXFILOP) = 64
Max appl. control heap size (4KB)    (APP_CTL_HEAP_SZ) = 64
Default application heap (4KB)            (APPLHEAPSZ) = 32
Package cache size (4KB)                  (PCKCACHESZ) = (MAXAPPLS*4)

Max storage for lock lists (4KB)            (LOCKLIST) = 64
Percent. of lock lists per appl.            (MAXLOCKS) = 10
Interval for checking deadlock (ms)        (DLCHKTIME) = 10000

Buffer pool size (4KB)                      (BUFFPAGE) = 6400
Database heap (4KB)                           (DBHEAP) = 256
SQL statement heap (4KB)                    (STMTHEAP) = 2048
Sort list heap (4KB)                        (SORTHEAP) = 256

Log buffer size                             (LOGBUFSZ) = 8
%  log file reclaimed before soft checkpoint (SOFTMAX) = 100
Group commit count                         (MINCOMMIT) = 1
Log file size (4KB)                        (LOGFILSIZ) = 1000
No. of primary log files                  (LOGPRIMARY) = 3
No. of secondary log files                 (LOGSECOND) = 2
Path to log files                = /db/dblogusr/NODE0010/SQL00001/SQLOGDIR/
Changed path to log files                 (NEWLOGPATH) =
First active log file                                  =
Next active log file                                   =

Database attributes                           (DBATTR) = 9
  Log retain for recovery enabled          (LOGRETAIN) = OFF
  User exit for logging enabled             (USEREXIT) = OFF
  Auto restart enabled                   (AUTORESTART) = ON

Index re-creation time                                 = SYSTEM (RESTART)

Number of segments                                     = 16
Maximum file segment size (4KB)                        = 32

Database State                                         = 1
  Database is consistent                               = YES
  Backup pending                                       = NO
  Rollforward pending                                  = NO
  Log retain for recovery status                       = OFF
  User exit for logging status                         = OFF

Restore pending                                        = NO

Database territory                                     = en_US
Database code set                                      = ISO8859-1
Database country code                                  = 1
Database code page                                     = 819
Database configuration release level                   = 0x0700

ADSM password                            (ADSM_PASSWD) =
```

Data Transfer Rate Considerations: The data transfer speed of the device you use to store off-line archived logs, and the software used to make the copies, must at a minimum match the average rate at which the database manager writes data in the logs. If the data transfer speed cannot keep up with new log data being generated, you may run out of disk space if logging activity continues for a sufficiently long period of time, determined by the amount of free disk space. If this happens, database processing will stop.

Minimizing log file loss is also an important consideration in setting the log size. Archiving takes an entire log. If you use a single large log, you increase the time between archiving. If the disk containing the log fails, some transaction information will probably be lost. Decreasing the log size increases the frequency of archiving but can reduce the amount of information loss in case of a media failure since the smaller logs before the one lost can be used.

For example, if we changed the LOGFILSIZ parameter value for 100, and the LOGPRIMARY parameter value for 10, then 10 new database logs of 100 pages each are formatted during the first connection to the database.

```
db2 ″update database configuration for michel using LOGFILSIZ 100″
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  All applications must disconnect from this database before the
changes become effective.

db2 ″update database configuration for michel using LOGPRIMARY 10″
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
DB21026I  All applications must disconnect from this database before the
changes become effective.

db2start
db2stop

db2 connect to michel

ls -l /db/dblogusr/NODE00010/SQL00001/SQLOGDIR

total 8160
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000000.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000001.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000002.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000003.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000004.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000005.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000006.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000007.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000008.LOG
-rw-------   1 dblogusr db2pegrp  417792 Dec 24 12:42 S0000009.LOG
```

A DB2 Parallel Edition c program is provided in *$HOME/sqllib/adsm* to query the ADSM server database and get the list of available backups for that database on that node. Copy *adsmqry.csmp* to *adsmqry.c* and compile it (see 4.7, “Using the ADSM API” on page 35). Its output will resemble the following:

```
sp2n10-dbusr /db -> adsmqry
Query for database SAMPLE1
   No DB2 Parallel Edition backup images found
   No log files found
   Query for database SAMPLE2
   Backup of NODE00010 taken at TIMESTAMP 19951215142008
        oldestLogFile is S0000000.LOG, catalog node number is 10
   No log files found
```

Chapter 6. Recovering a DB2 Parallel Edition Database

There are two aspects of DB2 Parallel Edition recovery namely:

- Restore
- Rollforward

These aspects of recovery are covered in detail in the *DB2 Parallel Edition for AIX Administration Guide and Reference,* SC09-1982.

DB2 Parallel Edition restore utilities use an exclusive connection to the node and a shared one on the catalog node. If the restore is done on the catalog node, the connection is exclusive. You cannot restore the catalog and data nodes concurrently. If only a few nodes have to be recovered, you can restore them and roll forward on them without restoring the entire database.

When restoring from an offline backup, rollforward recovery may not be required. Yet, if either the *LOGRETAIN* or the *USEREXIT* parameters are set on, restoring will put the database in the roll forward pending state. If you know you are restoring from an offline backup and need not apply any of the logs, add the *WITHOUT ROLLING FORWARD* option to your restore command:

```
db2_all ″<<+10< db2 restore database sample use adsm taken at
yyyymmddhhmmss WITHOUT ROLLING FORWARD WITHOUT PROMPTING″
```

6.1 Database Restore

In our scenario we want to restore the backup copy of the database that was taken at 13:53:02 on 12/24/95 (see above previous backups in 5.2, “DB2 Parallel Edition Log Achieving” on page 47). We choose to restore the backup copy in the same database for which the data was originally backed up. Note the `restore` command parameter *USE ADSM*, which indicates that the restore is being done from the ADSM server storage.

```
db2 restore database michel USE ADSM taken at 19951224135302 to /db

SQL2545W  Warning|  The backup image on the ADSM server is currently stored on
mountable media.  The time required to make it available is unknown.
Do you want to continue ? (y/n) y
SQL2539W  Warning|  Restoring to an existing database that is the same as the
backup image database.  The database files will be deleted.
Do you want to continue ? (y/n) y
DB20000I  The RESTORE DATABASE command completed successfully.
```

If you are using `Without Prompting Option`, this warning will cause restore to fail.

After a successful restore operation, a database that was configured for roll-forward recovery (that is having either of the *logretain*, or *userexit* database configuration parameters turned on) at the time the backup was taken, enters a *roll-forward pending* state, and is not usable until the ROLLFORWARD command has been run successfully.

Note: If you are restoring a full database backup image that was created using the offline option of the backup command, you can bypass this *roll-forward pending* state during the restore process. The restore command gives you the option (*without rolling forward*) to use the restored database immediately without rolling forward the database. Otherwise connecting to the database will not be possible as shown below.

```
$ db2 connect to michel

SQL1117N  A connection to database ″MICHEL″ cannot be made because of ROLL
FORWARD PENDING.  SQLSTATE=57019
```

6.2 Automating DB2 Parallel Edition Restore Using ADSM

If only one backup is available in the ADSM database (if you are using ADSM) or in the local DB2 Parallel Edition backup directory (if you are not using ADSM), you need not specify the *TAKEN AT* parameter. If more than one backup is available, that parameter is required.

The following script is only provided as an example of how automation can be achieved when using ADSM. It relies on the formatted *adsmqry* output. It assumes that the last backup should be restored.

```
for i in `db2_all ″<<+10< adsmqry″ | awk ′
/^rah:/ {count = 1}
/Backup of / && / taken at TIMESTAMP /  { if ( count == 1 ) { print $7}
                      count=(count + 1)} ′ - `
do
db2_all ″<<+10< db2 restore database alain use adsm taken at $i \
WITHOUT ROLLING FORWARD WITHOUT PROMPTING″
done
for i in `db2_all ″<<-10< adsmqry″ | awk ′
/^rah:/ {count = 1 ; node = $2}
/Backup of / && / taken at TIMESTAMP /  { if ( count == 1 )
          { print node }
                       count=(count + 1)} ′ - `
do
for j in `db2_all ″<<-10< adsmqry″ | awk ′
/Backup of / && / taken at TIMESTAMP /  { print $7 ; exit }′ - `
do
db2_all ″;<<-10< db2 restore database alain use adsm taken at $j \
WITHOUT ROLLING FORWARD WITHOUT PROMPTING″
done
done
```

6.3 Rollforward Recovery Using User Exit to ADSM

We ran the `rollforward` command until the end of logs, which means that all the logs (from the active log at the time the backup was taken, to the active log at the time the restore was started) will be applied to the restore backup copy. In our example, this corresponds to the following log file sequence:

```
S0000008.LOG S0000009.LOG S0000010.LOG S0000011.LOG S0000012.LOG
S0000013.LOG S0000014.LOG S0000015.LOG
```

The rollforward command must be run from the catalog node for the considered database as it shows in the following:

```
db2 ″rollforward database michel to end of logs on node (10)″

SQL6069N  The ROLLFORWARD DATABASE command cannot be submitted on a
non-catalog node.
```

Also note that we use the *and complete* parameter of the rollforward command, which completes the roll-forward recovery process after all the transactions satisfying the *TO* clause are successfully committed. Any incomplete transaction is rolled back. The roll-forward pending state of the database is turned off. This allows access to the database once again. The *ON NODE (10)* parameter specifies that the database is being recovered on node 10 only.

```
 db2 ″rollforward database michel to end of logs on node (10) complete″
DB20000I  The ROLLFORWARD command completed successfully.
```

This operation can be combined by using the and complete option as shown below.

```
 db2 ″rollforward database michel to end of logs on node (10) and complete″
DB20000I  The ROLLFORWARD command completed successfully.
```

Different snapshots of the contents of the database log path directory taken during the execution of the RESTORE command show the retrieval of the following sequence of log files:

```
S0000008.LOG S0000009.LOG S0000010.LOG S0000011.LOG S0000012.LOG
S0000013.LOG S0000014.LOG S0000015.LOG
```

This shows the different snapshots taken during the roll-forward recovery process:

```
total 0

total 816
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:20 S0000008.LOG
 ...
 ...
 ...
total 4080
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:20 S0000008.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:20 S0000009.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000010.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000011.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000012.LOG
 ...
 ...
 ...
total 5344
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:20 S0000008.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:20 S0000009.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000010.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000011.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000012.LOG
-rw-r--r--   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000013.LOG
-rw-r--r--   1 dblogusr db2pegrp  229376 Dec 27 06:21 S0000014.LOG
total 1264
-rw-r--r--   1 dblogusr db2pegrp  229376 Dec 27 06:21 S0000014.LOG
-rw------   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000015.LOG
total 816
-rw-------   1 dblogusr db2pegrp  417792 Dec 27 06:21 S0000015.LOG
```

Note that after the logs have been applied to the restored database they are deleted from the log path directory.

Once the roll-forward recovery process has completed, the last log file that was applied is being used as the active log file as shown by the database manager configuration parameters.

```
 ...
 ...
 First active log file                                 = S0000015.LOG
 Next active log file                                  = S0000015.LOG
 ...
 ...
```

The contents of the RETRIEVE.LOG.NODE00010 report file (generated by the user exit program) shows the sequence of calls to the db2uexit program that the database manager has made as the roll-forward recovery process was under way.

```
********************************************************************************
Time Started:      Wed Dec 27 06:20:52 1995

Parameters Passed: RETRIEVE MICHEL 10 /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/
S0000008.LOG
System Action:     RETRIEVE /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/S0000008.LO
G
Media Type:        ADSM
User Exit RC:      0
Time Completed:    Wed Dec 27 06:20:53 1995

********************************************************************************
   ...
   ...
   ...
********************************************************************************
Time Started:      Wed Dec 27 06:21:12 1995

Parameters Passed: RETRIEVE MICHEL 10 /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/
S0000013.LOG
System Action:     RETRIEVE /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/S0000013.LO
G
Media Type:        ADSM
User Exit RC:      0
Time Completed:    Wed Dec 27 06:21:13 1995

********************************************************************************
Time Started:      Wed Dec 27 06:21:16 1995

Parameters Passed: RETRIEVE MICHEL 10 /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/
S0000014.LOG
System Action:     RETRIEVE /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/S0000014.LO
G
Media Type:        ADSM
User Exit RC:      0
Time Completed:    Wed Dec 27 06:21:17 1995

********************************************************************************
Time Started:      Wed Dec 27 06:21:19 1995

Parameters Passed: RETRIEVE MICHEL 10 /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/
S0000015.LOG
System Action:     RETRIEVE /db/dblogusr/NODE00010/SQL00001/SQLOGDIR/S0000015.LO
G
Media Type:        ADSM
User Exit RC:      0
Time Completed:    Wed Dec 27 06:21:19 1995
********************************************************************************
```

Chapter 7. Using CLIO/S for Backup and Restore to Mainframe Tape

This chapter focuses on DB2 Parallel Edition database backup and restore operations using MVS tape drives supplied by CLIO/S to nodes of an RISC/6000 SP. Figure 13 on page 68 is an overview of the system environment we used in our work. A key feature of DB2 Parallel Edition is the ability to run backup or restore operations from multiple nodes of the same database simultaneously. In fact it is possible to perform the operation with all nodes of a given database in parallel providing there is a tape drive available for each node. It is also possible to develop strategies that backup smaller groups of nodes on a rotating basis, for example backup two nodes a day so that at the end of a week all ten nodes have been backed up. In our system environment, we mapped the ten nodes that contain the database by column into the Comm node at the top or each column of nodes. Nodes 1, 3, 5, 7 and 11 were handled by the node 13 ESCON adapter, and nodes 2, 4, 6, 8 and 12 were handled by the node 14 ESCON adapter. With that mapping we chose to pursue the minimum outage scenario and backup all nodes in parallel. We focused on getting the function to work and did not pursue alternative approaches that might lead to different outage durations or levels of daily operational workload. The catalog node for this database was on node 10 and we did not include catalog backup and restore strategies in the scope of our work.

When you implement CLIO/S tape with DB2 Parallel Edition in your environment, you will need to define an AIX user with sufficient authority to define and modify device characteristics such as `mkdev` or `chdev` in order to use the CLIO/S device commands including `cltalloc` and `cltunall`. A DB2 Parallel Edition authorized user will need to be defined in order to successfully issue DB2 backup and restore commands and use other DB2 Parallel Edition facilities like the `db2_all` command. Most of our implementation was done as root user.

Below are some thoughts about the performance aspects of the configuration we used in our work. When performing parallel backup or restore operations, it is important to have adequate monitoring facilities available in order to locate any bottlenecks. Without measurement data, it is not possible to conclude that more than one ESCON is required to backup or restore 10 DB2 Parallel Edition nodes in parallel. Our choice was purely based on what appeared to be a balanced distribution and was not based on any facts or beforehand knowledge. This is because there are several links in the end-to-end pathway between the database on a node and the tape drive on the MVS system at the other end. Each of these links has a maximum speed at which it will move data, and it is the slowest link in the entire path that determines the aggregate data rate for the entire path. Our conclusion at this point is that each installation and application needs to be measured in its environment to determine what the aggregate data rate will be.

The remainder of this chapter is divided into three scenarios. The first will discuss backing up a single DB2 Parallel Edition node to MVS tape. The second scenario will discuss restoring a single node from MVS tape. The third scenario will discuss considerations for establishing parallel backup and restore. It is assumed in all cases that the associated DB2 Parallel Edition catalog will be dealt with in compliance with recommendations in DB2 Parallel Edition documentation.

Note: DB2 Parallel Edition PTF U440995 is required to successfully perform these scenarios.

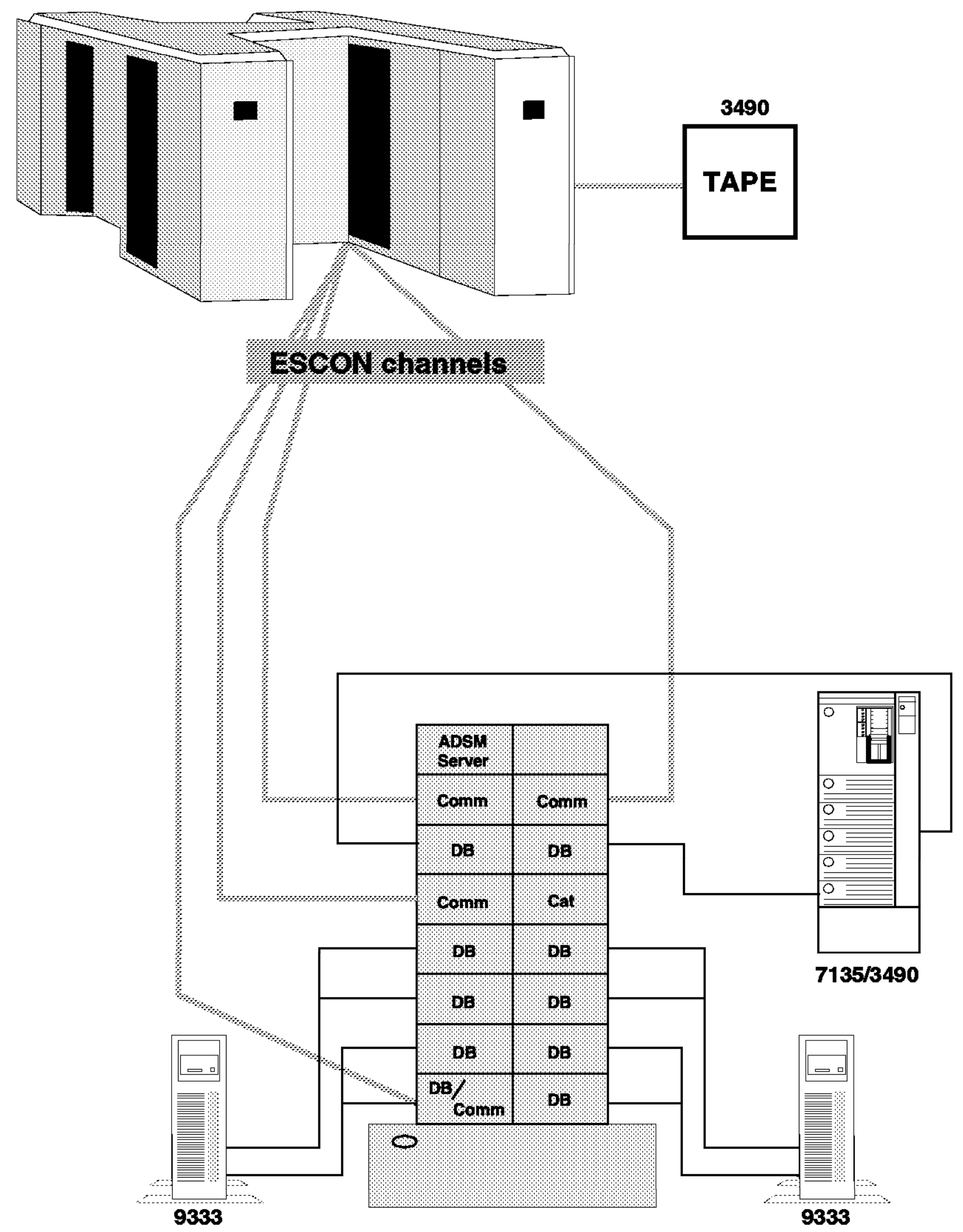


Figure 13. Mainframe to RISC/6000 SP Environment

7.1 Scenario Using a Tape Drive on MVS to Backup a Single Node

CLIO/S must be installed and running on all systems involved with the backup operation. At a minimum this means that CLIO/S must be running on the database node we are backing up as well as on the MVS system that manages the tape drive the data will be written to. If the node we are backing up to MVS tape is not directly connected to the MVS system, then CLIO/S must also be installed and active on all nodes that are part of the connection between the database node and the MVS system. In our example for this scenario, we are backing up the data on node 4 through the ESCON adapter on node 14 to the MVS system. In this case, we require CLIO/S active on node 4, node 14 and the MVS system. The data moves between node 4 and node 14 over the RISC/6000 SP High Performance Switch using TCP/IP protocol and from node 14 to the MVS system using the CLIO/S ESCON protocol. The system (node 4) containing the data to be backed up is the operational control point of the backup operation and its focal point is a pseudo tape device provided by the CLIO/S software. A pseudo tape needs to be defined on any node that will be the subject of a backup or restore operation. Various documentation sources will alternately refer to this device as a tape mount daemon as well as pseudo tape device. In this discussion we have chosen to refer to it as the CLIO-PST for consistency.

The CLIO-PST is typically created as a part of the CLIO/S install process. It is managed using typical AIX device management commands supplemented with some CLIO/S specific tape management commands. Typical AIX device management commands such as *rmdev* and *ifconfig* will fail if the appropriate CLIO/S tape management commands have not been issued beforehand and in the correct sequence. This occurs because CLIO/S introduces additional device states of *started* and *allocated* beyond the typical AIX device states of *Defined* and *Available*. Part of the challenge occurs because the CLIO/S device states are not presented by the customary AIX device display commands. You may however determine the status of a specific CLIO-PST device by examining its log file (**/tmp/cltclntd.pst***n*). The CLIO/S specific commands that deal with the special states of the CLIO-PST are shown below with a brief description of each:

CLTCLNTD Starts the device and makes it allocatable. The device must have been previously defined and made available through SMIT or AIX device commands.

CLTALLOC Assigns a started CLIO-PST device to a specific CLIO/S tape process and communicates the allocation parameters to the target system.

CLTUNALL Communicates intention to cease use of the CLIO-PST device to the target system. This must be accomplished before attempting to assign the device for another user. The `cltterm` command below must be issued if you intend to redefine the characteristics of the CLIO-PST device.

CLTTERM Removes allocatable status from the CLIO-PST device and returns the device to AIX available status. This must be accomplished before attempting to redefine the characteristics of the device using SMIT or AIX device commands.

The key thing to remember about the CLIO-PST is that it should be stopped (cltterm) and not allocated (cltunall) before you attempt to do anything with its characteristics using SMIT or AIX device commands.

Below is documented a sequence of commands that show the full life cycle of a CLIO-PST. In actual practice, it is not likely that the CLIO-PST will be created or destroyed during normal use. It is also not likely that its AIX status will need to be changed from Available to Defined. The scenario is presented as a full life cycle for completeness.

1. Create a CLIO-PST on the system (node 4) that has the data to be backed up:

   ```
   mkdev -c tape -s remote -t cliotape -a host='sp2sw14:db2mvs' -a allocp='UNIT=3490_NL_PRIV'
   ```

   Alternatively, the CLIO-PST could be created following the SMIT sequence shown here and filling in the panel as shown below: Via SMIT -> Devices -> Tape Drive -> Change/Show Remote Tape Drive -> /dev/pstn (where “n” is the appropriate CLIO-PST number). Ensure that you have indicated the desired allocation parameters and maximum blocksize.

   ```
                  Change / Show Characteristics of a Remote Tape Drive

    Type or select values in entry fields.
    Press Enter AFTER making all desired changes.

