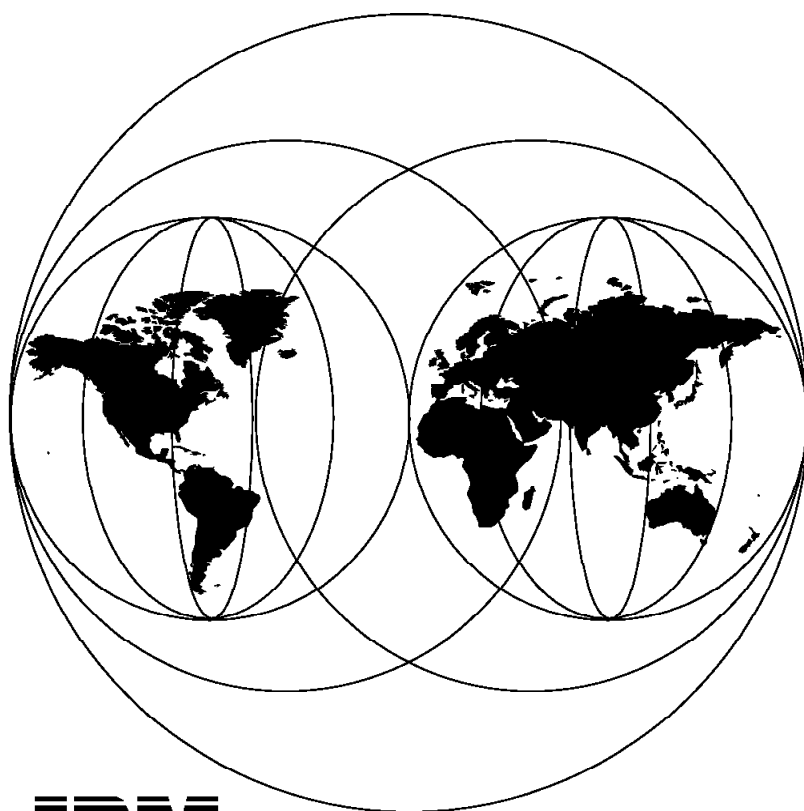


IBM DB2 for AIX and SAP R/3 Administration Guide

January 1997



**International Technical Support Organization
Austin Center**



International Technical Support Organization

SG24-4871-00

IBM DB2 for AIX and SAP R/3 Administration Guide

January 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, "Special Notices" on page 249.

First Edition (January 1997)

This edition applies to Version 2 Release Number 1.1 of IBM DATABASE 2 for AIX, Program Number 41H2128 and to R/3 Release 3.0 C from SAP for use with the AIX Operating System Version 4.1.4.

Comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. JN9B Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

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.

© **Copyright International Business Machines Corporation 1996. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	xi
Preface	xiii
How This Redbook Is Organized	xiii
The Team That Wrote This Redbook	xiv
Comments Welcome	xv
Chapter 1. Introduction to DB2 and R/3	1
1.1 Architecture of a SAP R/3 System	1
1.2 Client/Server R/3 Architecture	2
1.2.1 R/3 Advantages	5
1.3 DB2 Common Server	6
1.3.1 DB2 Servers	7
1.3.2 DB2 Components	7
1.3.3 DB2 Client Application Enabler (CAE)	8
1.3.4 DB2 Command Line Processor (CLP)	9
1.3.5 DB2 Administrator's Toolkit	10
1.3.6 Database Director	10
1.3.7 Visual Explain	12
1.3.8 Performance Monitor	13
1.4 DB2 Products	14
1.4.1 DB2 Server	14
1.4.2 DB2 Software Developer's Kit (SDK)	15
1.5 DB2 and R/3 Product Relationship	15
1.5.1 R/3 System and DB2 Products	16
Chapter 2. Getting Ready for R/3 and DB2	19
2.1 Installation Overview	19
2.2 R/3 and DB2 Configuration Guidelines and Planning	21
2.2.1 Minimum R/3 Memory and Disk Requirements	22
2.2.2 Physical Storage	22
2.2.3 AIX Facilities for Storage	24
2.2.4 Physical Storage Scenario for R/3 and DB2	27
2.3 Storage Concepts in DB2	28
2.3.1 Container	28
2.3.2 Tablespaces in DB2	28
2.3.3 When to Use DMS Tablespaces	32
2.3.4 Distributed DMS Tablespaces in R/3	32
2.3.5 R/3 Container and Tablespace Names	33
2.4 Data Placement	35
2.4.1 JFS and RDBMS Objects	35
2.4.2 R/3 and DB2 Directory Structure	37
2.4.3 SAP Directories	38
2.4.4 DB2 for AIX Directory Structure	39
2.4.5 Creating a Tablespace in R/3	41
2.5 DB2 Configuration Parameters	44
2.5.1 Memory Used by DB2 Database Manager	44
2.5.2 Database Manager Configuration Parameters	47
2.5.3 Database Configuration Parameters	50

2.6 Using DB2 Features in an R/3 System	54
2.6.1 Extent Size	54
2.6.2 Prefetching in DB2	55
2.6.3 I/O Cleaners	56
Chapter 3. Starting and Stopping SAP R/3	59
3.1 Starting R/3	59
3.2 Stopping R/3	60
3.3 DB2 for AIX Processes	61
3.3.1 DB2 Central Instance Processes (No Clients)	62
3.3.2 DB2 R/3 Central Instance with Clients Connecting to Database	63
3.3.3 Displaying R/3 Processes	65
3.3.4 DB2 for AIX and R/3 Processes	66
3.3.5 DB2 and R/3 Users	68
3.3.6 DB2 Users	69
Chapter 4. R/3 Database Administration	73
4.1 Tools and Utilities	73
4.1.1 DB2 Tools and Utilities	73
4.1.2 The R/3 Database Performance Monitor	79
4.1.3 R/3 DB2admin	83
4.2 Managing Tablespaces	86
4.2.1 Managing Tablespaces with DB2 Commands	87
4.2.2 Displaying Tablespace Information	87
4.2.3 Displaying Tablespace Information - DB2admin	89
4.2.4 Displaying Tablespace Information - R/3 Performance Monitor	89
4.2.5 Displaying Container Information	92
4.2.6 Monitoring Active Applications	99
4.2.7 Tablespace States	100
4.2.8 System Catalogs in DB2	101
4.2.9 Understanding and Changing the Size of DMS Tablespaces	103
4.2.10 Sizing Considerations for DMS Tablespaces	103
4.3 Altering Tablespaces	106
4.3.1 Adding a Container to a DMS Tablespace	107
4.3.2 Redefining the Containers in a Tablespace (Redirected Restore)	109
4.4 Data Maintenance	110
4.4.1 Analyzing Data's Physical Organization: REORGCHK	111
4.4.2 Interpreting the Output from REORGCHK	112
4.4.3 Interpreting the Index Output from REORGCHK	114
4.4.4 Using R/3 Performance Monitor to Check for Reorganization	116
4.4.5 Table Reorganization - REORG	119
4.4.6 RUNSTATS	124
4.4.7 Identifying Updated Statistics	125
4.4.8 Collecting Detailed Information about Indexes	126
4.4.9 Modelling a Production Environment	127
4.4.10 Using DB2admin to Reorganize Data	128
4.4.11 Summary	133
4.5 Database Monitoring	134
4.5.1 Snapshot Monitoring	135
4.5.2 Using R/3 to Display Monitor Information	136
4.5.3 The DB2 Performance Monitor	146
4.5.4 Event Monitoring	149
Chapter 5. Backup and Recovery	155
5.1 Logging Considerations	155

5.1.1	Types of Logging	156
5.1.2	Log Management Configuration Parameters	159
5.1.3	Terminology in DB2 and R/3	161
5.1.4	Enabling Archive Logging	162
5.1.5	Log File Information	163
5.1.6	Naming of Log Files in R/3	164
5.1.7	Log File Usage	165
5.1.8	DB2 R/3 Logging Summary	166
5.2	Configuring ADSM for DB2	166
5.2.1	Main Components	166
5.2.2	Installing ADSM	167
5.2.3	Customizing the ADSM Server	168
5.2.4	Customizing the ADSM Backup/Archive Client for R/3 with DB2	175
5.2.5	Setting up ADSM to Backup to Disk	176
5.2.6	Setting up ADSM to Save Archived Logs to Disk	186
5.3	R/3 DB2 Backup and Restore Utilities	191
5.3.1	Backup and Restore Considerations	192
5.3.2	How to Perform a DB2 Backup in an R/3 Environment	192
5.3.3	Performing an Offline Backup	193
5.3.4	Performing an Online Backup	195
5.3.5	Methods of Recovery	197
5.3.6	Recovery Using an Offline Backup from Disk without Roll Forward	199
5.3.7	Recovery Using an Offline Backup from ADSM (No Roll Forward)	200
5.3.8	Recovery Using an Offline Backup from Device with Roll Forward	201
5.3.9	Recovery Using an Offline Backup from ADSM with Roll Forward	203
5.3.10	Recovery Using an Online Backup from Disk	204
5.3.11	Recovery Using an Online Backup from ADSM	206
5.3.12	Roll Forward Pending	209
5.3.13	Roll Forward Database	210
5.3.14	Point-in-Time Recovery	213
5.3.15	Managing Backup and Restore Operations	214
5.3.16	Backup and Restore Considerations and Recommendations	220
5.4	Performing a Redirected Restore	221
5.4.1	Changing Containers in a Tablespace	222
5.4.2	Make Inactive Log Files Available for Roll Forward	229
5.4.3	Roll Forward Database	229
5.4.4	Removing Log Files	230
5.5	Managing Archive Log Files	230
5.5.1	Backing up the Archive Log Files to ADSM	230
5.5.2	Removing Files from log_archive File System with ADSM	232
5.5.3	Restoring Offline Archive Log Files	233
5.5.4	Determining Backed Up Log Files	234
5.5.5	Removing Used Log Files	234
Chapter 6. Monitoring and Troubleshooting		237
6.1	Monitoring	237
6.1.1	Alerts	237
6.1.2	System Logs	238
6.1.3	Database Monitoring	241
6.1.4	Database Alerts	241
6.1.5	R/3 Buffer Pools	242
6.2	Troubleshooting	242
6.2.1	Where to Look	242
6.2.2	Database Will Not Start	244
6.2.3	Database Will Not Stop	245

6.2.4 Archive Log Directory Fills Up	245
6.2.5 Tablespace is Full	245
6.2.6 Roll Forward Cannot Find Archive Log File	246
6.2.7 ADSM Recovery Log Fills Up	247
6.2.8 Cannot Archive Inactive Redo Logs	247
6.2.9 Redirected Restore with ADSM Does Not Complete	247
Appendix A. Special Notices	249
Appendix B. Related Publications	251
B.1 International Technical Support Organization Publications	251
B.2 Other Publications	251
How to Get ITSO Redbooks	253
How IBM Employees Can Get ITSO Redbooks	253
How Customers Can Get ITSO Redbooks	254
IBM Redbook Order Form	255
List of Abbreviations	257
Index	259

Figures

1.	Layered Architecture of the R/3 System	1
2.	Three-Tier R/3 Model	2
3.	Two-Tier R/3 System Configuration	3
4.	Three-Tier R/3 Configuration with Services Distributed among Systems	5
5.	DB2 Family of Database Servers	6
6.	DB2 Products and Components	8
7.	Remote Client Accessing DB2 Server Using CAE	9
8.	Command Line Processor	9
9.	Database Director - Tree View	11
10.	Visual Explain Output	12
11.	Performance Monitor Output	13
12.	DB2 Server with Remote Clients	14
13.	DB2 R/3 Product Relationship	17
14.	R/3 Installation Sequence	20
15.	Optimal Distribution of Data in an DB2 R/3 System	22
16.	LVM - Logical Volume Manager (No Mirroring)	24
17.	LVM - Mirroring	26
18.	R/3 Volume Group and Logical Volumes	27
19.	Containers in DB2	28
20.	Tablespace and Container One-to-Many Relationship	29
21.	Extent, Containers, and Tablespace in DB2	29
22.	R/3 Space Management in DB2 for AIX	31
23.	R/3 Container Names	34
24.	How DB2 Tables are Implemented Using JFS Files	36
25.	SAP Directories	38
26.	DB2 for AIX Instance Database Directories and Files	40
27.	The CREATE TABLESPACE Statement	42
28.	Create Tablespace Scenario in R/3	42
29.	Memory Used by the DB2 Database Manager	45
30.	DB2 Prefetching Data Using I/O Servers	55
31.	Process Model for DB2	61
32.	DB2 Process Model - No Database Connections	62
33.	DB2 for AIX Process Model with Connecting Clients	64
34.	Displaying R/3 Processes with the Database Performance Monitor	66
35.	DB2 for AIX and SAP R/3 Processes	67
36.	DB2 Interfaces	74
37.	Command Line Processor Option Settings	76
38.	CLP Input File - file.clp	77
39.	Tree View of the Database Director	77
40.	DB2 Database Director - Snapshot Monitor	78
41.	R/3 Entry Screen - Performance Monitor	80
42.	R/3 Database Performance Monitor, State Part	81
43.	R/3 Alert Monitor, Overview Screen	82
44.	Accessing DB2 Utilities through SMIT	84
45.	Administering DB2 via the SMIT Interface	85
46.	R/3 Database Performance Monitor - Tablespace Analysis	90
47.	R/3 Database Performance Monitor - Tablespace Selection Pop-Up Menu	91
48.	R/3 Database Performance Monitor - Detailed analysis	91
49.	Database Director - Table Spaces - Detailed View	92
50.	R/3 Database Performance Monitor - Obtaining Container Information	95

51.	R/3 Performance Monitor - Tablespace Container Information	96
52.	Database Director - Containers - Detailed View	97
53.	R/3 Alert Monitor - Monitoring Containers and Tablespaces	98
54.	Controlling Users within DB2	100
55.	Adding a Container in a DMS Tablespace	106
56.	Checking Tablespace Containers Before Adding a Container	107
57.	Container Details Before Adding Container	108
58.	DB2admin Tool - Changing Prefetch Size when Adding Container	109
59.	Results of Executing the REORCHK Utility	112
60.	REORGCHK Table Information	113
61.	REORGCHK Output Showing Exceeded Value for Overflowed Rows	114
62.	Logical Structure of an Index in DB2	114
63.	REORGCHK Index Information	114
64.	Detailed Analysis of Tables and Indexes	117
65.	Detailed Analysis of Tables and Indexes, Sorted by Overflow Rows	118
66.	Detailed Analysis of Table and Its Index	119
67.	Cluster Ratio Index	120
68.	Closer Look at Table and Index Statistics	121
69.	Reorganizing Table Using DB2admin	122
70.	Tables and Index Statistics after a REORG	123
71.	The Data Maintenance Process	127
72.	DB2admin - Update Statistics on Table with REORGCHK	130
73.	DB2admin - Using Current Statistics	130
74.	DB2admin - Displaying Results of REORGCHK before REORG	131
75.	DB2admin - Update Statistics on Table, REORGCHK	132
76.	DB2admin - Displaying Results of REORGCHK after REORG	132
77.	DB2admin - Displaying Results	133
78.	DB2admin - Displaying Results of REORGCHK after REORG	133
79.	Database Summary - Part One	137
80.	Database Summary - Part Two	138
81.	Detailed Database Performance Analysis	139
82.	Table Activity Statistics	141
83.	Tablespaces	142
84.	Active DB2 Applications	143
85.	Parameter Changes	145
86.	Performance History	146
87.	Performance Graph for a Database	147
88.	Example of Performance Details Display	148
89.	Archival Logging	157
90.	Overview of Database Backup and User Exit	158
91.	Enable Archive Logging	163
92.	Log File Information	164
93.	ADSM Graphical Interface	168
94.	ADSM Administrative Client GUI Login Screen	169
95.	ADSM Administrative Client Command Line Interface	170
96.	ADSM Administrative Client Graphical Interface	171
97.	Defined ADSM Administrators	171
98.	Adding an Administrator	172
99.	Registering a Client Node	174
100.	Setting the Password for a Client Node	175
101.	Sample ADSM Server and Client in an R/3 DB2 Environment	176
102.	Adding a Storage Pool for Backup	178
103.	Adding a Storage Pool for Backup	178
104.	Adding a Volume to a Storage Pool	179
105.	ADSM Administration Client GUI	180

106.	Backup Copy Groups - Details Screen	181
107.	Defining Storage Pool Destination for Backup	182
108.	Policy Sets - Details Screen	183
109.	Policy Sets - Activate Screen	183
110.	ADSM Client - Main Screen	184
111.	Using ADSM to Backup	185
112.	Storage Pool Volumes - Icons Panel	185
113.	List Volume Content to File Panel	186
114.	Defining Storage Pools for the Archived Log Files	187
115.	Adding a Storage Pool for the Archived Log Files	188
116.	Using Storage Pool Volumes	189
117.	Assigning a Logical Volume to a Storage Pool	189
118.	Setting the Storage Pool Destination	190
119.	Setting the Storage Pool Destination	191
120.	Offline Database Backup to Disk	194
121.	Offline Backup to ADSM	195
122.	Online Backup to Disk	196
123.	Online Backup to ADSM	197
124.	Three Methods of Recovery	198
125.	Recovery from Offline Backup without Roll Forward	200
126.	SMIT Warning Message	200
127.	Recovery from Offline Backup to ADSM without Roll Forward	201
128.	Recovery from Offline Backup to Disk with Roll Forward	202
129.	SMIT Warning Message	202
130.	Roll Forward Database after Recovery from Disk	203
131.	Roll Forward Database after Recovery from ADSM	204
132.	Error When Recovering from Online Backup (No Roll Forward)	205
133.	Restore Database from Device Panel	205
134.	Roll Forward Process Error Message: Archived Log File Unavailable	206
135.	Roll Forward Database Panel	207
136.	Error Message During Roll Forward Process	208
137.	Successful Roll Forward Process	209
138.	Roll Forward Database Screen in DB2admin	210
139.	Output Message from the ROLLFORWARD DATABASE Command	213
140.	Roll Forward Database Screen in DB2admin	214
141.	Show Protocol Screen in DB2admin Tool	219
142.	Using SMIT to Add a Logical Volume	223
143.	Recover Screen in Database Director (Redirected Restore Option)	225
144.	Redirected Restore using the Database Director	226
145.	Error Message During Redirected Restore	226
146.	Pausing to Redefine Containers	227
147.	Container Specification in Redirected Restore	227
148.	Saving Inactive Archive Log Files via ADSM	231
149.	Prompt to Continue Message	232
150.	Deleting Inactive Archive Log Files	233
151.	Restoring Archived Log Files with ADSM	234
152.	Cannot Locate Archived Log File	246

Tables

1. DB2 Product Packaging for R/3 (Central Instance)	16
2. DB2 R/3 Tablespaces	34
3. DB2 System CATALOG Views	102
4. DB2 Updatable System CATALOG Views	103
5. DB2 Monitor Switches	135
6. DB2 Event Types	151
7. Format of DB2 History File	216

Preface

This redbook covers the R/3 product from the German company SAP AG as implemented on DB2 for AIX.

The material this book contains is unique in its coverage of DB2 and R/3. We wrote it for R/3 database administrators who need to understand how DB2 and R/3 work together, and it is unique in its range of coverage. We look at how DB2 and R/3 interact together at an architectural level and cover everything from installation considerations to problem determination, with emphasis on what to monitor to avoid pitfalls. There is a large section that details backup and recovery using the ADSM product. Many concepts are presented for users that are new to DB2 and R/3; however, we provide tips and techniques that experienced users will find invaluable.

Some knowledge of AIX, R/3, and databases, especially DB2 databases, is assumed.

How This Redbook Is Organized

This redbook contains 261 pages. It is organized as follows:

- Chapter 1, "Introduction to DB2 and R/3"
This chapter provides an overview of the R/3 architecture and the DB2 product. It also looks at how the two products work together.
- Chapter 2, "Getting Ready for R/3 and DB2"
This chapter provides some guidelines to help in preparing for an R/3 DB2 installation. We also discuss some terminology that is used in both DB2 and R/3 that a database administrator needs to know. There are certain configuration parameters in DB2 that a database administrator can control and set to take advantage of his/her environment; these are discussed in this chapter.
- Chapter 3, "Starting and Stopping SAP R/3"
This chapter discusses how to start the database and the R/3 environment. Also covered are the processes created and the users you find in the DB2 R/3 environment.
- Chapter 4, "R/3 Database Administration"
This chapter describes administrative tools and utilities that assist the R/3 database administrator in day-to-day functions such as monitoring free space and performance.
- Chapter 5, "Backup and Recovery"
This chapter thoroughly discusses the areas of backup and recovery. Setting up and using ADSM is also discussed.
- Chapter 6, "Monitoring and Troubleshooting"
This final chapter provides tips and techniques on usage problems that might occur.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

CALENE JANACEK is a technical project manager at the International Technical Support Organization, Austin Center. She writes extensively and teaches IBM classes worldwide on all areas of DB2.

LAURA COOPER is a systems engineer for IBM who began working with SAP in September 1993. In March 1994 she joined the IBM/SAP Competency Center as a technical architect; her primary responsibilities are in providing technical (Basis) support for customers running SAP R/3 on the IBM RS/6000 platform.

SUSANNE HIRSCH is a software engineer at IBM Germany. She joined the IBM German Software Development Laboratory (GSDL) in 1987. Her areas of expertise include the development of client/server and database applications. In March 1994 she became a member of the SAP R/3 DB2 for AIX porting team. Her responsibilities included porting the R/3 buffer synchronization and the R/3 Database Performance Monitor. Since May of 1996, she has been working for the SDD/SAP Data Migration Center as part of the World Wide Fulfillment (WWF) project.

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

Marcus Brewer, Editor
International Technical Support Organization, Austin Center

Martin Mezger
R/3 on DB2 for AIX Development
IBM Germany

Frank Rusconi
International Technical Support Organization, Austin Center

Melanie Stopfer
IBM Education & Training

Dale McInnis
IBM Canada

There were many people who were kind enough to review the book and offer their comments, and we thank them all for their input. Our special thanks goes to:

Phil Hardy
IBM/SAP Competency Center

Dale McInnis
IBM Canada

Martin Mezger
R/3 on DB2 for AIX Development
IBM Germany

Albert Rodi

IBM/SAP Competency Center

Hal Rose
IBM/SAP Competency Center

Ming Wu
IBM Canada

Thanks to Peter Coldicott of the Austin RS/6000 Executive Briefing Center for reviewing the document, exercising extreme patience, and providing moral support.

We especially thank Stephen Michael Cooper, Laura's newborn son, for waiting until we had a working draft of the book before coming into the world.
Seton Hospital, Austin, Texas 10/05/96

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Introduction to DB2 and R/3

This chapter describes the architecture of an R/3 system. We look at those components as they might be implemented in an AIX environment. We briefly describe the DB2 family of products. However, we will detail those DB2 products and components that would be part of an R/3 DB2 client/server environment.

This chapter also starts to define terminology used by both DB2 and R/3. There are many terms that are used by both, but have different meanings. For example, the term *instance* is used by both DB2 and R/3. An instance in DB2 is defined as a unique database manager environment. An instance in R/3 is a collection of processes and resources on a single system providing R/3 resources to one or more end-users. Throughout this document, we look at the terminology that is relevant to a DB2 R/3 administrator.

1.1 Architecture of a SAP R/3 System

A SAP R/3 system is architecturally implemented as a layered model. There are two main layers:

- Basis
- Application

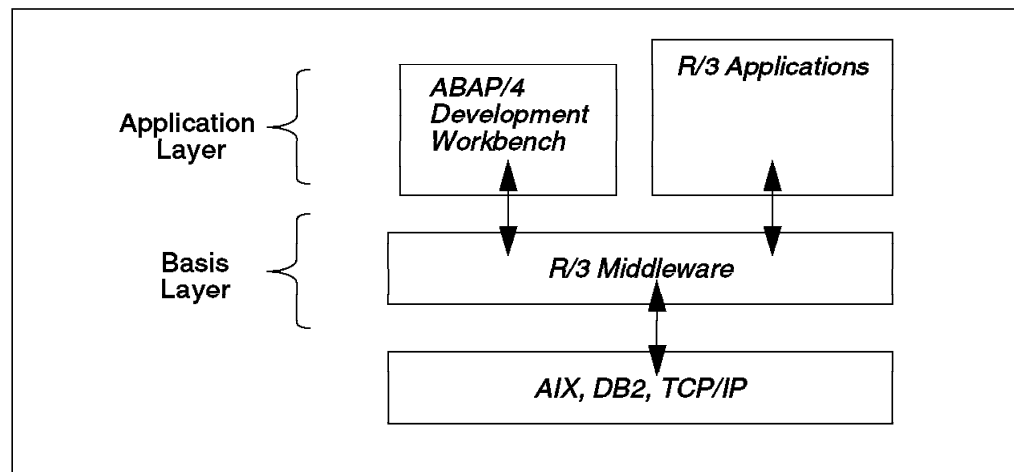


Figure 1. Layered Architecture of the R/3 System

The basis layer contains the R/3 System middleware. A portion of this middleware, or basis layer, is operating-system dependent. It is this middleware layer that makes applications independent of the hardware platform, including which operating system you have, which database product is installed, and the communication protocol that will be used. Figure 1 shows an AIX operating system using a DB2 database that uses TCP/IP as the communication-specific protocol to support remote clients. The R/3 middleware, or basis layer, is that portion of an R/3 system that is ported to specific environments. This document discusses only the basis layer of the R/3 system, in particular the basis layer of an R/3 system as it is implemented on an AIX DB2 database server that uses TCP/IP.

The ABAP/4 Development Workbench is the SAP programming environment used to develop client/server applications. The programming is done using the

SAP 4GL ABAP/4 language. The ABAP/4 Development Workbench supports the entire software development cycle with modeling tools, the 4GL language ABAP/4, definition of data and table structures and the design of graphical user interfaces.

The R/3 applications displayed in Figure 1 on page 1 are a library of business and general software components that can be integrated into your applications.

1.2 Client/Server R/3 Architecture

This section presents some R/3 terminology. We discuss some basic R/3 configurations with two-tier or three-tier models. Let's start with some basic terminology that you will find in a three-tier R/3 system. Figure 2 shows the R/3 system as a typical client/server system with three layers or tiers.

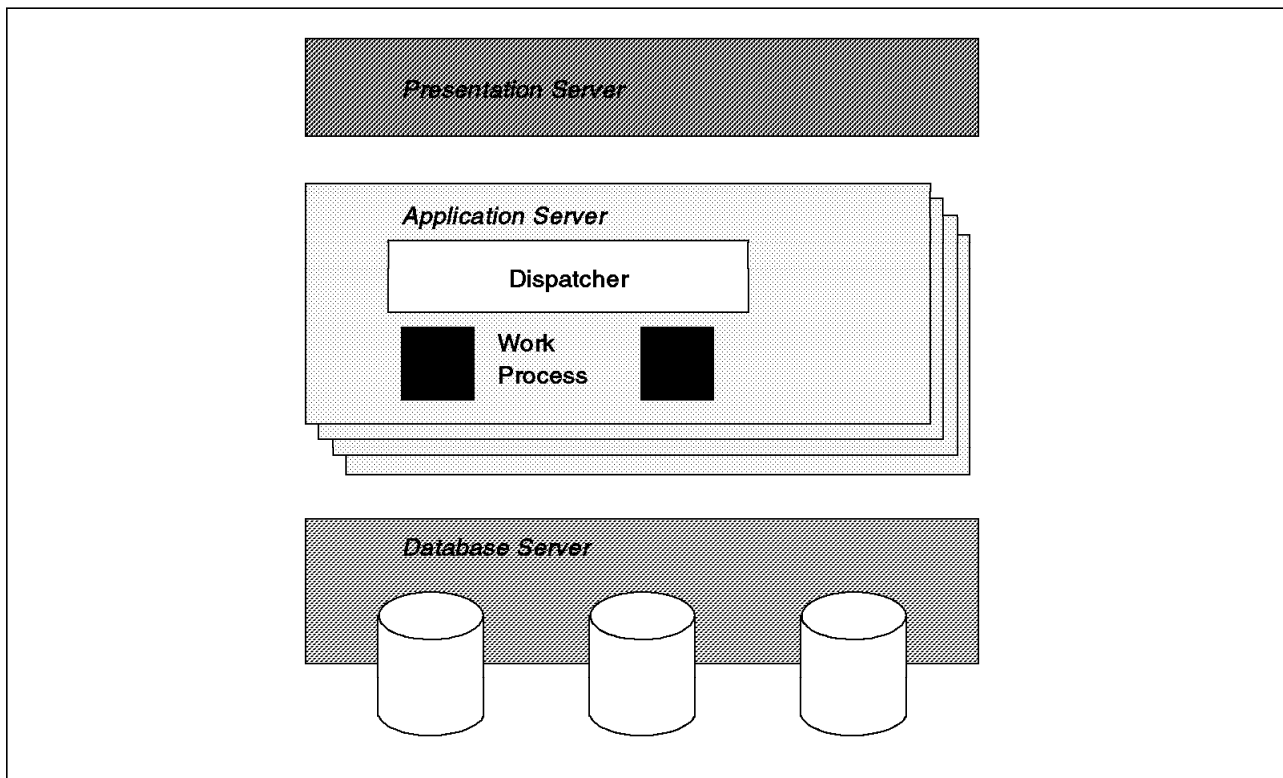


Figure 2. Three-Tier R/3 Model

- **Presentation Server**
The *Presentation Server* can execute on Motif, OS/2, Windows, Macintosh, Windows95, or WIN/NT systems. The SAP end-user has no database process executing at the client machine. This Presentation Server layer of R/3 is called SAPGUI.
- **Application Server**
The *Application Server* is, for example, an RS/6000 with AIX on which an R/3 instance exists. This instance is ready to service requests. Each Application Server has a single *Dispatcher* that manages the work load of the instance. The Presentation Server interacts with the dispatcher. End-user requests and units of work are assigned by the dispatcher to the work processes of the instance for completion. *Work processes* are jobs within the R/3 subsystem that actually perform the work. Each work process is assigned a

primary role by the dispatcher that will control, to a certain degree, what type of work will be performed by that work process. The number of work processes and the types that can exist for an instance is controlled by the instance profile. The duration of a work process is the time between screen input and screen output from the Presentations Server. The work process starts a "commit" before the output is displayed to the screen.

- Database Server
In our document, the Database Server is an RS/6000 on which the DB2 database is stored. The Database Server contains the information necessary to access the database from an instance. In the current supported implementation, a SAP R/3 System has only one database installed.

We have discussed a three-tier hardware configuration, which is a network of two or more RS/6000 machines. One machine in the network is a Database Server, but all other machines in the network are Application Servers. A Central Instance may also run on the Database Server. Let's look at an example of a two-tier R/3 system configuration.

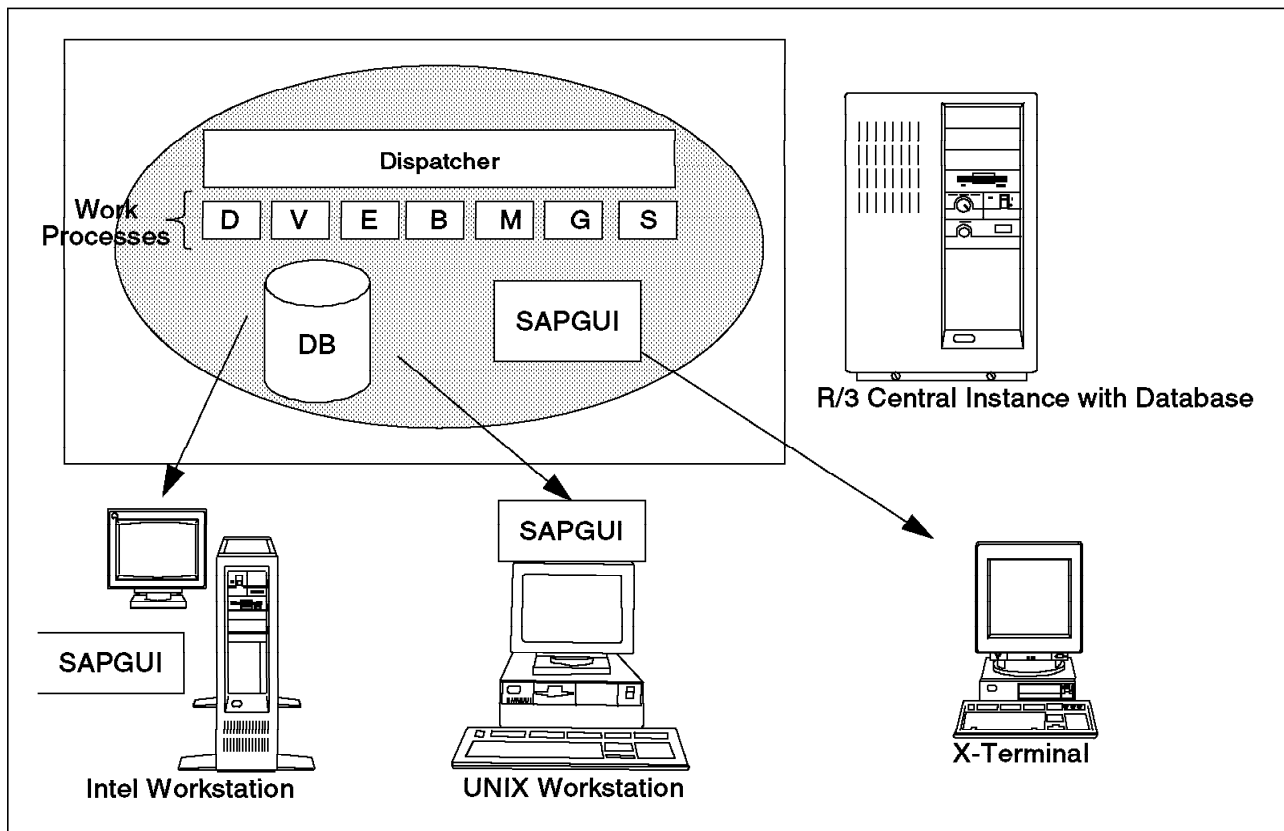


Figure 3. Two-Tier R/3 System Configuration

A two-tier configuration (Figure 3) is a hardware configuration in which a single machine serves as the Application Server, the Database Server and the Central Instance. From an R/3 viewpoint, this is a stand-alone machine that does not need to interact with another machine to service the R/3 end-user requests coming from the Presentation Server.