                                                           [Entry Fields]
      Tape Drive                                          pst0
      Status                                              Defined
      Tape drive type                                     cliotape
      Tape driver interface                               remote
      Description                                         CLIO/S remote tape
      Name of remote host                                [sp2sw14:db2mvs]
      Name of remote device / allocation parameter       [UNIT=3490_NL_PRIV]
      Maximum blocksize                                  [32768]
    #
      Unallocate on close                                 yes                   +
      Delayed Open on MVS                                 yes                   +
      Default volser for allocation on MVS               [DUMMY]
   ```

   The examples above create a new CLIO-PST device (/dev/pst0).

   The *host* parameter provides name of the target system and routing information if needed. Our example, *sp2sw14:db2mvs*, shows that the target system *db2mvs* will be accessed using an intermediate system named *sp2sw14*.

   The allocation parameter, *allocp*, provides information for the target system to properly locate and assign one of its tape drives to satisfy allocation requests for this pseudo tape device. in our example *UNIT=3490_NL_PRIV* is the information we have chosen to supply our target MVS system which will enable it to select a tape drive from its collection of 3490 tape drives and request mounting of an unlabeled tape. A full range of parameters can be specified here to clearly define to the target MVS system what sort of device and tape volume is required to satisfy allocation requests for the CLIO-PST device being defined by the mkdev command or the SMIT panel. A suggestion to consider would be to only code on the mkdev command the parameters that are not likely to change over time. Since it is possible to specify most of these allocation parameters as part of the cltalloc command, specifying only those that are stable on the mkdev command practically eliminates the need to modify the definition.

   Note: Creating a CLIO-PST using the mkdev command or SMIT results in a defined, available and started device. Defined and available are the customary AIX device states and started is the CLIO-PST device state required before issuing a CLTALLOC command.

2. Check that a CLIO-PST device is available by issuing the following command:

```
node04==> lsdev -Cc tape
```

All PST*n* devices should be available.

3. If you have not already done so you may wish to verify that CLIO/S connectivity exists between all systems involved in the operation we are about to begin:

```
node04==> parping sp2sw14:db2mvs
```

This command is equivalent of a TCP/IP PING command; however it also verifies that the CLIO/S server is running on each system. In our example we are going from our system to *sp2sw14* and from there on to *db2mvs*. If the response you receive does not indicate that the PARPING was successful, you should to make sure that PARSRV is running on each of the systems and that TCP/IP connectivity exists between them as well. An example of the response to PARPING appears below.

```
DB2MVS(0x090c1515):PING took 14.666080 ms
```

4. Allocate the CLIO-PST by issuing the following command:

```
node04==> cltalloc -p new -c -d ’TOTO.BACK01’ -f /dev/pst0
```

You will receive responses to this command that show the complete set of allocation parameters that were sent to the target system. You will also receive a results response once the target system has completed analysis and mounting of the tape. This may take a few minutes to complete since the operator at the target system may have to physically locate and mount the tape. A sample set of responses appear below:

```
cltalloc VolSer=DUMMY, DSName=TOTO.BACK01, Disp=NEW, Catalog=CATLG, Label=NL,
EpDate=, RetDate=, Device=/dev/pst0

cltalloc: DUMMY is now mounted on /dev/pst0
```

Now that the tape has been mounted, you are able to proceed with the next step.

5. Backing up the DB2 Parallel Edition Database to the allocated CLIO-PST:

```
node04==> db2 backup database michel to /dev/pst0 buffer 8
```

Note: The buffer parameter must correspond to the CLIO-PST maximum blocksize defined in SMIT. In the example above, buffer 8 is specified because 8 pages (32768 bytes) corresponds to the CLIO-PST maximum blocksize defined for 32768 bytes.

6. After the backup is complete, the `cltunall` command is used to remove the allocation and free the CLIO-PST for use by another task.

```
node04==> cltunall -f /dev/pst0
```

7. For completeness of the lifecycle scenario, we would follow the `cltunall` command with a `cltterm` command to terminate the device and return it to

the nominal AIX available state. In the available state, you are able to perform normal AIX device functions such as change device characteristics.

```
node04==> cltterm -f /dev/pst0
```

7.2 Scenario Using a Tape Drive on MVS to Restore a Single Node

Information presented in the previous section on the backup scenario relating to the creation (*mkdev*) and removal of a CLIO-PST should be reviewed prior to proceeding with this section. The scenario below begins with an existing CLIO-PST that is already started and ends with the return of that device to an allocatable status.

1. Allocate the CLIO-PST device by issuing the following command:

   ```
   node04==> cltalloc -f /dev/pst0 -p old -d ′iwao.tpcdb.backup′
   ```

   This invocation will allocate /dev/pst0 to an existing MVS tape data set named ′iwao.tpcdb.backup′.

   Wait for the reply indicating /dev/pst0 has been allocated.

2. Invoke the DB2 `restore database` command and specify the CLIO-PST (/dev/pst0):

   ```
   node04==> db2 restore database tpcdb from /dev/pst0 buffer 8
   ```

   Note: The buffer parameter must correspond to the CLIO-PST maximum blocksize defined in SMIT. In the example above, buffer 8 is specified because 8 pages (32768 bytes) corresponds to the CLIO-PST maximum blocksize defined for 32768 bytes.

3. Unallocate the CLIO-PST after the restore is complete and free it for use by another task.

   ```
   node04==> cltunall -f /dev/pst0
   ```

7.3 Considerations for Establishing Parallel Backup Using MVS Tapes

This section assumes that the reader is already familiar with the concepts covered in the previous sections of this chapter. In this discussion we are referring to Figure 13 on page 68 and will describe the considerations for backing up and restoring the 10 database nodes shown (nodes 1-8, 11, and 12) and splitting the nodes evenly across two ESCON connected nodes (nodes 13 and 14) using 10 MVS tape drives. For purposes of this discussion it is assumed that the database catalog node has already been backed up or will be restored as soon as the nodes have completed. Following our initial premise in the earlier scenarios of this chapter, CLIO/S must be up and running on nodes 1-8, 11-14, and the MVS system. Had our discussion included the catalog node (node 10) and we were backing it up using CLIO/S as well, then CLIO/S would have to be running there and on node 10′s associated ESCON node (node 9) as well.

We will pursue the full lifecylce of a CLIO-PST here again in order to demonstrate the use of the parallel command features of the RISC/6000 SP. In some cases we will simply remote shell (*rsh*) the command to each node specifically and in other cases we will use the distributed shell command (*dsh*). To make the dsh functionality work we have created a node list beforehand that contains the names of the database nodes we wish to send commands to. This list has been named *dbnodes* and is the subject of an EXPORT WCOLL=dbnodes command that we issued on node 1 before beginning this work. Refer to the RISC/6000 SP publications for details about the `dsh` command and the WCOLL environment variable.

Below is documented a sequence of commands that shows the full life cycle of a CLIO-PST. In actual practice, it is not likely that the CLIO-PST will be created or destroyed during normal use. It is also not likely that its AIX status will need to be changed from Available to Defined. The scenario is presented as a full life cycle for completeness. The working collection has been established to address all nodes except nodes 9, 10, 13-16. We will be executing all of our commands from node 1.

1. Create a CLIO-PST on the each node that has the data to be backed up:

   ```
   sp2n01==> dev -c tape -s remote -t cliotape -a host=’sp2sw14:db2m  vs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n02 mkdev -c tape -s remote -t cliotape -a host=’sp2sw14:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n03 mkdev -c tape -s remote -t cliotape -a host=’sp2sw13:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n04 mkdev -c tape -s remote -t cliotape -a host=’sp2sw14:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n05 mkdev -c tape -s remote -t cliotape -a host=’sp2sw13:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n06 mkdev -c tape -s remote -t cliotape -a host=’sp2sw14:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n07 mkdev -c tape -s remote -t cliotape -a host=’sp2sw13:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n08 mkdev -c tape -s remote -t cliotape -a host=’sp2sw14:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n11 mkdev -c tape -s remote -t cliotape -a host=’sp2sw13:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   sp2n01==> rsh sp2n12 mkdev -c tape -s remote -t cliotape -a host=’sp2sw14:db2mvs’ -a allocp=’UNIT=3490_NL_PRIV’
   ```

   The examples above create a new CLIO-PST device (/dev/pst0) on each of the 10 database nodes. Due to the variations between even numbered nodes and odd numbered nodes with regard to the assigned ESCON node, it is probably useful to consider using a script to create all the CLIO-PSTs or alternatively use minor command line edits of a retrieved command line. All nodes should respond with /dev/pst0 defined. Again this sequence of commands will probably only be executed once.

   The allocation parameter, *allocp*, provides information for the target system to properly locate and assign one of its tape drives to satisfy allocation requests for this pseudo tape device. in our example *UNIT=3490_NL_PRIV* is the information we have chosen to supply our target MVS system which will enable it to select a tape drive from its collection of 3490 tape drives and request mounting of an unlabeled tape. A full range of parameters can be specified here to clearly define to the target MVS system what sort of device and tape volume is required to satisfy allocation requests for the CLIO-PST device being defined by the `mkdev` command or the SMIT panel. A suggestion to consider would be to only code on the `mkdev` command those parameters that are not likely to change over time. Since it is possible to specify most of these allocation parameters as part of the `cltalloc` command, specifying only those that are stable on the `mkdev` command practically eliminates the need to modify the definition and avoids the potential pitfall of attempting to *chdev* CLIO-PST devices that have not been properly unallocated and stopped.

2. Check that a CLIO-PST device is available on each node by issuing the following command:

   ```
   node01==> dsh lsdev -Cc tape
   ```

All PST*n* devices should be available. Remember the dsh subject command *lsdev...* will be sent to all nodes in the list identified by the WCOLL environment variable on node01.

3. If you have not already done so you may wish to verify that CLIO/S connectivity exists between all systems involved in the operation we are about to begin:

```
node01==> dsh parping sp2sw13:db2mvs
node01==> dsh parping sp2sw14:db2mvs
```

   We did this twice to confirm that all nodes in our working collection could get through both ESCON nodes to the MVS system. Remember that we are planning to drive the even numbered nodes through node 14 and the odd numbered nodes through node 13. Ten responses should come back for each of the two commands above. If a node′s response did not come back, you should make sure that PARSRV is running on that system and that TCP/IP connectivity exists between them as well. An example of the response to PARPING appears below.

```
DB2MVS(0x090c1515):PING took 14.666080 ms
```

4. Allocate the CLIO-PSTs by issuing the following commands:

```
node01==> cltalloc -p new -c -d ′TOTO.BACK01′ -f /dev/pst0
node01==> rsh sp2n02 cltalloc -p new -c -d ′TOTO.BACK02′ -f /dev/pst0
node01==> rsh sp2n03 cltalloc -p new -c -d ′TOTO.BACK03′ -f /dev/pst0
node01==> rsh sp2n04 cltalloc -p new -c -d ′TOTO.BACK04′ -f /dev/pst0
node01==> rsh sp2n05 cltalloc -p new -c -d ′TOTO.BACK05′ -f /dev/pst0
node01==> rsh sp2n06 cltalloc -p new -c -d ′TOTO.BACK06′ -f /dev/pst0
node01==> rsh sp2n07 cltalloc -p new -c -d ′TOTO.BACK07′ -f /dev/pst0
node01==> rsh sp2n08 cltalloc -p new -c -d ′TOTO.BACK08′ -f /dev/pst0
node01==> rsh sp2n11 cltalloc -p new -c -d ′TOTO.BACK11′ -f /dev/pst0
node01==> rsh sp2n12 cltalloc -p new -c -d ′TOTO.BACK12′ -f /dev/pst0
```

   In this set we chose to rshell a command to each node giving a different data set name to each tape. Other parameters, such as volume serial, could be used to make each mount unique and more manageable from an operations perspective.

   You will receive responses to this command that show the complete set of allocation parameters that were sent to the target system. You will also receive a results response once the target system has completed analysis and mounting of the tape. This may take a few minutes to complete since the operator at the target system may have to physically locate and mount the tape. We would expect a set of responses for each of the 10 nodes that we are backing up. Only the first response set is shown below:

```
cltalloc VolSer=DUMMY, DSName=TOTO.BACK01, Disp=NEW, Catalog=CATLG, Label=NL,
EpDate=, RetDate=, Device=/dev/pst0

cltalloc: DUMMY is now mounted on /dev/pst0
```

   Now that the tapes have been mounted, you are able to proceed with the next step.

5. Backing up the DB2 Parallel Edition Database to the allocated CLIO-PSTs:

```
node01==> dsh db2 backup database michel to /dev/pst0 buffer 8
```

Alternative use of the DB2 Parallel Edition commands such as the *db2_all* script might be considered appropriate for your environment. Refer to DB2 Parallel Edition documentation for guidance.

You should receive confirmation messages from each node as it completes it backup.

6. After all backup is complete, the cltunall command is used to remove the allocation and free the CLIO-PST for use by another task.

```
node01==> dsh cltunall -f /dev/pst0
```

7. For completeness of the life cycle scenario, we would follow the cltunall command with a cltterm command to terminate the device and return it to the nominal AIX Available state. In the available state, you are able to perform normal AIX device functions such as change device characteristics.

```
node01==> dsh cltterm -f /dev/pst0
```

8. At this point, we could ifconfig and rmdev the CLIO-PSTs into a configured or an undefined state. A more likely occurrence would be to leave the CLIO-PSTs in the available state or at the started state by not issuing the cltterm command. During our implementation, we left our CLIO-PSTs in the started state.

7.4 Considerations for Establishing Parallel Restore Using MVS Tapes

Similar changes to the restore scenario above could be made to accommodate a parallel restore. This section assumes that you have read and understood the previous sections in this chapter.

1. Allocate tape drives and mount the tapes

```
node01==> cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.01'
node01==> rsh sp2n02 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.02'
node01==> rsh sp2n03 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.03'
node01==> rsh sp2n04 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.04'
node01==> rsh sp2n05 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.05'
node01==> rsh sp2n06 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.06'
node01==> rsh sp2n07 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.07'
node01==> rsh sp2n08 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.08'
node01==> rsh sp2n11 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.11'
node01==> rsh sp2n12 cltalloc -f /dev/pst0 -p old -d 'iwao.tpcdb.backup.12'
```

This invocation will allocate each node's CLIO-PST to an existing MVS tape dataset named 'iwao.tpcdb.backup.nn' which contains the backup image for that node. From node 1 we rshelled a different command to each node changing the data set name to provide MVS operations personnel with the information needed to correctly locate and mount the desired tapes. Remember, we are causing 10 tape mount messages to appear on the MVS system operators console and it may take some time for all of the tapes to be located and properly mounted. It might be a good idea to provide MVS Operations with some advance warning.

Once the CLIO-PSTs are allocated, we can proceed with the restore operation.

2. Start the restores

```
node01==> dsh db2 restore database tpcdb from /dev/pst0 buffer 8
```

This form of the DSHELL command assumes that the list of targeted nodes is identified by the current setting of the WCOLL environment variable. You could also use of the DB2 Parallel Edition *db2_all* command to perform similar operation.

3. Unallocate the CLIO-PST after all restores are completed.

```
node01==> dsh cltunall -f /dev/pst0
```

Chapter 8. Performance Considerations

This chapter will focus on some of the factors you need to account for to enhance the performance of your system environment during backup operation. These factors are:

- Network parameters
- ADSM parameters
- DB2 Parallel Edition related parameters

8.1 Configuring and Tuning the Network

The Maximum Transfer Unit(mtu) parameter should be set to the appropriate value.

Check the mtu size by issuing the `netstat` command.

- If you are using the switch, the mtu parameter should be set to 65520 across all nodes. Check that it is the case by issuing the following command:

```
dsh netstat -I css0
sp2n01: Name  Mtu   Network     Address   Ipkts Ierrs    Opkts Oerrs  Coll
sp2n01: css0  65520 <Link>0.0.0.0.0.0      0     0        0     0      0
sp2n01: css0  65520 9.12.6      sp2sw01    0     0        0     0      0
 .
 .
sp2n15: css0  65520 <Link>0.0.0.0.0.0      96932 0        47846 0      0
sp2n15: css0  65520 9.12.6      sp2sw15    96932 0        47846 0      0
```

- If you are using Ethernet, the mtu parameter should be set to 1500 across all nodes. Check that it is the case by issuing the following command:

```
dsh netstat -I en0
sp2n01: Name  Mtu   Network     Address   Ipkts Ierrs    Opkts Oerrs  Coll
sp2n01: en0   1500  <Link>10.0.5a.fa.18.cf 787     0      711     0    0
sp2n01: en0   1500  9.12.20     sp2n01     787     0      711     0    0
 .
 .
sp2n15: en0   1500  <Link>10.0.5a.fa.14.7c 545921  0      682376  0    0
sp2n15: en0   1500  9.12.20     sp2n15     545921  0      682376  0    0
```

If needed, the `ifconfig mtu` command will enable you to correct the mtu setting.

8.1.1 Network Option Parameters

The `no` command allows querying and setting of some other major network options.

If you are using the switch, ensure that the following network options are correctly set:

- TCP socket buffer sizes:
  - tcp_sendspace 655360
  - tcp_recvspace 655360
- Maximum memory allocated to mbuff and cluster mbuff pools:
  - thewall 100000

- Maximum system buffer size:
  - sb_max 1310720

To query network options across all nodes, issue the following command:

```
sp2n15-root / -> dsh -a ″no -a″
```

Make sure all nodes use the same settings. If needed, issue the following commands from the control workstation as root:

```
dsh no -o thewall=100000
dsh no -o sb_max=1310720
dsh no -o tcp_sendspace=655360
dsh no -o tcp_recvspace=655360
```

Note: Though all nodes are now using the correct values, these changes will not survive reboots. In order for those changes to be permanent, the various no options must be set in a file such as the /etc/rc.net file and ensure that this file is executed at boot time.

Note: The lowclust, lowmbuff and mb_cl_hiwat network option (no) parameters do not apply in AIX 4.1.

8.1.2 ADSM Performance Parameters

8.1.2.1 On the Server Side (dsmserv.opt)

Using Ethernet or HPS

- TCPWindowsize 0

8.1.2.2 On the Client Side (dsm.sys)

Using HPS

- TCPBuffsize 32
- TCPWindowsize 640
- TXNByteLIMIT 25600

Using Ethernet

- TCPBuffsize 32
- TCPWindowsize 64
- TXNByte 25600

More information regarding ADSM parameter settings can be found in *ADSM Using the UNIX Backup-Archive Clients Version 2*.

8.1.3 Configuring DB2 Parallel Edition

DB2 Parallel Edition sets the default backup/restore buffer sizes to 1024 pages, each being 4KB in length. This setting can be modified to suit specific needs.

1. Command Line Buffer Size Setting

   It is possible to specify a different buffer size than the default one on the DB2 Parallel Edition command line using the BUFFER parameter. This parameter can be used both with the BACKUP and RESTORE DB2 Parallel Edition commands.

   - To back up a database using a buffer size of 10240 pages (4K):

     ```
     BACKUP DATABASE sample BUFFER 10240
     ```

   - To restore a database using a buffer size of 10240 pages (4K):

     ```
     RESTORE DATABASE sample BUFFER 10240
     ```

   Specifying a buffer size this way does not affect the buffer sizes specified at database manager level (RESTBUFSZ and BACKBUFSZ values).
   If no buffer size is specified on the DB2 Parallel Edition command line, or if a buffer size of 0 is used, backups will use the BACKBUFSZ buffer size value while restores will use the RESTBUFSZ buffer size value.

   In order to find out what the backup buffer size is, issue the following command:

   ```
   db2 ″GET DATABASE MANAGER CONFIGURATION″ | grep BACKBUFSZ
   ```

   In order to find out what the restore buffer size is, issue the following command:

   ```
   db2 ″GET DATABASE MANAGER CONFIGURATION″ | grep RESTBUFSZ
   ```

2. Database Manager Buffer Size Setting

   It is also possible to modify the RESTBUFSZ and BACKBUFSZ values at the database manager level.

   The new backup buffer size will be used if no buffer size is specified in the DB2 Parallel Edition backup command, or if a buffer size of 0 is used.

   The new restore buffer size will be used if no buffer size is specified in the DB2 Parallel Edition restore command, or if a buffer size of 0 is used.

   Note: The RESTBUFSZ or BACKBUFSZ buffer sizes will be overridden if a non-zero BUFFER value is specified in the restore or backup DB2 Parallel Edition commands.

   - backbufsz sets the number of 4KB buffer pages used during the backup of a database. To set a backup buffer of 10240 pages:

     ```
     UPDATE DATABASE MANAGER CONFIGURATION USING BACKBUFSZ 10240
     ```

   - restbufsz sets the number of 4KB buffer pages used during the restore of a database. To set a restore buffer of 10240 pages:

     ```
     UPDATE DATABASE MANAGER CONFIGURATION USING RESTBUFSZ 10240
     ```

Chapter 9. High Availability for DB2 Parallel Edition on RISC/6000 SP

Generally speaking, the goal of these chapters about high availability is to list and suggest alternatives to those special issues raised by using DB2 Parallel Edition on the RISC/6000 SP. In this sense, the emphasis will be on High Availability Cluster Multi-Processing (HACMP) from a DB2 Parallel Edition perspective. No HACMP knowledge is required in order to understand this chapter and the subsequent chapters on this subject. This is written at a level that considers the interest of most database specialists with a need to implement HACMP/6000 in a DB2 Parallel Edition environment.

After reviewing the hardware and software configuration used for the tests, we will give an overview of the available design alternatives, then cover HACMP for DB2 Parallel Edition installation and configuration.

The approach followed in these chapters with HACMP is to start from an existing standard RISC/6000 SP environment with DB2 Parallel Edition installed, and to detail the changes necessary in order to configure HACMP into the existing RISC/6000 SP system. However, there are customers who may want to include the consideration of HACMP on their system prior to the installation of the RISC/6000 SP and DB2 Parallel Edition. Those readers interested in the latter approach should refer to the redbook *HACMP Cookbook*, SG24-4553 and use it with this book.

9.1 Hardware Configuration

The configuration used for DB2 Parallel Edition and HACMP testing consists of a four-node RISC/6000 SP (see Figure 14 on page 82) with 512 MB of main memory each. Nodes are wide nodes interconnected via a High-Performance Switch (HPS) model LC8. In addition, the standard administrative Ethernet (SP Ethernet) connects the nodes to the RS/6000 Model 370 control workstation.

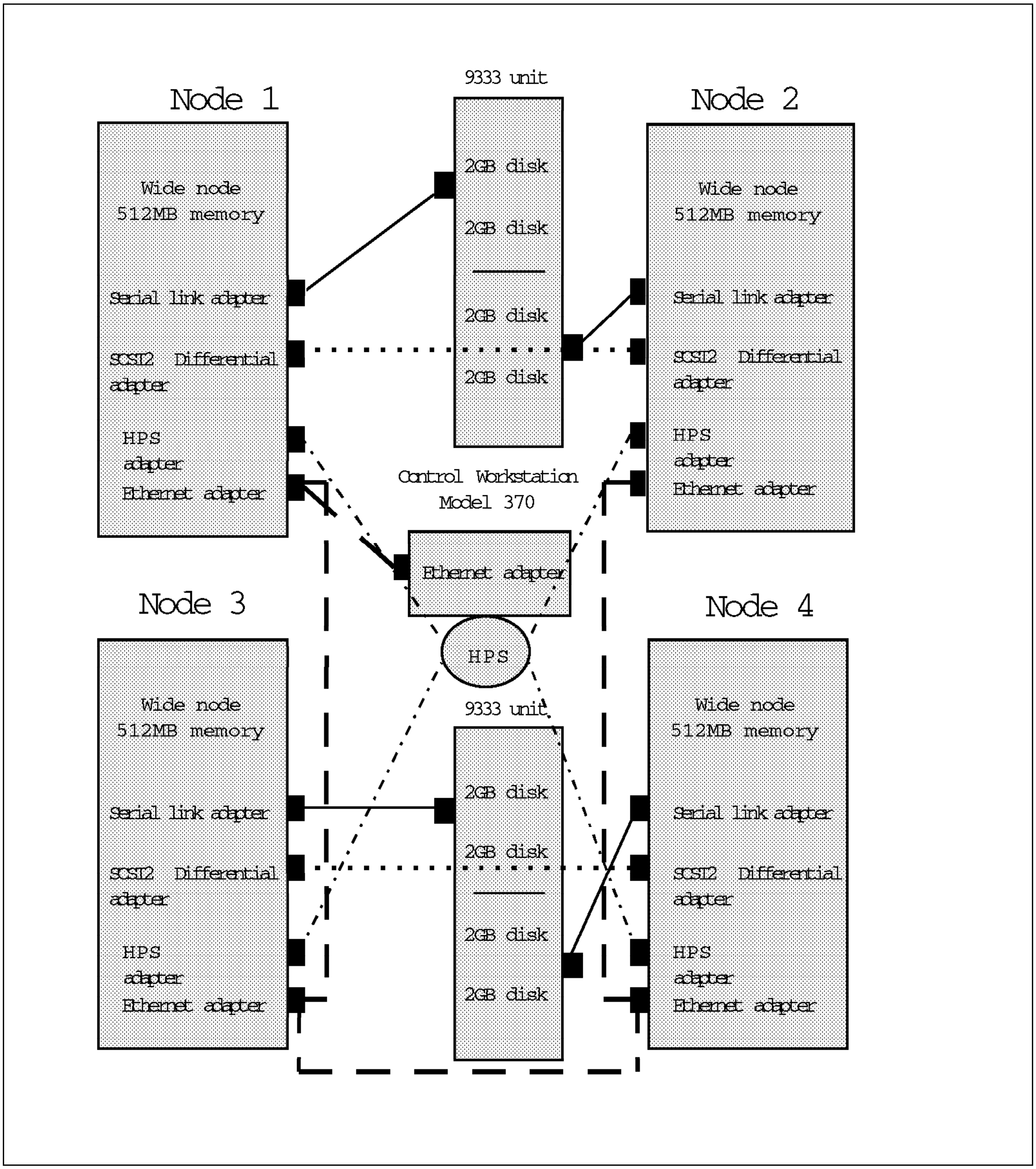


Figure 14. Hardware Configuration Used for the Tests

Two serial storage units type 9333-501 are connected to the four wide nodes in a twin-tailed configuration. Each unit contains four 2 GB serial disks. These disks are used to store any critical data, such as DB2 Parallel Edition table data, that needs to be protected against hardware failures.

Each node has a SCSI-2 Differential High-Performance External I/O Controller (FC 2420) used to enable target mode SCSI (tmscsi) communications between pairs of nodes for the HACMP heartbeat. For a two-node HACMP cluster, five cables and terminators are required, including two 52G7348 Y-cables, one 52G7349 system-to-system cable, and two 52G7350 terminators. Note that the two resistive terminators on each SCSI-2 Differential adapter have to be removed. Before the SCSI adapters can be cabled together, it is necessary to check that they have different SCSI identifiers by choosing an appropriate value for the Adapter card SCSI ID field in the menu displayed by the smit chgscsi command. Since each device on a SCSI bus needs to have a unique identifier and target mode SCSI connects two adapters together, the two adapters cannot have the same identifiers. It is recommended to select a value different from seven for each adapter′s SCSI identifier.

9.1.1 Software Levels

At the software level (see Figure 15), the control workstation and nodes run AIX V4.1.4 with PSSP V2.1, with HACMP V4.1.1 on the nodes only. Additional testing was performed with HACMP V3.1.1, PSSP V1.2, and AIX V3.2.5 on the nodes. Because HACMP V4.1.1 and HACMP V3.1.1 have the same functionality, the descriptions in this chapter can apply to both releases, the only differences being in some SMIT shortpaths having different names in the new version. The newer version has also some graphical administrative tools that will not be covered here, since they provide a lesser degree of functionality than SMIT.

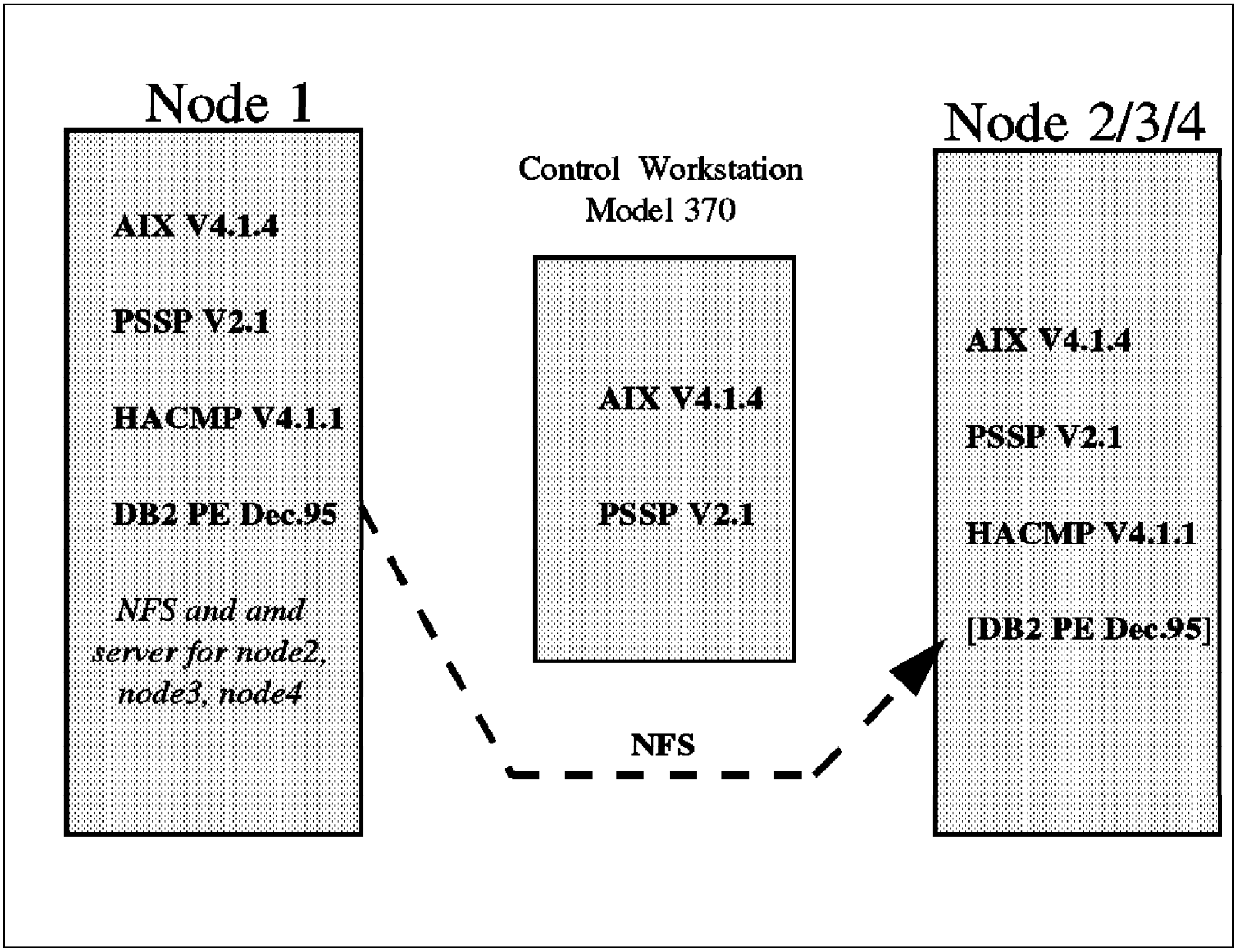


Figure 15. Software Configuration Used for the Tests

9.1.2 Network Interfaces

The four nodes are called node1 to node4. These names are the Ethernet interface network names. For the HPS, the names are switch1 to switch4. These are the names describing the switch adapters in the SDR. HACMP implementation of IP address takeover uses one switch adapter per node. As a result, it does not allow the use of the SDR switch names for IP address takeover. Each switch adapter has an alias address, called sw1 to sw4, on a different logical subnet (129.1.3 versus 129.1.2) than switch1 to switch4. Finally, four switch boot addresses, sw1_boot to sw4_boot, are defined on the same subnet as sw1 to sw4 for switch IP address takeover. For a summary of available network interfaces on each node, see Figure 16 on page 85.

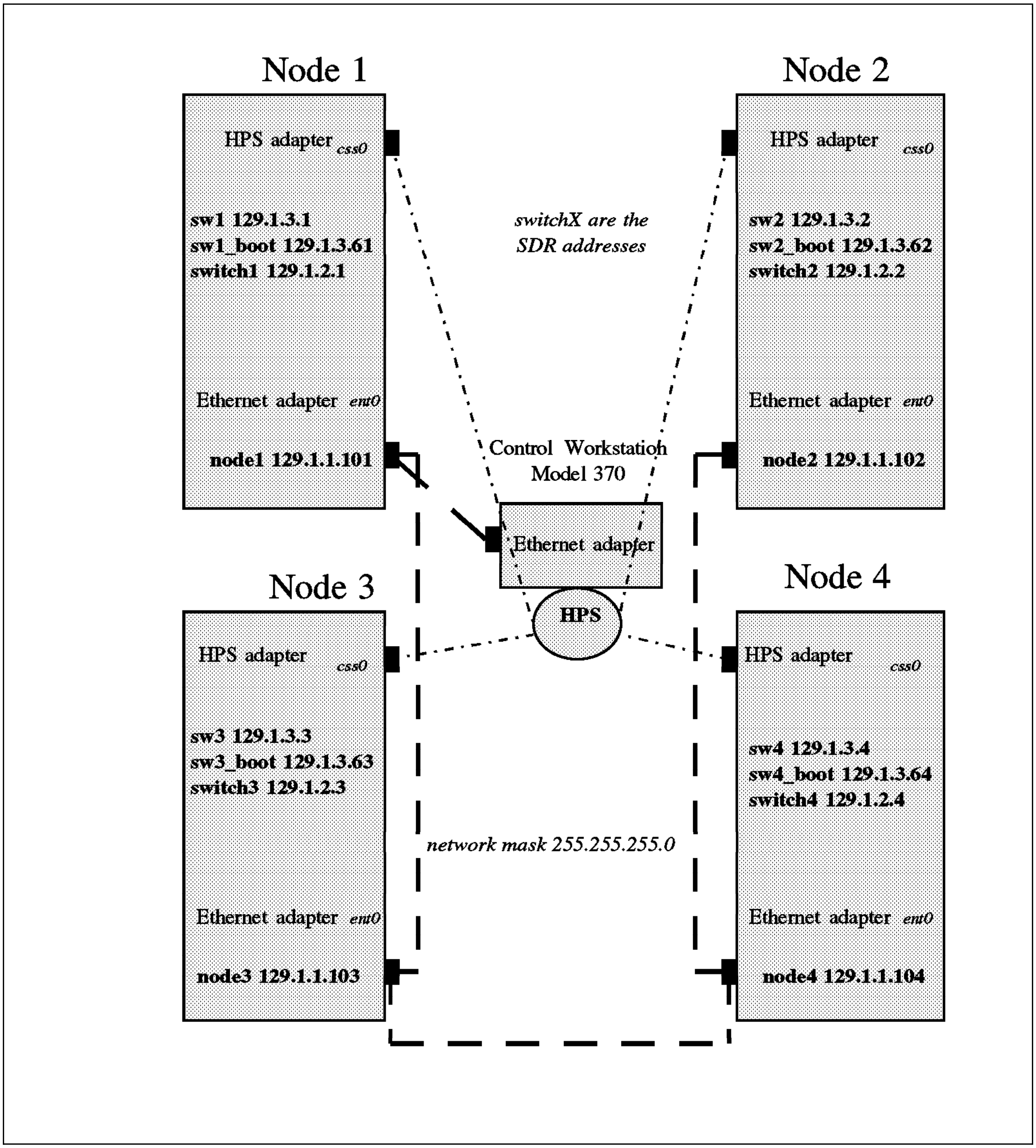


Figure 16. Network Interfaces Available on Each Node

9.1.3 DB2 Parallel Edition Configuration

DB2 Parallel Edition installable images are installed on node1 and exported to the other nodes via NFS. The instance owner is called db2 and belongs to the db2 group. The instance owner′s home directory is made available to the other nodes using amd. Two databases are defined, called SAMPLE and DUMB. Both databases contain the same tables with the same data, the difference between the two being that SAMPLE′s catalog node is node1, and DUMB′s catalog node is node2 (see Figure 17 on page 87). This was necessary in order to perform various failover tests. Three tables are defined in each database:

- Table1 has three integer fields (f1,f2,f3), and is partitioned on f1. It has one million rows, and is defined on all four nodes.
- Table2 has also three fields (g1,g2,g3), and is partitioned on g1. It has about 80000 rows, and was used in join tests together with table1. It is also defined on all nodes.
- Table3 has the same structure and data as table2, but is defined over two nodes only (node3 and node4).

DB2 Parallel Edition uses the sw1 to sw4 switch network addresses for inter-nodes communications in the db2nodes.cfg configuration file.

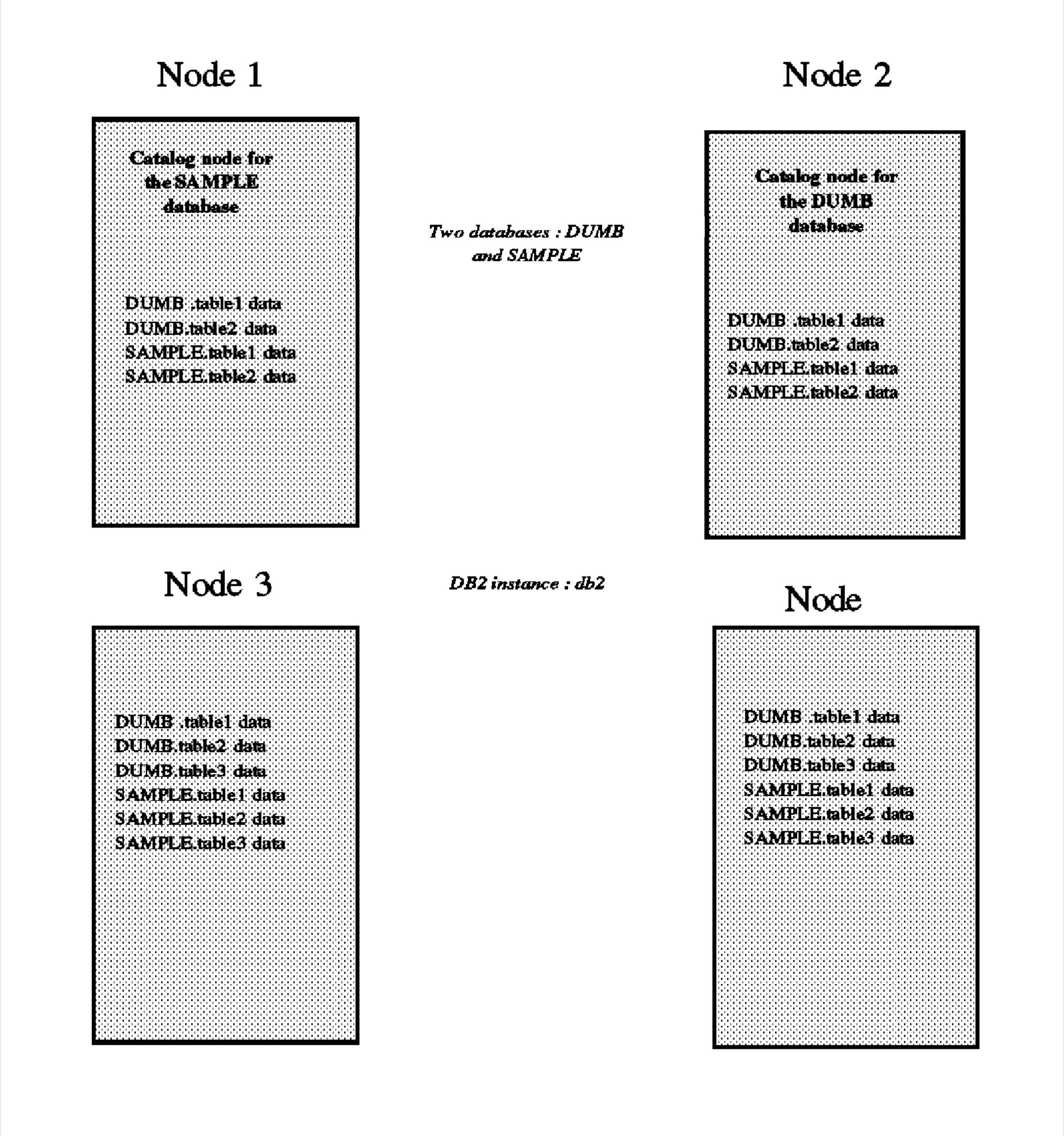


Figure 17. DB2 Parallel Edition Configuration Used for the Tests

Chapter 10. HACMP Considerations and Implementation Alternatives for DB2 Parallel Edition

This section provides several guidelines concerning what one should keep in mind in a HACMP environment involving DB2 Parallel Edition on the RS/6000 SP.

10.1 SP Ethernet Considerations

According to HACMP guidelines, the SP Ethernet should not be used for IP address takeover, since this could conflict with the node information stored in the SDR on the control workstation.

10.1.1 Non-IP Network Considerations

Even though it would be possible to use the SP Ethernet and the HPS as the only network for HACMP heartbeat, it is not recommended to do so. In case of TCP/IP failure on one node, several nodes may be trying to acquire the same shared resources at the same time due to network partitioning because of TCP/IP failure. Most customers will probably want to protect themselves against this type of problem, making the use of a non-IP link for heartbeat a necessity. This link can either be a serial RS-232 connection between nodes, requiring multiport serial adapters in each node, or a target mode SCSI connection between nodes. Actually, it is highly recommended to use two networks for heartbeat, one being a non-IP network, the other being the SP Ethernet. When the Estart command is issued, which happens each time the HPS needs to be restarted on one of the nodes and in particular for each HACMP takeover and cluster node reintegration, the switch becomes momentarily unavailable. For HACMP, if the SP Ethernet was not included in the configuration, this would mean that there would be no IP link available during Estart processing. To avoid possible problems caused by this, the SP Ethernet is included in the HACMP configuration as backup heartbeat network, which guarantees an IP communication between nodes at all times.

10.1.2 DB2 Parallel Edition Binaries Recovery

DB2 Parallel Edition binaries are located under /usr/lpp/db2pe_xx_xx. There are basically two ways to make these files available on all nodes:

1. Install DB2 Parallel Edition on all nodes
2. Export DB2 Parallel Edition binaries using NFS or AFS

   The second alternative makes software maintenance easier since DB2 Parallel Edition version upgrades can be done on one node only. For availability reasons, the exported file system should be placed on external shared disk hardware. Since having binaries accessed via NFS does not impact performance in most cases, this approach should be favored over the first one.

   In addition, because of the interdependencies between DB2 Parallel Edition binaries and the instance owner′s home directory, the latter being in all cases exported via NFS, it would make little sense to install binaries on all nodes. In case of the home directory not being available, it would no longer be possible to access the binaries, since access to the binaries goes through pointers in the instance owner′s home directory.

10.1.3 Cluster Size

HACMP supports cluster sizes ranging from two to eight nodes. The question is whether a specific size would be best suited for DB2 Parallel Edition. At the hardware, configuration, and maintenance level, there is probably no significant advantage in using small versus large clusters. Since it is possible to have multiple independent pairings of resource groups within one cluster, it might even be more convenient to manage large eight-node clusters on the RISC/6000 SP. Nevertheless, there are several drawbacks associated with two-node clusters:

- If both nodes fail, there is no backup node. However, the probability of this happening is minimal.
- In a mutual takeover configuration, when one node fails, the surviving node must support twice its usual load, possibly leading to performance degradations.
- Since two-node clusters are usually associated with cascading resources, database operations are interrupted twice. Once when a node fails, and once when it reintegrates the cluster and the surviving node needs to release the failing node′s resources.

Large HACMP clusters can be configured to eliminate these problems, as will be seen in the next topic.

10.1.4 Standby Nodes or Mutual Takeover

The question of using standby nodes versus mutual takeover is linked to the previous point. In small clusters, it is usually too expensive to waste up to 50% of the computing resources by configuring standby nodes. In this case, it is more sensible to implement mutual takeover with cascading resources. For large clusters, however, it can be advantageous to use rotating resources, allowing for one standby node. In this case, it is possible to avoid the problem of degraded performance and double database operation interruption associated with two-node clusters.

For these reasons, we recommend two types of HACMP configurations for DB2 Parallel Edition. For smaller clusters, we recommend two-node clusters in mutual takeover setups with cascading resources. For larger clusters, we recommend four- to eight-node clusters with one standby node and rotating resources. The last issue is how many logical DB2 Parallel Edition nodes should be configured per RISC/6000 SP node.

10.1.5 Number of DB2 Parallel Edition Logical Nodes Per RISC/6000 SP Node

The *DB2 Parallel Edition Administration Guide*, SC09-1982 gives on page 17 several examples of DB2 Parallel Edition environments involving several logical DB2 Parallel Edition nodes per RISC/6000 SP node (see Figure 18 on page 91). From a HACMP perspective, this implementation allows you to minimize the performance degradations normally associated with mutual takeover configurations. When a node fails, the two DB2 Parallel Edition logical nodes are restarted on two different RISC/6000 SP nodes, leading to a 50% load increase on each takeover node, instead of 100% in a standard setup with one DB2 Parallel Edition logical node per processor. There still remains the question whether DB2 Parallel Edition applications are well suited to having two or more logical nodes per RISC/6000 SP processor.

In particular, the catalog node is accessed more frequently than the other nodes, which could lead to performance problems in some environments if several logical nodes are running on the catalog node. For this reason, it can even be meaningful to keep the catalog node as a DB2 Parallel Edition datafree node by constraining the data on the non-catalog nodes. With most DB2 Parallel Edition applications being CPU bound, it seems that having two logical nodes per RISC/6000 SP node brings more contention than benefits. Bottomline is, today, it makes more sense to have only one logical node per processor, except in the case of non-CPU-bound DB2 Parallel Edition applications.

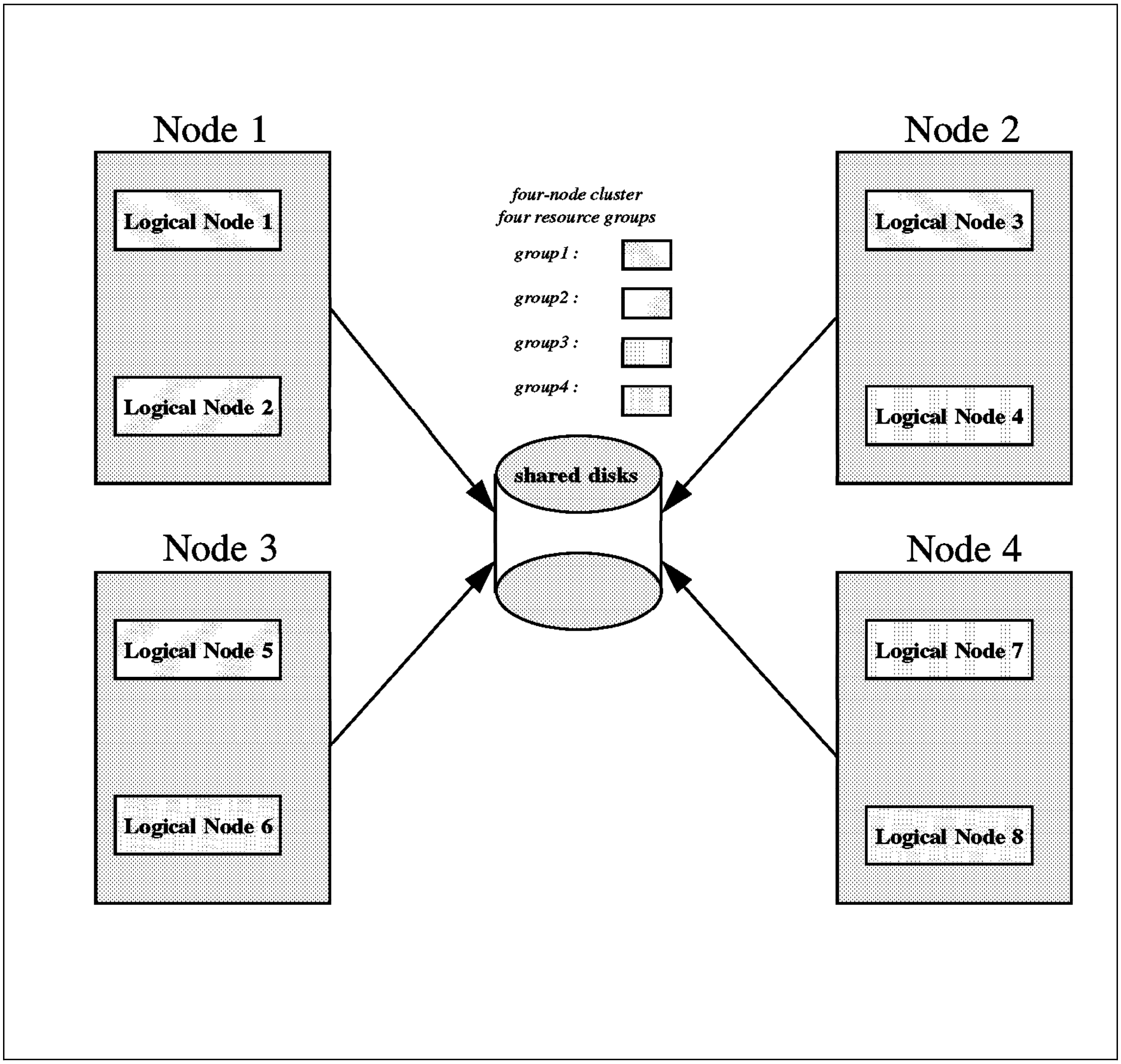


Figure 18. Four-Node Cluster with Two Logical Nodes Per Processor

10.1.6 Effect of Switch Restart on DB2 Parallel Edition

In a HACMP configuration, the Estart switch initialization command needs to be issued for each node takeover and each cluster reintegration. When Estart is issued, NFS and amd become shortly unavailable, since our test configuration mounts NFS and amd file systems over the switch. Running DB2 Parallel Edition transactions are delayed and resumed after Estart has completed. Connection to databases is not lost.

10.1.7 DB2 Parallel Edition Behavior in Case of Node Failure

This section gives some information on DB2 Parallel Edition behavior when one of the DB2 Parallel Edition nodes fails. This corresponds to the case where HACMP is not involved, and can serve as a reference for the minimum level of disruption one should aim at when implementing HACMP, seen from a DB2 Parallel Edition viewpoint. In 11.2.2.9, “DB2 Parallel Edition Logical Node Failure” on page 125, we will compare this best-case behavior to the actual one achieved in the test configuration.

10.1.7.1 Coordinator Node Failure

When the coordinator node for a transaction fails, the transaction needs to be restarted, and connection to the database is lost. When the node comes back up, the transaction is rolled back. Processing can be resumed after recovery has occurred, including database rollback and rollforward.

10.1.7.2 Data Node Failure

Since node recovery was not enabled in the DB2 Parallel Edition driver used for the tests, queries or updates will hang forever until the missing node is restarted and initiates recovery. Then, the user receives the SQL1229 (transaction rolled back because of a system failure) error message, and the transaction is rolled back. Refer to 11.2.2.1, “Amd Takeover” on page 121 for more information about the SQL1229 error message.

10.1.7.3 Data Node not Used in Transaction

If a node fails but it is not used in the current transaction, the transaction completes undisturbed. This is the case for instance if a transaction accesses a nodegroup that doesn′t include all nodes, and one of the non-used nodes becomes unavailable.

10.1.7.4 Catalog Node Failure

The current transaction receives the SQL1229 (see above) error message when the catalog node is restarted and initiates recovery. While the catalog node is down, system tables cannot be accessed, the redistribute command cannot be run, other nodes cannot be restored (the restore command needs a connection to the catalog node), access to static SQL packages is not possible, issuing DDL (Data Definition Language) statements is not possible, databases cannot be rollforwarded, and new connection attempts will return with SQL1229. This clearly shows how critical it is to place a special emphasis on the catalog node when implementing HACMP for DB2 Parallel Edition. With no catalog node, very little if anything can be done with DB2 Parallel Edition The connect reset command will timeout with SQL1475 and system error -25567 (connect reset successful but error occurred during termination). As far as existing connections to databases is concerned, they will be lost after MAX_CONNRETRIES (default 5) times CONN_ELAPSE (default 5s) seconds, which is 50s by default.

CONN_ELAPSE and MAX_CONNRETRIES can be modified through the db2 change database manager configuration command.

Chapter 11. Prerequisite for HACMP Installation and Installing HACMP

A number of preparation steps are required before installing HACMP. They include creating and modifying shared volume groups and file systems for HACMP, installing DB2 Parallel Edition, modifying amd, among other tasks.

11.1 Creating Shared Volume Groups

Each node holds a shared volume group named vgi_nodei, where i is the node number, from one to four. For instance, on node1, the commands used to create then activate a two-physical volume volume group on hdisk2 and hdisk3 would be, as user root:

```
mkvg -f -y vg1_node1 -n hdisk2 hdisk3
varyonvg vg1_node1
```

The -n option says that the volume group will not be automatically activated at system restart.

Similar commands would be used on the remaining nodes to create the corresponding volume groups. Refer to Figure 19 on page 96 for a description of the volume groups on the different nodes.

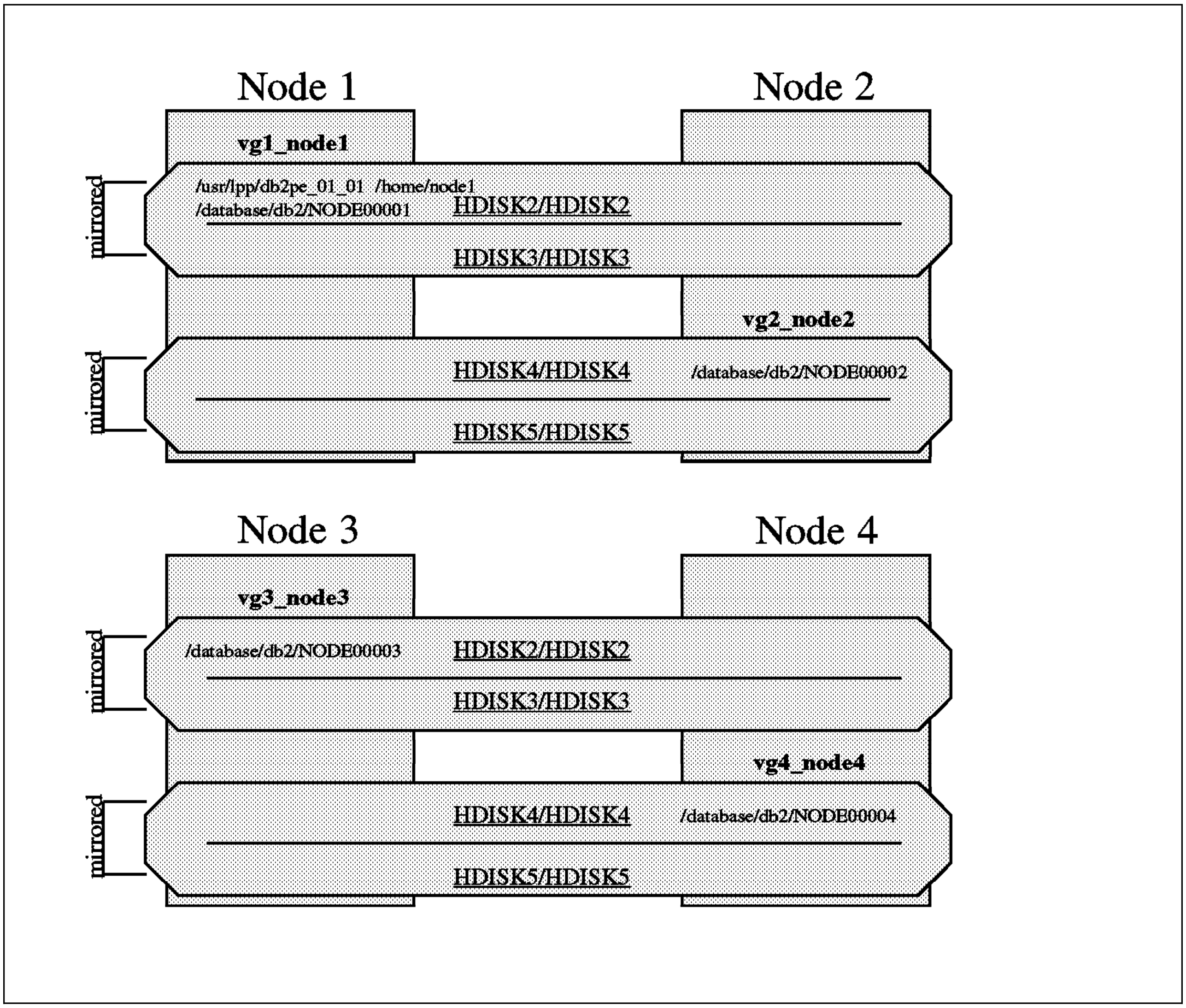


Figure 19. Shared Volume Group and File Systems Definitions

11.1.1 Creating Shared File Systems

On all nodes, we need to define a file system to store DB2 data. On node1, two additional shared file systems need to be defined: one to store DB2 Parallel Edition binaries, and one for the instance owner′s home directory (see Figure 19). Shared file systems should not have disk accounting activated.

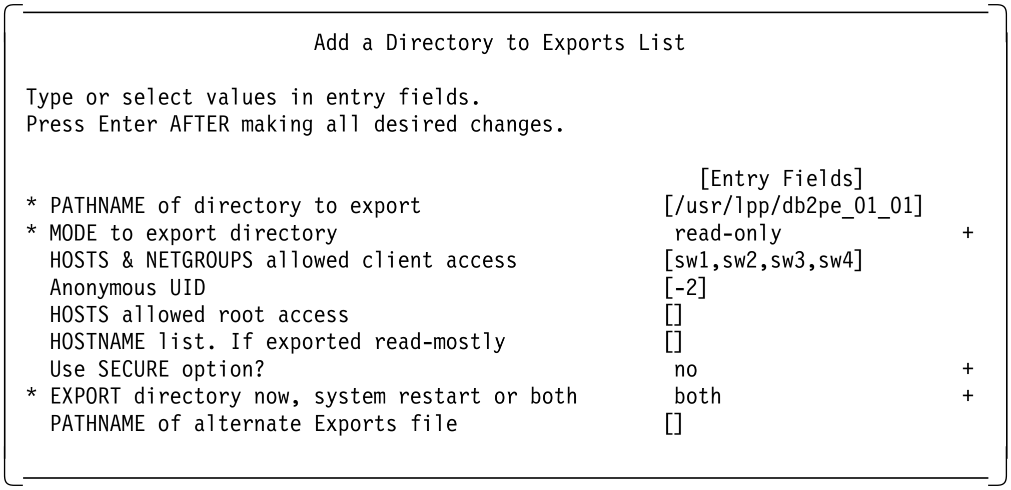
11.1.1.1 Creating the /usr/lpp/db2pe_01_01 File System

There are several methods of creating the file system and the procedure below is the way we chose to implement it in this chapter.

On node1, this file system can be created using the following command, as user root:

```
crfs -v jfs -g vg1_node1 -a size=120000 -m /usr/lpp/db2pe_01_01 \
-A no -p rw -t no -a frag=4096 -a nbpi=4096 -a compress=no
```

Note that we specified that the file system should not be mounted automatically at system restart. Since the file system will be NFS-mounted on the other nodes over the switch, we need to export it on node1 using the `smit mknfsexp` command:



```
                        Add a Directory to Exports List

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* PATHNAME of directory to export                    [/usr/lpp/db2pe_01_01]
* MODE to export directory                            read-only            +
  HOSTS & NETGROUPS allowed client access            [sw1,sw2,sw3,sw4]
  Anonymous UID                                      [-2]
  HOSTS allowed root access                          []
  HOSTNAME list. If exported read-mostly             []
  Use SECURE option?                                  no                   +
* EXPORT directory now, system restart or both        both                 +
  PATHNAME of alternate Exports file                 []
```

The main reason why we NFS-mount over the switch is not performance, but availability, since IP takeover cannot be configured for the SP Ethernet. However, as the number of nodes in the configuration increases, the switch will provide scalability.

On node2 to node4, make sure that the /usr/lpp/db2pe_01_01 directory entry exists. If it doesn′t, create it using the

```
mkdir /usr/lpp/db2pe_01_01
```

command.

11.1.1.2 Creating the /home/node1 File System

The /home/node1 file system is used by amd on node1 to store all RISC/6000 SP users′ default home directories. In our case, 50MB are enough. The file system can be created using a similar command as above :

```
crfs -v jfs -g vg1_node1 -a size=100000 -m /home/node1 \
-A no -p rw -t no -a frag=4096 -a nbpi=4096 -a compress=no
```

As previously, the file system should not be mounted automatically at system restart. It needs to be exported read-write to the other nodes using :

```
/usr/etc/mknfsexp -d /home/node1 -t rw -r sw2,sw3,sw4 -B
```

We could also have been through SMIT as shown above instead of typing the command explicitly.

11.1.1.3 Creating the /database/db2/NODE0000i File Systems

On node2, node3, and node4, there is only one shared file system, used to store DB2 data. This file system is rooted under the path specified in the DB2 `create database` command, called /database in our case. Because of the way DB2 Parallel Edition is designed, this directory needs to be defined on all nodes. However, when defining the shared file systems, we need to select different names on the different nodes. Otherwise it would not be possible to import the shared volume groups properly between two nodes, and importing the volume groups is a step required by HACMP. As a result, the naming convention for the database file systems is /database/db2/NODE0000i, with i going from one to four.

As an example, on node2, we create the /database/db2/NODE00002 file system by typing :

```
crfs -v jfs -g vg2_node2 -a size=100000 -m /database/db2/NODE00002 \
-A no -p rw -t no -a frag=4096 -a nbpi=4096 -a compress=no
```

The remaining file systems should be created accordingly on node1, node3, and node4.

11.1.2 Modifying amd

Amd should mount DB2′s home directory from node1 since the control workstation is not protected against failures in our setup. To protect the control workstation against failure, the possible solution will be to install the High Availability Control Workstation (HACWS). Since we do not have the HACWS option, we will attempt to become independent of the control workstation by eliminating possible points of failure associated with it. One of these points is amd.

On the control workstation, modify the amd.u file in the /etc/amd/amd-maps directory. The line concerning the DB2 user should read:

```
db2  host==node1;type:=link;fs:=/home/node1 \
         host!=node1;type:=nfs;rhost:=sw1;rfs:=/home/node1
```

This ensures that node1 becomes server for DB2′s home directory, and that the other nodes mount the file system over the sw1 switch interface. The modified map can then be propagated to the SP nodes by typing :

```
dsh -a /var/sysman/supper/update root.admin
```

Then, on node1, create the home directory for DB2 with the

```
mkdir /home/node1/db2
```

and

```
chown db2.db2 /home/node1/db2
```

commands. If the /home/node1 file system is not mounted, mount it first with the

```
mount /home/node1
```

command.

The last step is to refresh amd on all nodes by typing, on the control workstation:

```
dsh -a /etc/amd/refresh_amd
```

11.1.3 Enabling Disk Mirroring

Since the test configuration doesn′t involve RAID disks, it is necessary to protect ourselves against hardware disk failure. The way to achieve this is through AIX disk mirroring. The following file systems need to be mirrored :

- on node1 : /home/node1, /usr/lpp/db2pe_01_01, /database/db2/NODE00001
- on node2, node3, node4 : /database/db2/NODE0000i, with i going from 2 to 4

In addition, the log logical volume of each shared volume group vgi_nodei, i going from 1 to 4, has to be mirrored also. As an example, in order to mirror the /database/db2/NODE00001 file system on node1, defined over the lv00 logical volume, type:

```
mklvcopy lv00 2 hdisk2 hdisk3
```

This will create a copy of lv00 on disk hdisk3 from hdisk2. Repeat this operation for the other logical volumes on all the nodes.

Note that, in the test configuration, the serial link adapters and controllers are single points of failure. It would be possible to correct this by using multiple adapters and 9333 units and by mirroring the logical volumes across two adapters. In 11.2.1.16, “Using AIX Error Notification” on page 116, we will see that another way to eliminate this single point of failure is through AIX error notification.

Quorum Considerations

Quorum can be enabled or disabled. With quorum disabled, if a physical volume is not available, the volume group cannot be varied on unless varyonvg -f (force option) is used, leading to unpredictable results. On the other side, with quorum enabled, at least three disks are necessary in each volume group to authorize varyon after one disk failed, which might not always be feasible (like in the tested configuration). Moreover, since varyon is taken in charge by HACMP, disk failures can go undetected. HACMP guidelines are usually to disable quorum for non-concurrent access volume groups.

11.1.4 Enabling Target Mode SCSI

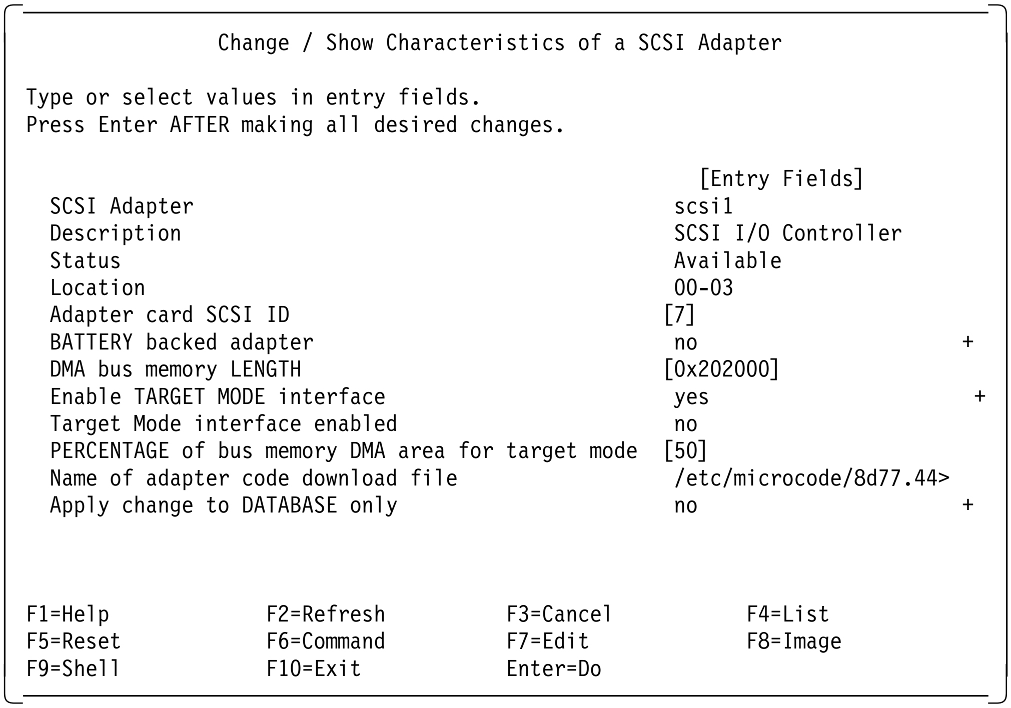
First, make sure the resistor blocks on the SCSI adapters have been removed and the adapters have different SCSI ids as explained in 9.1, “Hardware Configuration” on page 81. Then, on each node, put the SCSI adapter in the Defined state. The name of the SCSI adapter can be found by typing the following command:

```
lsdev -Cc adapter | grep scsi
```

Assuming that this command returned scsi1, enter :

```
rmdev -l scsi1
```

This will put the adapter in the defined state. To enable target mode SCSI, type `smit chgscsi`. Select your adapter, then press Enter. The following menu appears:



```
                 Change / Show Characteristics of a SCSI Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
  SCSI Adapter                                         scsi1
  Description                                          SCSI I/O Controller
  Status                                               Available
  Location                                             00-03
  Adapter card SCSI ID                                [7]
  BATTERY backed adapter                               no                    +
  DMA bus memory LENGTH                               [0x202000]
  Enable TARGET MODE interface                         yes                    +
  Target Mode interface enabled                        no
  PERCENTAGE of bus memory DMA area for target mode  [50]
  Name of adapter code download file                   /etc/microcode/8d77.44>
  Apply change to DATABASE only                        no                    +



F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Change the Enable Target Mode Interface field to yes, and press Enter. The next step is to make the adapter available :

```
cfgmgr
```

We should now have several files in the /dev directory, called respectively tmscsin.im for the initiator or sending interfaces, and tmscsin.tm for the target or receiving interfaces, with n going from 0 to 6.