In a three-tier hardware configuration, an Application Server is not a stand-alone machine (Figure 2 on page 2). It must work in conjunction with the Database

Server in order to provide the end-user with services. There may be multiple Application Servers in a three-tier configuration.

One of the advantages with the three-tier model is the ability to distribute the resources that are required on the R/3 System. In a two-tier model, all of the R/3 work processes are installed on an R/3 Central Instance on the same machine in which DB2 is installed. These Work Processes consist of the following (Figure 3 on page 3):

- D - Dialog
- V- Update
- E - Enqueue
- B - Batch
- M - Message Server
- G - Gateway
- S - Spool

The R/3 system kernel appears to AIX as a collection of processes executing in parallel. These processes are made up of the dispatcher and the work processes. The Work Processes are further divided into dialog, update, enqueue, batch, message server, gateway, and spool. A Central Instance (Figure 3 on page 3) includes the Message Server (or process) and the Enqueue Server. There is only one Central Instance per R/3 system. The Central Instance can either have the database installed on the same system or not.

The dispatcher distributes the work packages among these work processes depending on the type of work. The type of work processes and the number of each can be configured depending on the load on your system, the configuration of your system, and the amount of memory available. The number of jobs executed in batch is a configured parameter. The number of Batch Processes can be configured depending on the system load and/or time of day. The number of dialog processes can be increased as the number of users are increased. Users also may choose to run some large jobs in batch mode, rather than dialog mode, to off-load resources. The Dialog Process is interactive. It handles the input/output between the Application Server and the SAPGUI.

The Gateway Process is responsible for communication between an instance and processes outside of the instance. For example, the gateway is used to communicate between a Work Process and a program that is running external to the R/3 environment.

The Enqueue Server is a special Work Process that is responsible for handling the special locking requirements of R/3. The Enqueue Server handles logical locks. These are not the same kind of locks that a database administrator knows. Rather, these are locks that may need to be held across an entire R/3 system, instead of only within the database. There is only one Enqueue Server for an entire R/3 system, and it executes under the Central Instance, by default.

The Message Server is a special job within a Central Instance. It is responsible for facilitating communication between instances within the same R/3 system and for identifying services in the R/3 system to processes outside the SAP system. There is only one message server for an entire R/3 system, and it also executes under the Central Instance, by default.

In a two-tier model all the R/3 user front-ends are physically external to the central host. The SAPGUI is physically located on the R/3 Central Instance DB2 machine when the R/3 user frontend is an X-Terminal. Other remote workstations may have their SAPGUI programs located on an NFS directory that is available from the Central Instance DB2 machine.

When using a three-tier R/3 system configuration, it is also possible to split the services.

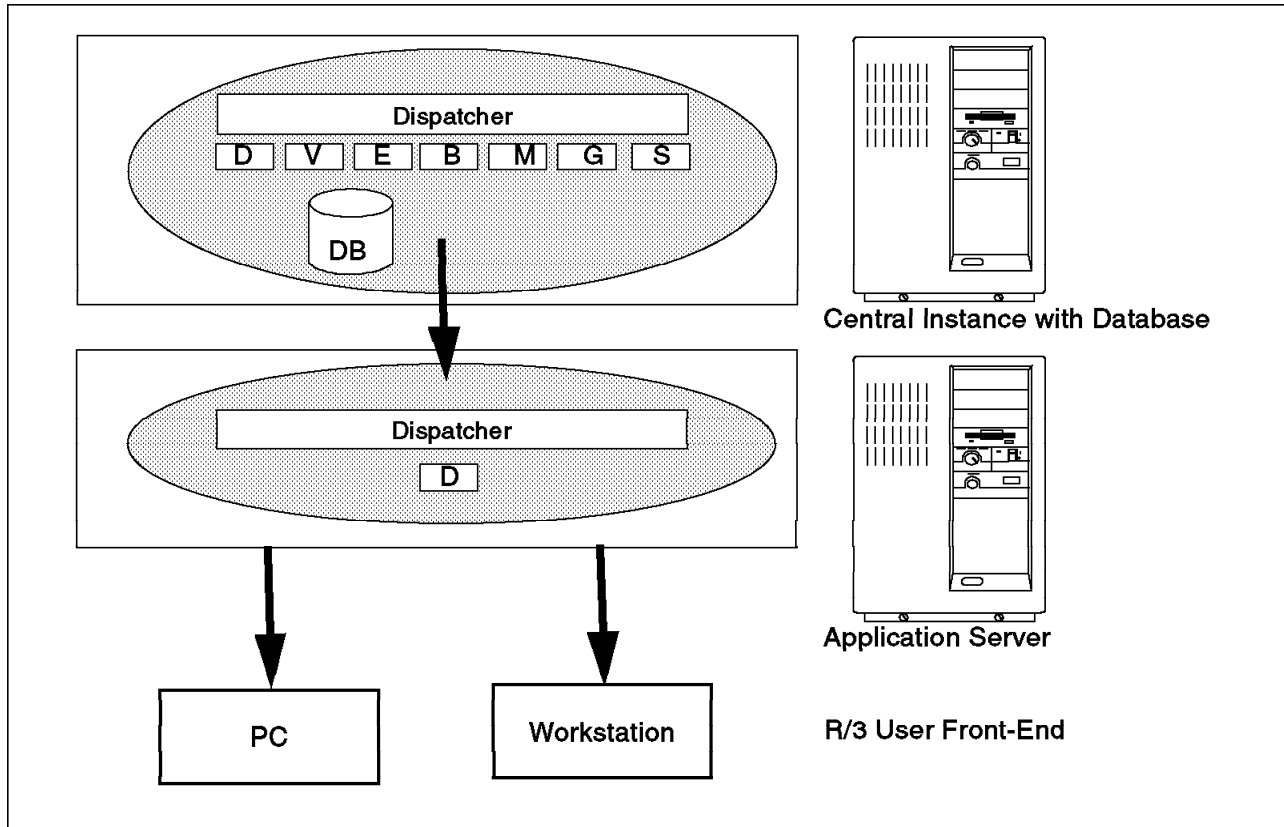


Figure 4. Three-Tier R/3 Configuration with Services Distributed among Systems

Figure 4 shows another example of a three-tier R/3 system configuration. There is an RS/6000 that has all the components and services of a standard R/3 Central Instance with a DB2 database. The second RS/6000 has only the Dialog Instance Service or Process installed on the Application Server. You may also place the Update, Spool, and Batch Processes on the same machine that has the Dialog Process. The clients (PCs and Workstations) may communicate with the R/3 Instance on the Database Server, the Application Server, or both. Any of the R/3-supported clients may be used; however, only a PC or workstation is shown in Figure 4.

1.2.1 R/3 Advantages

The multilayered architecture of the R/3 system makes it possible to separate the application logic from the database and the presentation logic. This separation allows load distribution, which is a requirement of client/server architecture. The flexibility of this architecture allows for a wide variety of implementations. The R/3 system can run on a single machine or be distributed on multiple machines to improve performance and increase work load.

Implementation of separate machines for specific work loads allows an increase in performance potential.

The R/3 architecture has the following advantages:

- Scalability: The flexible client/server architecture allows easy adjustment to increasing load. An R/3 system running on a single machine can be easily modified to run on multiple machines, allowing more users and increased performance. Because separate machines can be configured for a specific work load, these machines can be fine-tuned for the performance of their specific task.
- Portability: The R/3 system can run on many hardware and software platforms.
- Interoperability and openness: A distributed R/3 system can be run on multiple vendor platforms. The database can be on a different hardware platform with a different operating system to the application servers.
- Ease of customization: As the business needs change or grow, the R/3 architecture allows fast and simple modification of the implementation to meet the changing needs.
- Graphical user interface: A common graphical user interface is used. The SAPGUI can be run on many different workstation platforms.

1.3 DB2 Common Server

In this chapter, you are introduced to the DB2 family of products for the workstation environment. However, DB2 is available for many operating environments.

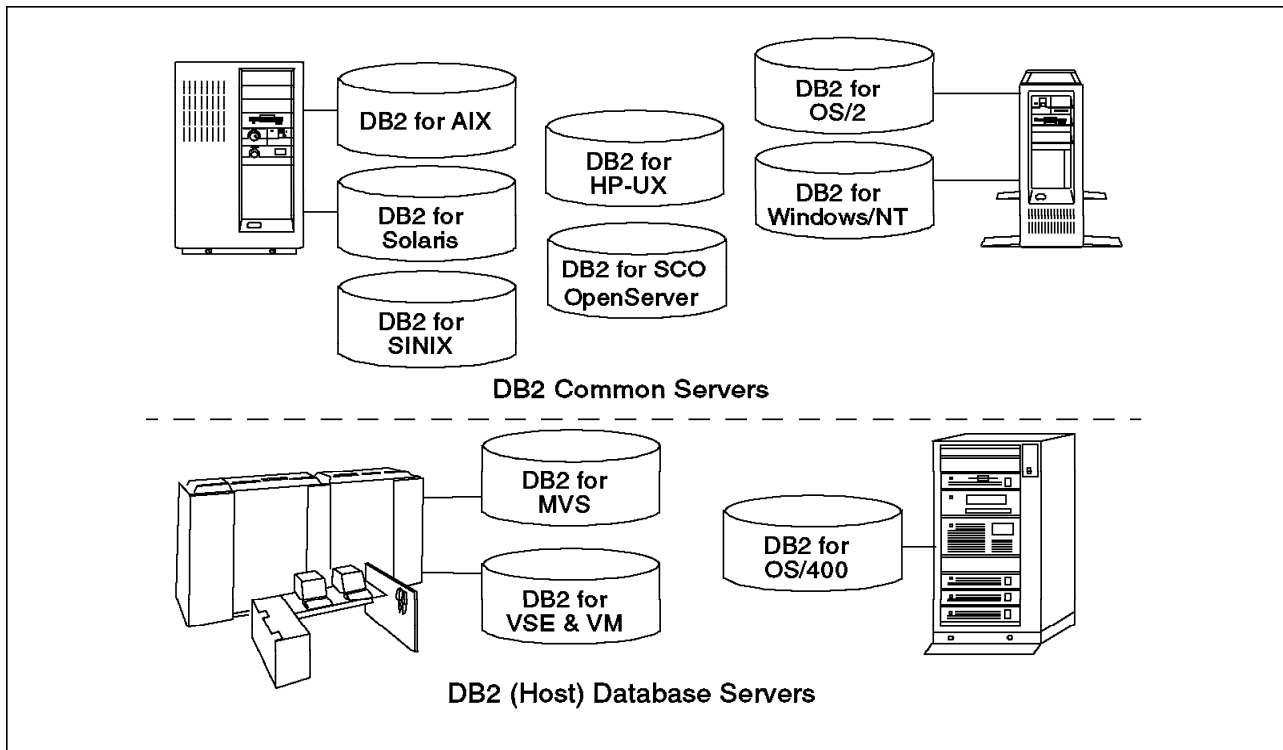


Figure 5. DB2 Family of Database Servers

The DB2 database products are collectively known as the DB2 Family. The DB2 Family may be divided into two main groups:

- DB2 for midrange and mainframe environments

This group includes:

- DB2 for OS/400
- DB2 for VM
- DB2 for VSE
- DB2 for MVS

- DB2 for workstation environments (common servers)

This group includes:

- DB2 for OS/2
- DB2 for Windows NT
- DB2 for UNIX (various versions)

For simplicity, we refer to the DB2 Common Server family of products as DB2 in this document.

The DB2 products described in this chapter are all part of the DB2 Common Server area of the DB2 Family, including:

- DB2 Server
- DB2 Software Developer's Kit (SDK)
- Distributed Database Connection Services (DDCS)

DB2 provides seamless database connectivity using various popular network protocols, including NetBIOS, TCP/IP, IPX/SPX, and APPC. The communication between a database application and the database server is handled by DB2.

1.3.1 DB2 Servers

In Figure 5 on page 6, all of the DB2 database servers are shown. The database servers located above the dotted line are collectively known as DB2 Common Servers. The R/3 system currently supports DB2 for OS/400 and DB2 for UNIX. Every database application utilizes DB2 resources, known as databases. These databases may reside locally (on the same computer) or on a DB2 database server located within a network.

The database server for R/3 is supported on UNIX, OS/400, and NT platforms. The frontend (SAPGUI) also runs on different platforms, such as Windows or OS/2. The operating environment of the client application is independent of the operating environment of the DB2 database server. Therefore, a single application may access one or more of the DB2 database servers shown in Figure 5 on page 6.

1.3.2 DB2 Components

Each DB2 product includes a series of components. These components provide a variety of functions. Most of the components are optionally installed; some are for database administration, and others are used for application development.

The components are packaged together within the DB2 products to provide the required functionality. For example, the user-interface tool known as the

Command Line Processor (CLP) is available with all DB2 (Common Server) products. DB2 components are provided within each of the DB2 products.

Component \ Product	DB2 Server	DB2 Single User	DB2 SDK	DDCS Single User	DDCS Multi-User
	CAE	✓	✓	✓	✓
Command Line Processor	✓	✓	✓	✓	✓
Database Director	✓	✓	✓	✓	✓
Visual Explain	✓	✓	✓		
Performance Monitor	✓				

Administrator's Toolkit {

Figure 6. DB2 Products and Components

The DB2 components are:

- Client Application Enabler (CAE)
- Command Line Processor (CLP)
- Database Director
- Visual Explain
- Performance Monitor

The relationship between DB2 products and components is shown in Figure 6. The functionality of the DB2 components may be slightly different. For example, the DB2 Command Line Processor (CLP) is a multipurpose tool used to connect to DB2 databases and perform various operations. The CLP component provided with the DB2 Software Developer's Kit (SDK) will provide for application development tasks. The CLP component is also provided with the Distributed Database Connection Services (DDCS) product, but the application development tasks cannot be performed using the CLP interface. The Command Line Processor found in an R/3 system is the same as that found in the DB2 Server product.

1.3.3 DB2 Client Application Enabler (CAE)

The DB2 Client Application Enabler (CAE) is a component common to all DB2 products. Once a DB2 application has been developed, the DB2 CAE component must be installed on each workstation. Otherwise, the application could not utilize the CAE component to provide communication support with the database server. Figure 7 on page 9 shows the CAE component relationship between the application and the database server. If the application and database are installed on the same workstation, the application is known as a local client.

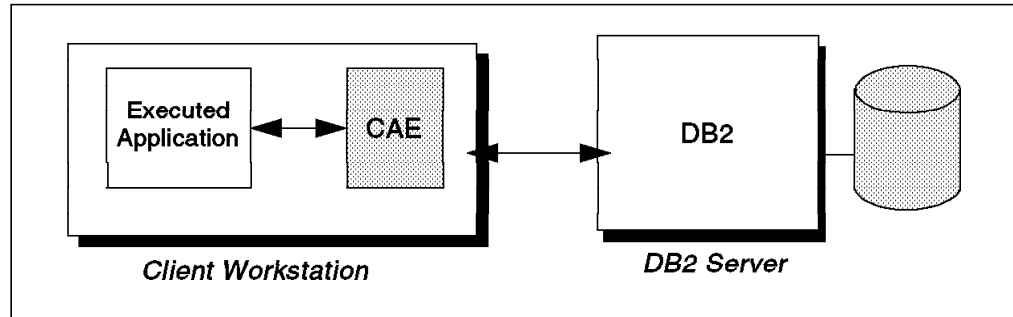


Figure 7. Remote Client Accessing DB2 Server Using CAE

For an R/3 system, the R/3 Application Server communicates with the DB2 CAE. You must install the operating-system specific CAE for your Application Server. For example, if you are using an RS/6000 as an Application Server, you must install the CAE for AIX.

1.3.4 DB2 Command Line Processor (CLP)

The Command Line Processor (CLP) is a component common to all DB2 products. It is a text-based application commonly used to execute SQL statements and DB2 commands. For example, you can create a database, catalog a database, and issue most of the supported SQL statements to retrieve data from a DB2 database.

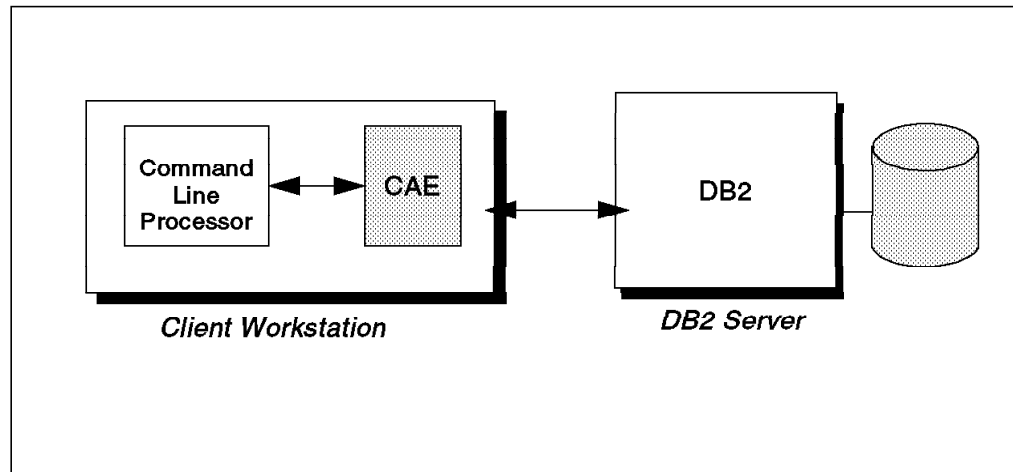


Figure 8. Command Line Processor

Figure 8 shows the Command Line Processor as an application that uses the CAE to communicate with the DB2 database server. The Command Line Processor may be used to issue interactive SQL statements or DB2 commands. The statements and commands can be placed in a file and executed in a batch environment, or they may be entered in an interactive mode.

All SQL statements issued from the Command Line Processor are dynamically prepared and executed on the database server. The output, or result, of the SQL query is displayed on the screen by default.

Many operations, such as making a backup image of the database, are known as DB2 commands. These commands may be issued from the CLP utility.

All of the DB2 commands are documented in the *DB2 Command Reference*.

1.3.5 DB2 Administrator's Toolkit

The DB2 Administrator's Toolkit contains a collection of tools that help manage and administer databases. You may optionally install the DB2 Administrator's Toolkit on any client or server workstation. The administration of the database can then be performed either locally at the server or remotely from a client by using the graphical interface provided.

The DB2 Administrator's Toolkit consists of the following components:

- Database Director
- Visual Explain
- Performance Monitor

1.3.6 Database Director

The Database Director provides a graphical interface to perform many administrative tasks within the database. These tasks are accomplished using the graphical interface because the database objects are represented by icons.

To invoke the Database Director from an R/3 System:

- Type `db2dd` at an operating system command line; or
- Access through SMIT
SMIT is the System Management Interface Tool that is part of AIX. To access the Database Director for an R/3 system, the following steps are needed:
 1. Type `smit` at the operating system prompt.
 2. Select **Applications**.
 3. Next select **IBM DATABASE 2 Database Director**.

You may have to set the `DISPLAY` variable within your AIX shell environment for the graphical tools that you want to use. The setting of the AIX environment variable is dependent on the shell that you use when logging in to AIX. For, example, if the address on which you want to display the Database Director is 9.3.1.13:

- ksh: `export DISPLAY=9.3.1.13:0`
- csh: `setenv DISPLAY 9.3.1.13:0`

The Database Director for an R/3 system is similar to the one shown in Figure 9 on page 11.

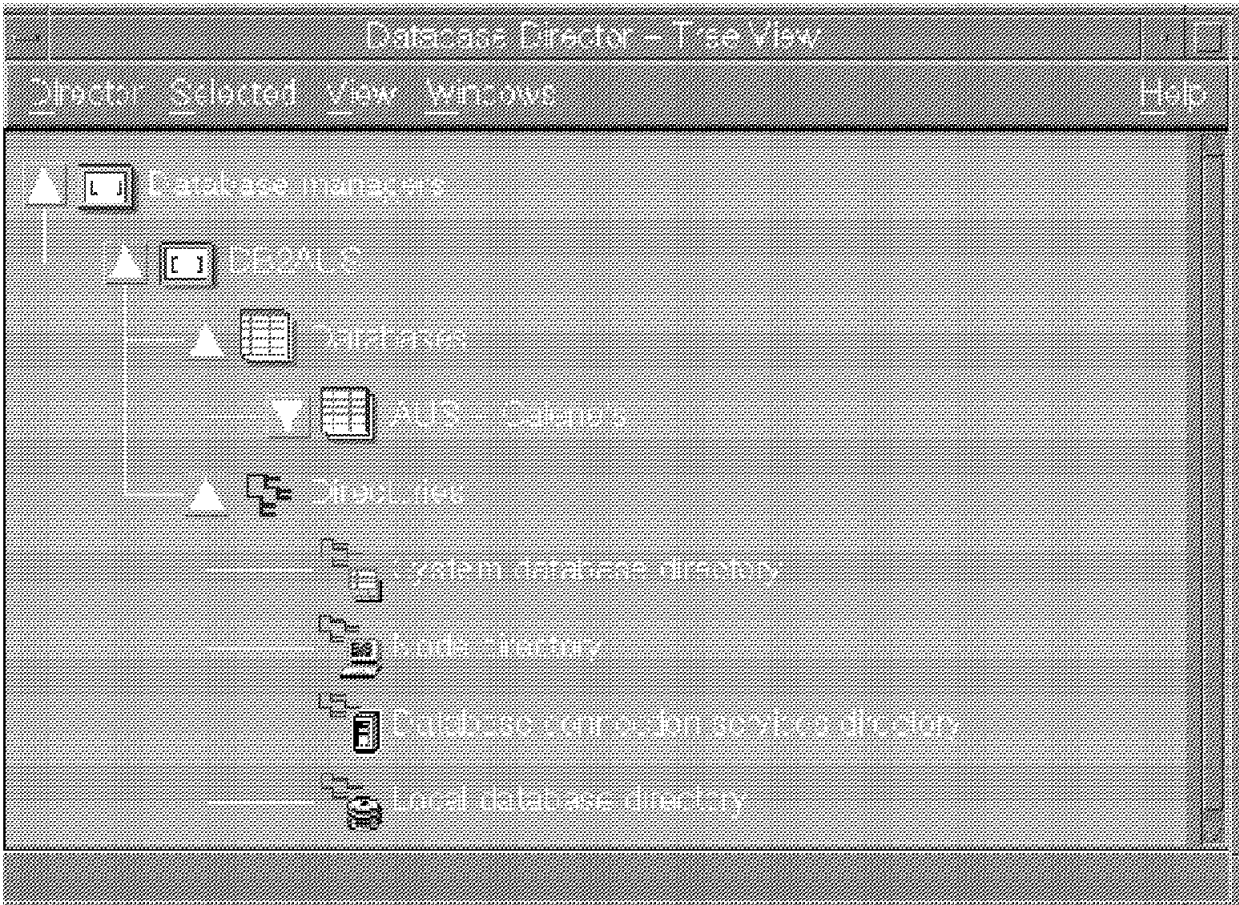


Figure 9. Database Director - Tree View

Each icon (DB2 object) is manipulated by using the menu bar or the pop-up menu. A single click of the mouse button on the icon will display the pop-up menu. Figure 9 shows the administration interface provided by the Database Director. Many of the commands that may be issued from the DB2 Command Line Processor may also be issued using the Database Director.

In Figure 9, the Database Director utility displays one database manager (known as a DB2 instance), DB2AUS. DB2 databases are created within a DB2 instance. We can see in Figure 9 that there is only one database called AUS created within the DB2 instance called DB2AUS. The level of R/3 code used in this document allows only one database in an DB2 instance within R/3.

The Database Director is a graphical interface used to issue DB2 commands. To contrast, the Command Line Processor is a text interface used to issue DB2 commands and SQL statements.

The Database Director may be used to perform the following tasks:

- Configure databases
- Configure instances
- Perform administrative tasks (back up and recover databases)
- Manage DB2 objects (tablespaces, tables)

The Database Director, by default, displays the database objects as a tree view. You may decide to change the view representation to be a list view. Using the

mouse, you may highlight an object and then open the object to perform various actions (these actions are different for each database object). A pop-up menu will be displayed when an object has been clicked once; from this menu you may perform an operation on the object. For example, to configure a database simply click on the database you wish to configure and select **Configure** from the pop-up menu. The Database Director component may be used to manage local and remote database resources.

1.3.7 Visual Explain

The Visual Explain is a graphical utility that provides a visual representation of the access plans that the DB2 server used to obtain the data for an SQL statement.

Visual Explain can be invoked from the Database Director or by entering the command `db2vexp` from an operating-system window. Figure 10 shows an example of the type of information that is displayed. In this example the table being accessed is called `test_taken`. An approximation of the cost of the query is also provided in the Visual Explain output. The query in Figure 10 has a cost of 452.92395. These costs are only an approximation, and they represent a relative cost of the query.

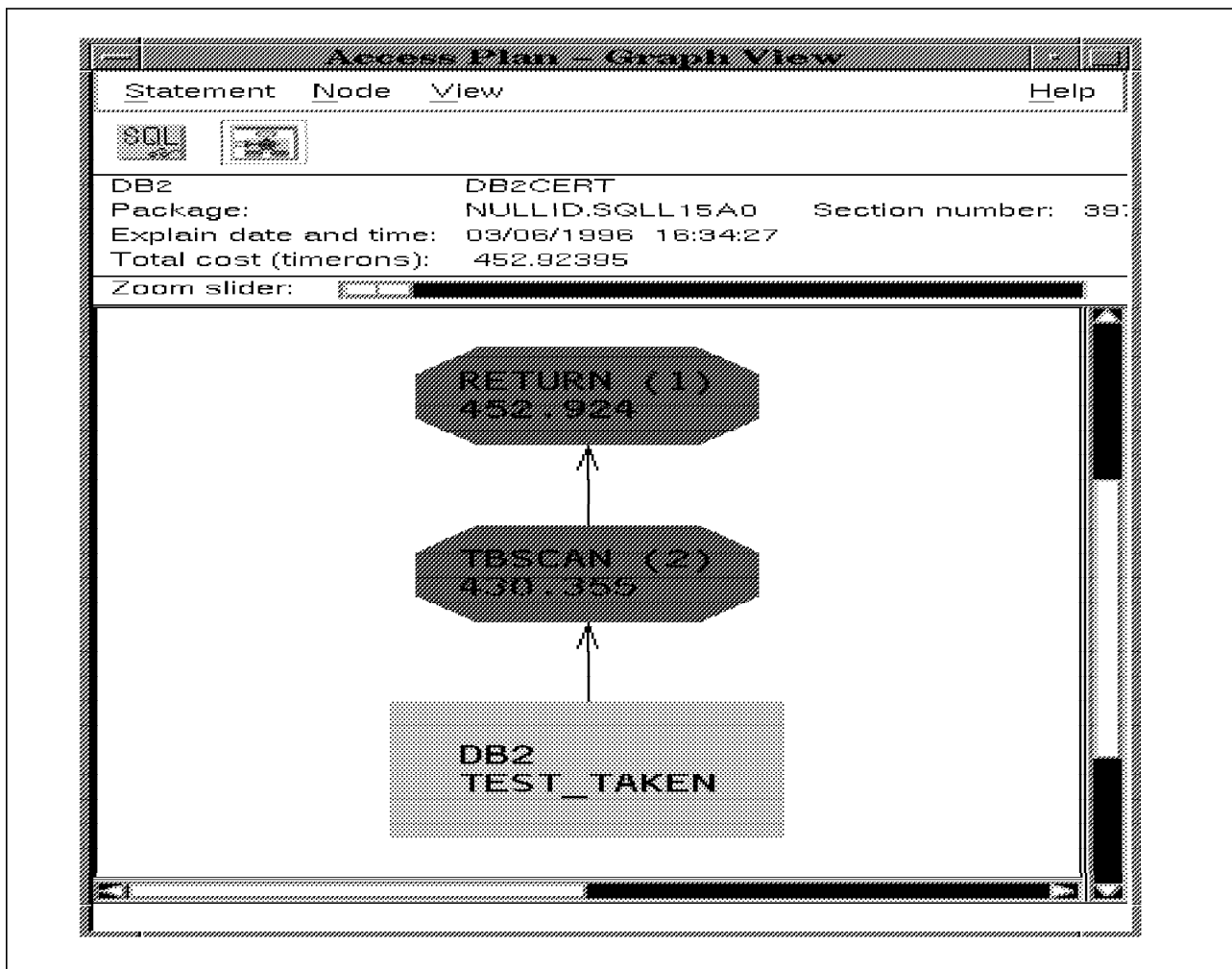


Figure 10. Visual Explain Output

1.3.8 Performance Monitor

The Performance Monitor is a graphical utility available on most of the DB2 operating systems. There are two basic monitoring facilities:

- Snapshot Monitor
- Event Monitor

The Snapshot Monitor captures information at specific intervals. The interval time and the data represented in the performance graph is configured.

Figure 11 is an example of output from the Snapshot Monitor that displays the amount of logical reads and logical writes that occurred in the buffer pool. The example shows the buffer pool hit ratio, which is at 100 percent. The information in the graph means that the data for a query was retrieved from memory (buffer pool), and disk I/O was not required. This tool can help to determine and analyze performance problems, tune SQL statements for better performance, and identify exception conditions based on limits or thresholds before they are defined.

The Event Monitor captures database activity as defined by the monitor definition. The Event Monitor records are usually stored on disk and then analyzed after the data has been captured. There is a graphical tool called the Event Analyzer provided with DB2 to help analyze the captured data.

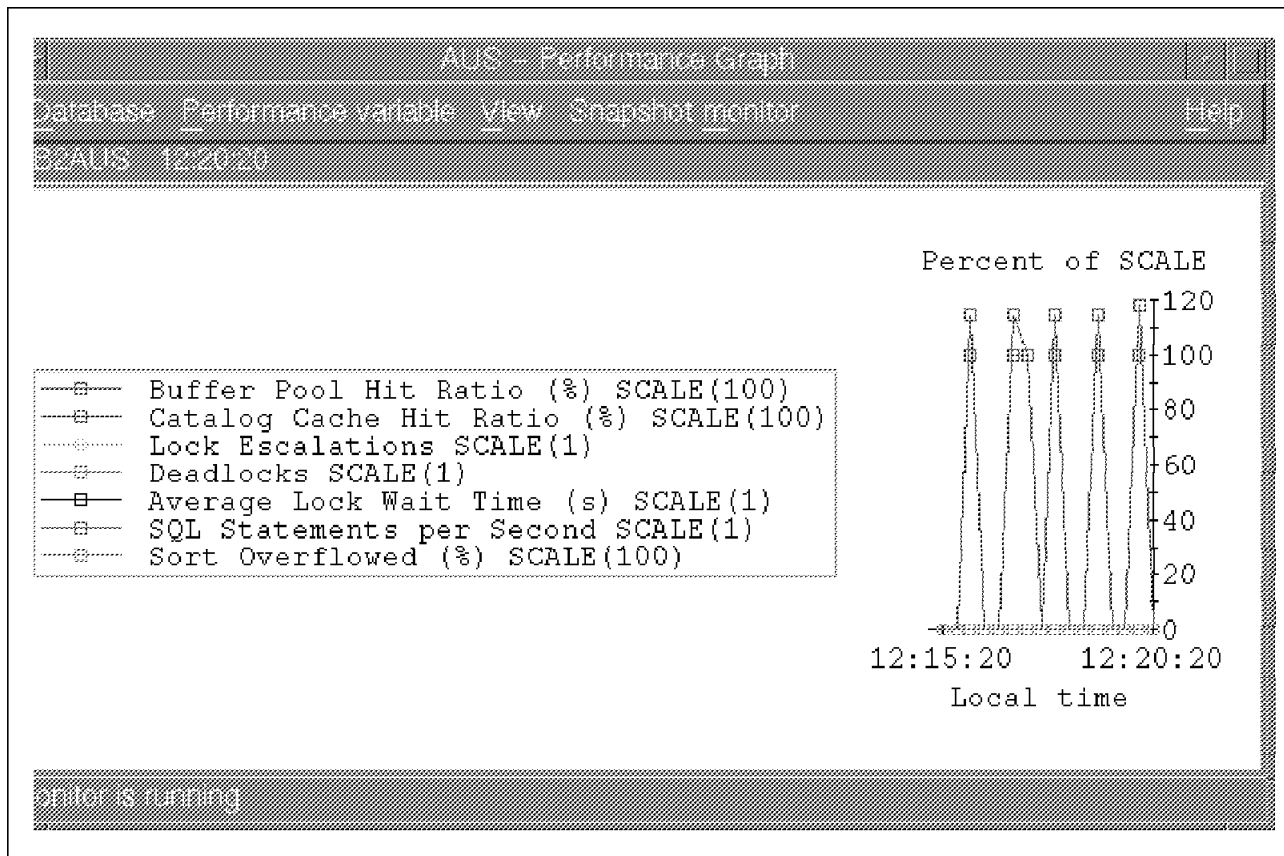


Figure 11. Performance Monitor Output

1.4 DB2 Products

The DB2 Common Server Family consists of a number of DB2 products and components. We have discussed the installable DB2 components, now let's examine the DB2 Products. There are three main DB2 products, including:

1. Distributed Database Connection Services (DDCS) - provides communication facilities to access host relational databases
2. DB2 Software Developer's Kit (SDK) - provides tools for application development
3. DB2 Server - provides a relational database engine

The database engine is available in a stand-alone version called DB2 Single-User and a multiuser version called DB2 Server. The functionality of the database engine in both products is identical. The only difference is in the ability to access incoming database requests. The engine includes a cost-based SQL optimizer, support for online transactions, and decision support systems. We briefly describe the DB2 products in this chapter. However, we focus on those DB2 products and components that are found in a R/3 system. 1.5.1, "R/3 System and DB2 Products" on page 16 lists the DB2 products and components that are installed on an R/3 Central Instance with Database Server and an Application Server.