After completing the above steps on all nodes, the connection between two nodes can be tested in the following way. As an example, on node3, we would type

```
cat < /dev/tmscsi0.tm
```

On node4, we can send data to node3 by typing:

```
cat /etc/hosts > /dev/tmscsi0.im
```

The contents of node4′s /etc/hosts file should be displayed on node3.

11.1.5 Renaming the Shared Logical Volumes

Since we are starting from an existing RISC/6000 SP environment where logical volumes probably have default names, it is necessary to change these. Each shared logical volume needs to have a unique name in the cluster. For instance, logical volume names on each node could end with the _<hostname> suffix, where <hostname> is the hostname of the node. In particular, it is necessary to rename the log logical volumes in each volume group, since they probably have the same default loglv00 name. In our configuration, we might have the following logical volumes :

- On node1 : loglv00, lv00, lv01, lv02
- On node2, node3, node4 : loglv00, lv00

They should be renamed into :

- On node1 : loglv00_node1, lv00_node1, lv01_node1, lv02_node1
- On node2, node3, node4 : loglv00_noden, lv00_noden, n from 2 to 4

As an example, the following commands will rename node4′s logical volumes :

```
chlv -n loglv00_node4 loglv00
chlv -n lv00_node4 lv00
```

After changing the name of the log logical volume, make sure you change the corresponding entries in the /etc/filesystems file.

11.1.6 Varying Off Shared Volume Groups on All Nodes

On node n, n going from 1 to 4, type:

```
varyoffvg vgn_noden
```

This will deactivate the shared volume groups on all nodes.

11.1.7 Importing Shared Volume Groups

Since we have two two-node clusters, we need to make node2′s volume group known on node1, and node1′s on node2, then repeat this step between node3 and node4. Assuming that vg3_node3 is defined on disks hdisk2 and hdisk3, as seen from node4, we would type :

```
importvg -y vg3_node3 hdisk2
varyonvg vg3_node3
```

This will import, then activate vg3_node3 on node4. On node3, we might have to specify :

```
importvg -y vg4_node4 hdisk4
varyonvg vg4_node4
```

This assumes that vg4_node4 is located on hdisk4 and hdisk5 as seen from node3. Similar commands need to be run on node1 and node2.

11.1.8 Changing Volume Groups on Destination Nodes

Shared volume groups should not be activated at system startup, since this operation is performed by HACMP. On node1, type:

```
chvg -a n Q y vg2_node2
```

On node2, type:

```
chvg -a n -Q y vg1_node1
```

Repeat these steps on node3 and node4 with the appropriate parameters.

11.1.9 Varying Off Volume Groups on Destination Nodes

On node1, type:

```
varyoffvg vg2_node2
```

On node2, type:

```
varyofvg vg1_node1
```

Then repeat these steps on node3 and node4 with the appropriate parameters.

11.1.10 Creating /.rhosts Files on All Nodes

On all nodes, create or add the following lines to the /.rhosts file as user root using your favorite text editor:

```
node1 root
sw1 root
sw1_boot root
switch1 root
node2 root
sw2 root
sw2_boot root
switch2 root
node3 root
sw3 root
sw3_boot root
switch3 root
node4 root
sw4 root
sw4_boot root
switch4 root
```

This /.rhosts file is required by HACMP for cluster startup and it is also used by the RISC/6000 SP. Make sure that the permissions on the .rhosts file are set to correct permission by using the following command:

```
chmod 600 /.rhosts
```

11.1.11 Updating the /etc/hosts File on All Nodes

Even if your environment has been set up to use a name server, it is recommended to include all boot and service adapters in the /etc/hosts file on all nodes to keep the cluster working in the event of the name server being unavailable. For the tested system, we would add the following lines to the /etc/hosts file on each node (refer to Figure 16 on page 85 for an overview of the network setup):

```
129.1.3.1     sw1
129.1.3.2     sw2
129.1.3.3     sw3
129.1.3.4     sw4
129.1.2.1     switch1
129.1.2.2     switch2
129.1.2.3     switch3
129.1.2.4     switch4
129.1.3.61    sw1_boot
129.1.3.62    sw2_boot
129.1.3.63    sw3_boot
129.1.3.64    sw4_boot
129.1.1.101   node1
129.1.1.102   node2
129.1.1.103   node3
129.1.1.104   node4
```

11.1.12 Creating DB2 Instance and Databases

The DB2 Parallel Edition instance is created on node1 using the DB2 `db2instance` command. The databases can then be created with the DB2 `create database` command, as user DB2. On node1 and node2, respectively, we could do :

```
db2 create database SAMPLE on /database
db2 create database DUMB on /database
```

11.1.13 HACMP Installation

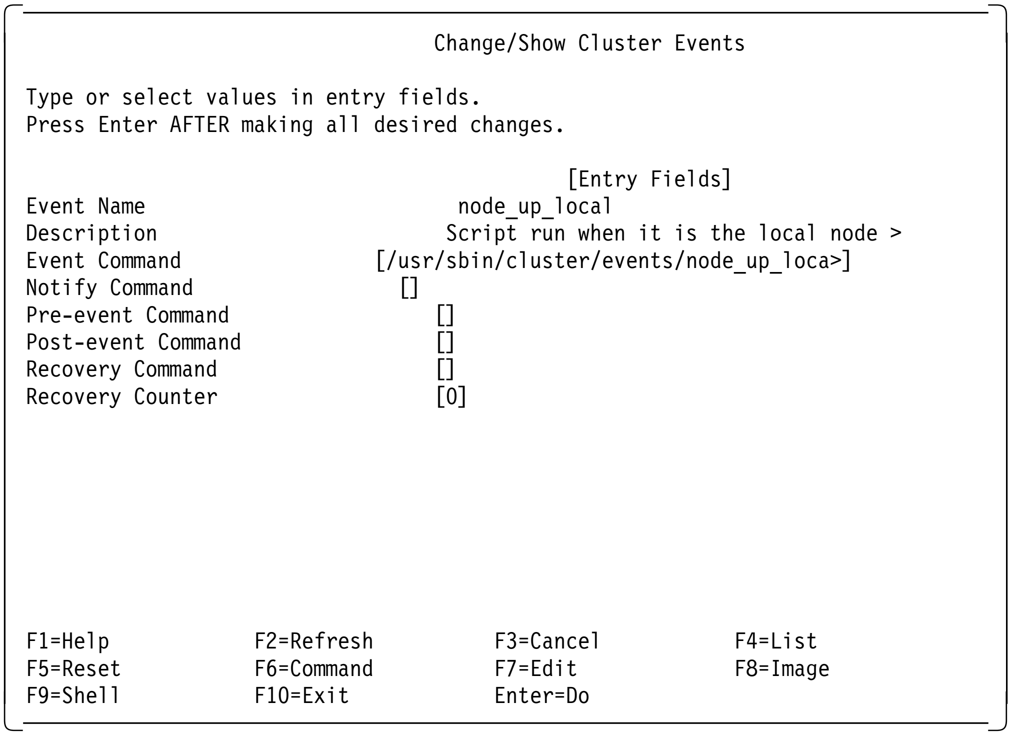
HACMP must be installed locally on each node. More specifically, the following products can be installed :

```
4.1.1.0  cluster.adt
        4.1.1.0  HACMP Client CLINFO Samples
        4.1.1.0  HACMP Client Clstat Samples
        4.1.1.0  HACMP Client Demos
        4.1.1.0  HACMP Client Demos Samples
        4.1.1.0  HACMP Client Include Files
        4.1.1.0  HACMP Client LIBCL Samples
        4.1.1.0  HACMP Server Demos
        4.1.1.0  HACMP Server Sample Demos
        4.1.1.0  HACMP Server Sample Images

4.1.1.0  cluster.base
        4.1.1.0  HACMP Base Client Libraries
        4.1.1.0  HACMP Base Client Runtime
        4.1.1.0  HACMP Base Client Utilities
        4.1.1.0  HACMP Base Server Diags
        4.1.1.0  HACMP Base Server Events
        4.1.1.0  HACMP Base Server Runtime
        4.1.1.0  HACMP Base Server Utilities

4.1.1.0  cluster.clvm
        4.1.1.0  HACMP for AIX Concurrent Access

4.1.1.0  cluster.man.en_US
        4.1.1.0  HACMP Client Man Pages - U.S. English
        4.1.1.0  HACMP Server Man Pages - U.S. English

4.1.1.0  cluster.vsm
        4.1.1.0  HACMP Visual System Management Configuration Utility
```

For our purposes, we need only cluster.base and cluster.man_en_US. Using `smit install_latest`, enter the name of the directory or device where the software resides, then select the products to be installed, and press Enter. You need about 5 to 10MB of free space in the /usr file system, depending on the options chosen. After the installation has completed, you should verify it using the /usr/sbin/cluster/diag/clverify utility. Select the software option, then follow instructions. If all nodes in the cluster have been installed from the same image, the verification step needs to be performed only once.

11.1.14 NFS-Mounting /usr/lpp/db2pe_01_01 on Node3 and Node4

To mount /usr/lpp/db2pe_01_01 on node3 and node4 over the switch from node1, we add a post-event script to the node_up_local HACMP script. Since other examples of pre- and post-event scripts will be given further below, we detail now the way this can be done using smit. By typing smit clcsclev.select, the list of HACMP event scripts is displayed. Select the event you want to add a post- or pre-event script to (node_up_local in our case). The following menu appears:



```
                                   Change/Show Cluster Events

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                              [Entry Fields]
Event Name                           node_up_local
Description                         Script run when it is the local node >
Event Command                [/usr/sbin/cluster/events/node_up_loca>]
Notify Command                 []
Pre-event Command                 []
Post-event Command                []
Recovery Command                  []
Recovery Counter                  [0]










F1=Help            F2=Refresh          F3=Cancel           F4=List
F5=Reset           F6=Command          F7=Edit             F8=Image
F9=Shell           F10=Exit            Enter=Do
```

In the Post-event Command field, add the name of the script you want to create. For maintenance reasons, it might not be a good idea to locate this script in the HACMP install directory, since its contents could be overwritten by applying a new release of the product. Instead, you might want to choose a separate directory to store the pre- and post-event scripts such as /usr/hacmp by creating the necessary entry with the mkdir command. The naming convention for the scripts could follow a simple scheme such as pre_ resp. post_<event name>. In the present case, the value of the Post-event Command field would be /usr/hacmp/post_node_up_local.

The contents of this script are:

```
#!/bin/sh
STATUS=0
/usr/sbin/cluster/events/utils/cl_activate_nfs 1 sw1 /usr/lpp/db2pe_01_01
if [ $? -ne 0 ]
then
        echo Failed to mount /usr/lpp/db2pe_01_01 from sw1
        echo Manual intervention required
        STATUS=1
fi
exit $STATUS
```

The script should be made executable by executing the `chmod +x /usr/hacmp/post_node_up_local` command, assuming that it belongs to the root user.

11.2 HACMP Configuration

In the following, we follow an incremental approach to high availability, starting from a relatively simple two-node mutual takeover configuration to progressively include more resources in the failover scenario. We define two HACMP clusters, one between node1 and node2, and one between node3 and node4. The first cluster is the most complex one, since it will be necessary to protect ourselves against NFS, amd, or switch Eprimary node failure. For this reason, we will start with the second cluster.

Each time it was necessary to change one of the HACMP scripts, we tried to do it through the use of post- and pre-event scripts. This makes software maintenance easier since these scripts are not affected by new releases of HACMP or PTFs as direct changes to the base HACMP scripts would.

11.2.1 Configuring the Cluster between Node3 and Node4

As a reminder, node3 and node4 share the same 9333 disk unit, with each node having its shared data mirrored on external serial disks. A SCSI cable between the nodes provides a non-IP heartbeat network. The configuration steps described below can be found with more detail in the *HACMP 4.1 for AIX Installation Guide*, SC23-2769.

11.2.1.1 Defining the Cluster ID and Name

On node3, enter `smit cm_config_cluster.add`. Select for instance 1 for Cluster ID and node3_node4 for Cluster Name, then press Enter.

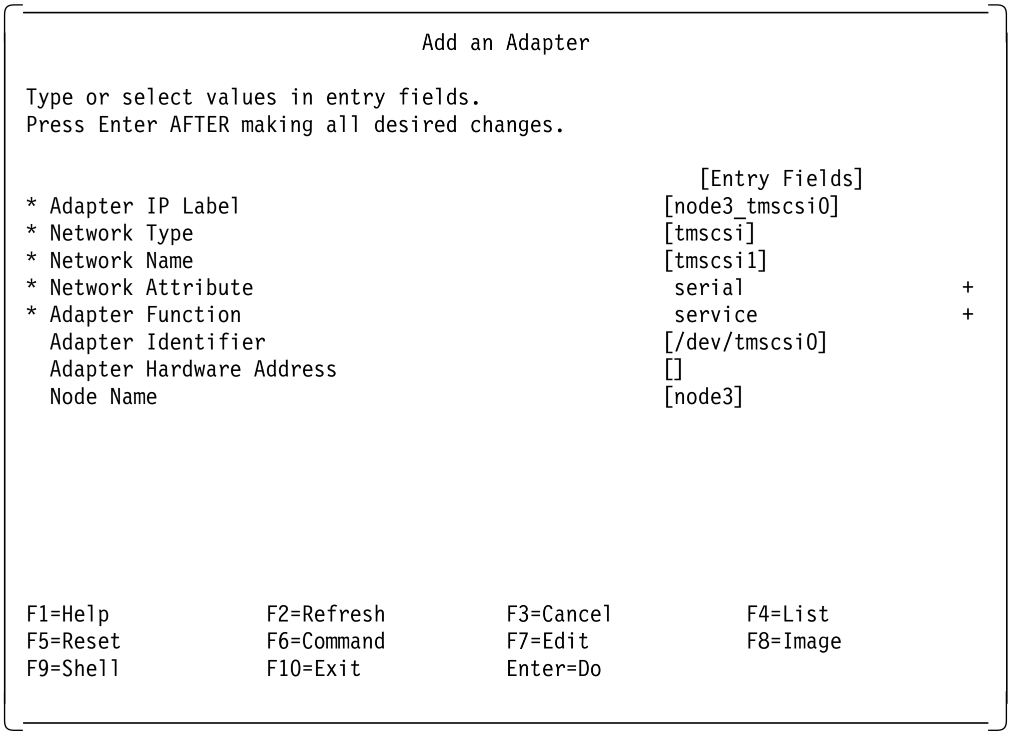
11.2.1.2 Defining Nodes

On node3, enter `smit cm_config_nodes.add`. Set Node Names to node3 node4, then press Enter.

11.2.1.3 Defining Adapters

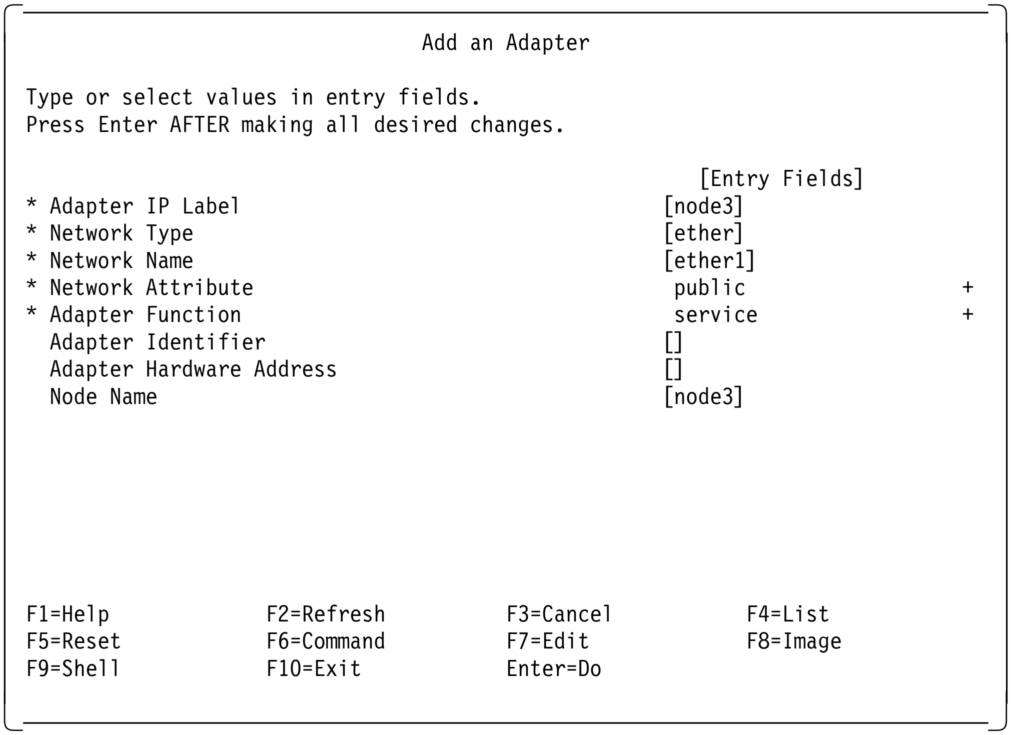
For this initial cluster, we have three adapters per node: one switch adapter, and one tmscsi and SP Ethernet adapter for heartbeat communications. The reason why we define two networks for heartbeat is explained in 10.1, “SP Ethernet Considerations” on page 89.

On node3, type `smit cm_config_adapters.add`. Since there are three adapters per node and two nodes, this command will have to be repeated six times. For node3′s tmscsi adapter, the fields should look like:



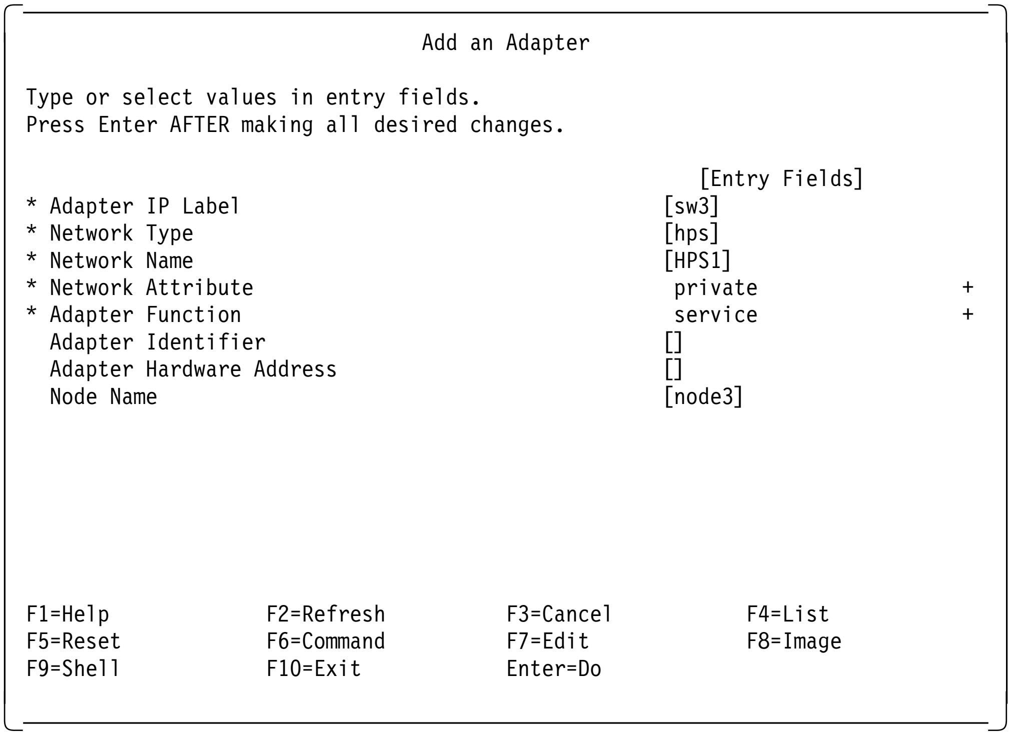
```
                                 Add an Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Adapter IP Label                                   [node3_tmscsi0]
* Network Type                                       [tmscsi]
* Network Name                                       [tmscsi1]
* Network Attribute                                   serial                 +
* Adapter Function                                    service                +
  Adapter Identifier                                 [/dev/tmscsi0]
  Adapter Hardware Address                           []
  Node Name                                          [node3]







F1=Help            F2=Refresh         F3=Cancel          F4=List
F5=Reset           F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

For node3′s Ethernet adapter, do:



```
                                 Add an Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Adapter IP Label                                   [node3]
* Network Type                                       [ether]
* Network Name                                       [ether1]
* Network Attribute                                   public                 +
* Adapter Function                                    service                +
  Adapter Identifier                                 []
  Adapter Hardware Address                           []
  Node Name                                          [node3]







F1=Help            F2=Refresh         F3=Cancel          F4=List
F5=Reset           F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