1.4.1 DB2 Server

DB2 Server is designed for use in a LAN environment. It provides support for both remote and local database client access. A workstation with DB2 Server installed can be connected to a network and participate in a client/server environment as shown in Figure 12.

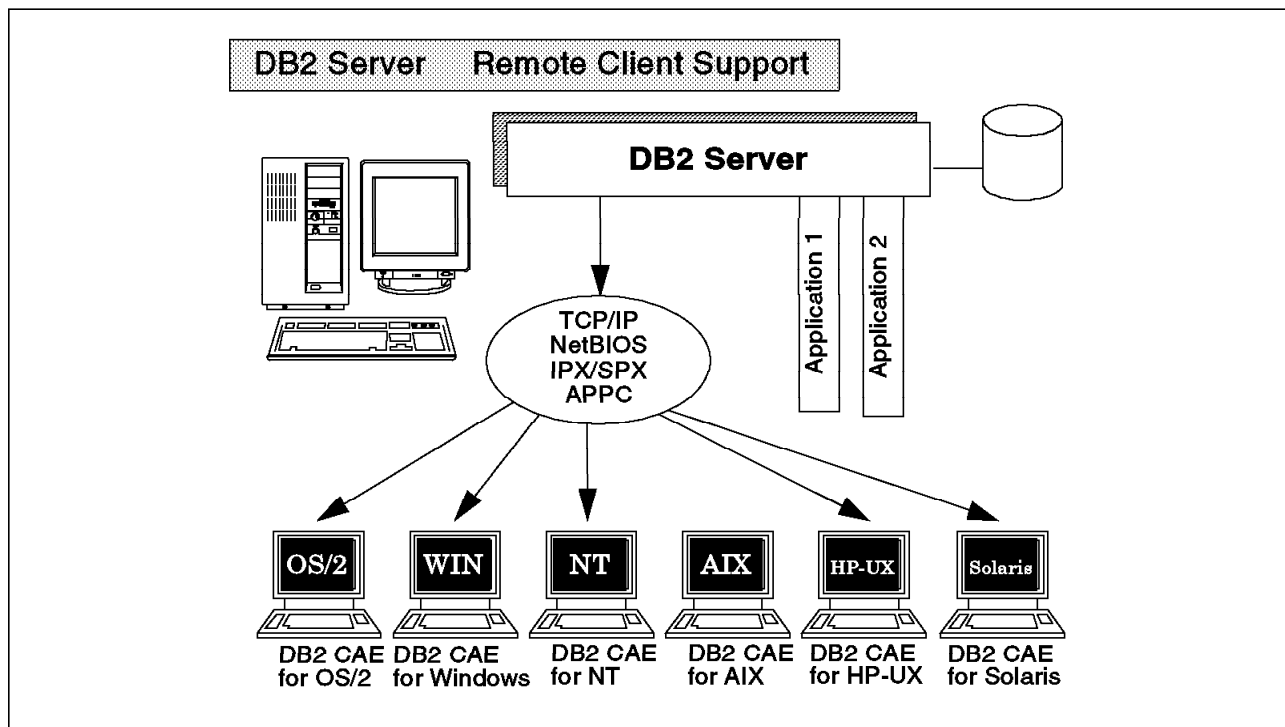


Figure 12. DB2 Server with Remote Clients

Here, application1 and application2 are local database users. Remote clients can also execute application1 and application2. The applications in this environment are split between server and client machines. Clients work without knowing the location of the physical database. Clients wanting remote access must communicate with the DB2 Server using a server-supported protocol and the CAE component. DB2 Server supports TCP/IP, NetBIOS, IPX/SPX, and APPC.

DB2 Server includes the Database Director, Visual Explain and Performance Monitor. In Figure 12 on page 14, the DB2 CAE component is not shown because the DB2 CAE component can only access DB2 Common Servers directly. The data flow from a DB2 CAE and a DB2 server is a private database protocol. This protocol can only be used to directly communicate with DB2 Common Servers.

1.4.2 DB2 Software Developer's Kit (SDK)

DB2 Software Developer's Kit (SDK) is a separate product that can be installed on either the server or on a client. It provides all of the necessary data access tools for embedded SQL applications and callable SQL interface applications.

The Software Developer's Kit also includes the Visual Explain tool, the Database Director and the Command Line Processor (CLP).

The Application Development Environment provided in the SDK allows application developers to write programs using embedded SQL, a callable SQL interface that is compatible with the Microsoft ODBC standard and various application programming interfaces (APIs) for accessing database utilities as well as the interactive SQL provided in the CLP.

The programming environment also includes the necessary programming libraries, header files, code samples, and precompilers for supported languages.

Several programming languages, including COBOL, FORTRAN, REXX, C, and C++ are supported for application development in the SDK product.

An application developed with SDK can be executed on a different client that has the CAE component installed. If the application development platform is different from the client platform where the application is executed, the application must be recompiled before executing on that client machine.

1.5 DB2 and R/3 Product Relationship

This section looks at the R/3 system and the DB2 products. When DB2 is installed on an UNIX system such as AIX, you will be able to selectively install components. When installing DB2 with R/3, the DB2 product is installed first. As part of the installation process, R/3 uses a utility called R3INST. R3INST checks that the most recent version of DB2 was installed. The following table lists those DB2 components found in an R/3 Central Instance with a Database Server installation.

IBM DATABASE 2 for AIX Version 2.1.1.1 Server	
db2_02_01.client	Client Application Enabler
db2_02_01.clp	Command Line Processor
db2_02_01.conv	DB2 Code Page Conversions
db2_02_01.cs.rte	DB2 Communications Support - Base with TCP/IP
db2_02_01.db2.misc	DB2 Utilities and Examples
db2_02_01.rte	DB2 Executables
db2_02_01.dd	DB2 Database Director
db2_02_01.doc.En_US.ipfx	DB2 Product Library - INF - En_US
db2_02_01.doc.En_US.pscript	DB2 Product Library - Postscript - En_US
db2_01_01.pm	DB2 Performance Monitor
db2_01_01.ve	DB2 Visual Explain
IBM DATABASE 2 Software Developer's Kit for AIX Version 2.1.1	
db2_02_01.sdk.c	DB2 C Language Include Files and Samples
db2_02_01.sdk.cli	DB2 Call Level Interface Samples
db2_02_01.misc	DB2 SDK Utilities and Samples

Table 1. DB2 Product Packaging for R/3 (Central Instance)

Note that these are not all of the DB2 components. For example, should you want a DB2 server to communicate via IPX/SPX, you would selectively install the db2_02_01.cs.ipx component.

When installing only the Application Server, the db2_02_01.client (Client Application Enabler) is the only DB2 component that is installed.

1.5.1 R/3 System and DB2 Products

This section looks at how the DB2 components fit into the R/3 implementation on a product basis. We'll look at the Central Instance with a Database Server and the Application Server.

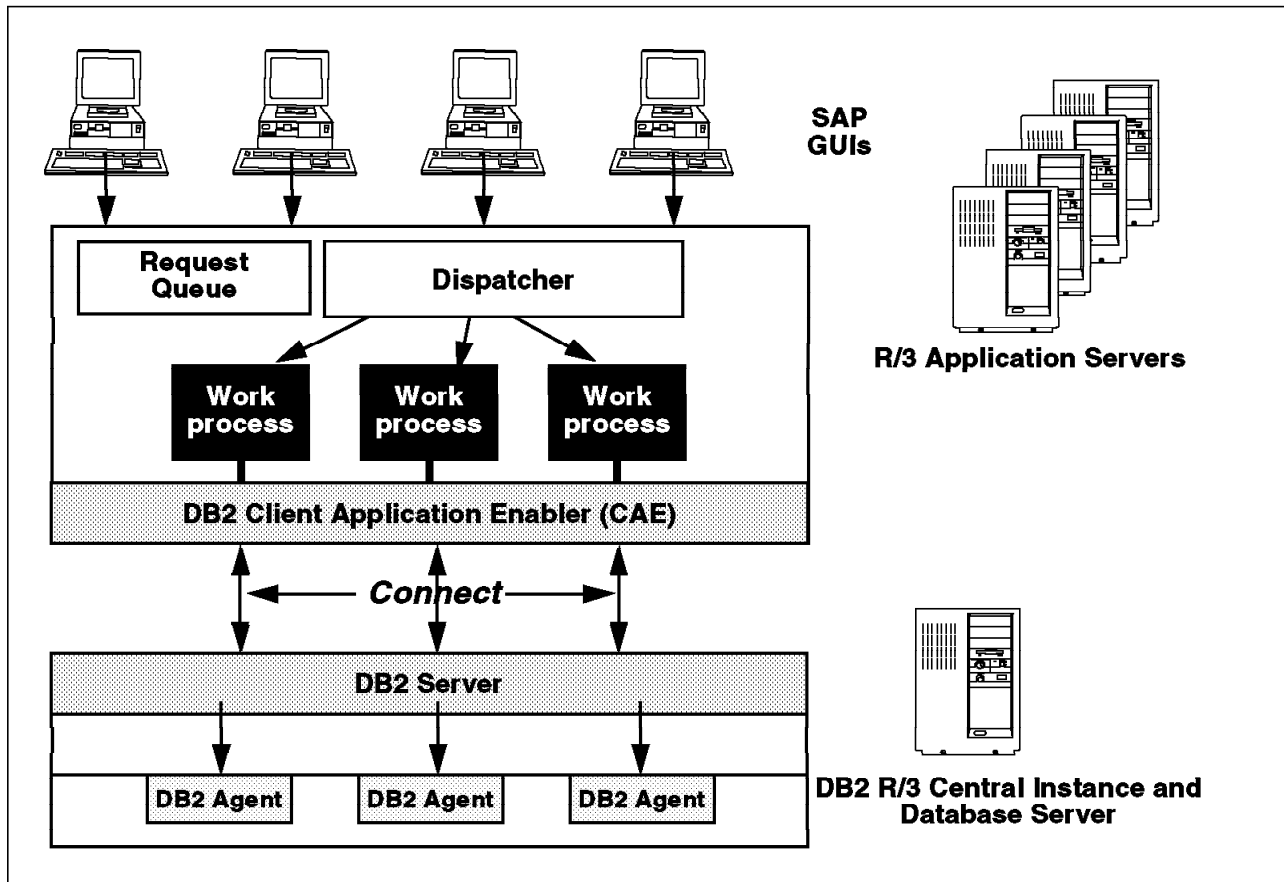


Figure 13. DB2 R/3 Product Relationship

Figure 13 shows a three-tier R/3 system. The Presentation Servers are represented as SAPGUIs. These are clients that access an R/3 Application Server. There may be one or more Application Servers.

The R/3 Application Server in this example shows the Request Queue and the Dispatcher as being on the Application Server. A request comes in from the SAPGUI. The Dispatcher distributes the work packages. The Work Process communicates with the Client Application Enabler where a connection to the DB2 R/3 Database Server is made. Each Work Process on the Application Server will have a corresponding DB2 Agent on the DB2 R/3 Server. A unique and independent agent process is assigned to handle different requests for service from the DB2 database manager. The type of service an agent might request could be a connection to a database, a request for a snapshot of an application or to perform some administrative function. Each DB2 agent process has its own memory, but shares the database manager and database manager resources such as the buffer pool, with other DB2 agents. The number of agents assigned to a database is a configurable parameter. For more information on configurable parameters, please see 2.5.2, "Database Manager Configuration Parameters" on page 47.

Chapter 2. Getting Ready for R/3 and DB2

This chapter looks at some of the considerations that an administrator needs to be aware of when preparing to install a DB2 R/3 Central Instance. We cover some of the terminology that is used both by DB2 and R/3. Though DB2 is found in many different hardware platforms and operating systems, we will specifically refer to DB2 for AIX and terms that would be particular to a DB2 AIX database server. This chapter is outlined as follows:

- **Installation Requirements** - Depending on your R/3 environment, you will need to prepare for the installation of R/3 and DB2. You will need to be aware of such items as how much memory and disk are required and where the products and data will be physically placed.
- **DB2 Storage Concepts** - This section looks at some of the DB2-specific terminology that is used when dealing with physical storage and logical database objects.
- **Data Placement** - You will need to have an understanding of what physical structures are created when the R/3 and DB2 products are installed and a database is created. This section also shows how to create a database object, called the tablespace.
- **Configuration Parameters** - This section looks at parameters within DB2, both at the instance level and the database level. Specifically, we discuss how certain parameters are used within R/3.
- **Unique DB2 Features** - There are many features in DB2 that a Database Administrator (DBA) can utilize. Understanding these is key to tuning. We discuss items such as asynchronous page cleaners and I/O prefetching.

2.1 Installation Overview

In preparing for installation, you'll need to install a Central Instance and Database Server before installing any Application Server or Presentation Server (frontends).

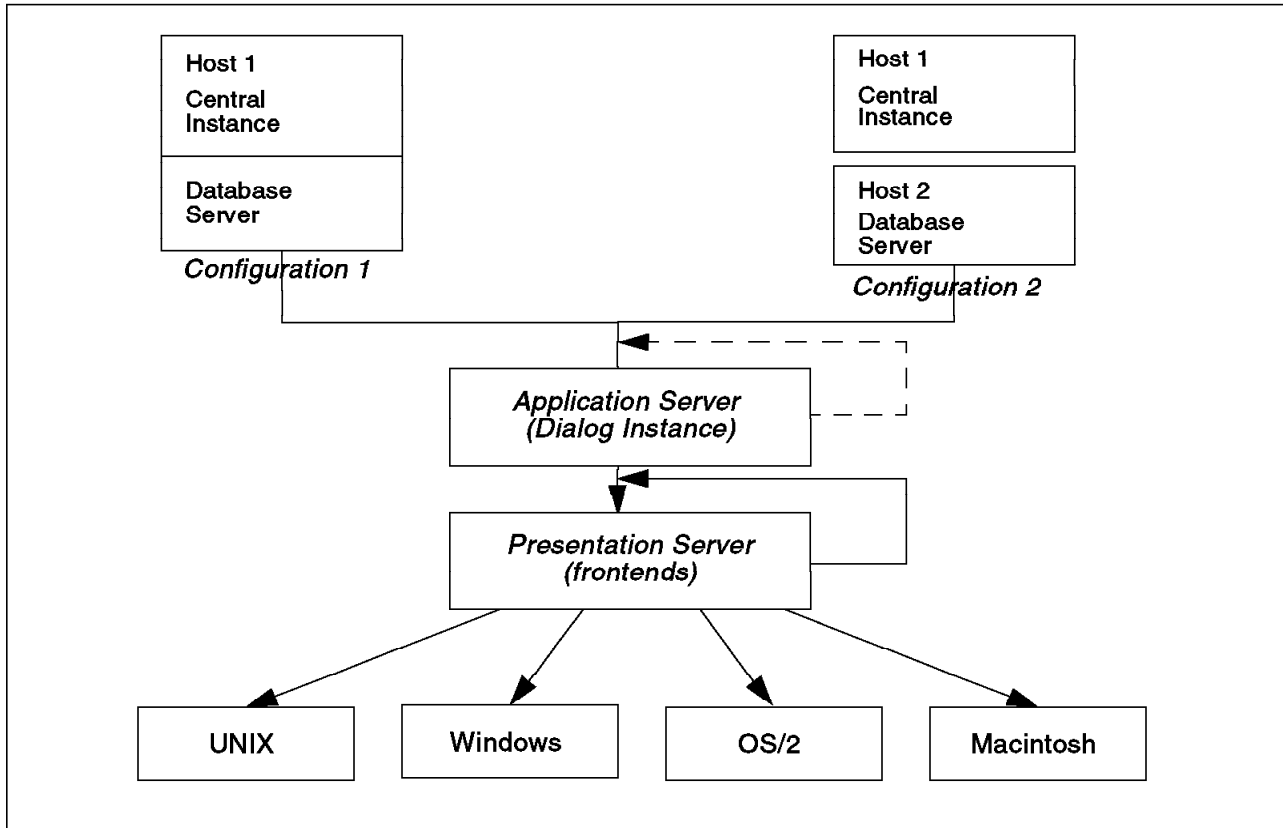


Figure 14. R/3 Installation Sequence

Figure 14 shows two possible configurations. Depending on your environment, your R/3 installation steps would be different. The Release 3.0 installation supports the following possible installations:

- A Central Instance and Database Server on one system (Configuration 1 - Host 1).
- Two machines where the Central Instance is installed on one host machine and the Database Server is installed on another machine (Configuration 2 Host 1 and Host 2).
- The Database Server as a stand-alone machine (Configuration 2 - Host 2 only)
- The Application Server. The dotted arrow in Figure 14 indicates there may be more than one system in which the Application Server is installed.
- UNIX Presentation Server frontends installation. There may be more than one system in which this is also installed.

There are numerous clients (PCs) that are possible using the UNIX Presentation Server.

Your installation sequence of R/3 depends upon your hardware configuration. Let's look at the steps for two different scenarios. In either case, the first component to be installed is the Central Instance.

- Central Instance only:
 1. Determine file system sizes. For this, you use SAPFS.TPL. One of the functions of SAPFS.TPL is to store the default layout of the R/3 file

system structure. You can modify SAPFS.TPL using R3INST. R3INST is a utility to assist with installation that is provided with R/3.

2. Perform the AIX-specific steps, such as creating file systems or extra paging space. For more information, consult the R/3 installation guide for DB2 for AIX.
 3. Execute R3INST. R3INST is provided as a tool to simplify the R/3 installation. With the installation of only a Central Instance, R3INST checks file systems and free space, creates users and groups, loads the R/3 software, and configures the Central Instance environment
- Central Instance with Database Server:
 1. Determine file system sizes. This step uses SAPFS.TPL.
 2. Perform the same AIX-specific steps. However, the size of the file systems or swap space may be different when installing the Central Instance with the Database Server.
 3. Install DB2 from the R/3 installation media. To install DB2, the recommended install method is using SMIT (System Management Interface Tool) rather than the command line. SMIT is found in AIX. The reason you must install DB2 prior to R/3 is that the R3INST utility for DB2 uses a dynamic library of DB2. This ensures that the most recent version of DB2 is used for the installation.
 4. Execute R3INST. Here R3INST performs all of the same functions as those found in a Central Instance with the installation of the database software, creating the database, loading the SAP-data into the database, and performing an import of the ABAP/4 report loads.

One of the first things you must decide when beginning an R/3 DB2 installation is the name you will give to your <SAPSID>. This <SAPSID> stands for SAP System ID. The current implementation of R/3 allows only one DB2 database. The database name and <SAPSID> are the same. This is a case-sensitive name that must be at least three characters long and begin with a letter. For example, we have used the <SAPSID> AUS. You will see how this SAP System ID or <SAPSID> is used in the R/3 implementation. For the remainder of this chapter (or document), wherever a <SAPSID> is required, we use AUS.

2.2 R/3 and DB2 Configuration Guidelines and Planning

This section looks at some of the planning considerations for installing an R/3 system. We do not cover the actual installation, but focus on what you need in preparation on your AIX system when installing an R/3 DB2 Central Instance. The following is a list of planning items that we cover in this section.

1. Make sure you have minimal memory and disk requirements for R/3 installation.
2. Decide how and where in AIX to physically place your database-related objects.

2.2.1 Minimum R/3 Memory and Disk Requirements

The following is a list of hardware requirements for a Central Instance with database:

- CD-ROM
This must be ISO9660 compatible.
- Tape Device
- 256 MB Memory
This amount of memory is the minimum required for a Central Instance in a productive system. You need this to support memory management for R/3 together with a large DB2 buffer pool. For more information on DB2 configuration parameters, see 2.5, “DB2 Configuration Parameters” on page 44.
- A minimum of three separate disks for database-related storage
Five physical disks is the recommended minimum number of physical disks for database-related storage. You will want to separate different kinds of database-related data. The next section talks about this in more detail.
- 6.5 GB of disk space
This amount depends on the release of R/3 you are installing. Remember to consider the number of disks and the minimum amount of disk space required together.
- 1.25 GB of Paging and Swap Space
This figure also depends on the current implementation of R/3. Check the *R/3 Installation Guide* for current numbers.

2.2.2 Physical Storage

There are a number of steps that an administrator needs to do prior to installation. One of the first is to determine the configuration in which you will install the R/3 product. In any RDBMS (Relational Database Management System), you'll want to protect your database files as much as possible.

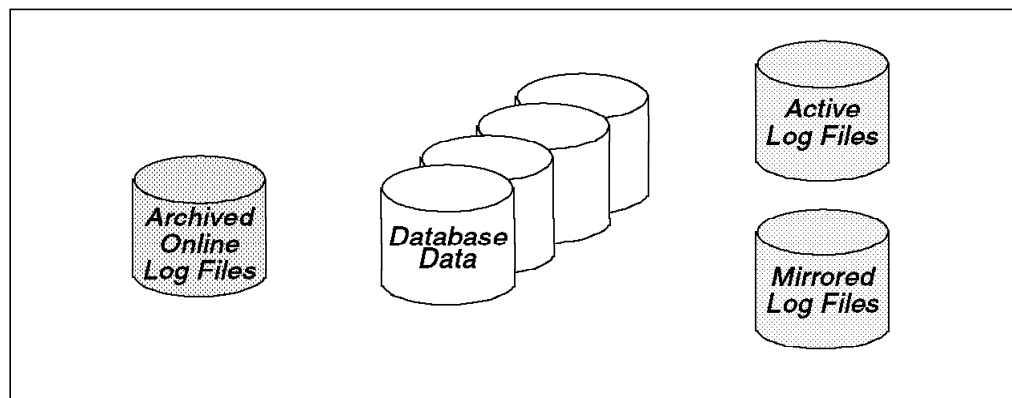


Figure 15. Optimal Distribution of Data in an DB2 R/3 System

Figure 15 shows an optimal distribution of database data. At a minimum, you would want your database data and log files on separate disks for recoverability purposes. Within AIX, it is possible to mirror files (Logical Volumes). You can have up to two copies of the same data (thus giving you three copies total) within AIX. Preferably, these mirrored copies are on separate physical disks. Log files are files that are used by DB2 to ensure the integrity of your database, even when the system crashes due to some unforeseen problem, such as a

power failure. A database will have different types of log files. Active logs contain information related to work (transactions) that have not been committed or rolled back. They may also contain information about transactions that have been committed, but not written to the database files. Online archived log files contain information related to completed transactions no longer required for crash recovery protection. In R/3, these online log files are moved from the current log directory to the archive log directory so that they do not fill a file system within AIX.

2.2.2.1 Recommendations

As part of your system planning, consider the following three items:

- Security
- Performance
- Size and Scalability

The DB2 log files should be mirrored in an R/3 production system. The mirroring of the log files can be done either by hardware or software. For example, you could implement a Redundant Array of Independent Disks (RAID) Level 5 disk array. However, a RAID Level 5 disk array may not be the best way to implement some of the DB2 storage structures. To mirror at the software level, you could use the facilities within AIX. This is covered in 2.2.3, “AIX Facilities for Storage” on page 24.

In a storage-constrained environment, you can install the log files on the same disk as the database files. However, this is not recommended. Should you experience a hardware failure, you will have to perform a complete media recovery if that disk is corrupted. However, in an R/3 production system, you must keep the archived online log files on a separate disk from the active log files.

For performance reasons, you should avoid placing non-DB2 files on the same physical disk as the DB2 data file. This would include your paging or swap space and non-R/3 applications. The log files are very I/O intensive. A desirable placement is on a small, fast, dedicated disk. DB2 allows you to separate regular table data from its indexes. It is recommended to place regular table data and indexes on separate physical devices.

The SAPFS.TPL configuration file that comes with the R/3 installation should be modified to take advantage of available disk space before the database is created. There are a (N) number of file systems created and called SAPDATA. SAPDATA is the location for the data within your R/3 DB2 database. If possible, place each one of the SAPDATA<N> file systems on a separate disk.

Finally, consider future growth of your R/3 system. There are a number of configuration parameters that you can change within DB2. For example, depending on the size of your database, you can increase the number and size of the primary log files. For more information about these parameters, see 2.5, “DB2 Configuration Parameters” on page 44. Allow for future growth of the file systems in which the log files are kept. There is a current limitation within AIX 4.1 of 64 GB per file system. Any one file in the file system cannot be more than 2 GB. In AIX Version 4.2 and higher, this limitation is removed.

2.2.3 AIX Facilities for Storage

If you are familiar with AIX, in particular the Logical Volume Manager (LVM) and the AIX Journaled File System (JFS), you can skip this section. We mentioned in the previous section that it is desirable to “mirror” your log files in an R/3 system. To do this within AIX, you need to understand the Logical Volume Manager or LVM. The LVM is an IBM enhancement to the UNIX file system. LVM is a layer of software between the application and the hardware that manages storage. You can access the LVM through SMIT or through the command line.

An AIX application performs I/O operations to disk in one of two ways:

1. Use the AIX Journaled File System (JFS).
2. The application uses its own input/output routines, bypassing most of those routines supplied by AIX. Instead, the application deals directly with the disk device.

A file system is a hierarchical structure of directories and files. AIX provides a native file system called the Journaled File System, or JFS. It uses a database journaling mechanism to maintain its structural consistency. One file system resides on one Logical Volume.

2.2.3.1 Logical Volume Manager (LVM)

In order to understand how the LVM can copy or mirror your data, there is some basic terminology that we discuss.

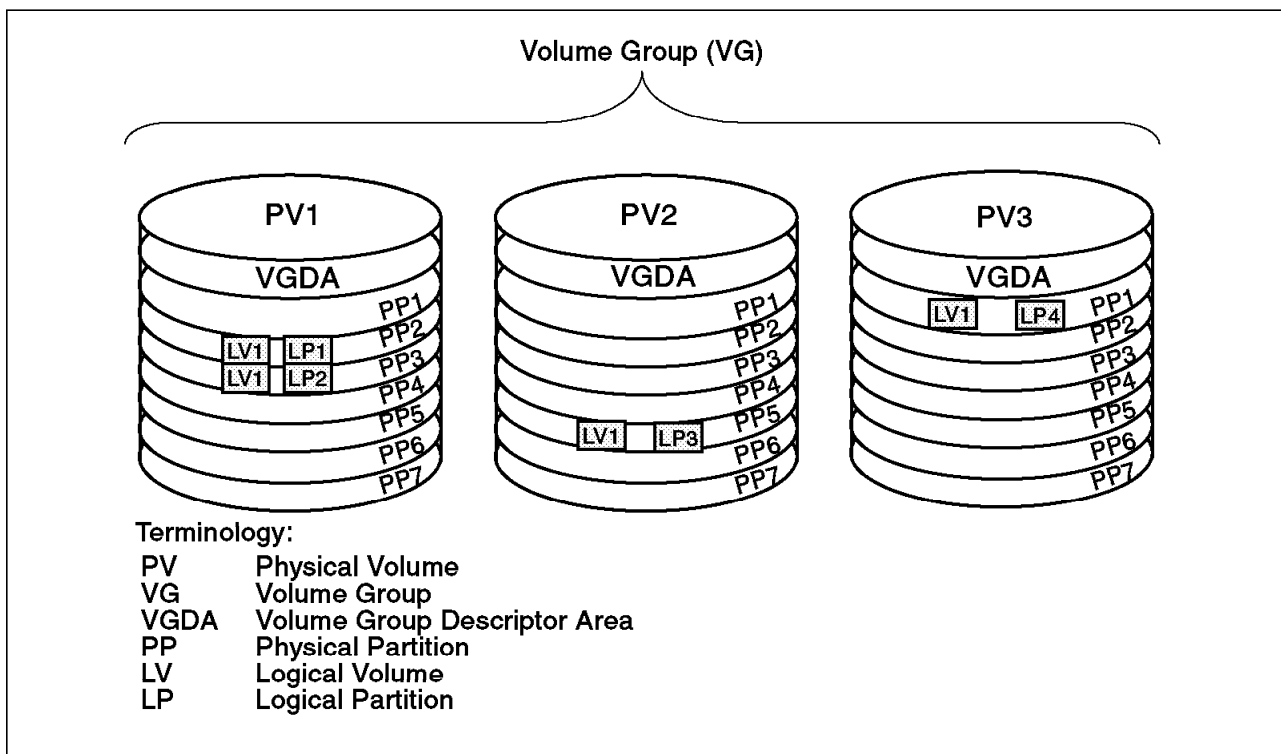


Figure 16. LVM - Logical Volume Manager (No Mirroring)

Figure 16 shows some of the terminology associated with the Logical Volume Manager in AIX. A collection of disks is grouped together into a *Volume Group* (VG). There can be a maximum of 255 VGs per system. A Physical Volume that supports removable media, RAID5 for example, should be assigned to a VG

containing itself and no other members. Each physical disk in the Volume Group is also referred to as a *Physical Volume* or PV. The Physical Volumes, or PVs, are numbered, starting at 1. In our diagram, there are three Physical Volumes, PV1, PV2, and PV3. There is a limit of 32 Physical Volumes per Volume Group.

Each Physical Volume or disk is divided into equal physical partitions. A *Physical Partition (PP)* is the smallest unit of disk allocated within the Volume Group. Physical partitions can be from 1 to 256 MB. The default size of a PP is 4 MB depending on the total size of your physical disk. If you are using a physical disk of 4 GB, the recommended PP size is 8 MB. The PP is a fixed size of contiguous bytes on a Physical Volume. The PP must be the same size across an entire Volume Group. There may be multiple VGs on a single system, each having a different PP size.

The VGDA is the Volume Group Descriptor Area. There is usually one VGDA per Physical Volume. The exceptions are when there is a VG of either one or two PVs. When there is only one PV, there are two VGDA's on that volume. When there are two PVs, there are two VGDA's on one and one on the other. The system administrator will use the varyonvg command to bring a Volume Group on-line and make it available to the system and its users. The varyoffvg command makes the VG unavailable. Only the root user has access to these commands. You must have what is referred to as a quorum within the VGs of the VGDA to varyonvg the Volume Group. A quorum is 51 percent of the VGDA's.

The LV is called the *Logical Volume*. The Logical Volume can span multiple Physical Volumes or disks within the Volume Group. However, it is contained within a single Volume Group. A file system sits on top of (is mounted over) a LV. A LV can be extended dynamically. There is a limit of 256 LVs per VG.

Logical Partitions are mapped one-to-one physically to a Physical Partition (PP), unless there is mirroring.

Figure 16 on page 24 shows one VG with three PVs. We can see from the diagram that there is one LV in which a file system resides. The file system is extended over the three Physical Volumes in the Volume Group. There is a total of 4 PPs in our LV, here called LV1. If our PP size is 4 MB, then there is a total of 16 MB available for use in the file system. Should we need more, we can extend, assuming that space is available within the Volume Group. In this scenario, there is a one-to-one correspondence between the Logical Partition (LP) and the Physical Partition (PP) because there is no mirroring. Next, we look at an example of the LVM that utilizes mirroring.

2.2.3.2 LVM - Mirroring

The Logical Volume Manager within AIX provides mirroring to protect against a disk failure. You are able to have up to three copies, one primary and one secondary, of a file system. In the event that the disk containing the primary copy fails, AIX continues working off of the secondary copies. All of this is transparent to DB2 and R/3.

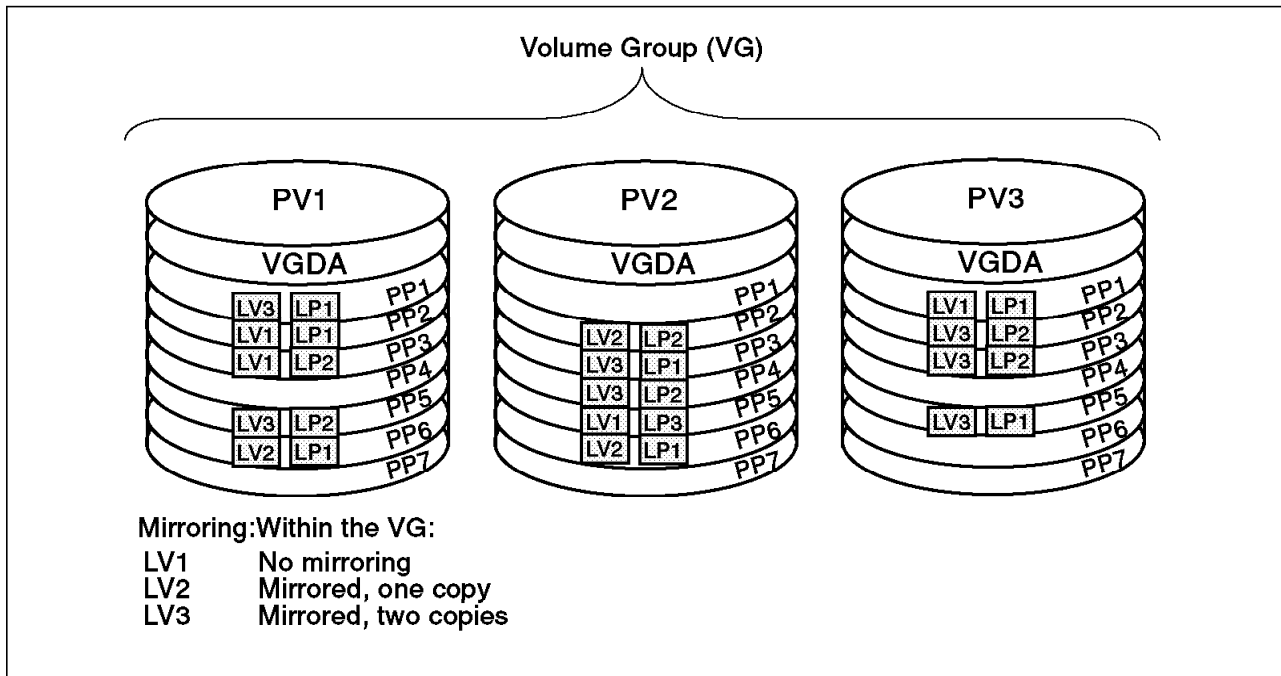


Figure 17. LVM - Mirroring

Mirroring is handled on the LP (Logical Partition) level from the LVM's perspective. However, the user requests mirroring on an LV (Logical Volume) level. Figure 17 shows three LVs (file systems). We have already seen LV1 from our previous example. There is no mirroring there. LV2 is mirrored with one copy. LV2 with LP1 can be found on both PV1 (PP6) and PV2 (PP6). LV2 with LP2 can be found on PV2 and PV3.

Next, LV3 is mirrored with two copies. LV3 with LP1 can be found on PV1 in PP1. The same information can also be found (LV3 - LP1) in PP3 on PV2 and in PP5 in PV3.

The user, when setting up the mirrored copies, can decide on the write policy to disk. You can request either that the write policy be sequential or parallel. A parallel write policy is one in which the mirrored copies are written to disk at the same time. A parallel write policy may be faster. A sequential write policy, since the writing of data is done in separate occurrences, may be slower than parallel, but could offer slightly more protection should a disaster occur during a write action where parallel write is chosen. During a read operation from disk, the fastest copy is read, if possible. If a bad block is detected, the LVM relocates it and fixes it if possible.

AIX can mirror file systems, not files. You can have dual logging in an R/3 DB2 system by creating a Logical Volume for the log files to control where the log files are written. You want to ensure that the logs and the database are on separate file systems.

2.2.4 Physical Storage Scenario for R/3 and DB2

This section looks at sample R/3 and DB2 installation from an AIX view. We will illustrate one possible physical disk layout of the R/3 and DB2 installation. This scenario covers the R/3 information and the DB2 environment. It does not show the actual placement of the DB2 products. DB2 for AIX is an LPP (Licensed Program Product). All LPPs in AIX are placed in /usr/lpp. For more information on the components installed with the DB2 product, please see 1.5, “DB2 and R/3 Product Relationship” on page 15. Here we show the result of a R/3 Central Instance installation with a DB2 database environment with a total of seven physical disks.

In our scenario, there are two Volume Groups, rootvg and sapr3vg. rootvg is where the AIX operating system and the DB2 product is installed. The other six Physical Volumes comprise the Volume Group, sapr3vg.

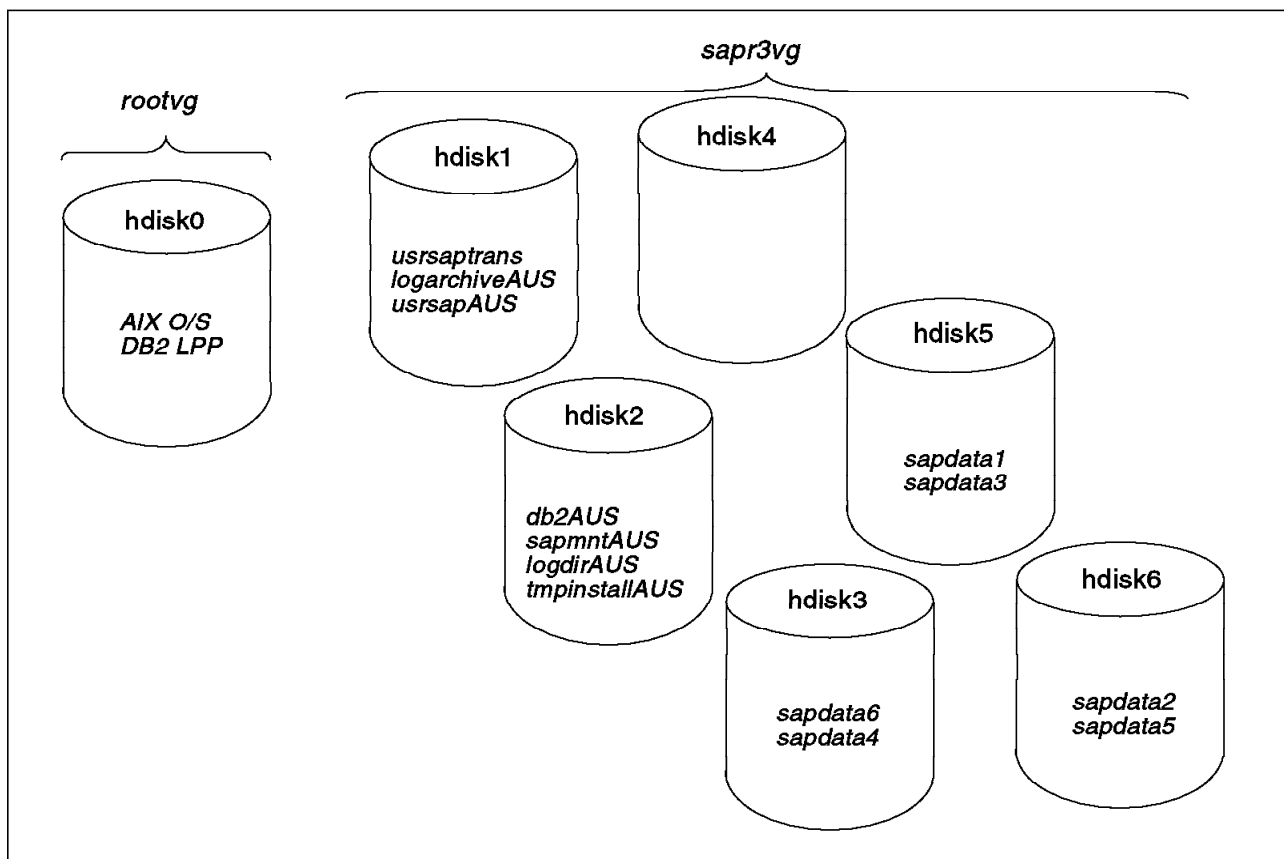


Figure 18. R/3 Volume Group and Logical Volumes

Figure 18 shows a sample layout of disks in an R/3 Central Instance Database Server. The AIX operating system and the DB2 database are contained in hdisk0. The other remaining disk are grouped together into a Volume Group, sapr3vg. We have separated the active log files from the archive log files, hdisk 2 and hdisk1. The concept of logging is discussed in more detail in 5.1, “Logging Considerations” on page 155. Optimally, we would like a separate disk for each Logical Volume (sapdata 1—sapdata6). We have chosen this distribution to allow for growth using hdisk4. We could also use hdisk4 to mirror the log files on hdisk2 (logdirAUS).

From here, we map the physical storage onto the DB2 and R/3 structures. First, we talk about some of the storage terminology that you will use with DB2 database objects.

2.3 Storage Concepts in DB2

The storage model in DB2 Version 2 led to new or different definitions of storage items than were found in earlier versions. The concepts this section discusses are:

- Container
- Tablespace
- Extent

2.3.1 Container

A container is a generic term used to describe the allocation of physical space.

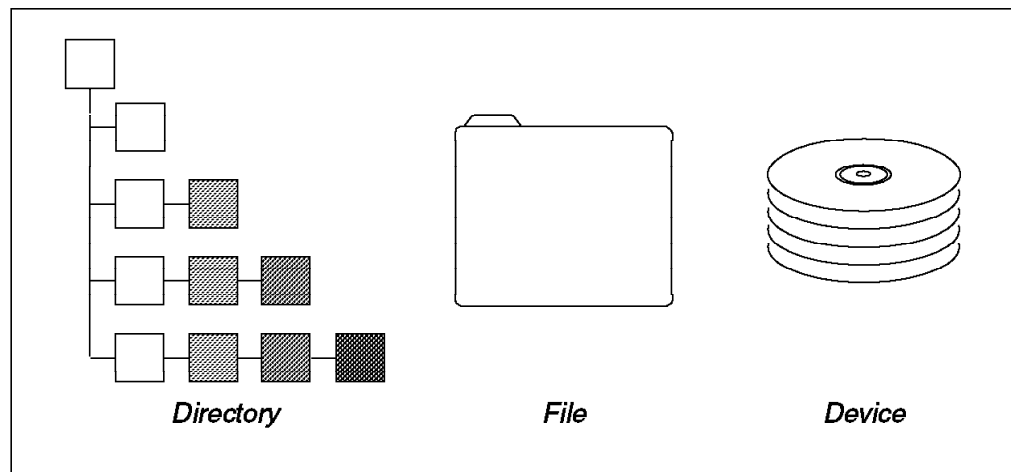


Figure 19. Containers in DB2

The type of container depends on the type of tablespace and the platform. For AIX, a container can be either a directory, file or device. The type of tablespace and operating system determines what type of containers you can use. When an R/3 system is installed, the containers, by default, are files in an AIX Journaled File System (JFS). For more information about containers and JFS, see 2.4.1, “JFS and RDBMS Objects” on page 35.

2.3.2 Tablespaces in DB2

Tablespaces exist in DB2 to provide you with a logical layer between your data and storage devices. All DB2 R/3 tables reside in a tablespace. This means you are able to control where data is stored. You can use different kinds of tablespaces to store different kinds of data. This gives you the ability to create a more detailed physical database design to fit your particular environment. For example, you can choose slower disks to store less-used data and faster disks to store indexes or frequently accessed data.

Recovery can be done at the tablespace level. Backup and recovery operations can be made for a tablespace. This gives you more granularity and control since you can back up or restore each tablespace individually.

2.3.2.1 Tablespaces and Containers

There is a one-to-many relationship between a tablespace and containers. Multiple containers may be defined for a tablespace. However, a container can only be assigned to one tablespace. Figure 20 shows this one-to-many relationship.

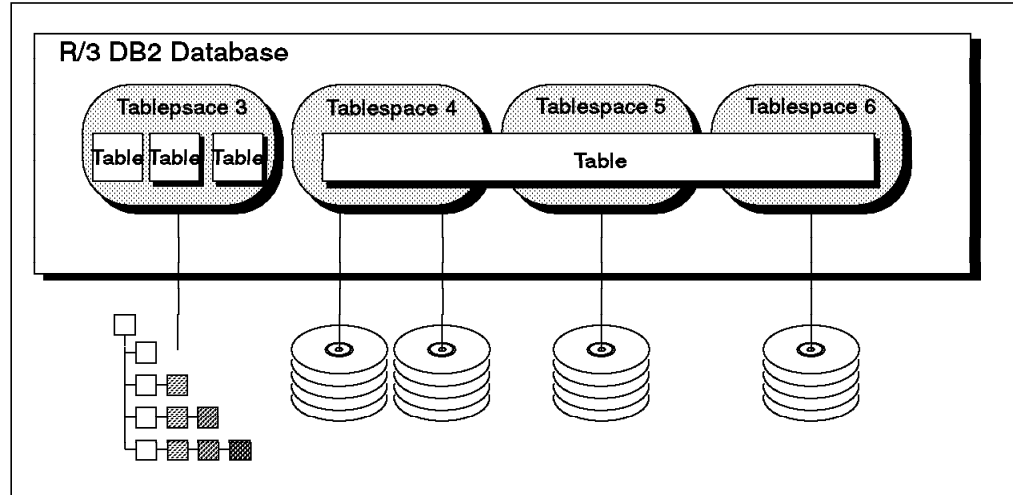


Figure 20. Tablespace and Container One-to-Many Relationship

Tablespace 3 has only one container assigned to it, a directory. Tablespace 4 has two containers assigned to it. The containers for Tablespace 4, Tablespace 5 and Tablespace 6 are devices. A mixture of containers is possible within the database. You may also mix container types within a tablespace, though it is not recommended for performance reasons. Notice that a table can span multiple tablespaces. In Figure 20 one table spans Tablespace 4, Tablespace 5, and Tablespace 6.

2.3.2.2 Extents

An extent is an allocation of space within a container of a tablespace. Database objects are stored in pages within DB2 (except for LOBs). These pages are grouped into allocation units called extents. The extent size is defined at the tablespace level. Once the extent size is established for the tablespace, it cannot be changed.

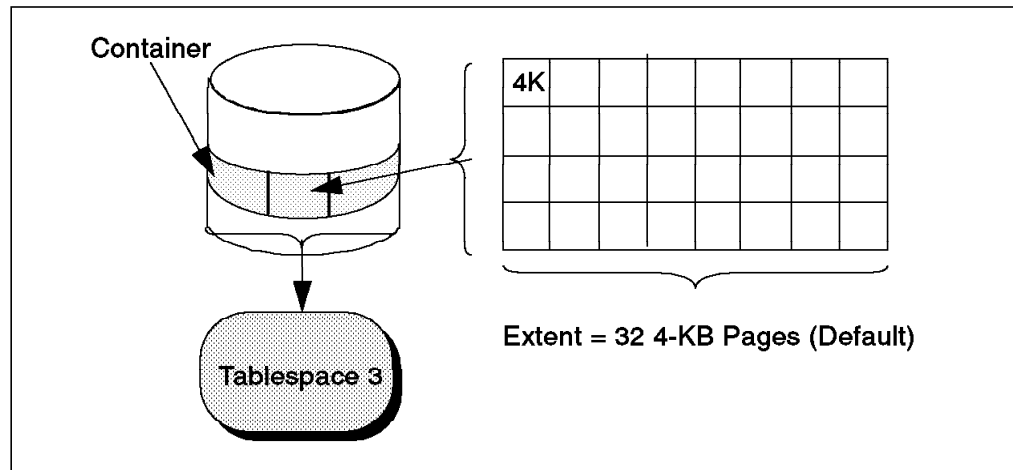


Figure 21. Extent, Containers, and Tablespace in DB2

Figure 21 shows the relationship between an extent, a container, and a tablespace. The tablespace is initialized when it is created. As part of this initialization, an allocation size is set for the tablespace. This allocation size is called the extent size. The tablespace page size is 4096 bytes (4K). The default extent size for a tablespace is 32 4-KB pages. The R/3 database uses extent sizes of 8, 16, 32, and 64. More information about the DB2 R/3 tablespaces can be found in 4.2.4, “Displaying Tablespace Information - R/3 Performance Monitor” on page 89.

2.3.2.3 Tablespace Types

DB2 supports two kinds of tablespaces:

- System Managed Storage (SMS) tablespace
- Database Managed Storage (DMS) tablespace

SMS Tablespaces and R/3

Since the release of R/3 2.2 G, SMS tablespaces are not supported. Check the product manual for more details.

System Managed Storage (SMS) tablespaces use the operating system’s file system. For AIX, the file system is the Journaled File System (JFS). Containers in SMS tablespaces have some of the following characteristics:

- The container in an SMS tablespace does not pre-allocate its storage. There will be some space used during tablespace creation for overhead.
- Containers cannot be added to a SMS tablespace after the tablespace is created.
- The total number of containers in a SMS tablespace is determined when the tablespace is created. (This is either when the database is created or when the tablespace is created.)
- SMS tablespaces can only use directories as containers.
- All the table objects must reside in the same SMS tablespace.

Database Managed Storage (DMS) tablespaces are characterized by tablespaces that are built on pre-allocated portions of storage. This storage container can either be a device or a file.

The database manager controls the storage space and allocates space when the container is created. When working with containers and DMS tablespaces, the following statements apply:

- If the container is a file, it is created when the tablespace is created and dropped when the tablespace is dropped.
- If the container is a Logical Volume in AIX, the container must exist before creating the tablespace. After dropping the tablespace, the Logical Volume still exists and must be removed.
- Storage is pre-allocated to a container when a container is created.
- Containers can be added to a tablespace after the tablespace is created.

2.3.2.4 Distributing Tables in DMS Tablespaces

One of the biggest differences and advantages to using DMS tablespace over SMS tablespace is the ability to span a table over multiple tablespaces. When creating a table in a DMS tablespace, you can decide to place certain objects of the table in different tablespaces. DMS tablespaces allow you the flexibility to store Large Object Data (Long Field (LF) and (BLOBs) Binary Large Objects) and indexes in different tablespaces from the regular table data. The tablespaces used to store the table are selected when the table is created.

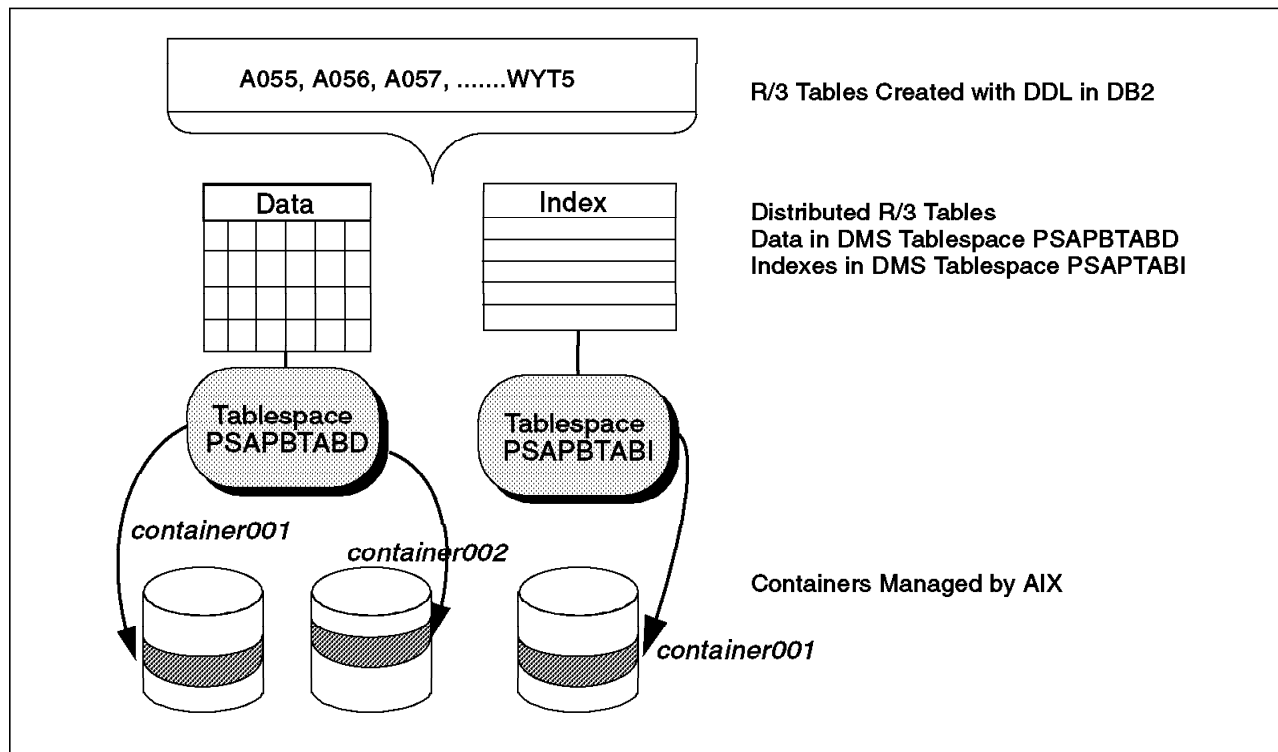


Figure 22. R/3 Space Management in DB2 for AIX

Figure 22 shows some of the R/3 tables being split among two DMS tablespaces. The regular table data is placed in Tablespace PSAPBTABD. Indexes for these tables are placed in Tablespace PSAPBTABI. R/3 has a naming convention to distinguish tablespaces that hold regular table data from tablespaces that hold the indexes for these tables. If the last character in the tablespace name ends with the letter "D", the tablespace contains regular data. Conversely, if the last character in the tablespace name ends with the letter "I", the tablespace contains indexes for the tables. In an R/3 environment, the naming of the tablespaces is descriptive. The name of the containers is derived from the tablespace name. We can also see from Figure 22 that Tablespace PSAPBTABD is created by using two containers. These containers do not have to be on separate disks. However, for performance reasons, it is better to distribute the containers to different physical devices if you can. This is covered in more detail in 2.6.2, "Prefetching in DB2" on page 55.

2.3.3 When to Use DMS Tablespaces

The DMS storage model has important benefits when compared to the SMS storage model. The following is a list of possible advantages:

- You have more control over the placement of database objects according to their type. Tables may be split across multiple DMS tablespaces, allowing the separation of table parts.
- You have more flexibility over administrative tasks such as backup and restore operations. You can control placement of less-frequently accessed items, like BLOBs, that may store images on separate tablespaces. These BLOBs, once created, may be data that is neither accessed nor frequently updated.
- There may be performance benefits in using DMS tablespaces because DB2 has more knowledge of the placement of the data. If using devices for DMS tablespaces, you can avoid the overhead of using the operating system's file system.
- DMS tablespaces provide you scalability. You can dynamically add containers to the tablespace online. Rebalancing of the data is done immediately and automatically when a container is added. Rebalancing is the process of evenly redistributing data amongst the containers.
- If you know the maximum size of your tablespace, consider using DMS tablespaces. DMS pre-allocates space as database objects are inserted. The database does not have to compete with other resources that may be used by other applications.

Tablespaces in the default R/3 installation are all DMS tablespaces. SMS tablespaces are not supported in R/3.

2.3.4 Distributed DMS Tablespaces in R/3

When examining the tablespaces installed with R/3, you will see 26 DMS tablespaces. There were 28 that were created, but two were removed. DB2, by default, will create three tablespaces to hold the following:

- System catalogs
This tablespace is called SYSCATSPACE in DB2 terminology. By default, it is a SMS tablespace. However, the R/3 installation creates it as a DMS tablespace. The tablespace that holds the system catalogs, SYSCATSPACE, cannot be dropped or changed after the database is created. There is only one tablespace for the system catalogs.
- Temporary space
Every DB2 database must have at least one temporary tablespace assigned to it. This tablespace, by default, is called TEMPSPACE1. There may be multiple tablespaces for temporary data. However, only one temporary tablespace will be used at any one time. This tablespace may be dropped as long as one temporary tablespace exists. The R/3 installation drops TEMPSPACE1 and creates a temporary tablespace called PSAPTEMP.
- User Data
By default, DB2 creates a tablespace for user data called USERSPACE1. This is dropped during the R/3 installation.

When tablespaces are created within a DB2 database, each tablespace is assigned a unique ID number. You can obtain information about tablespaces and containers within a DB2 R/3 system in a number of ways. We mention more

about database administration in Chapter 4, “R/3 Database Administration” on page 73. One of the ways that you can obtain information about tablespaces is with the `list tablespaces` command in DB2. The following is a portion of the output from that command:

```
Tablespaces for Current Database

Tablespace ID          = 0
Name                   = SYSCATSPACE
Type                   = Database managed space
Contents               = Any data
State                  = 0x0000
  Detailed explanation:
    Normal

Tablespace ID          = 3
Name                   = PSAPTEMP
Type                   = Database managed space
Contents               = Temporary data
State                  = 0x0000
  Detailed explanation:
    Normal

Tablespace ID          = 4
Name                   = PSAPBTABD
Type                   = Database managed space
Contents               = Any data
State                  = 0x0000
  Detailed explanation:
    Normal
....
```

The partial output from the command shows three tablespaces, SYSCATSPACE, PSAPTEMP, and PSAPBTABD. Notice the tablespace ID numbers: 0, 3, and 4. The ID numbers 1 and 2 were created during the R/3 installation (TEMPSPACE1 and USERSPACE1), but were deleted immediately.

2.3.5 R/3 Container and Tablespace Names

This section looks at the naming convention assigned to containers and tablespaces in an R/3 system. We look at this first from an R/3 point of view. In 4.2.5, “Displaying Container Information” on page 92, we tie in the naming convention given to R/3 containers within a DB2 database.

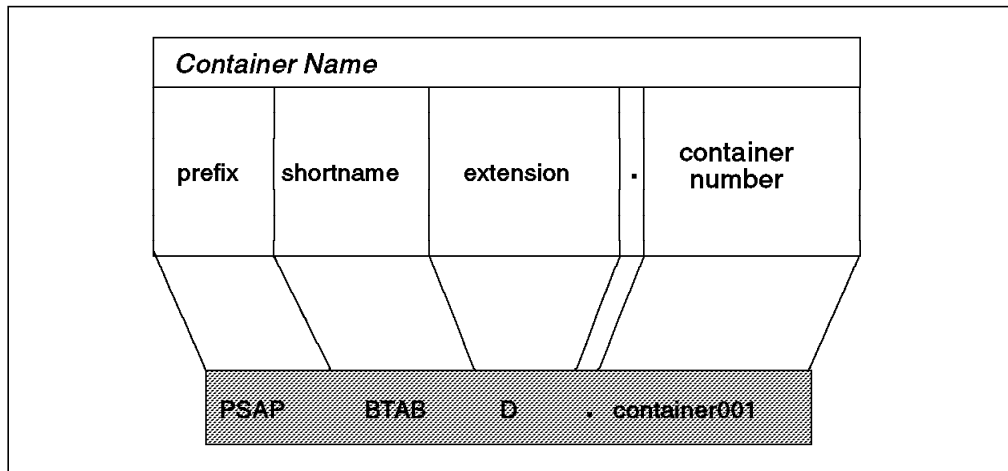


Figure 23. R/3 Container Names

Figure 23 shows the mechanism that R/3 uses for naming containers in DB2 tablespaces. The prefix for R/3 tablespaces is PSAP. The next part of the name is a descriptive part. It indicates the type of activity or information that the R/3 tablespace will have. The next part of the name indicates if the tablespace holds regular data or indexes. Finally, the last part of the container name is a number. The first container in the tablespace will be container001. The next container, if it exists, will be container002, and so on, in order. The container name in this example is PSAPBTABD.container001.

The following is a list of the DB2 R/3 Tablespaces that exist after installation.

Tablespace Type	Tablespaces	Description
DB2	SYSCATSPACE	DB2 System Catalogs
DB2	PSAPTEMP	Temporary Tablespace
SAP Basis	PSAPLOADD/I	Dynpro+Report Loads
SAP Basis	PSAPSOURCED/I	Dynpro+Report Sources
SAP Basis	PSAPDDICD/I	SAP Data Dictionary
SAP Basis	PSAPPROTD/I	Spool, Protocols, Converter
SAP Basis	PSAPES30CD/I	Repository Sources
SAP Basis	PSAPEL30CD/I	Repository Loads
SAP Application	PSAPCLUD/I	Cluster Tables
SAP Application	PSAPPOOLD/I	Pool Tables
SAP Application	PSAPSTABD	Master Data
SAP Application	PSAPBTABD/I	Transaction Data
SAP Application	PSAPDOCUD/I	Documentation, SAPscript, SAPfind
Customer Tables	PSPAUSER1D/I	Customer Tables

Table 2. DB2 R/3 Tablespaces

PSAPES30CD/I and PSAPEL30CD/I

The tablespace names, PSAPES30CD/I and PSAPEL30CD/I, are dependent on the release of R/3.

If the tablespace name has a D/I, there is a Data and an Index tablespace for each listed entry. For each tablespace, its data and indexes are, by default, in different SAPDADA directories. The most heavily used tablespaces are PSAPBTAB, PSAPCLU, and PSAPSTAB. However, PSAPBTABD/I will be the tablespace most likely to grow in size and therefore must be monitored. During development and customizing, PSAPSOURCE, PSAPLOAD, PSAPDDIC and PSAPDOCUD may increase in intensity. If you are upgrading your R/3 system, almost all tablespaces will experience more activity.

2.4 Data Placement

This section looks at of the placement or organization of data structures in an R/3 DB2 installation. We also look at the relationship between these structures, including:

- AIX storage objects to database objects
- R/3 and DB2 directories
- Default DB2 objects used by R/3
- Creating a tablespace within DB2

2.4.1 JFS and RDBMS Objects

When creating objects within a DB2 database, those objects reside in a tablespace. DB2 supports two kinds of tablespaces: SMS and DMS. However, R/3 only supports DMS tablespaces. This section looks at the relationship between the logical view and physical placement of database objects. We'll look at these objects from a user's perspective. A user's view of objects within a database is usually a two-dimensional table view:

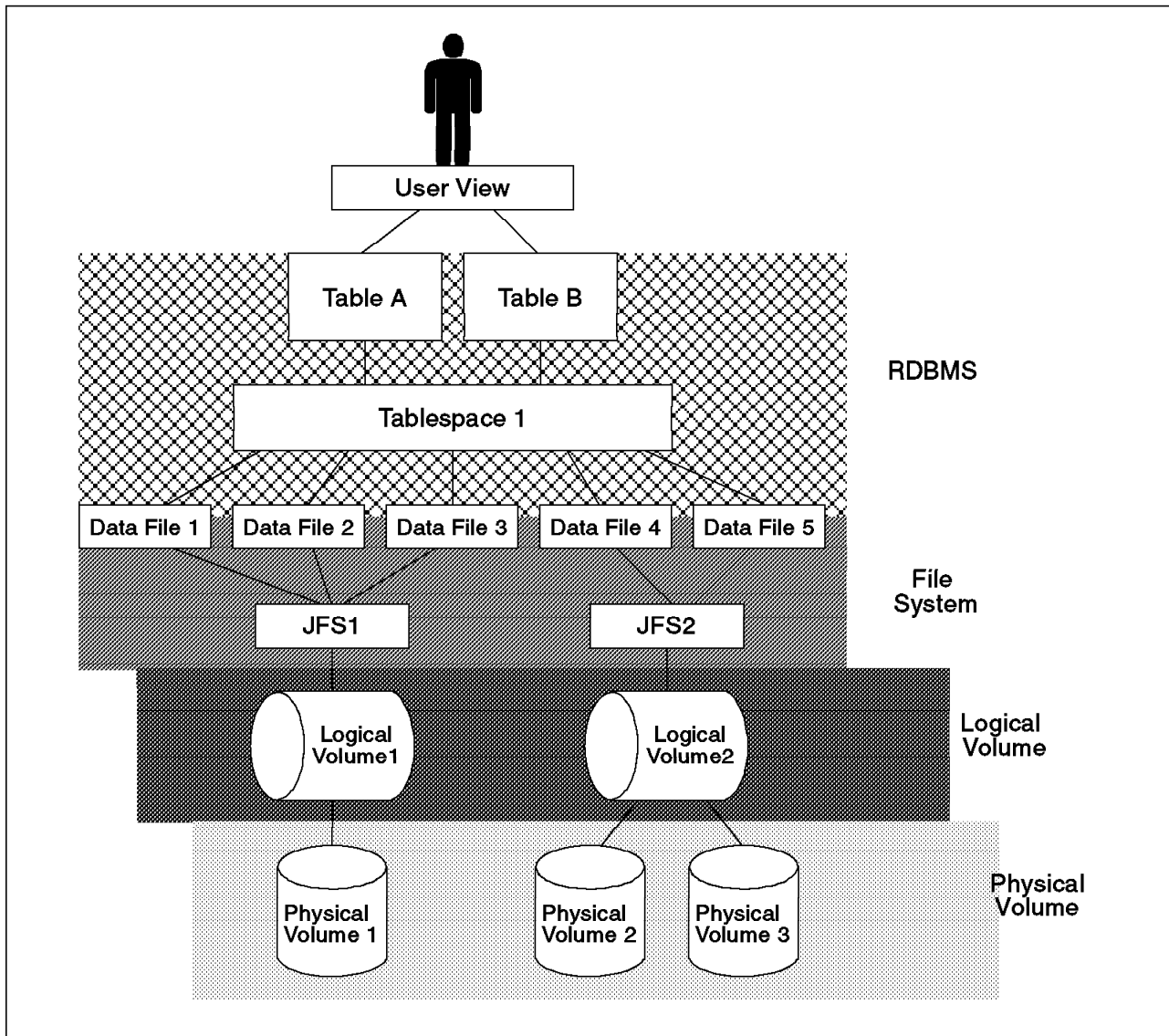


Figure 24. How DB2 Tables are Implemented Using JFS Files

Figure 24 shows the user's view at the top of the diagram. When users create a table, they create it within a tablespace. They may specify the name of the tablespace that they want to create, or the database manager will create the table in the default user tablespace.

The user's logical view of the data is represented by two tables, Table A and Table B. Tablespaces are logical concepts to the RDBMS. They provide a mechanism of separating the user's view of data from the way it is stored on disk.

Tablespace 1 provides the link between the logical views and the Logical Volumes within AIX. Within DB2, it is possible to eliminate the layer of the file system. In AIX, that is the JFS or Journaled File System. That is only possible with DMS tablespaces. However, the current release of R/3 for this documentation (R/3 Release 3.0C) does not support devices as containers in DMS tablespaces.

Points to note about Tablespace 1 are the following:

- The data for Table A may be placed in any of the five data files.
- Each of the data files may contain data for both Table A and Table B.
- Each data file may reside in a separate JFS.

The significance of these options of data placement within the RDBMS means that the only way to back up or recover tables in your database is with the facilities within DB2. In other words, you cannot use AIX commands to back up database objects.

Figure 24 on page 36 shows a one-to-one mapping between a JFS and a Logical Volume. However, it is possible to implement a one-to-many mapping between a Logical Volume and Physical Volumes. This concept is called mirroring and was discussed in 2.2.3, "AIX Facilities for Storage" on page 24.

2.4.2 R/3 and DB2 Directory Structure

This section gives an overview of the directory structure that you should find after installation of the DB2 database product and the db2<SAPSID> instance. For information on the R/3 directory structure after installation, please see the *R/3 Installation on UNIX: DB2 for AIX Database Manual* that comes with the R/3 product.

2.4.2.1 DB2 Directory Structure

When the DB2 products are installed, the following directories will be created on the R/3 Central Instance Database Server and placed under /usr/lpp/db2_02_01:

adm	System administrator executable files
adsm	ADSTAR Distributed Storage Manager files
bin	Binary executable files
cfg	Configuration files
dba	Database Director
deinstl	Files used to reject applied software
doc/%L	Postscript and online books in INF format
function	User-defined functions
include	C and FORTRAN include files
instance	Instance scripts
lib	Libraries
misc	Utilities and examples
msg/%L	Message catalogs
netls	iFOR/LS files
odbclib	Libraries and other files to support ODBC applications
readme/%L	README files
samples	Subdirectories are created under here, including c, cli, clp, and others.

Note, that the %L shown in the list above, is language dependent.