For node3′s switch adapter, do:



```
                                Add an Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Adapter IP Label                                 [sw3]
* Network Type                                     [hps]
* Network Name                                     [HPS1]
* Network Attribute                                 private            +
* Adapter Function                                  service            +
  Adapter Identifier                               []
  Adapter Hardware Address                         []
  Node Name                                        [node3]








F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Note that we don′t use the SDR switch address (switch3). Instead, HACMP knows only the sw3 address, which is defined on a different sub-network than the SDR switch addresses. Another point is that the network name for the switch should contain the string HPS, which will be required below when we enable IP address takeover. The network type should always be private for the HPS.

In the SMIT screens above, the Adapter Identifier field has been left blank. This is possible because HACMP looks up the address in the /etc/hosts file.

The remaining adapters can then be easily defined to HACMP in a way similar to that shown above.

SDR Switch Addresses

Don′t use the SDR switch address as boot address for HACMP. This might result in the nodes hanging with an error message.

11.2.1.4 Synchronizing Cluster Definition on All Nodes

This step will copy the ODM definitions entered on node3 to node4. Execute smit cm_cfg_top_menu, then select Synchronize Cluster Topology, and press Enter.

11.2.1.5 Configuring Application Servers

Since we want DB2 Parallel Edition to be restarted when one of the nodes in the cluster failed, we have to create a start script and a stop script for DB2 Parallel Edition. For availability reasons, we put these scripts in the instance owner′s home directory. Each node has its own set of start/stop scripts. As an example, we give template start and stop scripts for node3. They proved to work reliably in our configuration, but will probably have to be modified in order to be integrated into other environments. The start script looks like:

```
#!/bin/ksh
set -x
# DB2 home directory
HOME_DIR=/u/db2
cd $HOME_DIR

# which DB2 Parallel Edition node do we have to start
nodenum=3

DB2INSTANCE=su - db2 -exec env 2>/dev/null |grep DB2INSTANCE |awk -F= ’{print $2}’
if ps -fu $DB2INSTANCE |grep ″db2sysc  ${nodenum}″ |grep -v grep
then
# already running
exit 0
fi

# switch interface name
switch_name=netstat -i |grep -v Link |grep sw${nodenum} |awk ’/css0/ {print $4}’

# temp file used to issue the db2 start command
TEMP_FILE=/tmp/ha_${nodenum}
rm -rf $TEMP_FILE
AWK_FILE=/tmp/awk_${nodenum}
rm -rf $AWK_FILE

# what SP2 node are we on
hostname=hostname


echo Starting DB2 Parallel Edition node $nodenum on host $hostname through
interface $switch_name

# find hostname associated with nodenum
cat >| $AWK_FILE << EOF
/[0-9]*/   {if ((\$1 == $nodenum) && (\$3 == 0)) print \$2}
EOF
hostname_of_nodenum=cat sqllib/db2nodes.cfg |awk -f $AWK_FILE
if [[ $hostname = $hostname_of_nodenum ]] || [[ $hostname_of_nodenum = ″″ ]]
then
# case of a local start. Only one PE logical node per RISC/6000 SP node
port_number=0
else
# case of a remote start. We have 2 DB2 Parallel Edition logical nodes on the same
# RISC/6000 SP node
port_number=1
fi

# create temp file
cat >| $TEMP_FILE << EOF
j=0
while (( j == 0 ))
do
if ! db2start nodenum $nodenum restart hostname $hostname \
netname $switch_name port $port_number |grep SQL6036
# start or stop in progress
then
j=1
fi
```

```
done
EOF

# make temp file executable
chmod ogu+x $TEMP_FILE

# execute as user db2
su - db2 -exec ″$TEMP_FILE″
```

The core of the script is the db2 restart command. The syntax of this command is db2start nodenum W restart hostname X netname Y port Z. This means that DB2 Parallel Edition logical node W (corresponding to the first column of the db2nodes.cfg DB2 Parallel Edition configuration file) will be restarted on hostname X, using Y as network interface, on port Z. Typically, Z is either 0 or 1, the latter being selected when two logical nodes are started on the same hostname or RISC/6000 SP node. The db2 start with the restart option will update the db2nodes.cfg file and override the old values.

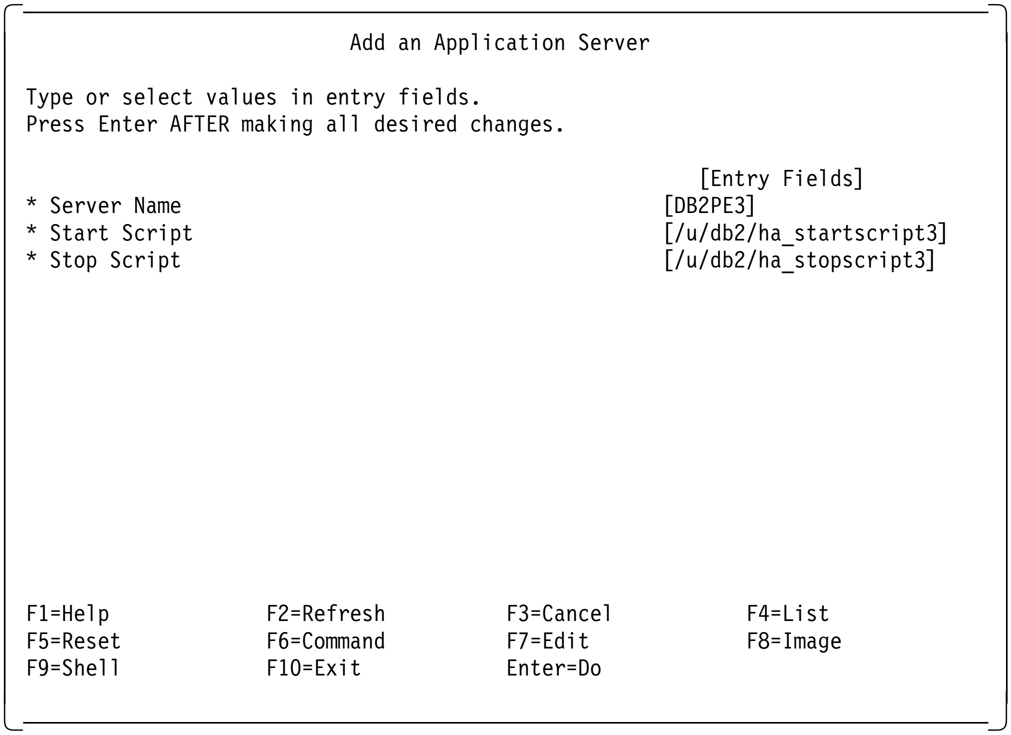
The stop script follows:

```
#!/bin/ksh
HOME_DIR=/u/db2
cd $HOME_DIR
nodenum=3
TEMP_FILE=/tmp/ha_${nodenum}
rm -rf $TEMP_FILE
hostname=hostname

DB2INSTANCE=su - db2 -exec env 2>/dev/null |grep DB2INSTANCE |awk -F= ′{print $2}′
if ! ps -fu $DB2INSTANCE |grep ″db2sysc  ${nodenum}″ |grep -v grep
then
# already stopped
exit 0
fi

echo Stopping DB2 Parallel Edition node $nodenum on host $hostname
cat >| $TEMP_FILE << EOF
# force out applications on all nodes
db2 force application all
# wait for applications to finish
j=0
while (( j == 0 ))
do
if db2 list applications |grep SQL1611 || db2 list applications | grep SQL1032
then
j=1
fi
done
# make sure no other start/stop in progress
j=0
while (( j == 0 ))
do
if ! db2stop nodenum $nodenum |grep SQL6036     # start or stop in progress
then
j=1
fi
done
EOF
chmod ogu+x $TEMP_FILE
```

```
su - db2 -exec "$TEMP_FILE"
```

Before issuing db2stop, the stop script forces all DB2 Parallel Edition applications out. The alternative would be to make the script a little bit more clever in order to force out only those applications that need access to the DB2 Parallel Edition logical node we wish to stop. Since most applications need access to all nodes, we felt that it was not worth checking whether the applications would need access to the local node or not. This wouldn't make any difference in most situations. This script highlights the benefits one would derive from using rotating resources instead of cascading ones. With cascading resources, all applications are forced out twice, the first time when a node fails, the second time when it reintegrates the cluster, which can be unacceptable for some critical environments.

To define the start and stop scripts to HACMP, we define two application servers, called DB2PE3 and DB2PE4, respectively on node3 and node4. As an example, for DB2PE3, enter `smit claddserv.dialog`:



```
                          Add an Application Server

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* Server Name                                         [DB2PE3]
* Start Script                                        [/u/db2/ha_startscript3]
* Stop Script                                         [/u/db2/ha_stopscript3]













F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

The application server information is propagated automatically to node4.

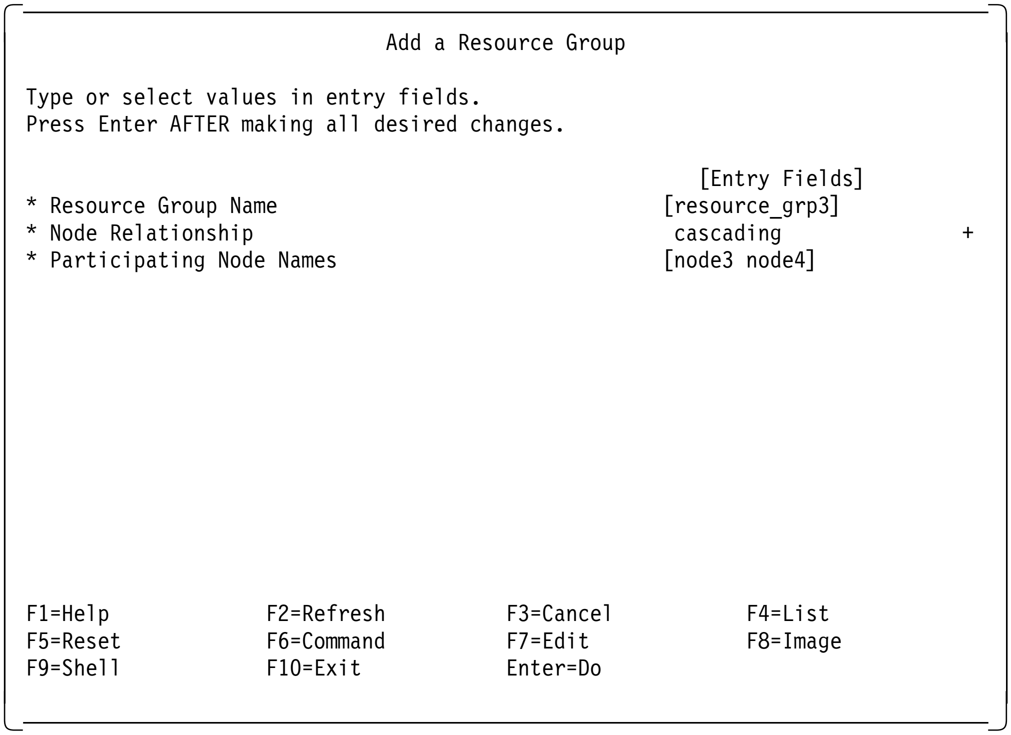
DB2 start problem

Make sure that the statd and lockd daemons are running on each node before starting DB2 Parallel Edition. Otherwise, SQL error 5005 will be returned. These daemons are part of the nfs group that can be started with the *startsrc -g nfs* command.

11.2.1.6 Configuring Resource Groups

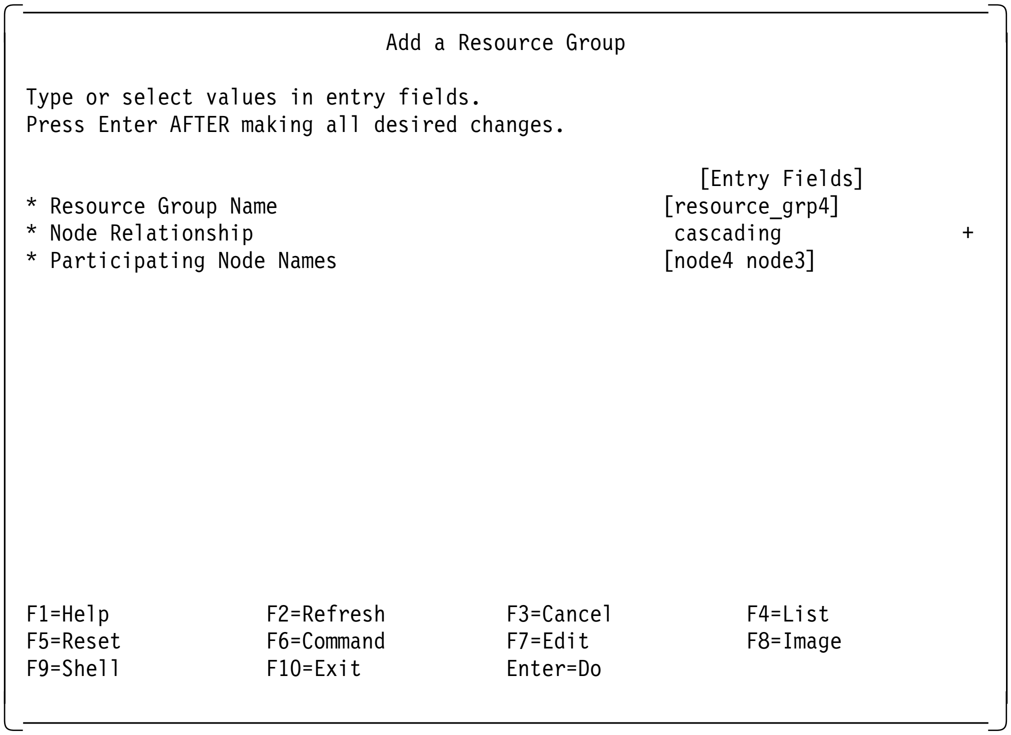
We have one resource group on each node, respectively called resource_grp3 and resource_grp4. The resource groups contain the shared disks and volume groups, file systems, network interfaces, and application servers for each node. Since we are not using standby nodes, required for rotating resource groups, nor the concurrent logical volume manager for concurrent access resource groups, the groups must be cascading resource groups.

To define the resource_grp3 resource group, type `smit cm_add_grp`, then enter the following information:



```
                               Add a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* Resource Group Name                                  [resource_grp3]
* Node Relationship                                     cascading              +
* Participating Node Names                             [node3 node4]









F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Repeat this step for the resource_grp4 resource group:

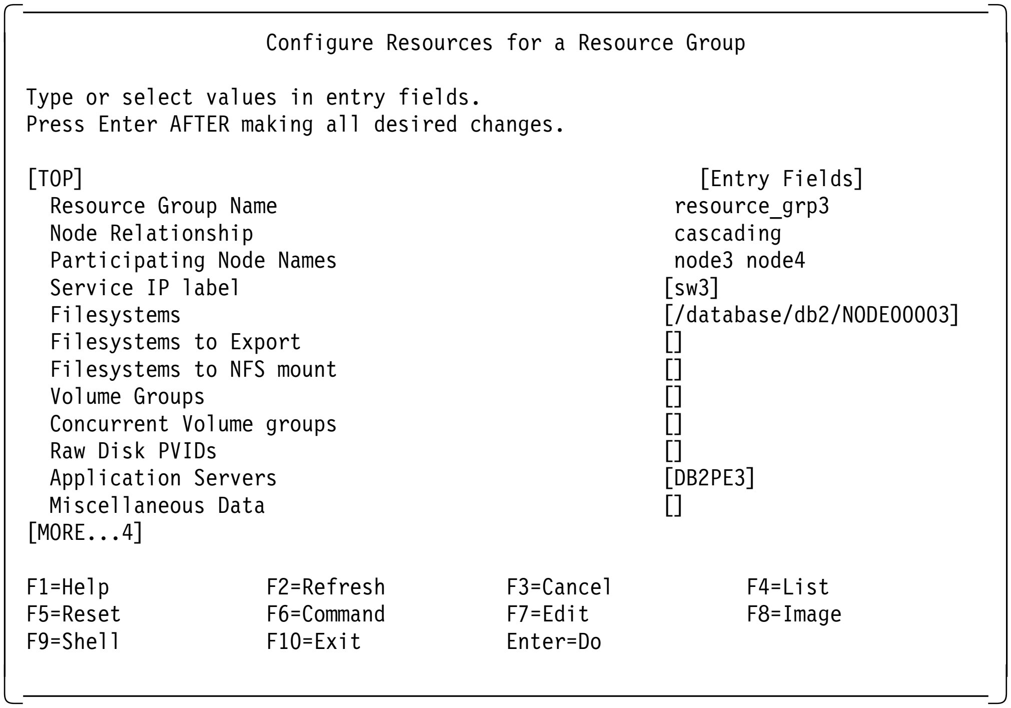


```
                              Add a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                           [Entry Fields]
* Resource Group Name                                   [resource_grp4]
* Node Relationship                                      cascading              +
* Participating Node Names                              [node4 node3]














F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Note that the first node in the node list specifies the high priority node, sometimes called the owner, of the resources.

11.2.1.7 Configuring Resources for Resource Groups

Resources need now to be configured for the resource_grp3 and resource_grp4 resource groups. For resource_grp3, type smit cm_cfg_res.select, select group, then enter the following data:



```
                    Configure Resources for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                      [Entry Fields]
  Resource Group Name                                    resource_grp3
  Node Relationship                                      cascading
  Participating Node Names                               node3 node4
  Service IP label                                      [sw3]
  Filesystems                                           [/database/db2/NODE00003]
  Filesystems to Export                                 []
  Filesystems to NFS mount                              []
  Volume Groups                                         []
  Concurrent Volume groups                              []
  Raw Disk PVIDs                                        []
  Application Servers                                   [DB2PE3]
  Miscellaneous Data                                    []
[MORE...4]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Repeat this step for resource_grp4, changing the resources to those appropriate for that resource group. Note that the Volume Groups and Raw Disk PVIDs fields need not be specified since we are using filesystems.

11.2.1.8 Synchronizing Node Environment

Propagate the above information to node4 by executing `smit cm_cfg_res_menu` and selecting the Synchronize Cluster Resources option.

11.2.1.9 Verifying Cluster Configuration

Run the *cluster* option of the /usr/sbin/cluster/diag/clverify utility on node3.

11.2.1.10 Configuring clinfo (Client Nodes)

The clinfo (cluster.client) program is required in order to update the arp caches after IP address takeover. It is also needed to use the clstat cluster status monitoring program. Clinfo uses a configuration file, located in /usr/sbin/cluster/etc/clhosts. Edit this file, comment out the line starting with 127.0.0.1, and add the following lines:

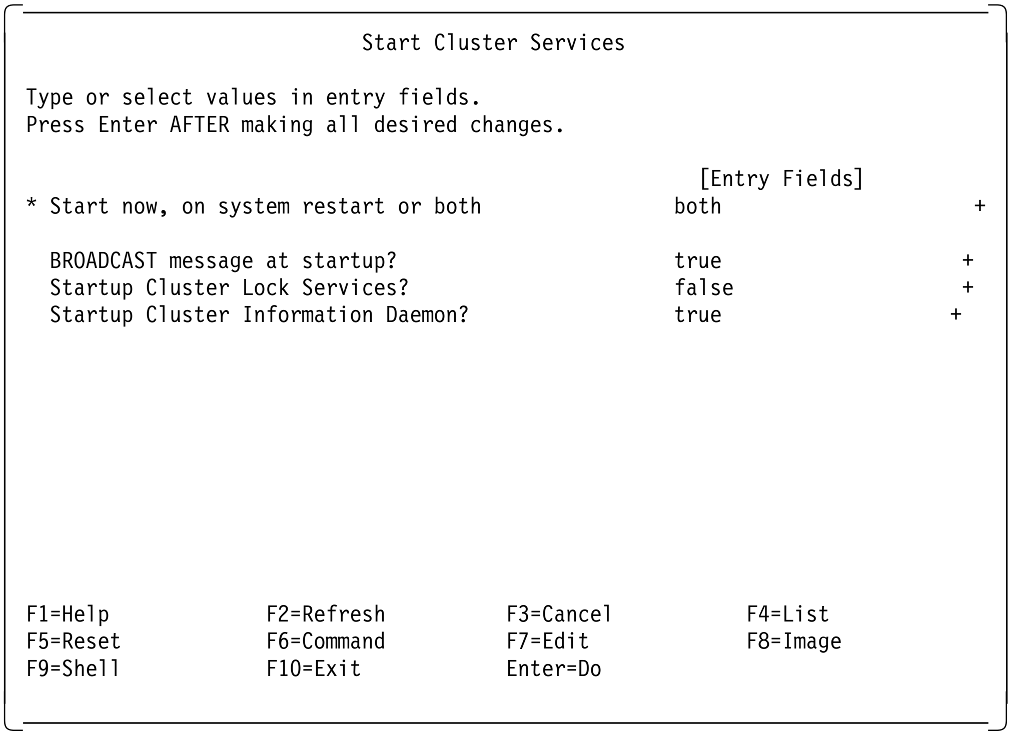
```
sw1
sw2
sw3
sw4
```

If we had clients, we would add their names to the PING_CLIENT_LIST variable in the /usr/sbin/cluster/etc/clinfo.rc file. You may want to include any service addresses in your system such as Ethernet addresses.

In order for HACMP to work properly, add /usr/sbin/cluster/utilities and /usr/sbin/cluster/events/utils to the PATH variable in the /.kshrc file on the control workstation. This assumes that the /.kshrc file is propagated to the SP nodes through the file collection mechanism, as is usually the case if file collections are enabled. If they are not, modify this procedure to fit your environment.

11.2.1.11 Starting Cluster Services

On node3 and node4, type `smit clstart.dialog`:



```
                          Start Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Start now, on system restart or both              both                  +

  BROADCAST message at startup?                     true                 +
  Startup Cluster Lock Services?                    false                +
  Startup Cluster Information Daemon?               true                +








F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Start only one node at a time, and let each node complete its startup before starting the next one. The startup is not complete when the SMIT displays OK. Instead, it is necessary to monitor the HACMP log files (/var/adm/cluster.log or /tmp/hacmp.out) to see when the startup is complete.

What have we achieved so far? We have a running two-node HACMP cluster with cascading resources in a mutual takeover configuration. From a DB2 Parallel Edition perspective, we are protected against:

- Disk failure, through mirroring
- TCP/IP failure for the heartbeat daemon, through tmscsi
- CPU failure, through shared file systems

We are not protected against:

- Switch adapter failure
- Switch primary node failure
- Switch power failure or global failure
- Node or frame power failure
- Node main memory failure
- Serial line failure
- Disk adapter or controller failure

These points will be addressed below. For the time being, in normal operating mode, each node (node3 and node4) has its own DB2 Parallel Edition logical node. In case of a node′s failure, the surviving node will take over the other node′s DB2 Parallel Edition data, releasing it when the failing node reintegrates the cluster. Time for takeover, assuming no DB2 Parallel Edition rollback and rollforward is required, is less than one minute. During node reintegration, the switch is reinitialized automatically by HACMP.