2.4.3 SAP Directories

The following is a list of SAP directories that are created with the default R/3 Central Instance Database Server installation:

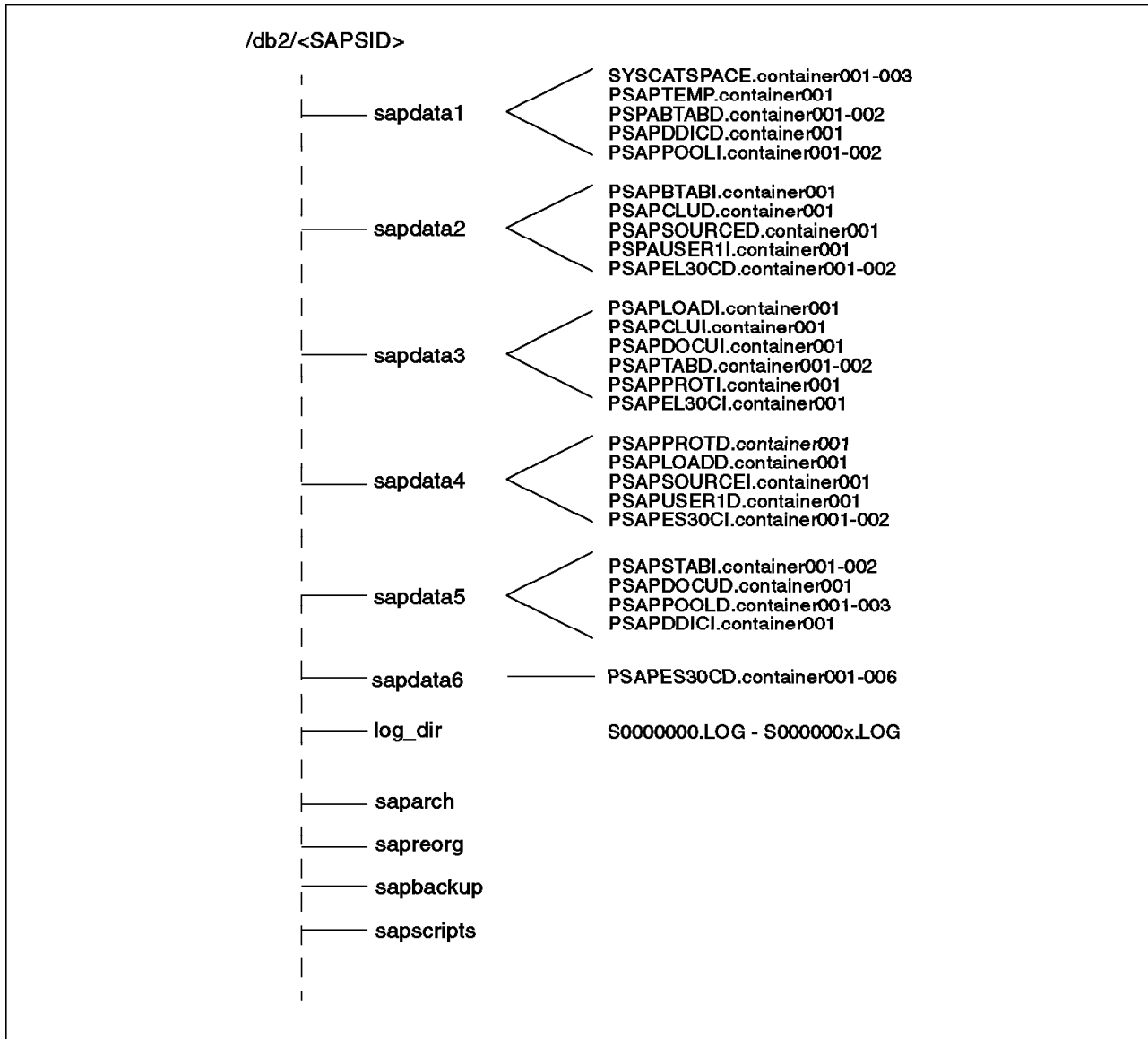


Figure 25. SAP Directories

Figure 25 shows a listing of the directories that are created as part of the R/3 installation. The sapdata1—sapadata6 directories have been created as Logical Volumes (JFS file system) within AIX. The file containers for the DB2 DMS tablespaces are also shown. Here is some further detail on the directory structure:

- db2<SAPSID>—This is the directory of the database instance from the R/3 system <SAPSID>. In our documentation, our <SAPSID> is AUS. The next section describes some of the directory and files that are placed here as a result of creating a database within the DB2 instance.
- sapdata1—6—This directory represents mount points. It includes all the DMS tablespace containers.

- log_dir—This directory represents the mount point for the DB2 log files. Logging is discussed in detail 5.1, “Logging Considerations” on page 155.
- saparch—This represents a mount point and includes database archive files and different log protocols. These protocols are explained in more detail in 5.1.3, “Terminology in DB2 and R/3” on page 161.
- sapreorg—This directory contains all files that are related to reorganization within R/3.
- sapbackup—There are specific R/3 database backup protocols that can be found here.
- sapscrips—This directory contains all the SAP scripts.

2.4.4 DB2 for AIX Directory Structure

This section looks at the default database files and objects that are created as a result of creating a database within DB2.

Figure 26 on page 40 shows the directory structure belonging to the R/3 DB2 instance owner. The directories are created either during the DB2 product installation or DB2 database creation or during the R/3 product installation. The files and directories are the following:

- batch—This is the R/3 batch directory.
- db2admin—This is the R/3 DB2admin tools installation directory.
- db2<SAPSID>—This directory and the files created within it are the result of creating the DB2 R/3 database.
- db2event—This directory is the default directory for DB2 Event Monitors.
- dbs—This is the R/3 BRARCHIVE profile directory.
- errors—This is an error log directory from R/3.
- log_archive—This is the R/3 mount point for the archived logs. It is either a separately mounted file system or a link to a mounted file system that contains the DB2 online archived log files that are used in database recovery.
- log_dir—This is the R/3 mount point for log files. It is either be a separately mounted file system or a link to a mounted file system that contains the DB2 online active log files that are used in database recovery.
- saparch—This contains the R/3 protocol files that are used to record the archiving of log files.
- sapbackup—This contains the R/3 protocol files that are used to restore archived log files.
- sapdata1—sapdata6—These are either separately mounted files systems or links to mounted file systems that contain the file containers used in the DMS tablespaces for the R/3 database.
- sapreorg—This is reserved for future use.
- sapscrips—This is an R/3 directory that includes SAP scripts.
- sqllib—This is the DB2 directory that is created when the DB2 instance is created. It contains the node and system database directory, the database manager configuration file, and executables and links to other product executables.

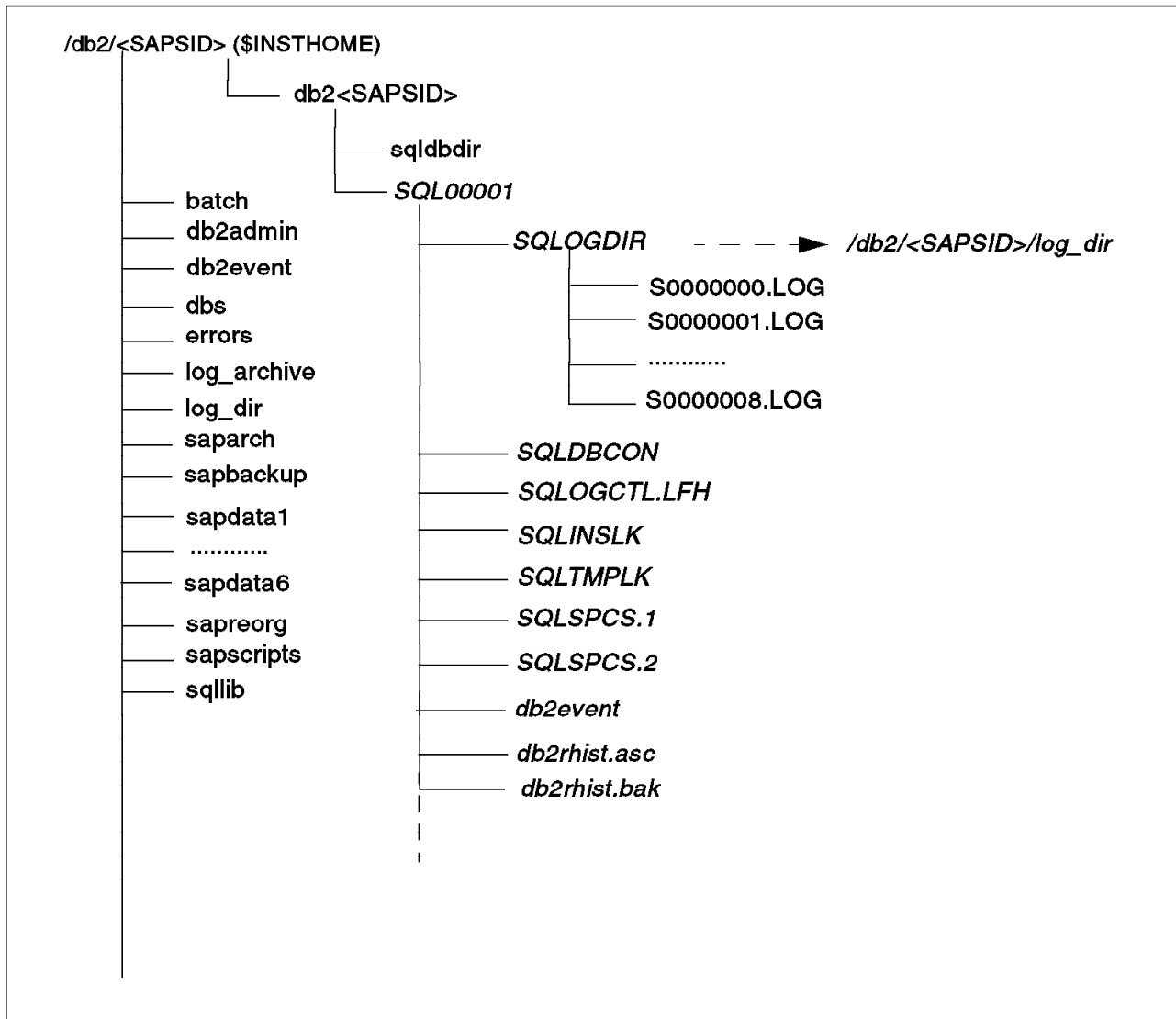


Figure 26. DB2 for AIX Instance Database Directories and Files

When DB2 creates a database, all data structures, log files, and internal structures for cataloging a database and clients are placed in a directory structure with the naming convention found in Figure 26. The exact location of the database directory may be altered at database creation, but the name of the directory is assigned by the DB2 database manager. The first (and only directory in R/3) will be SQL00001. If another database were created, the database directory would be SQL00002 and so on. This is independent of the type of tablespaces that are created in the database. Figure 26 shows the default objects associated with a database:

- **SQLDBCON** This is a file that stores the individual database parameters. When you issue the DB2 command, get db cfg for db_name, you will see the contents of this file. You cannot view the file with an editor; you must use one of the graphical interfaces or the DB2 command.
- **SQLINSLK** - This file is used to ensure that a database is only used by one instance of the database manager.
- **SQLTMPLK** - This file works along with SQLINSLK and has the same function.

- SQLLOGCTL.LFH - This file tracks and controls all the logs for a database.
- SQLOGDIR - This is a directory that contains the default log files. For more information on logging, see 5.1, “Logging Considerations” on page 155. DB2 creates three default log files with every database created. However, the default number of log files with a DB2 R/3 database is nine. They are numbered from S0000000.LOG to S0000008.LOG.
- SQLSPCS.1—This file contains the definition and current state of all the tablespaces in the database.
- SQLSPCS.2—This file is a copy of SQLSPCS.1 that is used as a backup should SQLSPCS.1 fail.
- db2event—This is the default directory that can be used for output files associated with an event monitor. Event monitors are covered in more detail in 4.5.4, “Event Monitoring” on page 149.
- db2rhist.asc—This file is used for the history of all backups and load operations.
- db2rhist.bak—This file is used with db2rhist.asc. The db2rhist.asc and db2rhist.bak files are ASCII files that are merged together and viewed as one entity by the user. The db2rhist.bak file is called a shadowed history file.

2.4.5 Creating a Tablespace in R/3

This section gives an example of how to create a tablespace. The R/3 installation creates 26 tablespaces during its installation. However, there may be an occasion where the db2<SAPSID> will have to create a tablespace. During an R/3 upgrade, you may have to perform this operation. Here are some considerations for tablespaces created in a DB2/R/3 database:

1. Only DMS tablespaces are used for DB2 for AIX under R/3. Remember that DMS means “Managed by Database.” DB2 also has SMS (Managed by System) tablespaces. SMS tablespaces should not be used in an R/3 database under any circumstances.
2. All of the containers belonging to a tablespace have the same size and, if possible, should be placed on separate disks for performance reasons.
3. R/3 tablespaces are organized in pairs:
 - PSAP<NAME>D only contains data.
 - PSAP<NAME>I contains the indexes for the tablespace that are found in PSAP<NAME>D>.

If possible, create the index tablespace on a separate disk than the tablespace containing the data. This will enhance performance.

4. If you are upgrading your release of R/3, you will need to create tablespaces with specific parameters for EXTENTSIZE and PREFETCHSIZE. (For information about EXTENTSIZE and PREFETCHSIZE, see 2.6, “Using DB2 Features in an R/3 System” on page 54.) This is because during the upgrade, the necessary number of pages is translated based on a reference that is expressed in Mbytes. Choose the following:
 - a. EXTENTSIZE 64 PREFETCHSIZE 16 for data tablespaces
 - b. EXTENTSIZE 32 PREFETCHSIZE 16 for index tablespaces

Let’s examine the syntax of the CREATE TABLESPACE statement. The user-supplied fields are shown in lowercase type.

```

CREATE {REGULAR | LONG | TEMPORY } TABLESPACE tablespace-name
MANAGED BY { SYSTEM USING ('string' [ {'string'} ... ] ) |
DATABASE USING ({FILE | DEVICE} 'string' number-of-pages
[ {, {FILE | DEVICE} 'string' number-of-pages} ... ] ) }
[EXTENTSIZE number-of-pages] [PREFETCHSIZE number-of-pages]
[OVERHEAD number-of-milliseconds] [TRANSFERRATE number-of-milliseconds]

```

Figure 27. The CREATE TABLESPACE Statement

Let's consider the following scenario. We use the <SAPSID> of AUS in our example. We want to create a tablespace, PSAPES30DD, with a size of 1650 MB during the upgrade, along with an index tablespace, PSAPES30DI, with a size of 600 MB.

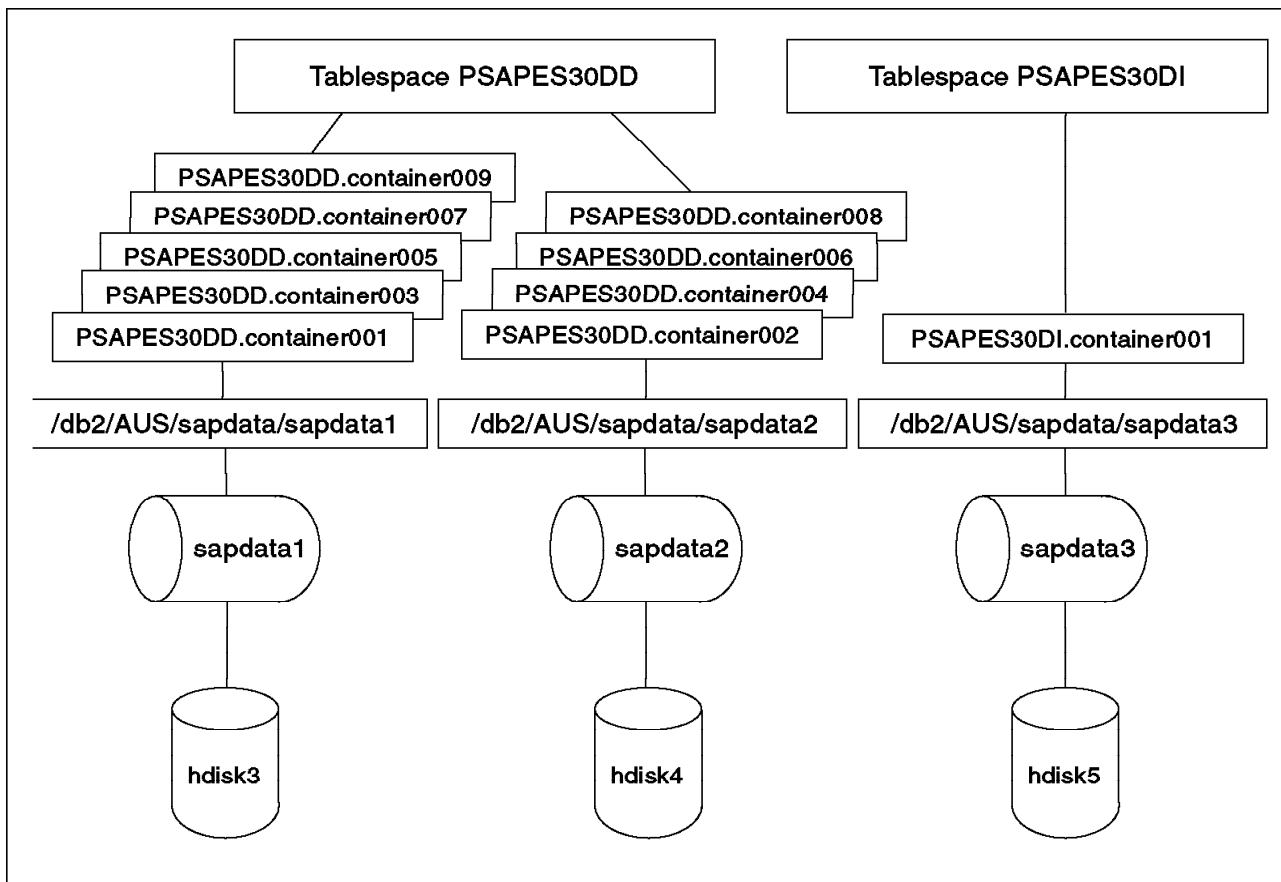


Figure 28. Create Tablespace Scenario in R/3

From Figure 28, we can see that there are three disks, hdisk3, hdisk4, and hdisk 5. The following steps are needed to create the tablespaces: (Not all the detail is shown here. Also, we are assuming that the Volume Group already exists.)

1. As the AIX system administrator, create Logical Volumes sapdata1, sapdata2, and sapdata3. The SMIT path to do this is the following:

SMIT->System Storage Management->Logical Volume Manager->Add a Logical Volume

2. Then create the file systems as the AIX system administrator. The SMIT path to do so is:

SMIT->System Storage Management->File Systems->Add/Change/Show/Delete File Systems->Journaled File Systems->Add a Journaled File System on a Previously Defined Logical Volume

3. Mount the file systems using the AIX mount command. For example:
mount /db2/AUS/sapdata/sapdata1
4. As the AIX system administrator, change the owner and permissions of the file systems so that the DB2 instance owner (db2<SAPSID>) owns the file systems. For example:
chown db2aus.sysadm /db2/AUS/sapdata/sapdata1
chmod 755 /db2/AUS/sapdata/sapdata1
5. As the DB2 system administrator, db2<SAPSID>, create a tablespace for regular data and one for index data. We will show the statements as a combination of the CREATE TABLESPACE and ALTER TABLESPACE statements. You can, however, use CREATE TABLESPACE only. The ALTER TABLESPACE statement is discussed in 4.3.1, "Adding a Container to a DMS Tablespace" on page 107. You can see that the number of pages allocated to each container is 5000. This gives us slightly more than we need. However, we must allow for the overhead of creating the tablespace and the objects within the tablespace. For more information on sizing containers in DMS tablespaces, please see 4.2.10, "Sizing Considerations for DMS Tablespaces" on page 103.

```
CREATE TABLESPACE PSAPES30DD MANAGED BY DATABASE USING
( FILE '/db2/AUS/sapdata/sapdata1/PSAPES30DD.container001' 5000 )
EXTENTSIZE 64 PREFETCHSIZE 16

CREATE TABLESPACE PSAPES30DI MANAGED BY DATABASE USING
( FILE '/db2/AUS/sapdata/sapdata3/PSPAES30DI.container001' 5200)
EXTENTSIZE 32 PREFETCHSIZE 16

ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata2/PSAPES30DD.container002' 5000 )
ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata1/PSAPES30DD.container003' 5000 )
ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata2/PSAPES30DD.container004' 5000 )
ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata1/PSAPES30DD.container005' 5000 )
ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata2/PSAPES30DD.container006' 5000 )
ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata1/PSAPES30DD.container007' 5000 )
ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata2/PSAPES30DD.container008' 5000 )
ALTER TABLESPACE PSAPES30DD ADD
( FILE '/db2/AUS/sapdata/sapdata1/PSPAES30DD.container009' 5000 )
```

As shown in Figure 28 on page 42 and the statements above, two tablespaces will be created, PSAPES30DD and PSAPES30I. There is only one container in the PSAPES30I index tablespace. Nine containers are distributed between two physical disks (two file systems) for the regular data.

Note that all the containers in the PSAPES30DD tablespaces are the same size 5000 pages.

2.5 DB2 Configuration Parameters

This section looks at some important configuration parameters within DB2. In particular, we will concentrate on those that specifically apply to R/3. It is important to understand these parameters if you are the R/3 DBA and need to monitor and/or tune your applications or database. The areas that we will discuss are the following:

- Memory used by DB2
- Database Manager Configuration Parameters
- Database Configuration Parameters
- Utilizing Features within DB2

2.5.1 Memory Used by DB2 Database Manager

One of the key purposes in monitoring database activity includes configuring various DB2 (instance) and database parameters to optimize memory utilization and increase performance.

Let's examine some of the key DBM and DB configuration parameters and how they relate to each other. Some of these parameters are used to determine the memory allocated for each DB2 instance, database, or application.

Database activity involves disk access (I/O) and memory access (CPU). Each of the DB2 configuration parameters affect either the memory or disk resources. Since disk access is much slower than memory access, the key database performance tuning criteria is to decrease the amount of disk activity. If you are able to eliminate I/O wait time, the database requests are CPU bound, and increasing performance would then require faster CPUs or multiple CPUs.

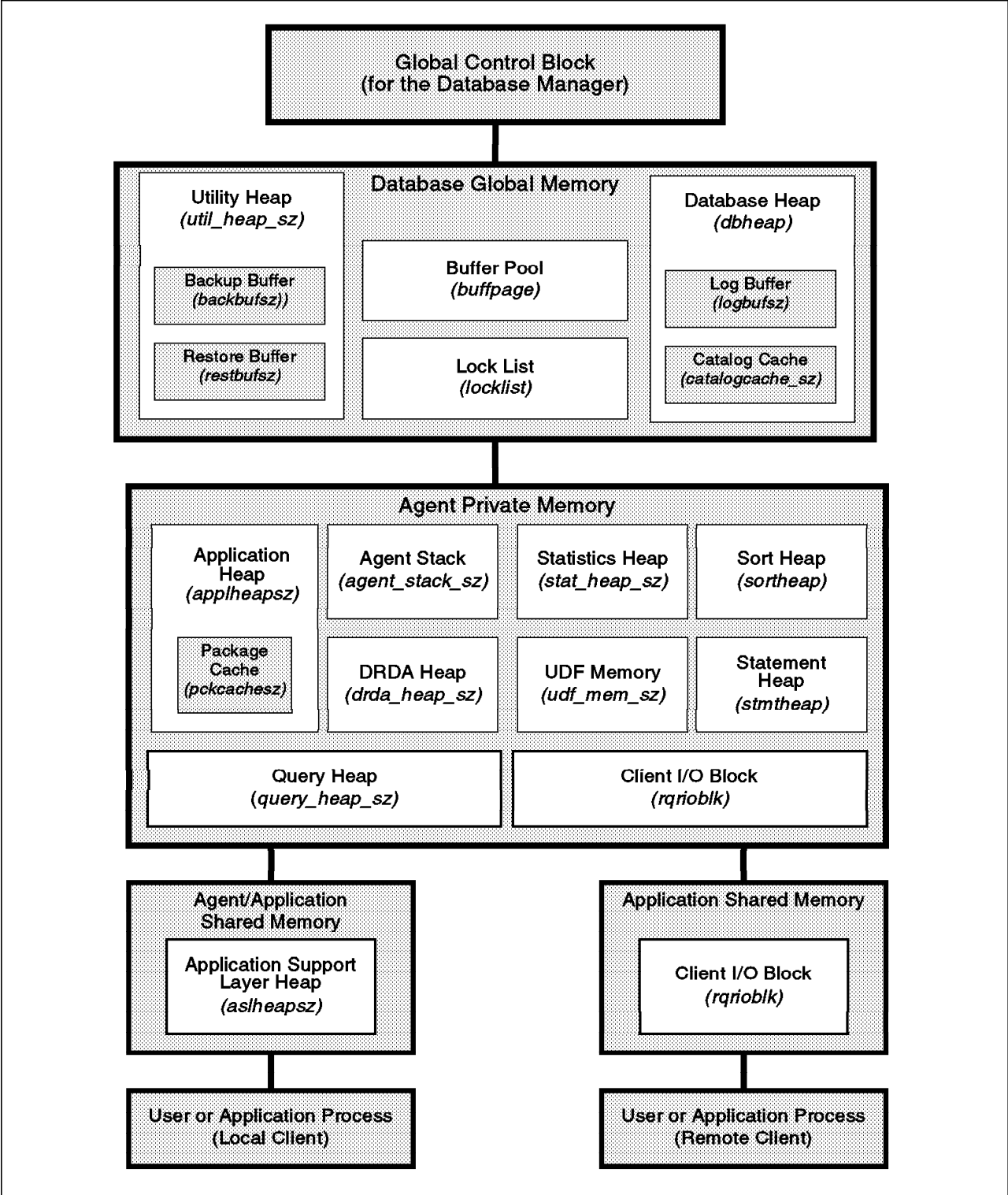


Figure 29. Memory Used by the DB2 Database Manager

Figure 29 shows the memory used by DB2 at the instance, database, and agent level.

Each DB2 application has an associated DB2 server agent. The database agent accesses the database resources on behalf of the application. Therefore, there are tuning parameters to adjust resource usage of the database agent. The

agent's private memory exists for each application connecting to the database, and the size is determined by the agent's private memory heap; the DBM parameter is called APPLHEAPSZ.

The memory area known as application shared memory is used to determine the amount of memory used to communication between the application and its DB2 agent process. Record blocking occurs within this memory area; the DBM parameter is ASLHEAPSZ.

2.5.1.1 Configuring Database Resources

The most important configuration parameter affecting database performance is the size of the buffer pool (BUFFPAGE). There is one buffer pool per database, and it will be used by all applications connected to the database. The buffer pool is a large data cache between the applications and the physical database files.

If there was no buffer pool, then all database activity would result in disk access. If the size of the buffer pool is too small, the buffer pool hit ratio will be low, and the applications will wait for disk access activity to satisfy SQL queries. If the buffer pool is too large, memory on the server will be wasted. If the buffer pool is larger than the physical memory available on the server, operating system paging (disk activity) will occur. Accessing a buffer pool that has been paged-out to disk is very inefficient.

Each page in the buffer pool is 4 KB in size.

The DB2 optimizer will utilize the buffer pool to achieve the best query performance. There is a parameter that provides the optimizer with information regarding the average number of active applications (AVG_APPLS). The AVG_APPLS parameter is used by the optimizer to determine how much of the buffer pool may be used for each application.

Another memory block shared at the database level is called the database heap (DBHEAP). Most of the database related resources are allocated out of the DBHEAP. There are many I/O caches which may be configured, including a log file cache (LOGBUFSZ) and a system catalog table cache (CATALOGCACHE_SZ).

The Log Buffer is used as a buffer for writing log records to disk. Every transaction involves writing multiple log records. To optimize disk write performance, the writes are buffered in memory and periodically flushed to disk.

The Catalog Cache is used to store the system catalog tables in memory. As an SQL statement is compiled or referenced, the database object information needs to be verified. If the information is in memory, then there is no need to perform disk activity to access the data.

Record blocking is a client/server caching technique used to send a group of records across the network to the client instead of a single record at a time. The decrease in network traffic increases application performance and allows for better network throughput.

The records are blocked by DB2 according to the cursor type and bind parameter. If the optimizer decides to return the query output in blocks, then the amount of data in each block is determined by the ASLHEAPSZ parameter.

If the DB2 client is configured with a different value for the RQRIOBLK parameter, the RQRIOBLK parameter is used as the record blocking size.

The application heap (APPLHEAPSZ) contains a number of memory blocks which are used by DB2 to handle requests for each application. The package cache (PCKCACHESZ) is allocated from this heap and is used to reduce the need to reload access plans (sections) of a package. This caching can improve performance when the same section is used multiple times within a program.

The access plans are cached for static and dynamic SQL statements in the package cache.

The sort heap (SORTHEAP) is allocated from the agent private memory and determines the number of private memory pages that may be used for each sort. This parameter is used by the optimizer to determine if the sorting can be performed in memory or on disk. DB2 will always attempt to perform the sort in memory. The SHEAPTHRES parameter is used to control the number of sort heaps allocated for a DB2 server.

Allocating half of the physical memory to the buffer pool is usually a good starting point when adjusting its size. This assumes a dedicated DB2 database server workstation and a single database active at any given time. For example, if the database server had 40 MB of RAM, then a buffer pool size of 20 MB would usually be effective.

2.5.2 Database Manager Configuration Parameters

This section looks at the configuration parameters at the instance or database manager level. We do not discuss all of them here. For more detail, see the *DB2 Administration Guide*. To view the current settings of the database manager configuration, issue the following command:

```
db2 get dbm cfg
```

You should see output similar to the following: (This is only a partial output.)

Database Manager Configuration

Node type = Database Server with local and remote clients

```
Database manager configuration release level          = 0x0600

CPU speed (millisec/instruction)                    (CPUSPEED) = 7.944251e-06
Max number of concurrently active databases          (NUMDB) = 3
Transaction processor monitor name                  (TP_MON_NAME) =

Diagnostic error capture level                      (DIAGLEVEL) = 3
Diagnostic data directory path                      (DIAGPATH) = /db2/AUS/sql1lib/db2dump

Default database monitor switches
  Buffer pool                                       (DFT_MON_BUFPOOL) = ON
  Lock                                             (DFT_MON_LOCK) = ON
  Sort                                             (DFT_MON_SORT) = ON
  Statement                                        (DFT_MON_STMT) = ON
  Table                                           (DFT_MON_TABLE) = ON
  Unit of work                                    (DFT_MON_UOW) = ON

Database monitor SQL statement size (bytes) (SQLSTMTSZ) = 4096

SYSADM group name                                 (SYSADM_GROUP) = SYSADM
SYSCTRL group name                               (SYSCTRL_GROUP) = SYSCTRL
SYSMAINT group name                              (SYSMAINT_GROUP) =

Database manager authentication                   (AUTHENTICATION) = SERVER

Default database path                             (DFTDBPATH) = /db2/AUS

Database monitor heap size (4KB)                 (MON_HEAP_SZ) = 128
UDF shared memory set size (4KB)                 (UDF_MEM_SZ) = 128

Backup buffer default size (4KB)                 (BACKBUFSZ) = 1024
Restore buffer default size (4KB)                 (RESTBUFSZ) = 1024

Sort heap threshold (4KB)                        (SHEAPTHRES) = 4096

Directory cache support                           (DIR_CACHE) = YES

Application support layer heap size (4KB)         (ASLHEAPSZ) = 96
Max requester I/O block size (bytes)              (RQRIOBLK) = 65000
Query heap size (4KB)                            (QUERY_HEAP_SZ) = 1000
DRDA services heap size (4KB)                    (DRDA_HEAP_SZ) = 128
...
```

Only db2<SAPSID> can change any of the parameters in the database manager configuration. To do so, you can use the following DB2 command:

```

UPDATE DATABASE MANAGER CONFIGURATION USING {config-keyword value ...}
config-keyword:  ADSM_NODENAME, ADSM_OWNER, ADSM_PASSWORD,
AGENTPRI, AGENT_STACK_SZ, ASLHEAPSZ, AUTHENTICATION, BACKBUFSZ,
CPUSPEED, DFT_ACCOUNT_STR, DFT_CLIENT_COMM, DFTDBPATH, DFT_MON_BUFPOOL,
DFT_MON_LOCK, DFT_MON_SORT, DFT_MON_STMT, DFT_MON_TABLE, DFT_MON_UOW,
DIAGLEVEL, DIAGPATH, DIR_CACHE, DIR_OBJ_NAME, DIR_PATH_NAME, DIR_TYPE,
DOS_RQRIOBLK, DRDA_HEAP_SZ, FILESERVER, INDEXREC, IPX_SOCKET, KEEPDAI,
MAXAGENTS, MAXCAGENTS, MAXDARI, MAX_IDLEAGENTS, MAXTOTFILOP,
MIN_PRIV_MEM, MON_HEAP_SZ, NNAME, NUMDB, OBJECTNAME, PIPENAME,
PRIV_MEM_THRESH, QUERY_HEAP_SZ, RESTBUFSZ, RESYNC_INTERVAL, ROUTE_OBJ_NAME,
RQRIOBLK, SHEAPTHRES, SPM_LOG_SIZE, SPM_NAME, SPM_RESYNC_AGENT_LIMIT,
SQLSTMTSZ, SVCENAME, SYSCTRL_GROUP, SYSMANT_GROUP,
TM_DATABASE, TP_MON_NAME, TPNAME, UDF_MEM_SZ, UDF_PROC_USER

```

Let's change the size of the amount of memory used to perform record blocking. The memory area used for record blocking is known as the application support layer heap (ASLHEAPSZ). The following command sets the record blocking to be 800 KB in size (the units are 4 KB):

```
db2 update database manager configuration using aslheapsz 200
```

Therefore, when the instance is restarted, records will be sent across the network from the DB2 server to the application in 200-KB blocks (likely more than a single row). If the average row length was 1 KB, then 200 records would be returned in a single block of data (assuming more than 200 records are in the final result table).

Record blocking occurs for remote and local DB2 client applications.

Any changes to the database manager configuration (instance) will not take effect until all of the applications have disconnected. The instance must then be stopped (db2stop command) and started using the db2start command. This can also be done by issuing the command through R/3. See Chapter 3, "Starting and Stopping SAP R/3" on page 59.

There are some database manager configuration parameters that are set during the installation or migration of the R/3 system. For example, all of the monitor recording switches are turned on. The SYSADM and SYSCTRL groups are determined at installation. For a complete listing of all the parameters, both database manager and database, that are set during installation, consult the *BC SAP Database Administration Guide: DB2 for AIX*.

The following is a brief description of some of the database manager configuration parameters that may affect performance:

- **MAXAGENTS**—This parameter indicates the maximum number of database manager agents available at any given time to accept application server requests. The value should be the sum of the values for MAXAPPLS that is set in the database configuration. Since there is only one database, the value of MAXAGENTS should always equal the value of MAXAPPLS.
- **MAXCAGENTS**—This parameter specifies the maximum number of database manager agents that can be concurrently executing a database manager transaction. This parameter does not limit the number of applications that can have connections to a database. It only limits the number of database manager agents that can be processed concurrently by the database manager at any one time, thereby limiting the usage of system resources during times of peak processing.

- **MAXDARI**—This parameter indicates the maximum number of DARI (Database Application Remote Interface) processes that may reside at the database server. In R/3, DARIs (stored procedures) are function calls from the database client, which is a defined API between the server procedure at the Database Server and the database client on the Application Server to get all the required information to the SAP Application Server. In R/3, MAXDARI is set to the same value as MAXAPPLS.
- **KEEPDARI**—This parameter indicates whether or not a DARI process is kept after a DARI call is complete. Keeping the DARI process will improve DARI performance, but will also result in additional system resources being consumed by the database manager for each DARI process that is activated. This parameter is set to **yes** during the R/3 installation.
- **BACKBUFSZ**—This parameter specifies the size of the buffer used when backing up the database or tablespace if the buffer size is not explicitly specified when calling the backup utility. When backing up a database/tablespace, the data is first copied to an internal buffer. Data is then written from this buffer to the backup media when the buffer is full.
- **RESTBUFSZ**—This parameter specifies the size of the buffer used when restoring the database if the buffer size is not explicitly specified when calling the restore utility. When restoring a database or tablespace, the data is first copied from the backup media to the internal buffer. Data is then written from this buffer to the target database when the buffer is full.
- **RQRIOBLK**—This parameter specifies the size of the communication buffer between remote applications and their database agents on the database server. This parameter is set to 32767 bytes, the default value.
- **ASLHEAPSZ**—The application support layer heap represents a communication buffer between the local application and its associated agent. This buffer is allocated as shared memory by each database manager agent that is started. The default value is 15 4-KB pages. However, it is set to 96 4-KB pages during the R/3 installation.

If you change the DBM (instance) configuration parameters, the new values will not be effective until the instance has been stopped and restarted.

2.5.3 Database Configuration Parameters

This section looks at the configuration parameters at the instance or database manager level. We do not discuss all of them here. For more detail, see the *DB2 Administration Guide*. To view the current settings of the database manager configuration, issue the following command:

```
db2 get db cfg for aus
```

You must supply the database name. Remember that in R/3 only one database is allowed. You should see output similar to the following: (This is only a partial output.)

Database Configuration for Database aus

```

Database configuration release level      = 0x0600
Database release level                   = 0x0600

Database territory                        = en_US
Database code page                       = 819
Database code set                        = ISO8859-1
Database country code                    = 1

Directory object name                    (DIR_OBJ_NAME) =

Backup pending                           = NO
Database is consistent                    = NO
Rollforward pending                      = NO

Log retain for recovery status           = YES
User exit for logging status             = YES

Number of frequent values retained       (NUM_FREQVALUES) = 10
Number of quantiles retained             (NUM_QUANTILES) = 20

Database heap (4KB)                      (DBHEAP) = 4000
Catalog cache size (4KB)                 (CATALOGCACHE_SZ) = 64
Log buffer size (4KB)                   (LOGBUFSZ) = 64
Utilities heap size (4KB)                (UTIL_HEAP_SZ) = 10000
Buffer pool size (4KB)                   (BUFFPAGE) = 10000
Max storage for lock lists (4KB)         (LOCKLIST) = 3200

Sort list heap (4KB)                     (SORTHEAP) = 512
SQL statement heap (4KB)                 (STMTHEAP) = 4096
Default application heap (4KB)           (APPLHEAPSZ) = 1500
Package cache size (4KB)                (PCKCACHESZ) = 256
Statistics heap size (4KB)              (STAT_HEAP_SZ) = 4384

Interval for checking deadlock (ms)      (DLCHKTIME) = 300000
Percent. of lock lists per application   (MAXLOCKS) = 100
Lock timeout (sec)                      (LOCKTIMEOUT) = -1

Changed pages threshold                  (CHNGPGS_THRESH) = 60
Number of asynchronous page cleaners     (NUM_IOCLEANERS) = 3
Number of I/O servers                   (NUM_IOSERVERS) = 3
Index sort flag                          (INDEXSORT) = YES
Sequential detect flag                   (SEQDETECT) = YES
Default prefetch size (4KB)             (DFT_PREFETCH_SZ) = 32

Default number of containers              = 1
Default tablespace extentsize (4KB)     (DFT_EXTENT_SZ) = 8

Max number of active applications        (MAXAPPLS) = 60
Average number of active applications    (AVG_APPLS) = 1
Max DB files open per application        (MAXFILOP) = 1950

```

Any modification to the database configuration file is not effective until all applications using the database are disconnected. The subsequent database connection uses the new database configuration parameters. You must be db2<SAPSID> to make any changes to the database configuration. To do so, you can use the following command:

```

UPDATE DATABASE CONFIGURATION FOR database-alias
USING {config-keyword value ...}
  config-keyword:
APPLHEAPSZ, AUTORESTART, AVG_APPLS, BUFFPAGE, CHNGPGS_THRESH, CATALOGCACHE_SZ,
COPYPROTECT, DBHEAP, DFT_EXTENT_SZ, DFT_LOADREC_SES, DFT_PREFETCH_SZ,
DIR_OBJ_NAME, DLCHKTIME, INDEXREC, INDEXSORT, LOCKLIST, LOCKTIMEOUT, LOGBUFSZ,
LOGFILSIZ, LOGPRIMARY, LOGRETAIN, LOGSECOND, MAXAPPLS, MAXFILOP, MAXLOCKS,
MINCOMMIT, NEWLOGPATH, NUM_FREQVALUES, NUM_IOCLEANERS, NUM_IOSERVERS,
NUM_QUANTILES, PCKCACHESZ, REC_HIS_RETENTN, SEQDETECT, SOFTMAX, SORTHEAP,
STAT_HEAP_SZ, STMTHEAP, USEREXIT, UTIL_HEAP_SZ

```

For example, suppose you wanted to change from the default of circular logging to log retain. You would issue the following command:

```
db2 update db cfg for aus using logretain on
```

You will have to supply the database name in the parameter of the command. Also, you must terminate all connections from the database before the next connection will take effect. This is a special case, however. You must also perform a backup of the database when enabling LOG RETAIN or USEREXIT. For more information on backup and recovery, please see Chapter 5, “Backup and Recovery” on page 155.

We briefly discuss some of the database parameters. For more detail, consult the *DB2 Administration Guide*. The *BC SAP Database Administration Guide: DB2 for AIX* tells you those database parameters that have been changed or modified from defaults as part of the R/3 installation.

The following is a partial list of parameters. They are some of the parameters that can have a high impact on performance:

- **BUFFPAGE**—This configuration parameter is the most important parameter affecting database performance. There is one buffer pool per database, and as shown in Figure 29 on page 45, it resides in global memory that is available to all applications using the database. The buffer pool is the area of storage into which database pages are read and in which they are changed. Buffer pool pages are written to disk either by the database manager agents (synchronous writes) or by I/O Cleaners (asynchronous writes).

Buffer pool pages are read from disk synchronously (by database manager agents) or are prefetched asynchronously by I/O Servers. Activities of the I/O Cleaners and I/O Servers should be monitored as they correlate with the buffer pool data. (You can use the graphical interface in R/3 to display this information. See Figure 81 on page 139.)

The buffer pool hit ratio should lie at about 90 percent for the newly started R/3 system. If this is not the case, you should consider increasing the size of the buffer pool.

- **MAX_APPLS**—This parameter specifies the maximum number of concurrent applications that can be connected (both local and remote) to a database. Each application that attaches to a database causes some private memory to be allocated; so a large number of concurrent applications will use more memory.

- **AVG_APPLS**—This parameter is used by the DB2 SQL optimizer to help estimate how much buffer pool will be available at run time for the access plan chosen. Increasing the parameter can influence the optimizer to choose an access plan for queries that will be more conservative in its buffer pool usage.
- **DBHEAP**—There is one database heap per database, and the database manager uses it on behalf of all applications connected to the database. It contains control block information for tables, indexes, and tablespaces, as well as space for the log buffer (LOGBUFSZ). The size of the heap will be dependent on the number of control blocks stored in the heap at a given time. The control block information is kept in the heap until all applications disconnect from the database. The default value in DB2 for AIX is 1200 4-KB pages. During the R/3 installation, it is set to 4000 4-KB pages.
- **LOGBUFSZ**—This parameter allows you to specify the amount of the database heap (defined by the DBHEAP parameter) to use as a buffer for log records before writing these records to disk. The default size in DB2 for AIX is 8 4-KB pages. It is set to 64 4-KB pages during installation.
- **UTIL_HEAP_SZ**—This parameter indicates the maximum amount of memory that can be used simultaneously by the BACKUP, RESTORE and LOAD and load recovery utilities. If the parameter is set too low, you may not be able to concurrently run utilities. When setting this parameter, you should note that the utility heap (UTIL_HEAP_SZ) is allocated from the same shared memory segment as the buffer pool (BUFFPAGE), database heap (DBHEAP) and lock list (LOCKLIST). The default value is 5000 4-KB pages, reset at installation to 10000 4-KB pages.
- **LOCKLIST**—This parameter indicates the amount of storage that is allocated to the lock list. There is one lock list for the database, and it contains the locks held by all applications concurrently connected to the database. You may consider increasing the size of the LOCKLIST database parameter if the lock list utilization is high or if lock escalations occur. A lock is escalated when the total number of locks held by an application reaches the maximum amount of lock list space available to the application.

The amount of lock list space available is determined by the LOCKLIST and MAXLOCKS database-configuration parameters. When an application reaches the maximum number of locks allowed and when there are no more locks to escalate, the application uses space in the lock list allocated to other applications. When the entire lock list is full, an error occurs. There is a graphical interface in R/3 that captures information about locks being held and lock escalation. For more information, please see 4.5.4, “Event Monitoring” on page 149.

- **MAXLOCKS**—This parameter defines a percentage of the lock list held by an application that must be filled before the database manager performs escalation.
- **NUM_IOCLEANERS**—This parameter specifies the number of asynchronous page cleaners for a database. These page cleaners write changed pages from the buffer pool to disk before the space in the buffer pool is required by a database agent. For more information, see 2.6.3, “I/O Cleaners” on page 56.
- **CHNGPGS_THRESH**—This parameter specifies the percentage of changed pages at which the asynchronous page cleaner will be started.

- **NUM_IOSERVERS**—This parameter specifies the number of I/O servers for a database. I/O servers are used on behalf of database agents to perform prefetch I/O and asynchronous I/O by utilities such as BACKUP and RESTORE. Please see 2.6.2, “Prefetching in DB2” on page 55 for more detail.
- **SORTHEAP**—This parameter defines the number of private memory pages to be used for each sort. During the R/3 installation, this value is doubled from the default of 256 4-KB pages. The R/3 graphical interface can help you monitor many database parameters (Figure 81 on page 139). You can get an indication if the total number of sorts ran out of sort heap and required disk space for temporary storage. When a sort overflows, data is written to disk to the freespace in the sort heap. This disk I/O results in reduced performance for the sort.
- **APPLHEAPSZ**—This parameter defines the number of private memory pages available to be used by the database manager on behalf of a specific agent process.
- **PCKCACHESZ**—This parameter controls the amount of application heap memory to be used for caching a package’s static and dynamic SQL statements.
- **MINCOMMIT**—This is the number of commits to group. This value indicates that grouping of commits issued by multiple applications is to be attempted, if the value is set greater than 1. The range is 1—10. The default value is 1. If the number of applications connected to the database is greater than or equal to this parameter, commits will not cause immediate physical writes of the log. The writes will be delayed 1 (one) second or until the number of commits requested by all applications equals this value. Grouping commits can enhance the performance of databases servicing multiple connects with high change activity, such as the R/3 database. It is recommended to set this value to 3 for your R/3 database.

2.6 Using DB2 Features in an R/3 System

This section discusses how some of the features in DB2 that may require changing parameter values can affect performance.

2.6.1 Extent Size

The extent size is a database parameter. By default, the extent size that is used when a tablespace is created is 32 4 KB pages. You can change this default value at the database level by using the UPDATE DATABASE command, as long as the tablespace you wish to change the setting for has not been accessed. Once the extent size is set for the tablespace, it cannot be changed. If you want to specify a different value for the extent size of a tablespace you are about to create, you need to use the EXTENTSIZE option of the CREATE TABLESPACE statement. The range of EXTENTSIZE is from 2 to 256 4-KB pages.

A smaller extent size than the default can be selected if your tablespace will not store large table objects. A smaller extent size will save space, but it will require more frequent allocation of extents. An extent size of 8 or 16 4-KB pages is usually selected for tablespaces storing user data. A larger extent size is recommended for long tablespaces. This will help to avoid overhead derived from allocating a large number of extents each time a new entry is made.

However, if LOBs are smaller (less than 100 KB), the default size of 32 for the extent size will suffice.

The extent size has a direct impact on the amount of prefetching that will take place.

2.6.2 Prefetching in DB2

Query processing is divided into parallel activities: CPU processing and I/O. The CPU processing programs ask the I/O servers for data, thus sending the so-called prefetch requests. A prefetch request contains a description of the I/O needed to satisfy the anticipated data needs. I/O prefetching increases the performance of queries that retrieve a large amount of data.

Prefetching allows CPU and I/O overlap, thus reducing the query execution time. Prefetching is expressed by the CREATE TABLESPACE parameter, PREFETCHSIZE. This parameter determines the amount of data that is “read-ahead”.

Figure 30 shows two types of reads, a synchronous read and two prefetch reads.

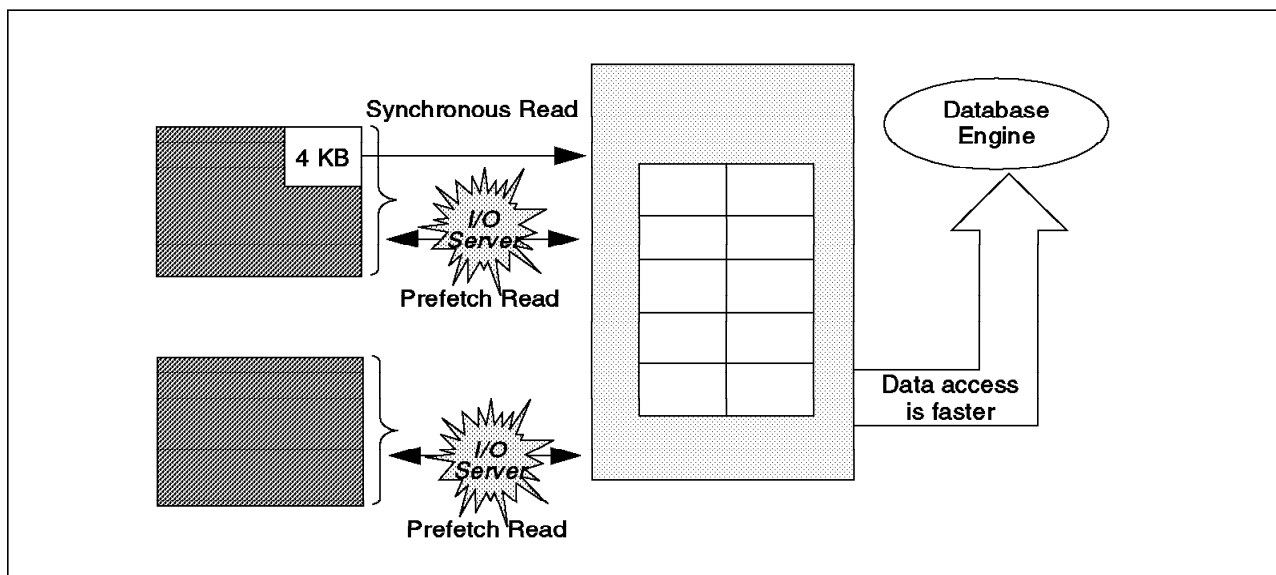


Figure 30. DB2 Prefetching Data Using I/O Servers

A big-block read is one in which more than 4-KB of data is read in a single I/O operation. The typical I/O unit will be the same size as the value defined in the EXTENTSIZE parameter of the tablespace.

Prefetching will help to enable parallel I/O if the prefetch size is a multiple of the extent size and the extents being prefetched are on separate disks. The prefetch size is determined when the tablespace is created. If no value is specified when the tablespace is created, the prefetch size will be set to the default value found in the database parameter, DFT_PREFETCH_SZ. The default size is 32 4-KB pages. This is the amount of data that is read-ahead at any one time. If the tablespace is created using multiple container definitions, especially if those definitions are found on multiple disk drives, this will assist in performance. The optimal configuration would be to have multiple container definitions and the PREFETCHSIZE set to a multiple of the EXTENTSIZE.

There are two categories of prefetch:

- Sequential prefetch is the mechanism that reads consecutive pages into the buffer pool before the pages are needed by the application.
- List prefetch (or list sequential prefetch) is a way to access data pages efficiently, even when the data pages needed are not consecutive.

There are situations when it is not immediately obvious whether sequential prefetch will improve performance. In this case, the database manager can monitor I/O and if sequential page reading is occurring, the database manager can activate prefetching. Prefetching here can be activated and deactivated by the database manager as necessary. This type of sequential prefetch is called sequential detection. To activate it, set the SEQDETECT database parameter to YES. The default value for SEQDETECT is YES.

During program execution, an SQL statement that requires a large amount of data is executed. The database manager identifies that it would be beneficial to use I/O prefetch. At db2start, the database engine starts separate threads of control. These are called I/O Servers. In AIX, the I/O Servers are separate processes, not threads. The number of threads/processes to be created is specified in the database configuration parameter, NUM_IOSERVERS. By using I/O prefetch, CPU and I/O overlap will occur, thus reducing query execution time.

The database engine informs the I/O Server of the application's anticipated data needs. This process is called the prefetch request. I/O is fetched in units larger than a single 4KB page. These units are called big-blocks.

For any prefetching to occur, the buffer pool also must be set to 1100 4-KB pages or higher. The default value for the buffer pool in AIX is 1000 4-KB pages.

2.6.3 I/O Cleaners

The objective of having I/O Cleaners is to help ensure that there are enough "clean" pages in the buffer pool when a transaction is started. In this way, the transaction will not have to wait until previously used pages are written out to disk. DB2 moves pages of data from disk to the buffer pool in order to read and/or modify data. If a page has been modified, it should be written back to disk. With previous versions of DB2, this was done when a database agent needed some pages in the buffer pool but discovered that the slots contained changed pages. So the agent had to wait for an I/O operation to continue. DB2 uses page cleaner agents to handle modified pages.

The purpose of the page cleaners is to write out most changed pages to disk so that the regular database agents can find empty slots and do not have to write pages out to disk. This means that the agents will not have to wait for additional I/O, and the user transaction should execute faster. The page cleaners are processes external to the database.

The number of asynchronous page cleaners for a database is specified by the database configuration parameter, NUM_IOCLEANERS. In a query-only database, this parameter should be set to 0, as no pages will need to be written to disk. In a transaction environment, the parameter should be set between 1 and the number of physical disk drives used by the database.

The other database parameter we mention here is CHNGPGS_TRHESH. This is the changed pages threshold. This threshold determines at which percentage of

changed pages the asynchronous page cleaners will be started. Page cleaners write all modified pages to disk and become inactive until the threshold is again exceeded. This parameter is defined at the database level, and its default value is 60 percent.

Chapter 3. Starting and Stopping SAP R/3

This chapter looks at what is involved in starting and stopping a DB2 R/3 Central Instance. We discuss the processes that are created for the DB2 database and instance and for the R/3 Central Instance. We will discuss how you start and stop the R/3 system. There are two programs within R/3 that handle these functions: `/home/<SAPSID>adm/startsap` and `/home/<SAPSID>adm/stopsap`. These programs are responsible for other programs starting, such as the DB2 database manager.

We will also look at some of the environment-specific items that a SAP DBA should be aware of. You can find out more information by reading the SAP Database Administration Guide: *DB2 for AIX* and the *SAP Installation Guide*.

3.1 Starting R/3

To start R/3, log in to AIX as the SAP system administrator, `<SAPSID>adm`. The database must be running before R/3 can be started. To start the database and R/3, enter the following command on the AIX command line:

```
startsap all
```

The default parameter for the `startsap` script is `all`. You may also enter:

```
startsap
```

To start only the database, enter the following command on the AIX command line:

```
startsap db
```

If the database manager is already running, you may start only R/3. If you run `startsap` or `startsap all`, you will receive a message that the database is already running. You may also use the `r3` option to start only R/3. To start R/3 after the database is already running, enter the following command at the AIX command line:

```
startsap r3
```

If R/3 successfully started, you will receive messages indicating that the system is started and that log files are being written into. The log file is located in the `<SAPSID>adm`'s home directory. The log file will be written in `/home/<SAPSID>adm/startsap_<hostname>_SAPSYSTEMNUMBER.log`. For example, in our installation the complete path for the log file was `/home/ausadm/startsap_f01n07e.itsc.austin.ibm.com_00.log`.

When `startsap` executes, it does the following:

- Starts the R/3 instance
- Performs SAP license check
- Calls `/usr/sap/<SAPSID>/SYS/exe/run/startdb`. This script initializes the DB2 environment and calls the `db2start` command to start the database manager. It also checks the DB2 license, updates the database monitor switches, checks that the database is consistent so that a connection is possible and makes sure that the database is enabled for log retain. (Log retain is discussed in 5.1, "Logging Considerations" on page 155.)

```

ausadm> startsap

Starting SAP-Collector Daemon
-----
Collector already running ... don't launch
saposcol already running

Starting SAP R/3 AUS Database
-----
Startup-Log is written to /home/ausadm/startdb.log
Database started

Starting SAP R/3 Instance
-----
Startup-Log is written to /home/ausadm/
startsap_f01n07e.itsc.austin.ibm.com_00.log
Instance on host f01n07e started

```

The R/3 processes that are started with startsap are described in 3.3, “DB2 for AIX Processes” on page 61.

3.2 Stopping R/3

R/3 should be shut down before the database is shut down. To stop R/3 and/or the database, log in to the SAP system administrator user ID, <SAPSID>adm. To shut down only R/3 and leave the database running, enter the following command at the AIX command line:

```
stopsap r3
```

To shut down R/3 and then shut down the database, enter the following command at the command line:

```
stopsap all
```

The default parameter of the stopsap script is all. You may also enter:

```
stopsap
```

When the R/3 system is shut down, various functions are performed. Let's look at the messages that you might see upon a successful shutdown of R/3.

```

ausadm> stopsap
Stopping the SAP R/3 AUS Processes
-----
Shutdown-Log is written to
/home/ausadm/stopsap_f01n07e.itsc.austin.ibm.com_00.log
Instance on host f01n07e stopped
Waiting for cleanup of resources.....
Stopping SAP R/3 AUS Database
-----
Shutdown-Log is written to /home/ausadm/stopdb.log
Database stopped

```

As with the startsap program, there is a log file that is written to with the stopsap program. The directory path is /home/<SAPSID>adm/stopsap_<hostname>_<SAPSYSTEMNUMBER>.log.

For our installation, the file was placed in /home/ausadm/stopsap_f01n07e.itsc.austin.ibm.com_00.log, where f01n07e.itsc.austin.ibm.com was the hostname of our R/3 system. As part of the stopsap program, the R/3 instance is shut down, the database manager instance is stopped (db2stop), the IPC (Inter-Process Communication) resources are released, and a database log file switch is performed before shutting down the database. If applications are still connected to the database, stopsap will fail. You can check the status of the DB2 processes with the following command:

```
db2 list applications
```

You can then make a determination if you want to continue with the shutdown or not. For more information on how to monitor DB2 applications, see 4.2.6, “Monitoring Active Applications” on page 99.

3.3 DB2 for AIX Processes

The process model for DB2 is known as an “n-n process model” (pronounced n-to-n). The main feature of this architecture is its ability to ensure database integrity. It isolates all database applications from critical database resources. These resources could include database control blocks and critical database files.

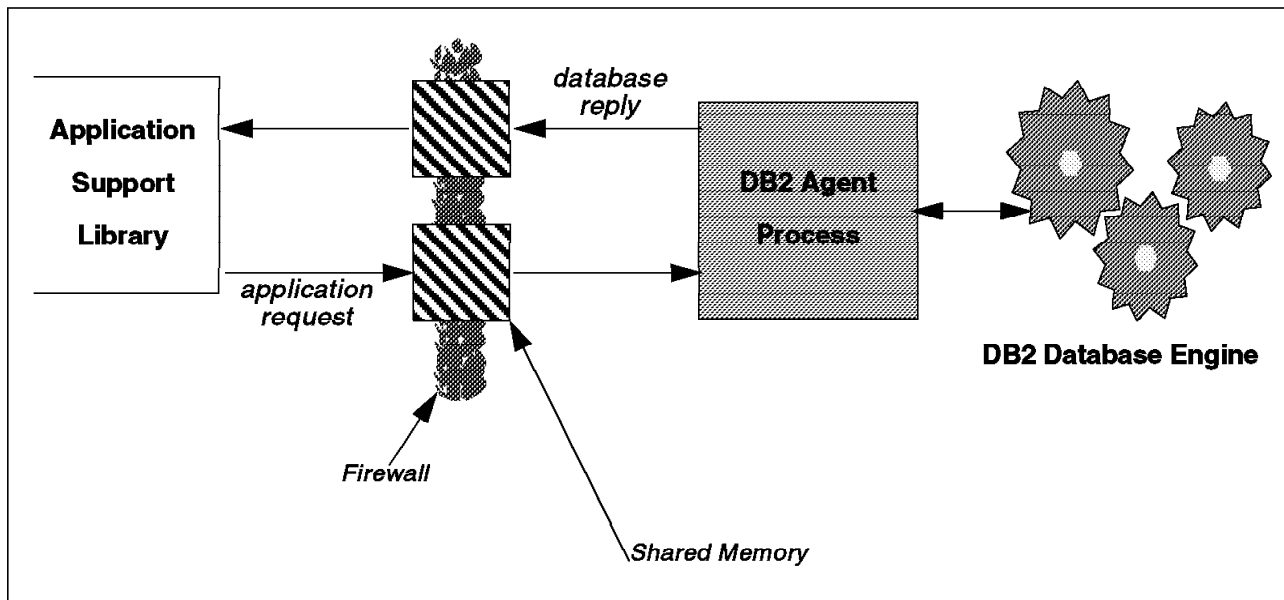


Figure 31. Process Model for DB2

Figure 31 shows the DB2 process model. There is a one-to-one mapping of application processes and database application processes. Each database agent works on behalf of an application process. They communicate with each other using Inter-Process Communication (IPC) techniques (message queues, shared memory, and semaphores). This architecture provides a firewall to protect the database and the database manager from an errant database application.

The firewall provides the following benefits:

- It maintains the integrity of the data in the database. An application programming error cannot cause an internal buffer of the database manager to be overwritten.

- It improves the reliability of the database manager. An application programming error cannot crash the database manager or any of the other user application processes, such as R/3.
- It provides optimal performance between the database application and the database engine.
- The number of clients increases. Database servers have a limited number of clients they can serve effectively. By reducing the amount of memory each database agent requires, the total number of clients supported by a single database server increases.

Database agents are created when the Application Server issues a CONNECT statement to the database.

3.3.1 DB2 Central Instance Processes (No Clients)

This section describes the DB2 processes that are executing when the database manager is started and no connections have been issued. The processes are as follows:

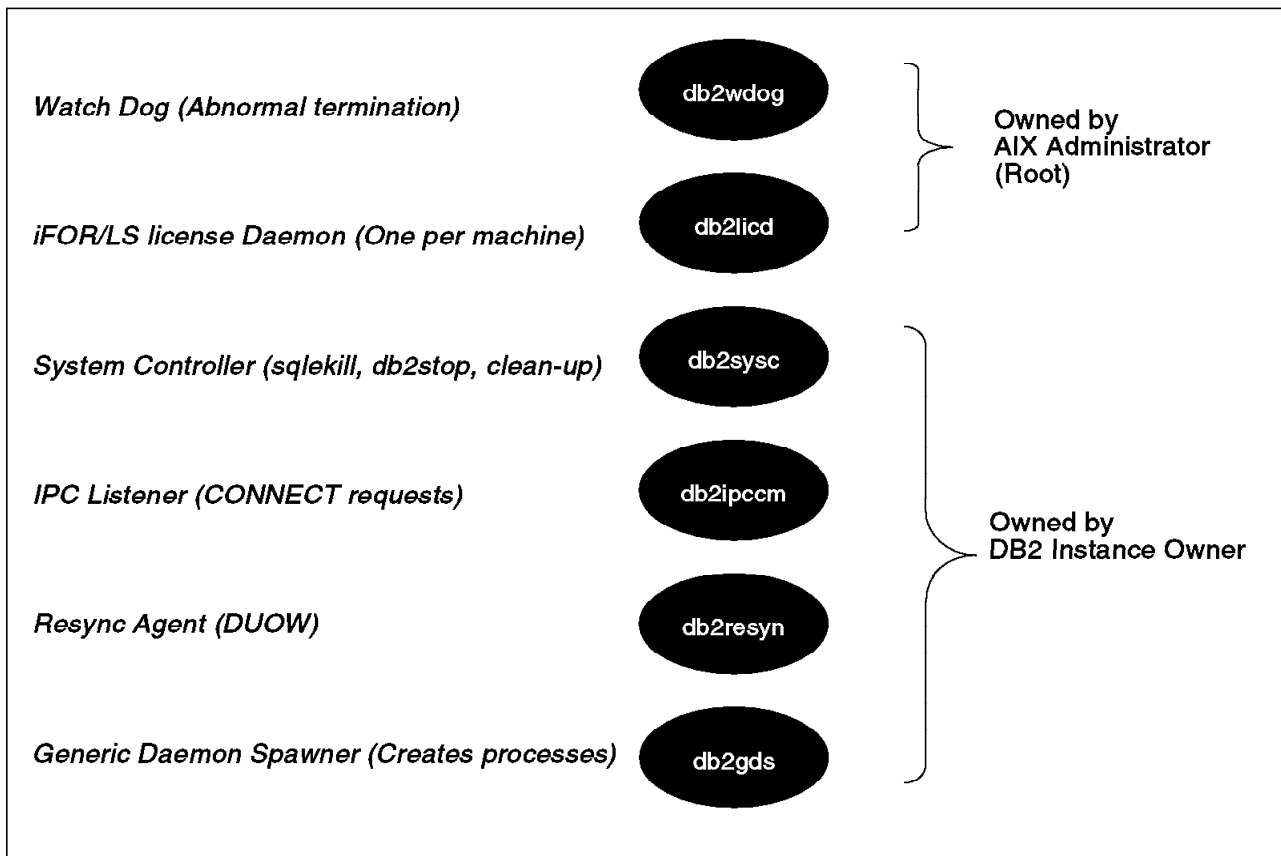


Figure 32. DB2 Process Model - No Database Connections

There are a number of processes created when the database manager is started. The starting of the DB2 instance can be done with the DB2 command db2start or from <SAPSID>adm with the startsap command. Figure 32 shows the processes that are created when only the database manager is started. This means there are no connections to the database at this time. There are two processes that are owned by the AIX system administrator (root). These are the

db2wdog and the db2licd processes. The other processes are owned by the DB2 instance owner. The description of the processes are as follows:

- **db2wdog**
The purpose of the watch dog process is to handle abnormal terminations. This process is the parent of the system controller (db2sysc) process. The watch dog process runs with root authority on the AIX system. The watch dog is needed because SIGKILL signals cannot be caught. It runs as root so it is immune to a `kill -9 -1` that might be issued by the instance owner.
- **db2licd**
The purpose of the DB2 license daemon is to handle license administration for all instances of the DB2 products on one machine. However, remember that an R/3 DB2 Central Instance will probably have only one DB2 instance.
- **db2sysc**
This is the System Controller. The purpose of the System Controller in DB2 is to handle the start up, to create the DARI process, and to stop the database instance. It will handle calls for abrupt shutdown via the `sqlikill()` API.
- **db2ippcm**
This is the IPC (Inter-Process Communication) listener. The IPC listener is responsible for handling the creation of a database agent on the server to handle the requests of an application.
- **db2resyn**
This is the Resynchronization (Resync) Agent. The purpose of the Resync Agent is to scan the global resync list to perform TM (Transaction Monitor) or RM-initiated (Resource Manager) resynchronization for transactions that are involved in DUOW (Distributed Unit of Work).
- **db2gds**
This is the Generic Daemon Spawner. The purpose of the Generic Daemon Spawner is to fork new processes.

You can check the DB2 processes that are executing by issuing the following command in AIX:

```
ps -ef | grep db2
```

3.3.2 DB2 R/3 Central Instance with Clients Connecting to Database

This section looks at the additional processes that are created with clients connecting to the database. We will also look at some of the processes that are created with other database activities, such as backup or restore. Keep in mind that there are more DB2-specific processes than will be covered in this document. For a complete description of DB2 processes, check the *DB2 Administration Guide*.

All connection requests, whether they are local or remote, are allocated to a corresponding database agent. Once the database agent has been created, all database requests are performed by the database agent on behalf of the application. These corresponding agents in DB2 are called db2agent.