Troubleshooting

To avoid being asked by the operating system to change DB2′s password at logon time, which would interfere with the execution of the DB2 Parallel Edition start and stop scripts, delete the line starting with *flags=* in the /etc/security/passwd file on the control workstation (if you are using file collections) for the DB2 user.

11.2.1.12 Switch Primary Node Failure

In order to activate Eprimary node takeover, execute the `/usr/sbin/cluster/events/utils/cl_HPS_Eprimary manage` command on node3. This feature ensures switch availability even in the case of the Eprimary node being unavailable. Should node3 be defined as Eprimary (by executing the Eprimary node3 command on the control workstation) and fail, HACMP will move the Eprimary node to node4. The Eprimary resource is defined as a rotating resource, so that node4 would keep the Eprimary function until it becomes unavailable and node3 becomes Eprimary again.

As we will see below, it makes more sense to define the Eprimary function in the other cluster including node1 and node2. If the switch is not running, HACMP cannot start properly. If NFS is down, it cannot work either. In our setup, NFS files are mounted over the switch. Since node1 is the NFS server, it should also be the Eprimary node. The unmanage option of the cl_HPS_Eprimary command can be used to move the Eprimary takeover function to another cluster. Only one cluster can have this feature enabled.

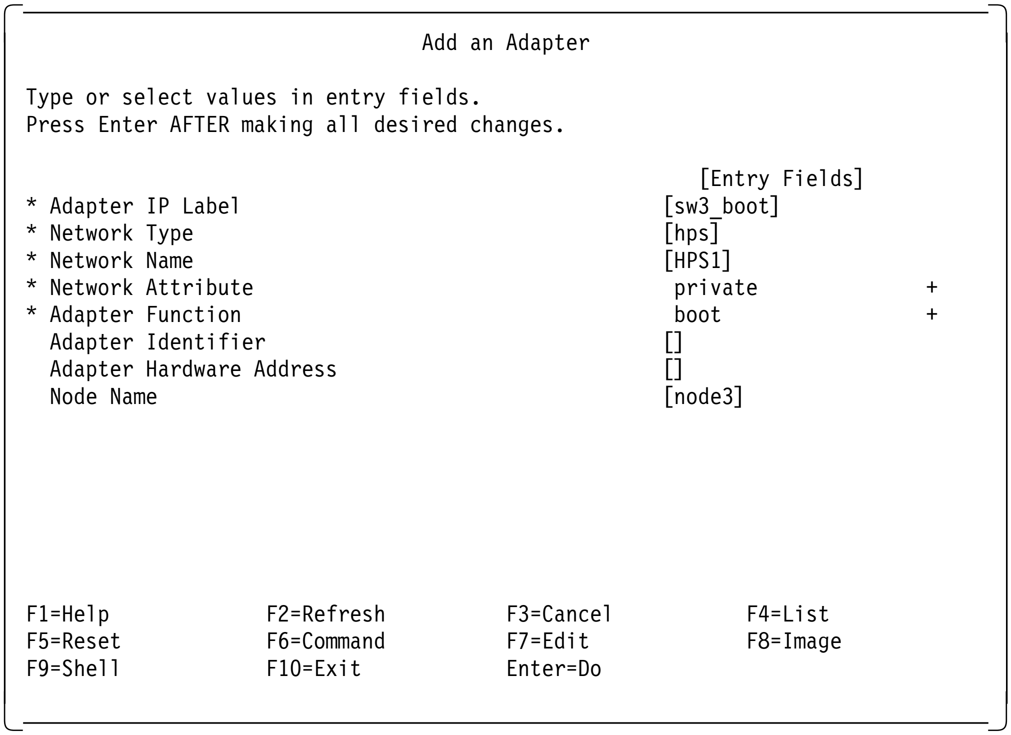
11.2.1.13 Switch IP Address Takeover

Because of the way the DB2 start with the restart option works, DB2 Parallel Edition doesn′t really need switch IP address takeover. The failing DB2 Parallel Edition logical node is restarted through another network interface on the surviving node. For clients and applications, however, it can certainly be useful to implement IP address takeover. The following points need to be considered:

- ARP must be enabled for the HPS network. You can check whether this is the case by executing `smit list_node_switch` on the control workstation. The RISC/6000 SP documentation recommends to enable ARP in all situations. If ARP is not enabled, change the boot response field to customize, enable ARP, and press Enter. Then, network boot the SP nodes by selecting the global commands submenu from the spmon -g interface.
- HACMP HPS network names must contain the HPS string.
- HPS networks must be private networks.
- Standby addresses are not required. Since IP takeover cannot use the SDR switch addresses, we use the swn and swn_boot addresses as service and boot addresses, with n ranging from 1 to 4. These addresses are on a different subnet than the SDR switch addresses.

11.2.1.14 Add Boot Interface Definitions

With `smit cm_config_adapters.add`, add the HACMP adapter definitions for sw3_boot and sw4_boot. The network name is the same as previously (HPS1). For sw3_boot, we would enter the following information on node3:



```
                              Add an Adapter

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Adapter IP Label                                   [sw3_boot]
* Network Type                                       [hps]
* Network Name                                       [HPS1]
* Network Attribute                                   private              +
* Adapter Function                                    boot                 +
  Adapter Identifier                                 []
  Adapter Hardware Address                           []
  Node Name                                          [node3]







F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Repeat this step for sw4_boot.

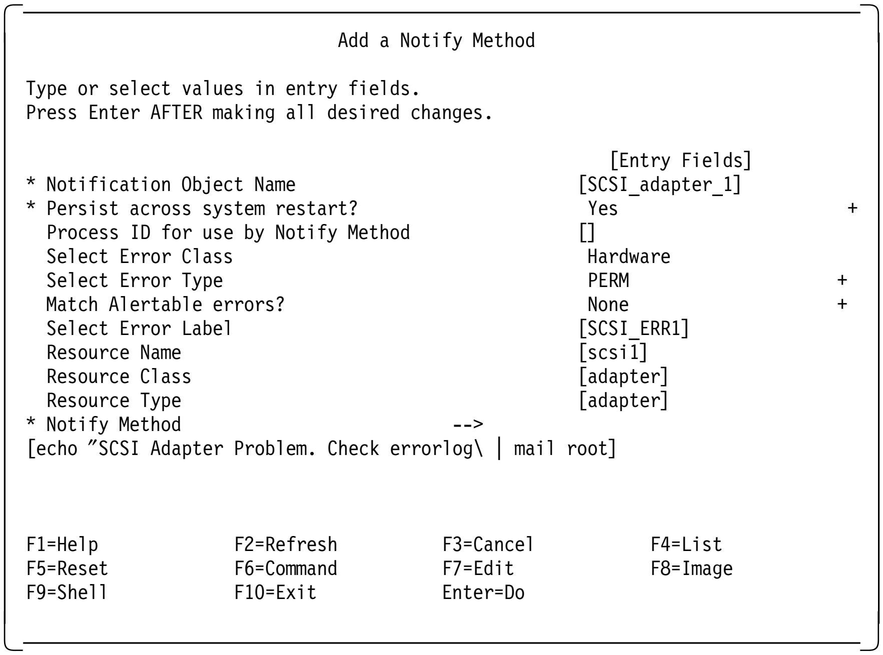
11.2.1.15 Synchronizing Cluster Topology

Execute smit cm_cfg_top_menu to propagate the node information to node4. At this point, when node3 fails, node4 assumes node3′s service address sw3, ending up with three defined addresses on the switch adapter: switch4, sw4, and sw3, on two separate subnets.

11.2.1.16 Using AIX Error Notification

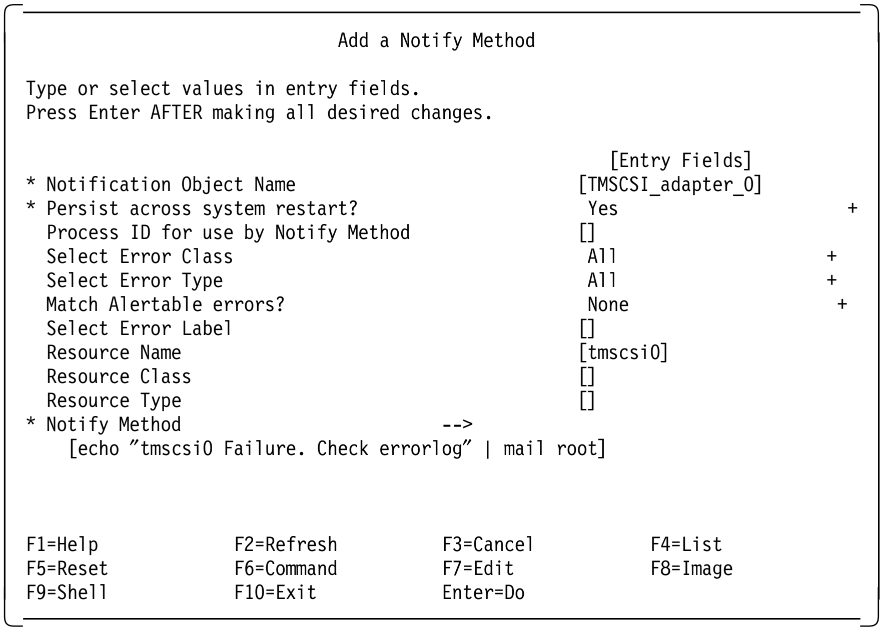
HACMP provides a way to associate user-defined scripts (so-called Notify methods) with errors logged by the AIX error notification services. The steps shown below should be repeated on all cluster nodes. In each case, the information to be entered can be accessed by typing `smit cm_add_notifymeth.dialog`. More error messages can be found in the *IBM RISC System/6000 Scalable POWERparallel Systems Diagnosis and Messages Guide*, GC23-3899 and in the *AIX Version 4.1 Problem Solving Guide and Reference*, SC23-2606 to suit other environments′ needs.

- For SCSI failure on the tmscsi serial line, we sent a mail message to the system administrator to make him/her aware of the problem. Since failure of the tmscsi is not critical as long as there is a TCP/IP connection available, it is acceptable to wait for some off-shift period until the adapter will be serviced.



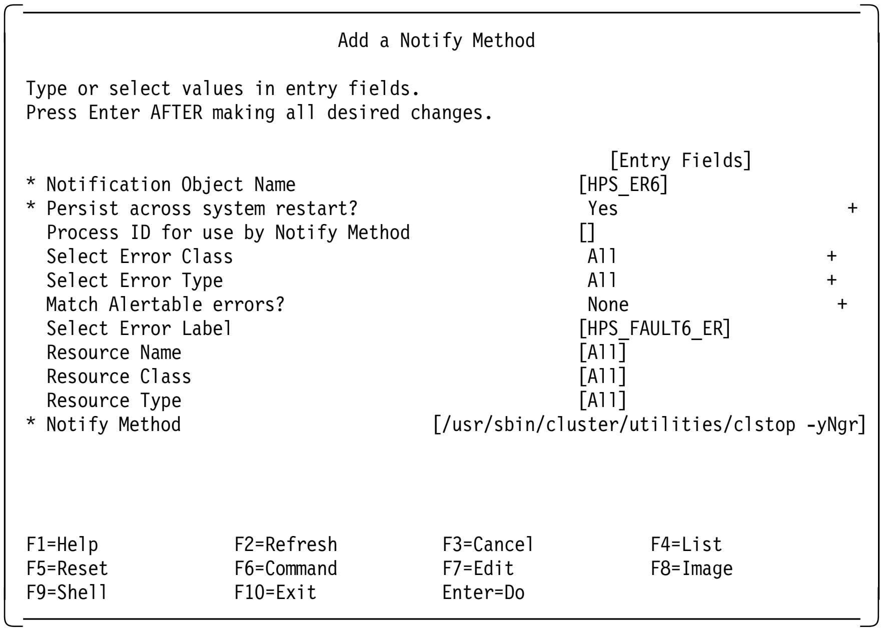
```
                               Add a Notify Method

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Notification Object Name                         [SCSI_adapter_1]
* Persist across system restart?                    Yes                    +
  Process ID for use by Notify Method              []
  Select Error Class                                Hardware
  Select Error Type                                 PERM                  +
  Match Alertable errors?                           None                  +
  Select Error Label                               [SCSI_ERR1]
  Resource Name                                    [scsi1]
  Resource Class                                   [adapter]
  Resource Type                                    [adapter]
* Notify Method                              -->
[echo "SCSI Adapter Problem. Check errorlog\ | mail root]



F1=Help            F2=Refresh         F3=Cancel          F4=List
F5=Reset           F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

- The same approach as above can be used for failures of the tmscsi0 serial interface.



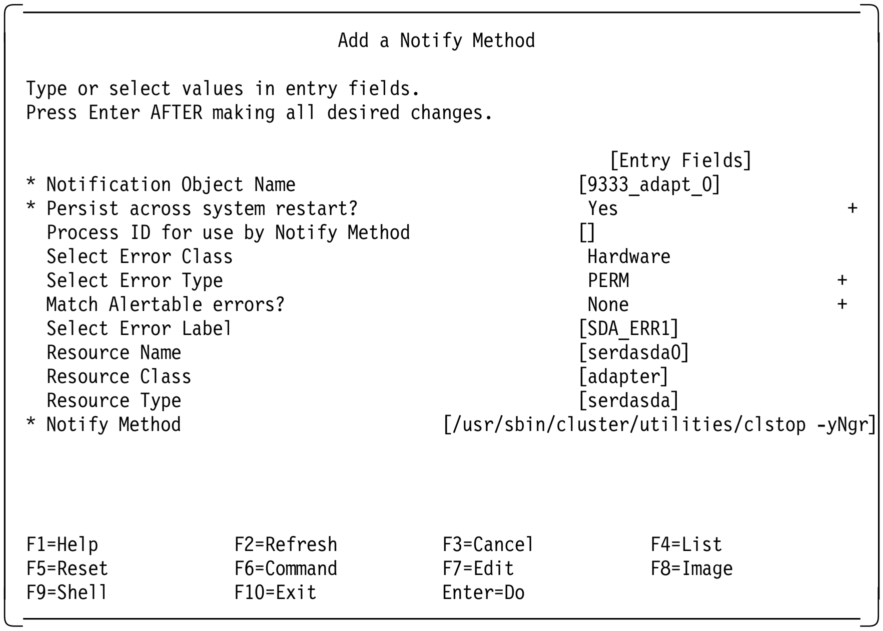
```
                               Add a Notify Method

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Notification Object Name                         [TMSCSI_adapter_0]
* Persist across system restart?                    Yes                    +
  Process ID for use by Notify Method              []
  Select Error Class                                All                 +
  Select Error Type                                 All                 +
  Match Alertable errors?                           None                  +
  Select Error Label                               []
  Resource Name                                    [tmscsi0]
  Resource Class                                   []
  Resource Type                                    []
* Notify Method                              -->
    [echo "tmscsi0 Failure. Check errorlog" | mail root]



F1=Help            F2=Refresh         F3=Cancel          F4=List
F5=Reset           F6=Command         F7=Edit            F8=Image
F9=Shell           F10=Exit           Enter=Do
```

- As far as switch failures goes, there are two interesting error messages: HPS_ER9, for switch adapter failures, and HPS_ER6, for switch adapter, micro-channel bus slot, or external clock source failures, leading to termination of the Worm process. In both cases, it is necessary to determine whether the failure happened on both cluster nodes, or only locally. In the first case, we should perform a network failover, or send a message to the root user if there is no backup network available (like in our sample configuration). In the second case, graceful shutdown with takeover is performed, which allows DB2 Parallel Edition to re-establish communications with the failing logical database node on the surviving RISC/6000 SP node. For the second alternative, one would do, for HPS_ER6:



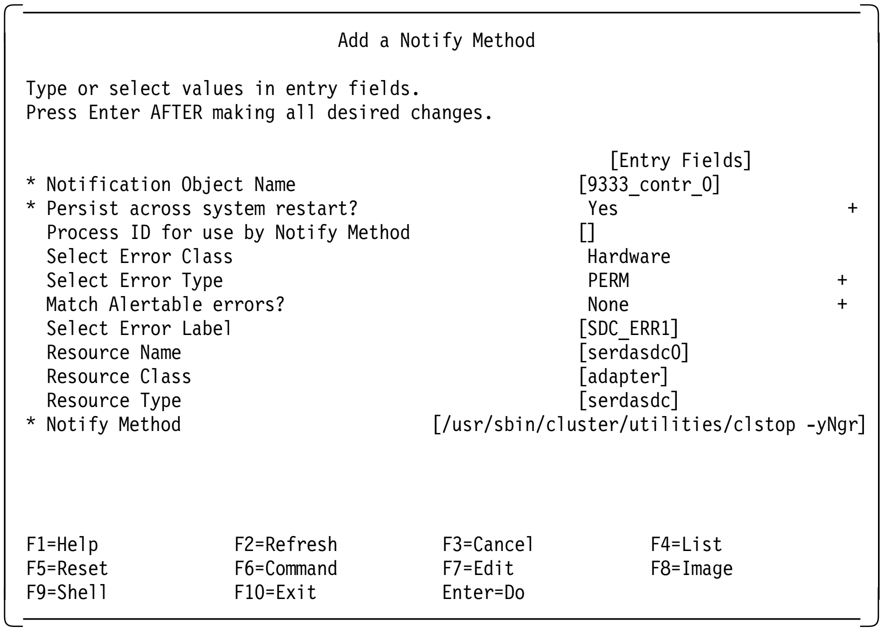
```
                              Add a Notify Method

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* Notification Object Name                          [HPS_ER6]
* Persist across system restart?                     Yes                    +
  Process ID for use by Notify Method               []
  Select Error Class                                 All                  +
  Select Error Type                                  All                  +
  Match Alertable errors?                            None                  +
  Select Error Label                                [HPS_FAULT6_ER]
  Resource Name                                     [All]
  Resource Class                                    [All]
  Resource Type                                     [All]
* Notify Method                       [/usr/sbin/cluster/utilities/clstop -yNgr]




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

- For 9333 adapter failures, we do a graceful shutdown with takeover like previously, to make sure that resources are migrating to the backup node.



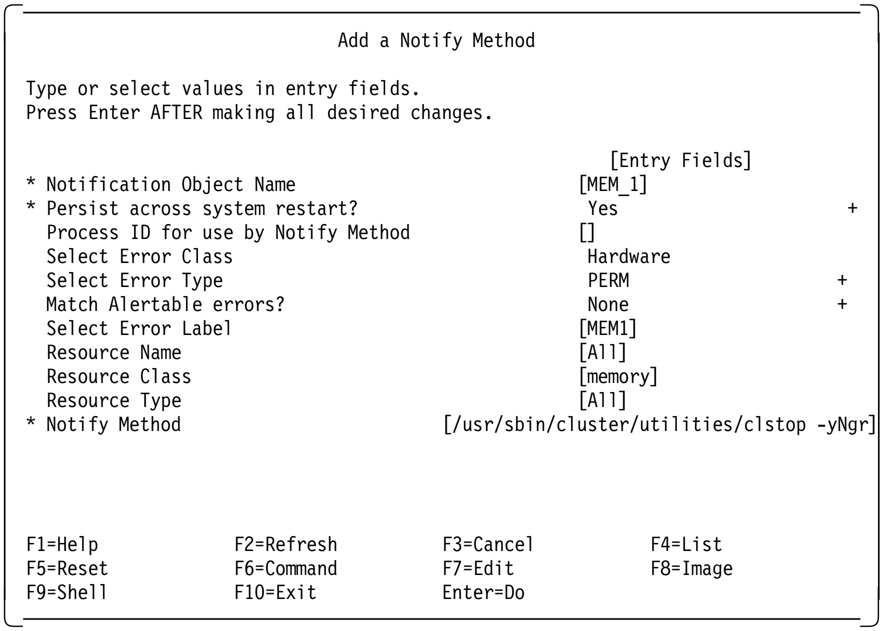
```
                              Add a Notify Method

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
* Notification Object Name                          [9333_adapt_0]
* Persist across system restart?                     Yes                    +
  Process ID for use by Notify Method               []
  Select Error Class                                 Hardware
  Select Error Type                                  PERM                  +
  Match Alertable errors?                            None                  +
  Select Error Label                                [SDA_ERR1]
  Resource Name                                     [serdasda0]
  Resource Class                                    [adapter]
  Resource Type                                     [serdasda]
* Notify Method                        [/usr/sbin/cluster/utilities/clstop -yNgr]




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

- For 9333 controller failures, do:



```
                              Add a Notify Method

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Notification Object Name                           [9333_contr_0]
* Persist across system restart?                      Yes                       +
  Process ID for use by Notify Method                []
  Select Error Class                                  Hardware
  Select Error Type                                   PERM                     +
  Match Alertable errors?                             None                     +
  Select Error Label                                 [SDC_ERR1]
  Resource Name                                      [serdasdc0]
  Resource Class                                     [adapter]
  Resource Type                                      [serdasdc]
* Notify Method                        [/usr/sbin/cluster/utilities/clstop -yNgr]




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

- For memory failures, do:



```
                              Add a Notify Method

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Notification Object Name                           [MEM_1]
* Persist across system restart?                      Yes                       +
  Process ID for use by Notify Method                []
  Select Error Class                                  Hardware
  Select Error Type                                   PERM                     +
  Match Alertable errors?                             None                     +
  Select Error Label                                 [MEM1]
  Resource Name                                      [All]
  Resource Class                                     [memory]
  Resource Type                                      [All]
* Notify Method                         [/usr/sbin/cluster/utilities/clstop -yNgr]




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

The same procedure should be used for the MEM2 and MEM3 error labels. MEM1 indicates the absence of a memory card out of a memory card pair, MEM2 indicates the failure of up to two SIMMs on a memory card, and MEM3 indicates the failure of a memory card out of a memory card pair.

11.2.1.17 Node and Frame Power Recovery

RISC/6000 SP frames contain between one and three AC/DC 48 volt power supplies, depending on the type of frame. If the N+1 feature is installed, this implies that there are at least two power supplies per frame. A failure with one of these power supplies will not interrupt the operation of the RISC/6000 SP because of the backup that will be provided by the other power supply. Furthermore, the defective power supply is hot_pluggable and can be replaced without interruption to the system. Each power supply can service up to eight nodes. With this in mind, if you have more than eight nodes per frame, it becomes necessary to consider configuring three power supplies to protect your system against unforeseen power supply failure. However, there is only one power cord from the external power network to the SP. Customers should implement uninterruptible power sources (UPS) to guard themselves against a global power loss. The same is true concerning the power source for the external disks.

11.2.1.18 Activating I/O Pacing

To avoid HACMP having to compete with I/O bound applications for the CPU, it is a good idea to activate I/O pacing. With smit chgsys, set the HIGH water field to 33, and the LOW water field to 24. This will guarantee a correct failover behavior for HACMP in most environments.