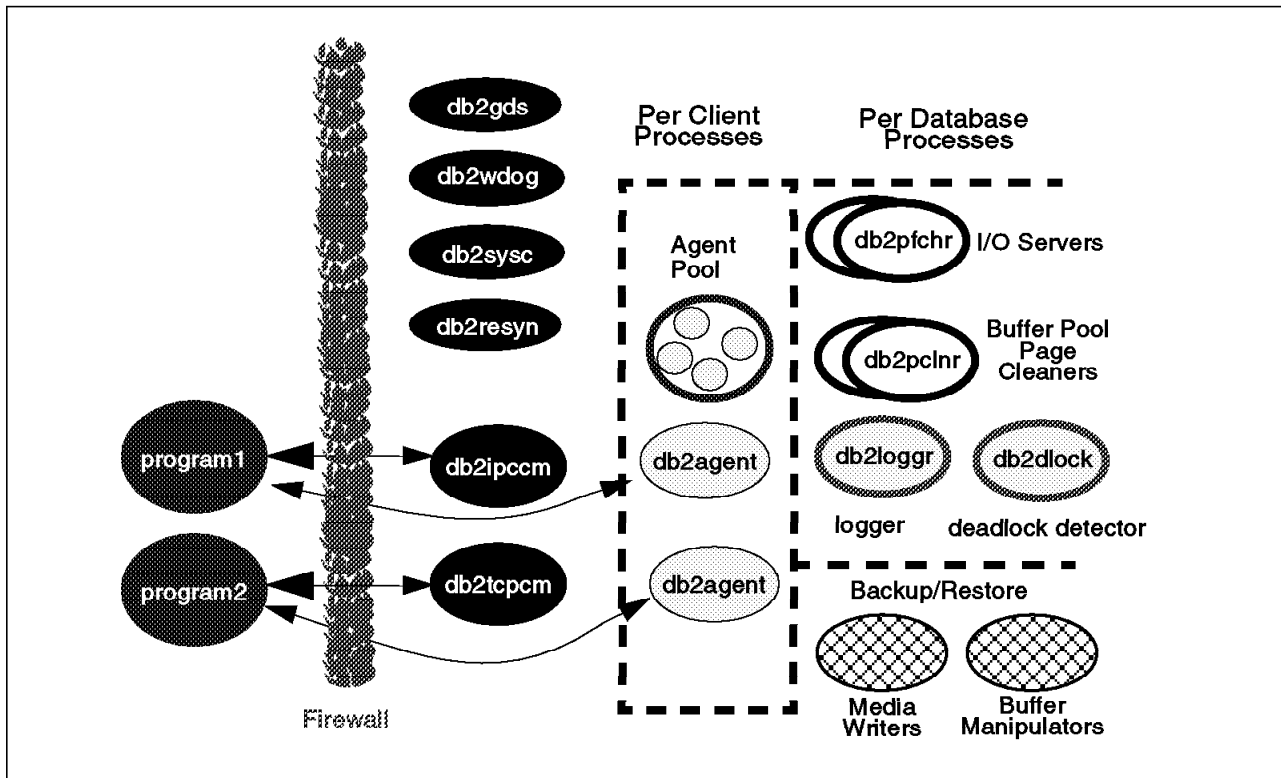


Figure 33. DB2 for AIX Process Model with Connecting Clients

Figure 33 shows some of the DB2 processes that occur when the database is active. An agent pool is created from which a db2agent is dispatched. A db2agent process may be:

- Idle—Initialized and available for a connect request.
- Connected to a database with an alias—For example, db2agent(AUS). This agent is connected to the database alias called AUS.
- Connected to an instance—For example, db2agent(instance). This agent is connected to an instance.

There are also processes that exist on the database level. These processes handle database operations such as buffer manipulation, login, deadlock detection, and the backup or restore of a database. Some of the other database processes are the following:

- db2dlock** The database deadlock detector process looks for any deadlock situations in a database. If one is determined, the database manager takes action to resolve it.
- db2loggr** The database logger process handles all of the log I/O operations for the database.
- db2pclnr** The page cleaner process also helps to increase efficiency by asynchronously writing dirty pages when the CPU would otherwise be idle. The number of page cleaners is configurable.
- db2pfchr** The prefetch process performs read-ahead, big-block, and parallel I/O. This allows for more efficient processing of data.

All of the processes just mentioned are at the database level. There are other database-level processes that may be running, such as db2dart and db2cart.

The db2dart process handles log retain, and db2cart is the database process that handles user-exit functions.

3.3.3 Displaying R/3 Processes

You can display the R/3 processes from AIX with the command:

```
ps -ef | grep sap
```

The processes will look similar to the following:

```
ausadm 16620 28202 0 18:18:03 - 0:05 dw.sapAUS_DVEBMGS00
pf=/usr/sap/AUS/SYS/profile/AUS_DVEBMGS00_f01n07e.itsc.austin.ibm.com
```

```
ausadm 18932 16620 0 18:18:04 - 1:23 dw.sapAUS_DVEBMGS00
pf=/usr/sap/AUS/SYS/profile/AUS_DVEBMGS00_f01n07e.itsc.austin.ibm.com
(This is DIA 0 in Figure 34 on page 66. Notice how the parent process ID is 16620.)
```

```
ausadm 19742 16620 0 18:18:05 - 0:00 dw.sapAUS_DVEBMGS00
pf=/usr/sap/AUS/SYS/profile/AUS_DVEBMGS00_f01n07e.itsc.austin.ibm.com
(This is DIA 4 in Figure 34 on page 66. The parent process of this process has an ID of 16620.)
```

```
ausadm 22254 28202 0 18:18:03 - 0:13 co.sapAUS_DVEBMGS00 -F
pf=/usr/sap/AUS/SYS/profile/AUS_DVEBMGS00_f01n07e.itsc.austin.ibm.com
```

(Note that all processes are not shown.)

You can display the R/3 "dw" or Work Processes in R/3. The "dw" stands for "disp + work". (Remember, the Dispatcher forks various works processes and then begins its dispatching routine. These forked processes recognize themselves as Work Processes, and start to work as either dialog, batch or update processes. For more information, please see Figure 3 on page 3. (In AIX, these are dw.sap<SAPSID> processes.) To display these processes, we entered the transaction code SM50. The PID column will match up to the process ID in the AIX process stack. These processes are all dw processes. One additional dw process will be shown on the AIX process stack. This process is the parent of all the other dw processes and is known as the Dispatcher.

The number of processes and the type of processes is defined in the R/3 profiles located in /usr/sap/<SAPSID><SAPSID>/SYS/profile. In our case, we had the following configuration: (All of these processes can be seen in Figure 34 on page 66.)

```
rdisp/wp_no_dia=5 - type DIA
```

```
rdisp/wp_no_vb=2 - type UPD
```

```
rdisp/wp_no_vb2=1 - type UP2
```

```
rdisp/wp_no_enq=1 - type ENQ
```

```
rdisp/wp_no_btc=2 - type BTC
```

```
rdisp/wp_no_spo=1 - type SPO
```

No.	Ty.	PID	Status	Reason	Start	Err	Sem	CPU	Time	Program	Cl.	User
0	DIA	18932	running		Yes					RSMON000	100	CALENE
1	DIA	47122	waiting		Yes							
2	DIA	45846	waiting		Yes							
3	DIA	32026	waiting		Yes							
4	DIA	19742	waiting		Yes							
5	UPD	25634	waiting		Yes							
6	UPD	44838	waiting		Yes							
7	ENQ	20522	waiting		Yes							
8	BTC	21038	waiting		Yes							
9	BTC	34866	waiting		Yes							
10	SPO	53046	waiting		Yes							
11	UP2	31034	waiting		Yes							

Figure 34. Displaying R/3 Processes with the Database Performance Monitor

3.3.4 DB2 for AIX and R/3 Processes

This section looks at the processes that are created from an AIX, R/3, and DB2 perspective. Some of these processes were discussed earlier in this chapter.

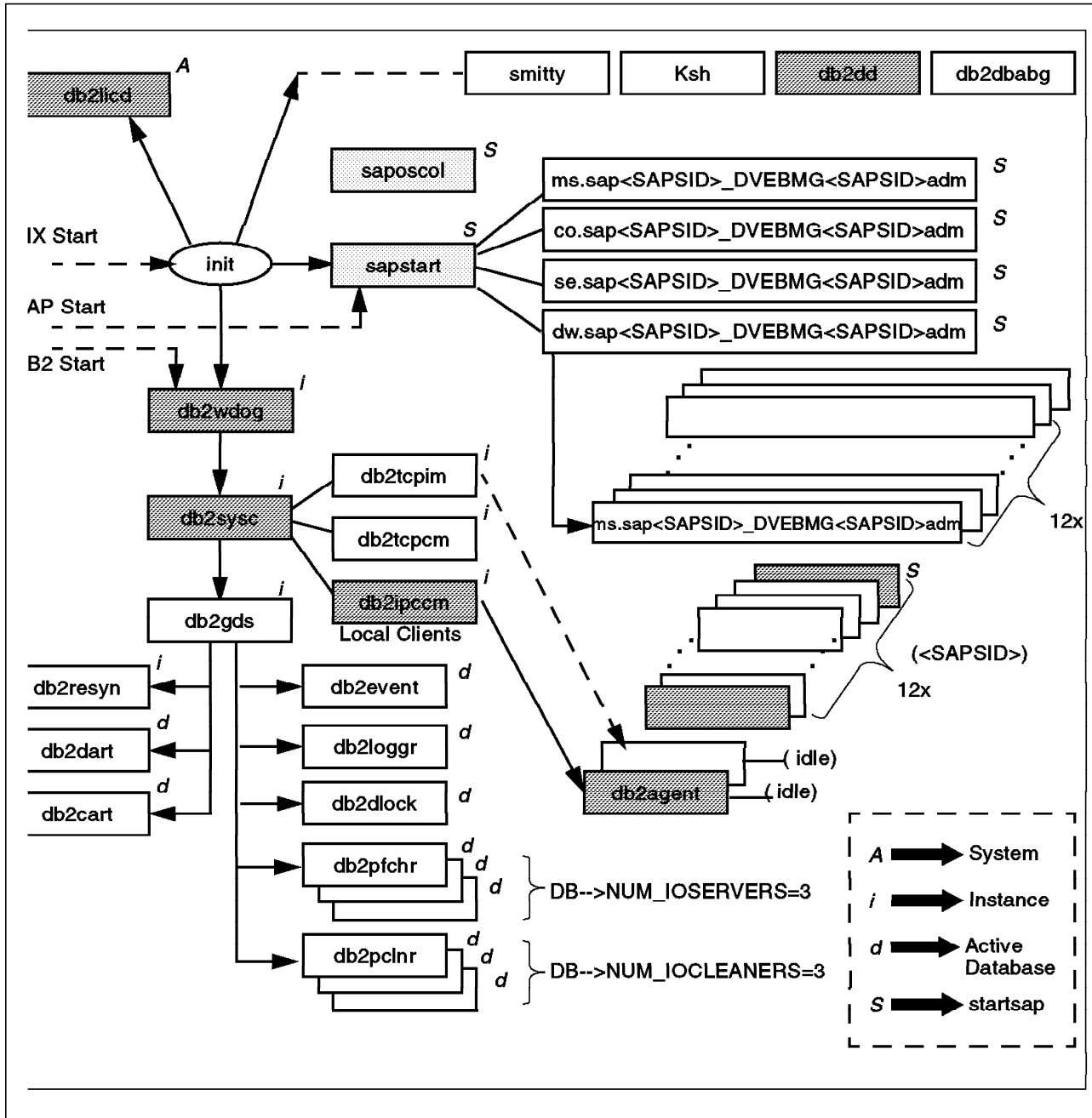


Figure 35. DB2 for AIX and SAP R/3 Processes

Figure 35 shows the process relationship between AIX, DB2, and R/3. The legend in the figure indicates which processes belong to which portion:

- A This indicates that this is an AIX process.
- db2licd This is the license daemon for the DB2 product.
- i This indicates that this is a DB2 process that is created at the DB2 instance level. Remember that there is only one DB2 instance that is created with the R/3 installation. Some of these we have mentioned previously in this chapter.
- db2tcpim This process is for the interrupt handler for TCP/IP remote clients. For example, an interrupt occurs when a remote user process issues a CTRL+C while waiting for a long query.

db2tcpm	This process works along with db2tcpim. It is the communication process for remote clients using TCP/IP. This process listens for a connection request from the remote TCP/IP client and forks a db2agent to handle the TCP/IP clients request when a connection request is detected.
db2ipccm	This is the communication process for local clients.
d	This means that this process is at the database level.
db2event	This process belongs to the DB2 Event Monitor.
db2pfchr	This is a configurable parameter within the database. It is used for read-ahead, big-block, and parallel I/O operations.
db2pclnr	This is the buffer pool cleaner that helps to increase the efficiency of the buffer pool writes to disk. The buffer pool pages can be written to disk asynchronously to maintain a larger buffer pool memory. This is also a database tunable parameter.
db2dart	This
db2cart	This process is for user-exit functions.
db2agent	This is the process that is working on behalf of the application to correspond with the database.
db2dd	This is the Database Director.
db2dbabg	This process is forked from the Database Director. It is responsible for the Redirected Restore option within DB2.
S	These processes are part of the processes that are created during the execution of startsap.
sapocol	(SAP Operating System Collector) This process collects system information and puts it into the SAP system database.
ms	This is the message server.
co	This is the collection process. This process runs on the instance where the central log is to be maintained.
se	This is the send process. This process runs on each instance of your system, including the Central Instance. The send process forwards messages from each instance to the central log.
dw	This is the Dispatcher and Work Processes.

3.3.5 DB2 and R/3 Users

This section looks at the different users accessing the DB2 and/or R/3 system. We look at these users from both a database and R/3 point of view. During the installation of R/3 and DB2, you are prompted to enter a <SAPSID>. Users are created for you as part of the R3INST program in AIX. We show these users with a sample <SAPSID>. The <SAPSID> we use throughout this document is AUS.

3.3.6 DB2 Users

There are three DB2 users that are created in AIX:

- db2aus This is the DB2 instance owner (SYSADM).
- ausadm This is the R/3 administrator (SYSCTRL).
- sapr3 This is the database user in R/3.

SYSADM and SYSCTRL are authorization levels in DB2. In R/3, the <SAPSID> user is the instance and database administrator in DB2. The R/3 administrator is <SAPSID>adm. You must change password for R/3 users within DB2admin tool. The reason for this requirement is that they changed not only in AIX but also in R/3. If you change R/3 user password through AIX, those changes are not passed on to R/3. There are more levels than SYSADM and SYSCTRL. The different levels of authority in DB2 are:

- SYSADM - System Administration Authority
- SYSCTRL - System Control Authority
- SYSMANT - System Maintenance Authority
- DBADM - Database Administration Authority

We only discuss SYSADM and SYSCTRL in this document as they are part of the default R/3 installation. However, you may want to give other R/3 users certain privileges within the database. Authorization levels in DB2 provide you with the hierarchy for the database administration capabilities. An authority is a set of privileges covering a set of database objects. These authorities are assigned to a group of users. Each member of the group has the same DB2 authority, unless they are explicitly removed. Let's look at some of the tasks that the SYSADM can perform.

SYSADM is at the top of the hierarchy. SYSADM is the DB2 system administrator. SYSADM is able to perform any of the DB2 administration operations as well as select any information from the database. SYSADM is the only user able to make changes in the database manager configuration.

System control (SYSCTRL) provides the ability to perform almost any administration command within DB2. For example, SYSCTRL in R/3 can start the database instance. That is why the R/3 administrator can start the database manager. However, a member of the SYSCTRL group does not have the authority to access user information or modify the database manager configuration. SYSCTRL offers almost complete control of database objects defined in the DB2 instance, but cannot access user data directly, unless granted the privilege to do so. A user with this authority, or higher, can perform the following functions within DB2:

- Update the database, node and DCS directory entries
- Update database configuration parameters (Not database manager configuration parameters)
- Force applications using the DB2 force application command
- Run the RESTORE/BACKUP/ROLLFORWARD commands within DB2
- Create or drop a tablespace

The group authorities are also related to the security mechanisms of the AIX operating system. For example, an AIX user who is placed in the same AIX

group as SYSCTRL has their access controlled within DB2 according to the SYSCTRL authority and within AIX as their file system access control group name.

As part of the R/3 installation, there are two groups created within AIX. The group for SYSADM is sysadm. The AIX group that is created during R/3 installation for SYSCTRL is sysctrl. If you place a user in the same AIX group as sysadm or sysctrl, the user will inherit the DB2 authorizations. Be careful in doing so that you understand the implications of such a move. For example, you might want a user to be able to run the BACKUP and RESTORE utilities. However, if you place that user in sysadm within AIX, know all the functions that SYSADM can perform within DB2 before doing so. For more information, please see the *DB2 Version 2 Planning Guide for Database Administrators*.

3.3.6.1 DB2 Privileges

We have discussed authorization levels within DB2. This section briefly discusses privileges within DB2. A privilege is the right to create or access a database resource. Both DB2 authorities and privileges on database objects are hierarchical in nature.

There are three types of privileges within DB2:

1. Ownership or control privileges—For most objects, the user or group who creates the object has full access to that object. CONTROL privilege is automatically granted to the creator of an object. There are some database objects, such as views, that are exceptions to this statement. Having Control privilege is very much like having ownership of the object in that you have the right to access an object and even grant privileges to other users or groups on that object. Privileges are controlled by users with ownership or administrative authority.
2. Individual privileges—These are privileges that allow you to perform a specific function, sometimes on a specific objects. These privileges include: SELECT, DELETE and INSERT operations within tables of the database.
3. Implicit privileges—As an example, a user who can execute a package where that package involves other privileges obtains those privileges while executing the package. This is known as the EXECUTE privilege.

As part of your R/3 installation and customization, you will be responsible for determining which users will have which access on your R/3 applications. Information about privileges is maintained in four system catalog views:

- SYSCAT.DBAUTH—Contains database privileges
- SYSCAT.TABAUTH—Contains table and view privileges
- SYSCAT.PACKAGEAUTH—Contains package privileges
- SYSCAT.INDEXAUTH—Contains index privileges

Let's say we want to look at all the DB2 R/3 users that have database privileges. We could issue the following SQL statement:

```
db2 "select * from syscat.dbauth"
```

The output would be similar to the following:

GRANTOR	GRANTEE	GRANTEETYPE	DBADMAUTH	CREATETABAUTH	BINDADDAUTH	CONNECTAUTH	NOFENCEAUTH
SYSIBM	DB2AUS	U	Y	Y	Y	Y	Y
DB2AUS	SAPR3	U	N	Y	Y	Y	N
DB2AUS	AUSADM	U	Y	Y	Y	Y	Y

3 record(s) selected.

From the output, we can see that there are two users who have been granted DBADM authority: DB2AUS and AUSADM. The user SAPR3 can do various database activities such as connect to the database and create tables within the database.

Chapter 4. R/3 Database Administration

R/3 is a system for a large number of business applications that are supplemented by software development tools that are handling large amounts of data. This data requires many gigabytes of disk storage. A SAP database administrator has to be able to manage that data in terms of performance and space. You want to be able to anticipate the growth of your data to avoid running out of storage that might result in users having to wait for you to administer the system. You also need to be aware of the organization of the data, especially after database operations such as inserts, updates and deletes. The performance of your application could be affected by fragmentation of space when executing these typical database operations.

This chapter explores the tools and utilities provided to assist a SAP database administrator. Some of these utilities are provided in the R/3 product, some in DB2, and some in AIX. We look at how there are multiple ways of obtaining the same type of information. Each SAP administrator is able to decide the method that is appropriate for his/her R/3 environment. As we explain the utilities, we highlight those parameters that you can change within DB2 that will affect your R/3 environment.

4.1 Tools and Utilities

This section is an introduction to tools and utilities that you can use to manage a DB2 database on your R/3 system. We use these utilities throughout the remainder of this document. This section shows how to invoke the tool or utility, some of the initial screens that a user might need to navigate, and an overview of the functions available in each.

4.1.1 DB2 Tools and Utilities

There are two basic interfaces within DB2:

- Command Line Processor (CLP)
- Database Director

Figure 36 on page 74 shows the two DB2 interfaces we describe in this section and some of the basic differences between them.

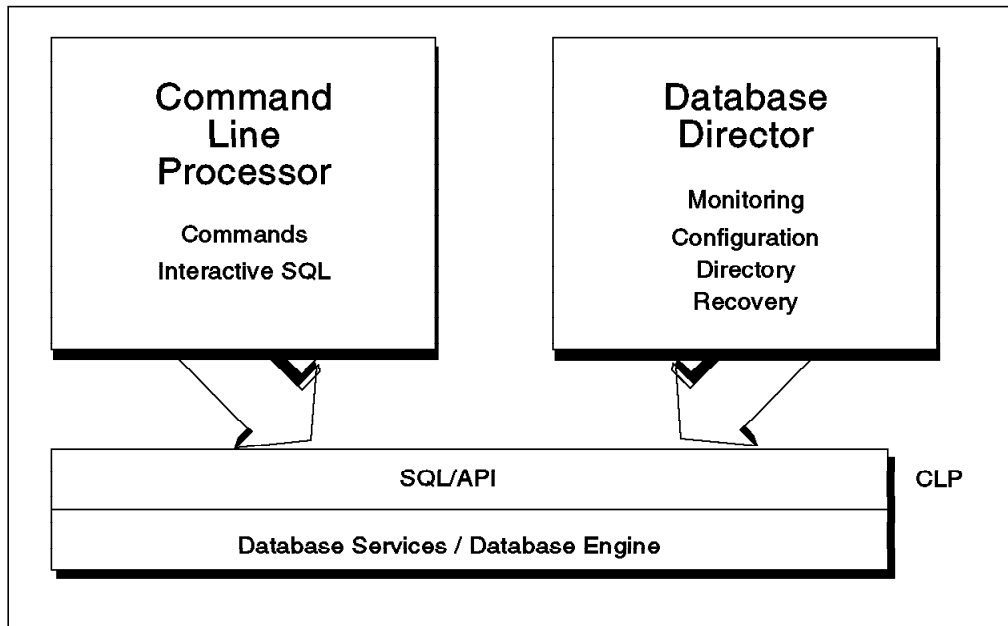


Figure 36. DB2 Interfaces

4.1.1.1 Using the Command Line Processor

The Command Line Processor is installed with all the DB2 products. Using the CLP, you can manage a database environment by entering DB2 commands and SQL statements in an interactive mode or from command files. The following are some of the functions that are possible using the CLP:

- Maintain a history file of all requests
- Redirect the output of requests
- Access both local and remote databases
- Request the command syntax for DB2 utilities
- Request information on DB2 messages

The following is a complete list of commands that can be entered using the CLP:

DB2 Commands Available in CLP

ACTIVATE DATABASE	ECHO	LOAD QUERY
ATTACH	EXPORT	PREP/PRECOMPILE
ATTACH TO	FORCE APPLICATION	PRUNE HISTORY
BACKUP DATABASE	GET AUTHORIZATIONS	QUERY CLIENT
BIND	GET CONNECTION STATE	QUIESCE TABLESPACES
CATALOG APPC NODE	GET INSTANCE	QUIT
CATALOG DATABASE	GET MONITOR SWITCHES	REBIND
CATALOG DCS DATABASE	GET SNAPSHOT	REGISTER
CATALOG GLOBAL DATABASE	GET/RESET/UPDATE DB CFG	RELEASE
CATALOG IPXSPX NODE	GET/RESET/UPDATE DEM CFG	REORG TABLE
CATALOG LOCAL NODE	HELP	REORGCHK
CATALOG NETBIOS NODE	IMPORT	RESET MONITOR
CATALOG TCPIP NODE	INVOKE	RESTART DATABASE
CHANGE DATABASE COMMENT	LIST APPLICATIONS	RESTORE DATABASE
CHANGE SQLISL	LIST BACKUP/HISTORY	ROLLFORWARD DATABASE
CONNECT	LIST COMMAND OPTIONS	RUNSTATS
CONNECT RESET	LIST DATABASE DIRECTORY	SET CLIENT
CONNECT TO	LIST DCS APPLICATIONS	SET CONNECTION
CREATE DATABASE	LIST DCS DIRECTORY	TERMINATE
DB2START/DB2STOP	LIST INDOUBT TRANSACTIONS	UNCATALOG DATABASE
DEACTIVATE DATABASE	LIST NODE DIRECTORY	UNCATALOG DCS DATABASE
DEREGISTER	LIST PACKAGES/TABLES	UNCATALOG NODE
DETACH	LIST TABLESPACE CONTAINERS	UPDATE COMMAND OPTIONS
DISCONNECT	LIST TABLESPACES	UPDATE HISTORY
DROP DATABASE	LOAD	UPDATE MONITOR SWITCHES

All of the DB2 commands may be issued from the CLP. You may also issue dynamic SQL statements from the CLP. You must preface all DB2 commands or SQL statements that are entered from the command line with `db2` followed by the command or the SQL statement.

An alternative method is to issue DB2 commands or SQL statements from an interactive CLP session. To enter the interactive CLP mode issue the command `db2`. The Command Line Processor has two parts, a front-end process and a back-end process. The front-end process is called `db2` and the back-end is `db2bp`. The back-end process will maintain a connection to the database. Therefore to release this connection, the terminate command should be used. To end an interactive CLP session, issue the quit command. This ends the CLP session, but does not release the database connection.

When using the Command Line Processor be careful that the shell you are using in your operating system does not mistake SQL statements for operating system commands. If you enclose any SQL statements or DB2 commands within double quotation marks (" "), it will ensure that they are not interpreted as operating system commands. From within the Interactive Command Line Processor, you can issue operating system commands by prefacing them with an exclamation mark "!" before the operating-system command. For example:

```
!<operating-system-command>
```

If a command that you are entering exceeds the limit allowed by the operating system, use a backslash "/" as the line continuation character.

The complete syntax and explanation of all SQL statements is documented in the *DB2 SQL Reference* manual. The complete syntax and explanation of all DB2 commands is documented in the *DB2 Command Reference*.

However, you can obtain syntax and information for all DB2 commands from the Command Line Processor. The db2<SAPSID> user ID defaults to the C shell.

- db2 ?—displays a list of all DB2 commands
- db2 ? command—displays information about a specific command
- db2 ? SQLnnnn—displays information about a specific SQLCODE
- db2 ? DB2nnnn—displays information about an error generated by the CLP

If you issue the command db2 list command options, the settings for the Command Line Processor will be displayed. Figure 37 shows the default settings. These settings can be updated for each CLP session or globally by using the DB2OPTIONS environment variable.

Command Line Processor Option Settings		
Backend process wait time (seconds)		(DB2BQTIME) = 1
No. of retries to connect to backend		(DB2BQTRY) = 60
Request queue wait time (seconds)		(DB2RQTIME) = 5
Input queue wait time (seconds)		(DB2IQTIME) = 5
Command options		(DB2OPTIONS) =
Option Description		Current Setting

-a	Display SQLCA	OFF
-c	Auto-Commit	ON
-e	Display SQLCODE/SQLSTATE	OFF
-f	Read from input file	OFF
-l	Log commands in history file	OFF
-o	Display output	ON
-p	Display interactive input prompt	ON
-r	Save output to report file	OFF
-s	Stop execution on command error	OFF
-t	Set statement termination character	OFF
-v	Echo current command	OFF
-w	Display FETCH/SELECT warning messages	ON
-z	Save all output to output file	OFF

Figure 37. Command Line Processor Option Settings

- db2 ? update command options—displays all the available options, the possible values for each option and the usage of the update command
- db2 update command options using <options....>—to change the value of one or more options.

You can also create a file with SQL statements and DB2 commands that you wish to execute using the CLP. When you create the text file with CLP commands you do not require the db2 prefix. For example, suppose you have a file called *file.clp*, as shown in Figure 38 on page 77. Every DB2 command or SQL statement in the file is terminated with a semicolon (;), which is the default terminating character. You may change the terminating character if you wish by using the -t option. To execute the CLP input file in Figure 38 on page 77, enter db2 - tvf file.clp from the command line.

```

connect to AUS;
create table tab1
(name varchar(20) not null,
 phone char(4),
 salary dec(7,2));
select * from tab1;
commit work;
connect reset;

```

Figure 38. CLP Input File - file.clp

4.1.1.2 The Database Director

The Database Director is a graphical database administration tool. You can invoke it by typing db2dd at the system command prompt or from the database administrator's ID (db2<SAPSID>), or you may access it through the SMIT interface under the **Applications** entry from the main SMIT screen.

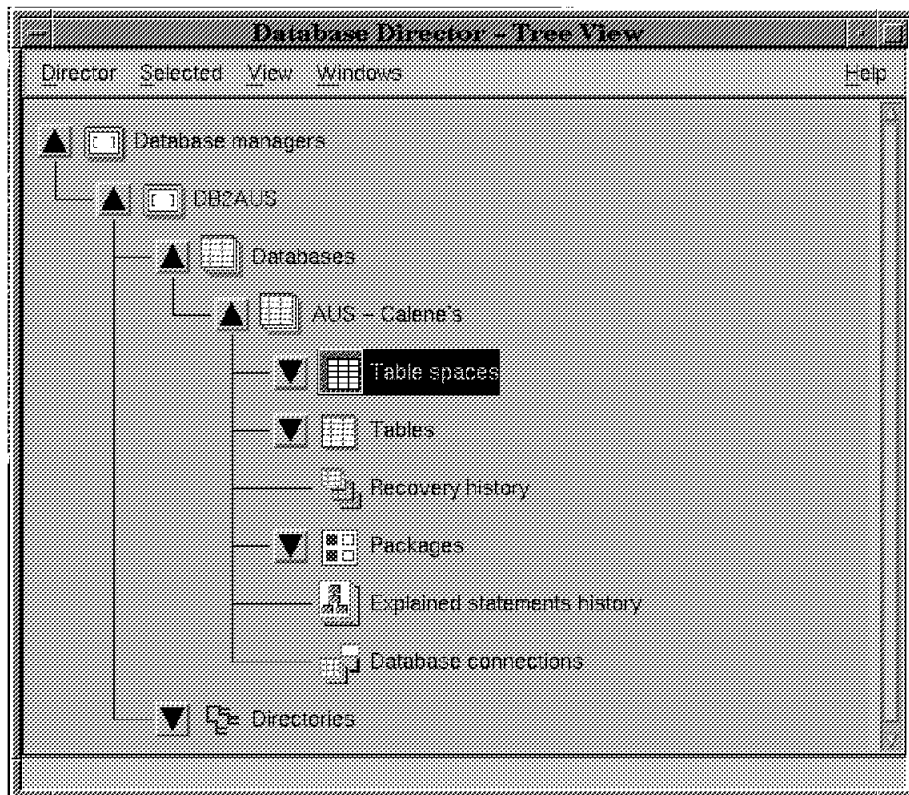


Figure 39. Tree View of the Database Director

Using the Database Director, you can perform the following tasks:

- **Configuration** - You can display and alter the settings of the resources allocated to your database. You can set the size of the buffer pool, the log files, and the sort buffer. You can specify the number of concurrent application programs that can connect to each database. You can also enable a database for roll forward recovery.

- **Recovery**—You can back up, restore, or roll forward a database or a tablespace.
- **Directory**—You can manage directories for accessing local and remote databases, create a database, catalog or uncatalog a local or remote node or database, drop a database, and list information contained in the node, database, and dcs directories.
- **Managing Media**—You can create, drop, or change tablespaces. You can modify the storage assigned to tablespaces.

4.1.1.3 Performance Monitor

You can access the DB2 Performance Monitor through the Database Director. To get the Snapshot Monitor screen, click on the database that you want to monitor. Hold down the right mouse button to get the pull-down menu. From there, you will see the option "Start monitoring". The screen is similar to the following:

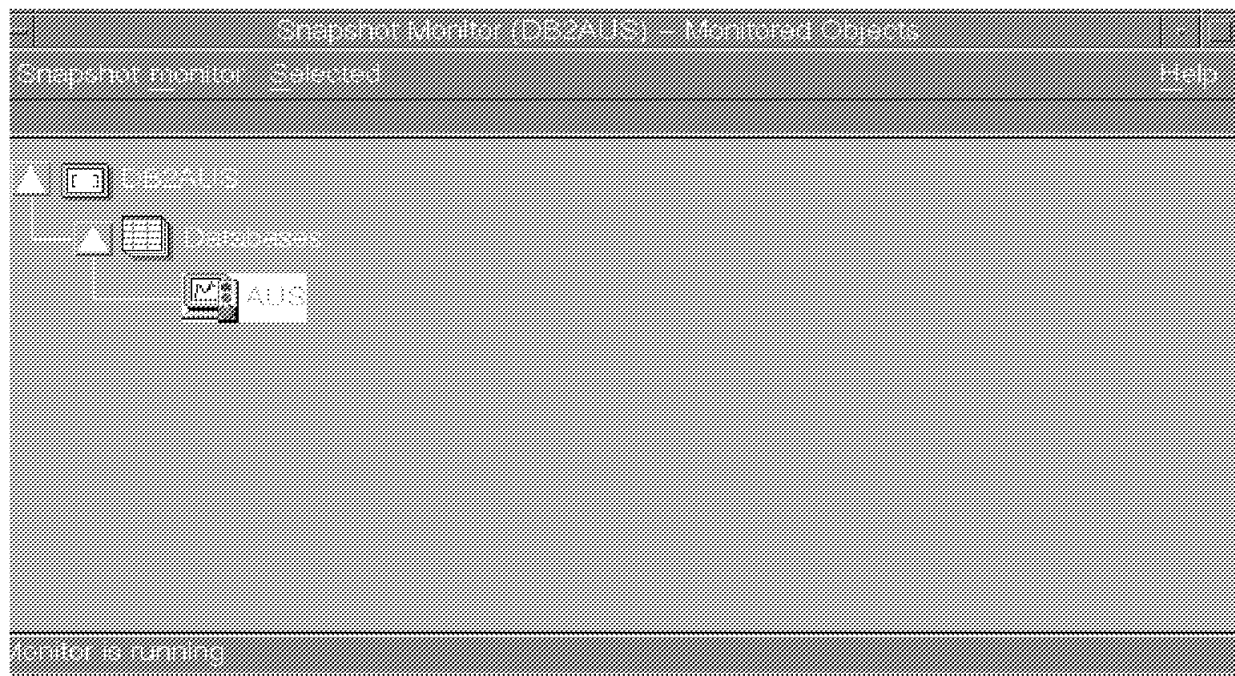


Figure 40. DB2 Database Director - Snapshot Monitor

When the screen first appears, there will be a question mark (?) next to the database. Double-click on the database that you want to monitor. It will take a few seconds to begin the monitoring. The question mark will be replaced by a green traffic light indicating that the monitor is running. (You will also see a similar text message displayed on the bottom of the screen.)