11.2.1.19 Starting amd

Since DB2′s home directory is mounted over the switch from sw1, it is necessary that HACMP be started before amd is started. By default, amd is started in the /usr/lpp/ssp/install/bin/services_config script in /etc/rc.sp. The following five lines in this script should be commented out as follows:

```
if [ -x $SSP_INSTALL_BIN/amd_config ]; then
        $SSP_INSTALL_BIN/amd_config ; fi
if [ $? -ne 0 ]; then
          $SPMSG sminstall $MSGCAT emsg042 ′%1$s: 0016-042 Msg not found.\
Error found while executing the %2$s configuration script.\n′ $0 ″amd_config″
fi
```

Then, create (on node1 and node2) or modify (on node3 and node4) the post_node_up_local post-event script with the following contents:

```
#!/bin/sh
STATUS=0
# start amd
SSP_INSTALL_BIN=″/usr/lpp/ssp/install/bin″
MSGCAT=″/usr/lib/nls/msg/C/sminstall.cat″
SPMSG=″/usr/lpp/ssp/bin/spmsg_basic″
AMD_CONFIG=TRUE
export AMD_CONFIG
if [ -x $SSP_INSTALL_BIN/amd_config ]; then
        $SSP_INSTALL_BIN/amd_config ; fi
if [ $? -ne 0 ]; then
          $SPMSG sminstall $MSGCAT emsg042 ′%1$s: 0016-042 Msg not found.\
Error found while executing the %2$s configuration script.\n′ $0 ″amd_config″
STATUS=1
fi
exit $STATUS
```

The steps described above should be repeated on all cluster nodes. Refer to 11.1.14, “NFS-Mounting /usr/lpp/db2pe_01_01 on Node3 and Node4” on page 104 for how to configure post-event scripts for HACMP using smit.

11.2.2 Configuring the Cluster between Node1 and Node2

To start with, the steps presented above for the cluster comprised of node3 and node4 should be repeated for node1 and node2. The differences are:

- Section 11.1.14, “NFS-Mounting /usr/lpp/db2pe_01_01 on Node3 and Node4” on page 104 should not be completed on node1 and node2.

- Assuming the resource groups are now called resource_grp1 and resource_grp2, for resource_grp1, we have to specify:

  ```
  Filesystems to Export = /usr/lpp/db2pe_01_01 /home/node1
  Filesystems to NFS mount = /usr/lpp/db2pe_01_01
  ```

- The vg1_node1 volume group must have the same major number on node1 and node2, because it contains NFS-mounted file systems. This number can be specified when importing and creating the volume group. Available major numbers are given by the lvlstmajor AIX command. To change an existing volume group′s major number, it is possible to export it then import it and specify the major number in the `importvg` command.

- As said in 11.2.1.12, “Switch Primary Node Failure” on page 115, the Eprimary node should be node1 since node1 is also the NFS server. Assuming that the node3_node4 cluster is still managing Eprimary takeover, type:

  – on node3:

    ```
    /usr/sbin/cluster/events/utils/cl_HPS_Eprimary unmanage
    ```

  – on node1 :

    ```
    /usr/sbin/cluster/events/utils/cl_HPS_Eprimary manage
    ```

  – on the control workstation:

    ```
    Eprimary node1
    ```

11.2.2.1 Amd Takeover

In the current setup, node2 will take over node1′s file systems in the event of node1′s failure. For amd, this means that /home/node1 will be exported from node2 instead of node1. With IP takeover enabled, and considering that the amd map is the same as when node1 is up and running, node2 will mount /home/node1 over NFS from himself through the sw1 network interface. As a result, /home/node1 will be present twice on node2 : once locally, and once as NFS-mounted file system from himself. For those who find this unacceptable, it would be necessary to modify the amd map for /home/node1 before restarting amd on node2. This certainly would generate several tricky problems which can be easily bypassed by taking the proposed approach.

From a DB2 Parallel Edition perspective, if the amd server goes down, no command can be issued as user db2. Running transactions are normally not affected since the db2 executable is already in the local memory. Commands can be issued again when the server comes back up or takeover has occurred. Connection to DB2 is not lost. With HACMP, however, transactions are terminated with SQL error message SQL1229 when takeover takes place, corresponding to the DB2 Parallel Edition logical node being restarted. This is because, when DB2 Parallel Edition starts, it first needs to rollback all uncompleted transactions, then rollforward to the last commit. For the sake of completeness, we give here the text of error SQL1229, showing the behavior of various utilities when they are rolled back :

```
 SQL1229N The current transaction has been rolled back
          because of a system error.
Explanation:  A system error, such as node failure or
connection failure, has occurred.  The application is rolled
back to the previous COMMIT.
Note that in the case of DB2 utility functions, the behavior
is described below:
```

```
Import - The application is rolled back. If the COMMITCOUNT
       parameter was used, the operation is rolled back to a
       previous committed point.
Reorg  - The operation is aborted and must be resubmitted.
Redistribute - The operation is aborted, however, some of
       the operation may have been successful. Issuing the
       request again will restart the operation from the
       point of failure.
Rollforward - The operation is aborted and the database is
       still in rollforward pending state. The command must
       be resubmitted.
Backup/Restore - The operation is aborted and must be
       resubmitted.
User Response:  Try the request again.  If the error
persists, you can find more information about the problem in
the syslog files.  It may be necessary to contact the system
administrator for assistance since the most common reason
for this error is that a node has failed.
Note that in an SP environment where the high speed switch
is used, this error can be a symptom of a failure in the
high speed switch.
The sqlerrd[5] field of the SQLCA will indicate the node
number that detected the node failure.  On the node that
detected the failure a message will be placed in the syslog
that identifies the failed node.
```

11.2.2.2 NFS Takeover

If a node comes up while node1 is down, /usr/lpp/db2pe_01_01 will be automatically mounted as soon as node1 becomes available. The same holds for amd. Users will be able to sign on as db2 as soon as node1 becomes available.

From a DB2 Parallel Edition perspective, if the NFS server goes down, all db2 commands hang, since they are accessed via NFS. They are resumed when the server becomes available again, or when takeover has occurred. Connection to DB2 is not lost. Running transactions are usually not affected since they normally don′t need access to /usr/lpp/db2pe_01_01 to complete. With HACMP, however, transactions will terminate with SQL1229 as explained in the previous point.

In the tested configuration, assuming that no DB2 Parallel Edition recovery is required, failover time in case of NFS/amd node (node1) failure is about 20 seconds.

11.2.2.3 Modifying the cl_activate_fs Script

Because node2 normally mounts /home/node1 from node1 through the sw1 network interface, HACMP won′t mount /home/node1 locally during takeover, since it believes that the file system is already mounted. To correct this, the /usr/sbin/cluster/events/utils/cl_activate_fs script should be modified on node2 only. The line starting with

```
mount | awk ′{ print $2 }′ | fgrep -s -x ″$fs″
```

should be replaced with

```
V1=df | awk ′{ print $7 }′ | grep ″$fs″
V2=df | grep ″$fs″ | awk ′{ print $1 }′ | grep ″^\/dev\/″
if [ ″$V1″ != ″″ ] && [ ″$V2″ != ″″ ]
```

The next line, `if [ $? -eq 0 ]`, must be commented out by adding a # sign at the beginning of the line. This is one of only two occurrences throughout these chapters about HACMP where it is necessary to change the HACMP scripts directly because the use of post- or pre-event scripts is either not possible or quite complex.

Note that it is not convenient to specify /home/node1 as one of the resources Filesystems to NFS Mount in the resource_grp1 resource group, since /home/node1 is mounted by amd and not by NFS directly. The approach taken here is easier to implement.

11.2.2.4 Modifying the cl_deactivate_nfs Script

The node_down_remote script, called by the surviving node when the other one has failed, executes the cl_deactivate_nfs script. By default, cl_deactivate_nfs will try to unmount the /usr/lpp/db2pe_01_01 file system by first killing all processes having open file descriptors in this file system, then issuing a umount -f command in a never ending loop. For DB2 Parallel Edition, this means that HACMP is going to kill node2′s logical node when node1 goes down. To avoid this, comment out the line in cl_deactivate_nfs (located under /usr/sbin/cluster/events/utils) containing the cl_nfskill command on node2 :

```
#         cl_nfskill -k -u $fs
```

Then, replace the last seven lines of the script with the following ones:

```
        if [ $? -eq 0 ]
# changed for DB2 Parallel Edition
        then
          umount -f $fs
          # needed to get exit code = 0
          echo ″...″
        fi
#        if [ $? -eq 0 ]
#        then
#          until umount -f $fs
#          do
#               sleep 2
#          done
#        fi
```

Instead of looping on the umount command, we do it only once. If the filesystem is not busy, it will be unmounted. If it is busy, it won′t work any better if we try to unmount it several times. We had to modify the HACMP event script directly, since it is not possible to implement a post- or pre-event script in this particular case.

11.2.2.5 Adding a Post-Event Script to the node_up_remote_complete Script

When node1 reintegrates the cluster, the cl_deactivate_fs script is executed on node2. This script will call the AIX fuser command, which will kill those processes having open file descriptors in the filesystems to be unmounted. For DB2 Parallel Edition, this means that the logical node is going to be killed. Since applications are forced out during cluster reintegration, stopping node2′s logical node won′t make any difference for most applications. To restart DB2 Parallel Edition on node2 after node1′s reintegration, create, as explained in 11.1.14, “NFS-Mounting /usr/lpp/db2pe_01_01 on Node3 and Node4” on page 104, the post_node_up_remote_complete script as a post-event script to the node_up_remote_complete event as follows:

```
#!/bin/sh
STATUS=0
     # added forDB2 Parallel Edition
     i=0
     while (( i < 240 ))
     do
     if ! su - db2 -exec date 2>&1 |grep ″Unable to change″
     then
     break
     fi
     sleep 3
     (( i = i + 1 ))
     done
     if (( i == 240 ))
     then
     echo Problem with amd...Manual intervention required
     STATUS=1
     fi

/u/db2/ha_startscript2

if [ $? -ne 0 ]
then
     STATUS=1
fi
exit $STATUS
```

The lines before the call to ha_startscript2 prevent starting DB2 Parallel Edition too early, before amd has stabilized. This assumes that we know the name of the instance.

11.2.2.6 Adding a Post-Event Script to the stop_server Script

As explained in the previous section, by default, HACMP will kill node2′s logical node when node1 reintegrates the cluster. To avoid a brute force kill of DB2, create the post_stop_server post-event script to the stop_server event as explained in 11.1.14, "NFS-Mounting /usr/lpp/db2pe_01_01 on Node3 and Node4" on page 104 on node2:

```
#!/bin/sh
STATUS=0
     /u/db2/ha_stopscript2

     if [ $? -ne 0 ]
     then
          STATUS=1
     fi
exit $STATUS
```

11.2.2.7 Adding a Pre-Event Script to start_server

We create the pre_start_server script as a pre-event script to the start_server event on node2, to make sure that start_server is not issued before amd has stabilized (refer to 11.1.14, "NFS-Mounting /usr/lpp/db2pe_01_01 on Node3 and Node4" on page 104 for how to configure pre- and post-event scripts using smit):

```
#!/bin/sh
STATUS=0
        i=0
        while (( i < 30 ))
        do
        if ! su - db2 -exec date 2>&1 |grep ″Unable to change″
        then
        break
        fi
        sleep 3
        (( i = i + 1 ))
        done
        if (( i == 30 ))
        then
        echo Problem with amd...Manual intervention required
        STATUS=1
        fi
exit $STATUS
```

11.2.2.8 Adding a Post-Event Script to node_down_remote_complete

During failover on node2, we observed that, sometimes, restarting logical node 1 on node2 would cause node2′s logical node to terminate. To avoid this, we create the post_node_down_remote_complete script as a post-event script to the node_down_remote_complete event as follows:

```
#!/bin/sh
/u/db2/ha_startscript2
```

This has no effect if node2′s logical node is already running, as it should.

11.2.2.9 DB2 Parallel Edition Logical Node Failure

This section describes the tested system′s behavior in the event of a DB2 Parallel Edition logical node′s failure, depending on node type. This should be compared to the behavior of DB2 Parallel Edition without HACMP.

- Coordinator node failure: all applications need to reconnect to the databases. The behavior is the same as without HACMP.
- Data node failure: after takeover (within one minute of failure), DB2 Parallel Edition needs to recover. Then the SQL1229 (transaction rolled back) error message is issued when the logical node is restarted, and transactions are rolled back. Because the ha_stopscript stop script forces all applications out, connections to databases are lost. Until the logical DB2 Parallel Edition node is restarted, transactions hang, just as when HACMP is not used.
- Data node not used in transaction: when the failure occurs, transactions proceed undisturbed. During cluster reintegration, however, the ha_stopscript will force all applications out, leading to transactions being rolled back (SQL1229) and connections being lost.
- Catalog node failure: at the time the failure occurs, behavior is the same as without HACMP. After takeover, new transactions and connects are possible again. When the catalog node reintegrates the cluster, the force applications command issued by the ha_stopscript script will force all applications out and connections will be lost. If takeover occurs within CONN_ELAPSE*MAX_CONNRETRIES seconds, connections are not lost during takeover. It might be sensible to increase the value of MAX_CONNRETRIES (default 5) to avoid loosing connections during takeover.

11.2.2.10 Other SP Components Failures

In this section, we give a quick overview of the influence of some RISC/6000 SP component failures on DB2 Parallel Edition operation as defined in the tested configuration:

- RISC/6000 SP Ethernet failure: since the RISC/6000 SP Ethernet is used as a backup heartbeat network in our setup, it is important that it be available. However, DB2 Parallel Edition operation is not directly impacted by possible failures of the RISC/6000 SP Ethernet.
- PSSP software failure: the only relevant component for DB2 Parallel Edition is the Worm switch daemon (fault_service_Worm_RTG). It has to run in order for the switch to be usable. If this daemon dies on any of the nodes, DB2 loses communications. As seen in 11.2.1.16, "Using AIX Error Notification" on page 116, it is possible to use HACMP′s support of AIX error notification mechanism to alleviate this problem.
- Control workstation failure: in normal operation, the control workstation is not used. However, to restart the switch or restart amd on a node, the control workstation needs to be running. This means that HACMP takeovers and node cluster reintegrations are not possible if the control workstation is down. For this reason, is should be protected with the HA-CWS software.
- Serial line between the nodes and the control workstation: failure of this line has no influence on DB2 Parallel Edition.

11.2.2.11 Failover and Cluster Reintegration Times

Time for failover of node1 is about one minute to two minutes, assuming no DB2 Parallel Edition recovery is necessary. If transactions need to be rolled back that have been running for N minutes before the crash, add N minutes for a rough estimate of the time needed for takeover. Cluster reintegration is a little bit longer, 3 minutes 30 seconds to 4 minutes, the difference coming from the fact that it takes about two minutes to stop node2′s logical node when node1′s logical node has been running on node2.

Appendix A.   HACMP Scripts

This appendix lists those HACMP scripts that had to be modified for operation with DB2 Parallel Edition on node2. They are located under /usr/sbin/cluster/events and /usr/sbin/cluster/events/utils.

A.1 cl_activate_fs

```
PROGNAME=″$0″
STATUS=0
if [ ″$VERBOSE_LOGGING″ = ″high″ ]
then
    set -x
fi
set -u
if [ $# -ne 0 ]
then
    FILELIST=for i in $*; do /bin/echo $i; done | /bin/sort
    for fs in $FILELIST
    do
       # -s says only return status, -x says exact match
       # we use awk instead of cut because mount outputs
       # lots of leading blanks that confuse cut
#      mount | awk ′{ print $2 }′ | fgrep -s -x ″$fs″
# changed for amd
       V1=df | awk ′{ print $7 }′ | grep ″$fs″
       V2=df | grep ″$fs″ | awk ′{ print $1 }′ | grep ″^\/dev\/″
       if [ ″$V1″ != ″″ ] && [ ″$V2″ != ″″ ]
#      if [ $? -eq 0 ]
       then
           cl_log 11 ″$PROGNAME: Filesystem $fs already mounted.″ $PROGNAME $fs
       else
           # Perform quick fsck first
           fsck -f $fs
           # Try to mount filesystem
           mount $fs
           if [ $? -ne 0 ]
           then
              # Mount failed, run fsck before retrying mount
              fsck -p $fs
              if [ $? -ne 0 ]
              then
                  cl_log 13 ″$PROGNAME: Failed fsck -p of $fs.″ $PROGNAME $fs
                  STATUS=1  # note error and keep going
              else
                  mount $fs
                  if [ $? -ne 0 ]
                  then
                     cl_log 10 ″$PROGNAME: Failed mount of $fs.″ $PROGNAME $fs
                     STATUS=1      # note error and keep going
                  fi
              fi
           fi
       fi
    done
else
              cl_echo 12 ″usage: $PROGNAME filesystems_to_mount″ $PROGNAME
       exit 2
fi
exit $STATUS
```

A.2 cl_deactivate_nfs

```
PROGNAME=″$0″
MOUNTED=″false″
SLEEP=″2″
if [ ″$VERBOSE_LOGGING″ = ″high″ ]
then
    set -x
fi
set -u
if [ $# -ne 0 ]
then
    FILELIST=for i in $*; do /bin/echo $i; done | /bin/sort -r
    for fs in $FILELIST
    do
       # Is the filesystem mounted?
       # -s says only return status, -x says exact match
        # we use awk instead of cut because mount outputs
        # lots of leading blanks that confuse cut
       # This line has been replaced (/etc/mount to mount) as a result of bug # 5287
       # This line has been replaced ($2 to $3) as a result of bug # 5633
       #/etc/mount | awk ’{ print $2 }’ | fgrep -s -x ”$fs”
       mount | awk ’{ print $3 }’ | fgrep -s -x ”$fs”
        if [ $? -eq 0 ]
        then
              # At least one filesystem is mounted
              MOUNTED=”true”
              # This filesystem is mounted
              # Send a SIGKILL to all processes having open file
              # descriptors within this logical volume to allow
              # the umount to succeed..
# cl_nfskill commented out for DB2 Parallel Edition
#             cl_nfskill -k -u $fs
       fi
    done
else
    cl_echo 26 ’usage: $PROGNAME filesystems_to_unmount’ $PROGNAME
    exit 2
fi
# Make sure all processes have time to die
# This function split in two so that only one sleep is necessary
# independent of number of filesystems mounted.
if [ ”$MOUNTED” = ”true” ]
then
        sleep $SLEEP
fi
FILELIST=for i in $*; do /bin/echo $i; done | /bin/sort -r
for fs in $FILELIST
do
       # Check to see if filesystem is mounted.
       # This line has been replaced (/etc/mount to mount) as a result of bug # 5287
       # This line has been replaced ($2 to $3) as a result of bug # 5633
       #/etc/mount | awk ’{ print $2 }’ | fgrep -s -x ”$fs”
       mount | awk ’{ print $3 }’ | fgrep -s -x ”$fs”
        if [ $? -eq 0 ]
# changed for DB2 Parallel Edition
        then
              umount -f $fs
              # needed to get exit code = 0
              echo ”...”
        fi
#        if [ $? -eq 0 ]
#        then
#             until umount -f $fs
#             do
#                    sleep 2
#             done
#        fi
done
```

List of Abbreviations

ADSM	ADSTAR Distributed Storage Manager
ADSTAR	Advanced Storage And Retrieval (now SSD - Storage Systems Division)
AIX	advanced interactive executive (IBM′s flavor of UNIX)
API	application program interface
ARP	address resolution protocol
ASCII	American National Standard Code for Information Interchange
CD-ROM	(optically read) compact disk - read only memory
CLIO/S	IBM Client Input/Output Sockets (licensed program)
CPU	central processing unit
DDL	data definition language
DMA	direct memory access
DSM	data systems manager
DSM	distributed systems management
DSM	distribution services manager
Gb	gigabit (10**9 bits or 1,000,000,000 bits) not recommended, use Gbit
GB	gigabyte (10**9 bytes or 1,000,000,000 bytes) case should be Gb
GET	grand elapsed time
GUI	generalized end-user interface
GUI	graphical user interface
HACMP	high availability cluster multi-processing (AIX)
HPS	high performance switch
I/O	input/output
IBM	International Business Machines Corporation
INEWS	information news facility (IBM)
IP	internet protocol (ISO)
ITSC	International Technical Support Center (IBM)
ITSO	International Technical Support Organization
Mb	megabit (million bits) not recommended, use Mbit
MB	megabyte, 1,000,000 bytes (1,048,576 bytes memory) case should be Mb
MVS	Multiple Virtual Storage node $NFSHOSTS″ $PROGNAME $NFSHOSTS
NFS	network file system (USA, Sun Microsystems Inc)
PC	Personal Computer (IBM)
PIOFS	Parallel I/O File System
POK	Poughkeepsie, NY
PSSP	AIX Parallel System Support Programs (IBM program product for SP1 and SP2)
PTF	program temporary fix
RISC	reduced instruction set computer/cycles
SCSI	small computer system interface
SDR	system data repository
SMIT	System Management Interface Tool (see also DSMIT)
SMP	shared multiprocessor
SP	IBM RS/6000 Scalable POWERparallel Systems (RS/6000 SP)
SQL	structured query language
TCP	transmission control protocol (USA, DoD)
TCP/IP	Transmission Control Protocol/Internet Protocol (USA, DoD, ARPANET; TCP=layer 4, IP=layer 3, UNIX-ish/Ethernet-based system-interconnect protocol)
TCPIP	transmission control protocol internet protocol
UNIX	an operating system developed at Bell Laboratories (trademark of UNIX System Laboratories, licensed exclusively by X/Open Company, Ltd.)

UPS	uninterruptible power supply/system
URL	Uniform Resource Locator
URL	Universal Resource Locator

Index

ITSO Redbook Evaluation

International Technical Support Organization
Backup, Recovery and Availability with
DB2 Parallel Edition on RISC/6000 SP
April 1996

Publication No. SG24-4695-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____

Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) Are you an employee of IBM or its subsidiaries: Yes____ No____

b) Do you work in the USA? Yes____ No____

c) Was this redbook published in time for your needs? Yes____ No____

d) Did this redbook meet your needs? Yes____ No____

If no, please explain:

____

____

What other topics would you like to see in this redbook?

____

____

What other redbooks would you like to see published?

____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

______________________ ______________________
Name Address

______________________ ______________________
Company or Organization

______________________ ______________________
Phone No.

ITSO Redbook Evaluation
SG24-4695-00

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

IBM International Technical Support Organization
Mail Station P099
522 SOUTH ROAD
POUGHKEEPSIE NY
USA 12601-5400

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

SG24-4695-00