From Figure 40, we can tell that there is one database in the DB2AUS instance. The database name is AUS. When the traffic light is green, it indicates that monitoring is being performed on that particular database. We cover the Snapshot Monitor in more detail in 4.5.1, "Snapshot Monitoring" on page 135.

4.1.2 The R/3 Database Performance Monitor

The Database Monitor is an integrated R/3 system tool that can assist the DBA in monitoring database activities, disk usage of database files, database parameters, and historical information about the database. The Database Performance Monitor takes advantage of the Command Line Processor (CLP), Call Level Interface (CLI) within DB2, the programming APIs, and DB2 stored procedures. Some of the data that is analyzed is database statistics that have been collected using the Snapshot and Event Monitors within DB2 and DB2 utilities, such as RUNSTATS and REORGCHK.

Within R/3, you can monitor network performance and AIX resource usage, such as memory used by R/3. The various Performance Monitors are integrated in what R/3 calls Computing Center Management System (CCMS). They can also be activated by entering transaction codes in a window box in the R/3 screen.

There are two ways to get to the R/3 Database Performance Monitor entry screen (see Figure 41 on page 80).

1. From the main R/3 screen, select **Tools- > Administration- > Monitoring- > Performance.**
2. From the SAPGUI, you can enter a transaction code. You execute the STUN transaction by entering the code in the window box that is located to the right of the check mark.

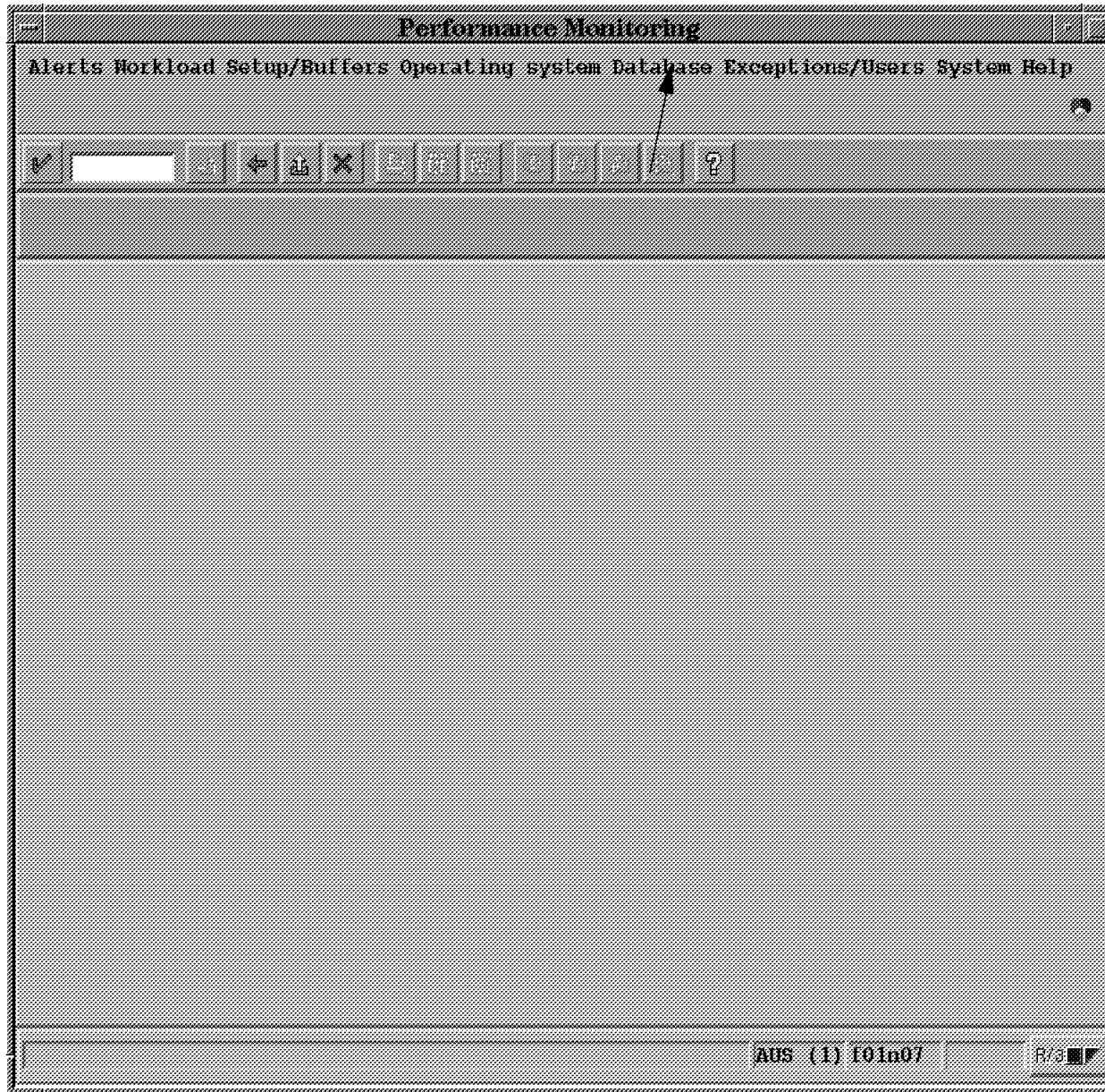


Figure 41. R/3 Entry Screen - Performance Monitor

From the menu bar, click on **Database**. The pull-down menu will show four options:

1. Activity
2. Exclusive lockwaits
3. Tables/Indexes
4. Parameter changes

Suppose you wanted to monitor the tablespaces within your R/3 DB2 database. You could select **Tables/Indexes** from the submenu of the screen shown in Figure 41. What appears next is the first screen of what SAP calls the "State Part" of the R/3 Database Performance Monitor (Figure 42 on page 81).

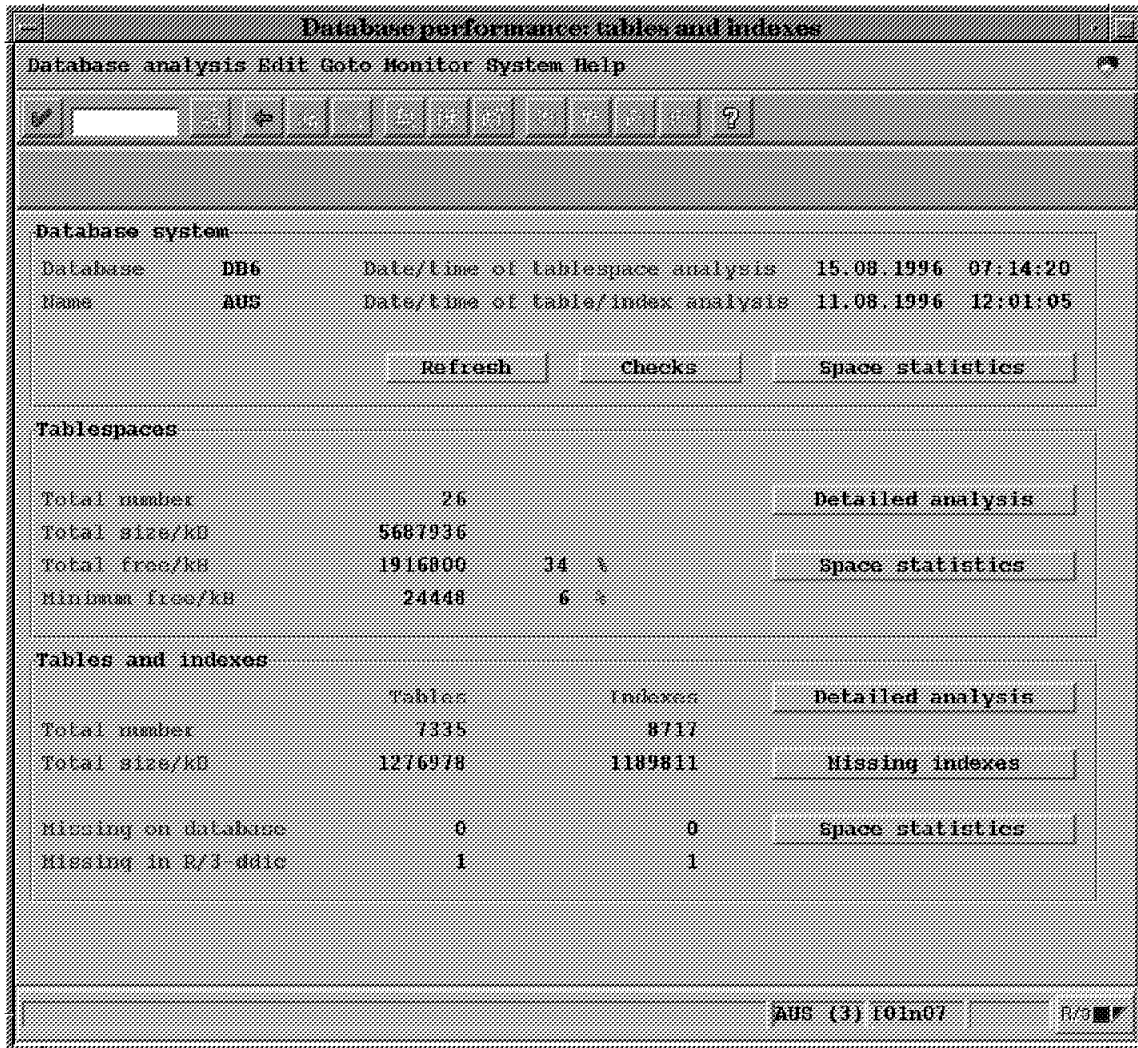


Figure 42. R/3 Database Performance Monitor, State Part

4.1.2.1 The R/3 Alert Monitor

The Alert Monitor is a graphical tool that displays database parameters, several operating system values, and network parameters. To open the Alert Monitor, start the R/3 sapwin application from the directory in which the software executables for the interface are installed. (In our installation, /sapmnt/AUS/exe/sapwin). To start the Alert Monitor from the Performance Monitor Entry Screen (Figure 41 on page 80), select **Alerts->Global->Database_system**.

The main parameters that can be monitored for DB2 for AIX are displayed.

R/3 Alert Monitor

The actual displaying of the Alert Monitor may be affected by the release of R/3. From our experience (We used R/3 Release 3.0C.), when bringing up the Alert Monitor, make sure that you are in the same directory where the executable is stored. In our installation, it was /sapmnt/AUS/exe. Remember that AUS is our <SAPSID>. This requirement may or may not be necessary in your installation.

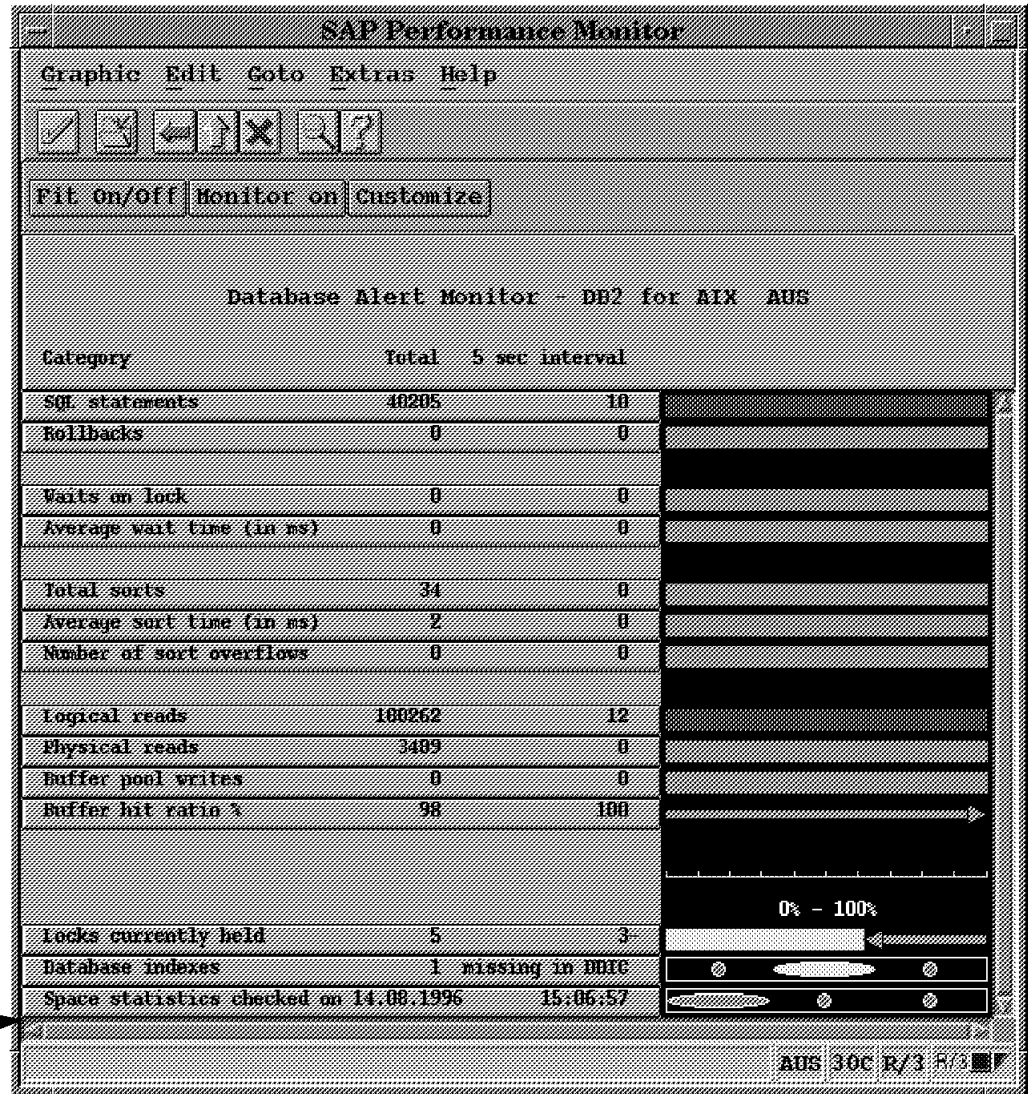


Figure 43. R/3 Alert Monitor, Overview Screen

The Alert Monitor uses several types of symbols to display the status of the database activity. You should be able to see two-color bars that display the level of activity in relation to a user-defined threshold that should not be significantly exceeded. To set the parameters, you click on **Customize**. From there, you can change the parameters by clicking on the **upper** or **lower** button. This increases or decreases the values, respectively.

Depending on the values that you have set, you'll see either the colors, green and red. This indicates that the activity is below/above the threshold value that

you have defined. The size of the bar corresponds to the current value. This is displayed numerically to the left of the bar. The green part indicates the area up to the level of the threshold value. The red part shows how far that value has exceeded the threshold. If the bar is red and filled-in, this indicates that the current value is far above the threshold defined and is critical. The color red might also indicate an improperly set threshold.

If you see the colors green and yellow, this indicates that the activity is below the threshold value that you have defined. The entire bar represents the area up to the threshold value. The green portion indicates the area up to the current value. The yellow part shows how far the value falls below the threshold. If a yellow bar turns green, it indicates that the current value equals the threshold. If the current value is equal to 0 and the threshold value is larger than 0, then the entire bar will turn yellow.

For example, say you want to monitor the free space in the tablespaces within your DB2 database (indicated by the arrow in Figure 43 on page 82), you will probably want to look at two values, SPACESTAT_%%_OK and SPACESTAT_%%_CRIT. These values can be interpreted as follows:

- SPACESTAT_%%_OK

A value of 0 means the parameter is not set. A value greater than 0 and with SPACESTAT_%%_CRIT not set means that if the free space of the smallest container within the tablespace is less than or equal to the SPACESTAT_%%_OK value, the green bar of the tablespace value changes to yellow in the detailed screen. If the value for the free space in the container is less than or equal to the SPACESTAT_%%_OK value, the traffic light becomes yellow.

- SPACESTAT_%%_CRIT

A value of 0 means that the parameter is not set. If the value is greater than 0 and the SPACESTAT_%%_OK is set, the bar for the tablespaces becomes red if the freespace is equal to or greater than that value. The traffic light for the database in the first screen becomes red if the free space in the container is less than or equal to the SPACESTAT_%%_CRIT value and the SPACESTAT_%%_OK value was set.

The values that you see displayed in SPACESTAT_%%_OK and SPACESTAT_%%_CRIT are obtained from the DB2 Performance Monitor. When you exit the Alert Monitor, these threshold values will be lost. Also, if you do a refresh in the Performance Monitor, the values in the Monitor become active when you restart with the STUN transaction.

You can set the Alert Monitor in R/3 to do periodic monitoring and reporting. To do this, select the button **Monitor on**. You obtain output every five seconds. However, the Alert Monitor may adversely impact performance if done every five seconds. It is recommended to use the Alert Monitor to monitor free space in critical situations only.

4.1.3 R/3 DB2admin

This section talks about using the R/3 utility called DB2admin that is invoked from the AIX SMIT interface. DB2admin is used to help manage your R/3 DB2 database. SMIT (System Management Interface Tool) is IBM's utility to assist AIX administrators with system administrative activities without having to remember UNIX commands and parameters. SMIT can be used either in graphic

or ASCII mode. For each dialog screen, the user sees several elements, such as default values, to assist them in their selections. Online help gives detailed descriptions of every screen function.

From the SMIT main menu, if you select **Applications**, you'll see the following screen:

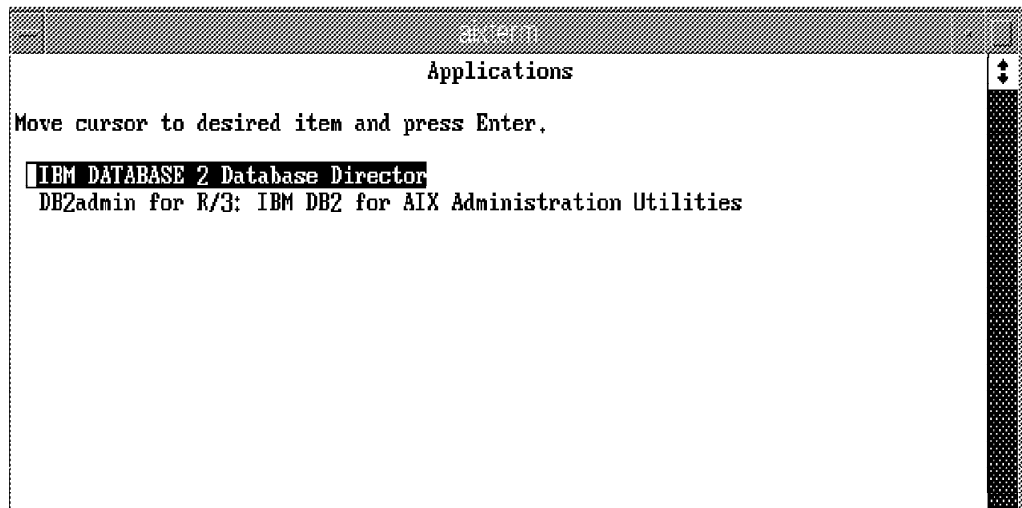


Figure 44. Accessing DB2 Utilities through SMIT

From Figure 44, you can see that there are two entries: IBM DATABASE 2 Database Director and DB2admin for R/3: IBM DB2 for AIX Administration Utilities. If you select the Database Director, you will be using the DB2 Database Director. You will be prompted for your DISPLAY variable within AIX. Then the DB2 Database Director will be displayed at your terminal.

DB2admin is independent of the R/3 front-ends. DB2admin is dependent on the SAP database administrator functions. It assists in the management of DB2 for AIX. Like other functions called through SMIT, the user does not have to know all the commands and parameters. Another advantage of using DB2admin is that a SAP DBA can initiate an action without having to know all the steps required in the procedure.

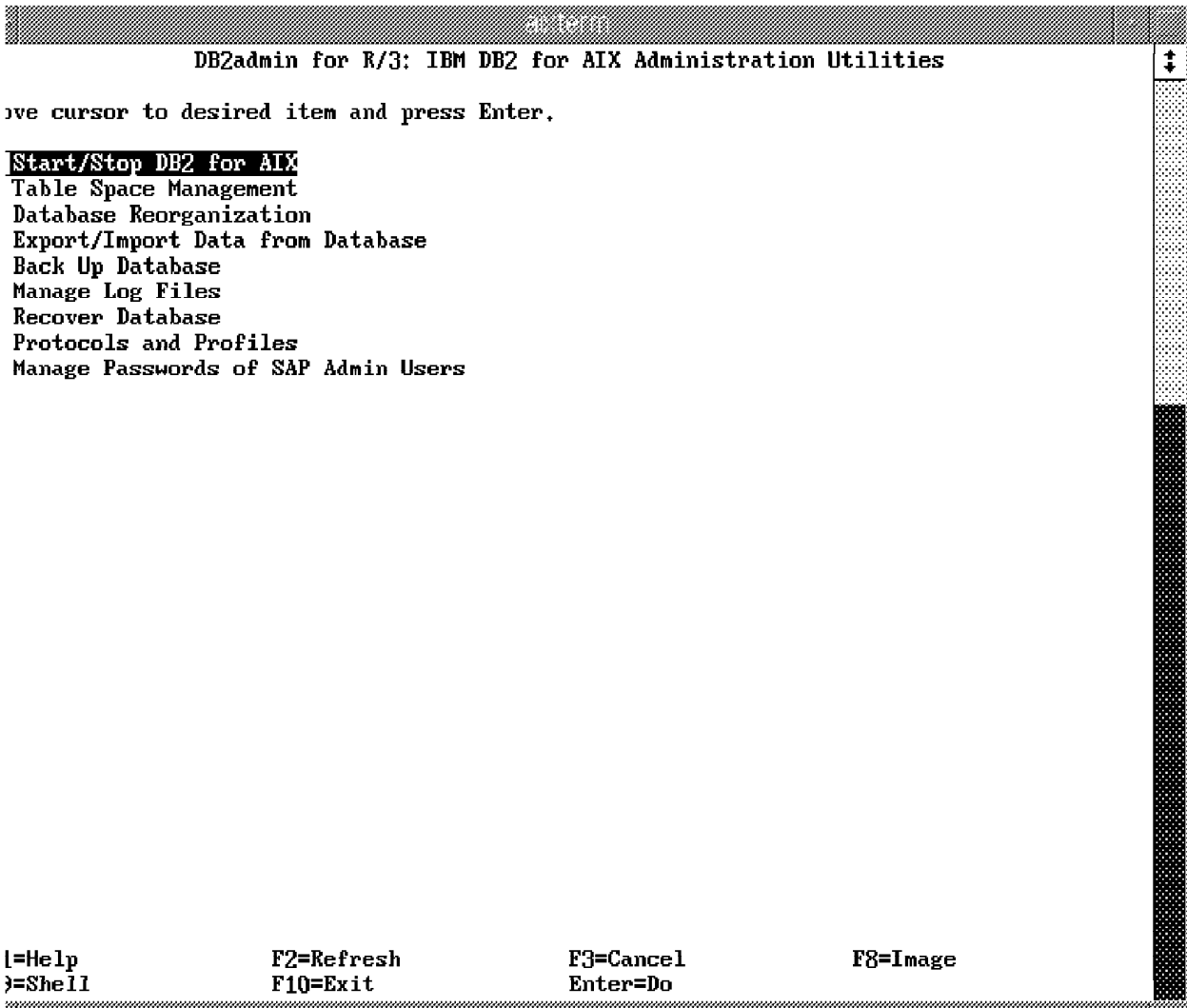


Figure 45. Administering DB2 via the SMIT Interface

Here are the basic functions. We will explain some of these in this chapter.

- Start/Stop DB2 for AIX
This starts and stops the DB2 database manager. The only user who can execute this function is the database administrator. This user is db2<SAPSID>.
- Tablespace Management
From a submenu, you can access various menu options that execute specific DB2 tablespace-management commands. For more information, please see 4.2.1, “Managing Tablespaces with DB2 Commands” on page 87.
- Database Reorganization
With this menu, all tasks related to database reorganization can be accessed. You need to have at least CONTROL privilege on the tables.
- Export/Import Data from Database
You can use the DB2 IMPORT/EXPORT utility to exchange data between different databases, between a database and other tools, or for backup purposes.
- Backup Database

This option is an interface to the backup utilities in DB2. You can look at the existing ADSTM backups, execute a backup of the database, and register a backup as a batch job in the corresponding operating system table.

- Manage Log Files

Here, you can manage the archival of the log files that are used to roll forward a recovered database to a state prior to a previously damaged one so that you can rebuild the database. The management of archiving log files in R/3 is critical so that the database has storage available and the log files that are necessary for rebuilding a database are not lost.

- Recover Database

If this option is selected from DB2admin, all the actions necessary to rebuild a database are done. This includes restoring a database from a backup copy, rolling forward the necessary log files with changes from the active log files, and performing an autorestart of the database manager to replay log files in the event of a system failure.

- Protocols and Profiles

For DB2 R/3 database administration, there are two kinds of log files: the log files written by the database manager that can be used in recovery situations and the history log files written by the DB2admin tool to store information for the database administrator. For example, you can find out when a backup or restore operation was performed and the return code associated with the operation. To distinguish the log files in DB2 from the history log files in R/3, the history log files for the database administrator are called protocol files in the DB2admin tool. You can display and edit profiles used with these protocol files. You can display and edit the user exit profile and the archive profile using the DB2admin tool.

- Manage Password of SAP Admin User

The sub-menus here lead you to dialogs where you can change the password for the R/3 users and <sap>adm. This screen is the only way to change the password in parallel to maintain encryption within both AIX and R/3.

There are files associated with DB2admin. The directory structure shows the default values for DB2admin. The values can be individually changed in the .db2uexit profile. For more information, please see the *BC SAP Database Administration Guide: DB2 for AIX*.

4.2 Managing Tablespaces

This section looks at the ways that you can manage tablespaces within a DB2 R/3 system. We show you that there are a number of ways that you can obtain some of the same information. As a SAP DB2, you will want to know how to monitor free space within tablespaces. We also discuss how to extend the size of your tablespace, hopefully before you exceed the current allocation of space. This section is outlined as follows:

- Commands within DB2 for managing tablespaces
- Understanding tablespace states with DB2
- Using the R/3 Database Performance Monitor
- Sizing considerations for DMS tablespaces
- Creating or dropping a tablespace

- Adding a container to a DMS tablespace
- Using the Redirected Restore option within the Database Director

4.2.1 Managing Tablespaces with DB2 Commands

Tablespace management includes the tasks of creating, deleting, modifying and monitoring table spaces and containers. DB2 provides commands and utilities to perform these tasks. The commands and statements available to the database administrator are the following:

- **LIST TABLESPACES**—The LIST TABLESPACES command lists all the tablespaces contained in the database. For each tablespace, it shows information about type, data type contained, state, size (only for DMS tablespaces) and extent, and prefetch size.
- **LIST TABLESPACE CONTAINERS**—The LIST TABLESPACE CONTAINERS command lists all the containers for a specific tablespace. It shows information about the name, type, and size (only DMS table spaces) of the containers of this tablespace.
- **ALTER TABLESPACE**—The ALTER TABLESPACE statement enables the database administrator to add containers to a DMS tablespace. (This is not supported with SMS tablespaces.) It also allows modification of the PREFETCHSIZE, OVERHEAD, and TRANSFERRATE of a tablespace.
- **CREATE TABLESPACE** —The CREATE TABLESPACE statement creates a new tablespace (SMS or DMS) within an existing database, assigns containers to the tablespace, and records the tablespace definition and attributes in the system catalogs.
- **DROP**—The DROP statement deletes, among other table objects, an index, table, tablespace, or view. Objects that are directly or indirectly dependent on that object are also deleted or marked as inoperative.

4.2.2 Displaying Tablespace Information

This section looks at the various ways that you can obtain information about your tablespaces. We show multiple ways to obtain some of the same type of information. We show the various tools, DB2 commands, DB2admin, and Performance Monitor within R/3 and point out some of the similarities and differences between the output that is displayed.

4.2.2.1 The LIST TABLESPACES Command

One of the easiest ways to obtain information regarding the status of tablespaces within a DB2 database, is to issue the DB2 LIST TABLESPACES command. The syntax for the list tablespaces command is as follows:

```
LIST TABLESPACES [SHOW DETAIL]
```

The partial output from the command is similar to the following:

Tablespaces for Current Database

```

Tablespace ID          = 0
Name                   = SYSCATSPACE 1
Type                   = Database managed space 2
Contents               = Any data 3
State                  = 0x0000
    Detailed explanation:
        Normal
Total pages            = 30000
Useable pages         = 29952 4
Used pages             = 24672
Free pages             = 5280
High water mark (pages) = 25664
Page size (bytes)     = 4096 5
Extent size (pages)   = 32
Prefetch size (pages) = 16 6
Number of containers  = 3

Tablespace ID          = 3
Name                   = PSAPTEMP
Type                   = Database managed space
Contents               = Temporary data
State                  = 0x0000
    Detailed explanation:
        Normal
Total pages            = 16000
Useable pages         = 15968
Used pages             = 64
Free pages             = 15904
High water mark (pages) = 64
Page size (bytes)     = 4096
Extent size (pages)   = 32
Prefetch size (pages) = 16
Number of containers  = 1

Tablespace ID          = 4
Name                   = PSAPBTABD

```

If the SHOW DETAIL option is specified, the command will list all tablespaces for the current database, their name, type, contents, extent size, number of containers, prefetch size, total pages, usable pages, used and free pages, and the tablespace state. Let's explain some of the information displayed with the command:

1. SYSCATSPACE has a tablespace ID of 0 since it is the first tablespace that is created in the database. Remember that SYSCATSPACE is the tablespace that contains the system catalogs for the database.
2. This is a DMS or Database Managed Storage tablespace.
3. The tablespace State is normal as indicated by the hexadecimal number 0x0000. Tablespace states are explained in 4.2.7, "Tablespace States" on page 100.
4. Total pages reflects the number of pages that were defined when the container(s) was created. Notice that there is a difference of 48 pages between total pages and usable pages for the container(s). When a container is created, one page is used per container for a container tag.

Depending on the extent size of the tablespace, the remaining pages are divided into allocations of the extent size. Any other remaining amounts less than the extent size are not used. The total number of pages that can be divided into equal extent size numbers are the usable pages. Used pages are those that are used for creation of the tablespace and data. Sizing is covered in more detail in 4.2.10, "Sizing Considerations for DMS Tablespaces" on page 103.

5. The extent size for this tablespace is 32 4-KB pages. The prefetch size is 16 4-KB pages.
6. There are three containers assigned to this tablespace.

4.2.3 Displaying Tablespace Information - DB2admin

You can also obtain this same information from the DB2admin utility. The path to follow is **SMIT->Applications->DB2admin->Tablespace Management->List Tablespaces**. The output is the same as that displayed in the previous section. If you select **F6** while in DB2admin from the List Tablespaces dialog screen, you'll see that a .ksh file is executed. That file is located in the db2admin subdirectory under the DB2 instance owner's home directory. In that subdirectory, db2admin, you'll find the .ksh files that are executed from the DB2admin SMIT interface. The file that is being executed in this example is sddb6lts.ksh. One of the commands that is executed from the file is the LIST TABLESPACES SHOW DETAIL command within DB2.

4.2.4 Displaying Tablespace Information - R/3 Performance Monitor

From the Performance Monitor in R/3, you can obtain similar information. From the State Part (Figure 42 on page 81) click on the first **Detailed analysis** button. The screen should be similar to the following:

Database Performance Tablespace						
Database analysis Edit Goto Monitor System Help						
<input type="text"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="↵"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="⋮"/>						
Analysis		Sort				
Date/time of this analysis: 11.08.1996 16:06:22						
Tablespace	Type	Current Size (kB)	Free (kB)	%-Used	Container	Status
SYSCATSPACE	DMS	98688	21120	82,37	3	Normal
PSAPTEMP	DMS	256	63616	0,40	1	Normal
PSAPBTABD	DMS	271808	48064	84,97	2	Normal
PSAPDDICD	DMS	41728	8256	83,48	1	Normal
PSAPOOLI	DMS	356000	43936	89,01	2	Normal
PSAPBTABI	DMS	160992	50976	75,95	1	Normal
PSAPUSER1I	DMS	96	49888	0,19	1	Normal
PSAPCLUD	DMS	20736	89088	18,88	1	Normal
PSAPSOURCED	DMS	15872	40960	27,93	1	Normal
PSAPEL30CB	DMS	149120	120832	55,24	2	Normal
PSAPLOADI	DMS	4224	34688	10,86	1	Normal
PSAPCLUJ	DMS	8832	5120	63,30	1	Normal
PSAPDOCUI	DMS	26240	13696	65,71	1	Normal
PSAPSTABD	DMS	294016	225920	56,55	2	Normal
PSAPPROTI	DMS	10368	3584	74,31	1	Normal
PSAPEL30CI	DMS	5824	24128	19,44	1	Normal
PSAPPROTD	DMS	23040	72832	24,03	1	Normal
PSAPLOADD	DMS	8960	58880	13,21	1	Normal
PSAPSOURCEI	DMS	7680	6272	55,05	1	Normal
PSAPUSERID	DMS	192	99776	0,19	1	Normal
PSAPES30CI	DMS	319616	100352	76,10	2	Normal
PSAPSTABI	DMS	189088	150880	55,62	2	Normal
PSAPDOCUD	DMS	69376	20608	77,10	1	Normal
PSAPPOOLD	DMS	429952	0	100,00	2	Normal
PSAPDDICI	DMS	14976	12992	53,55	1	Normal
PSAPES30CB	DMS	1338112	461696	74,35	6	Normal
Total		3.865.792	1.828.160	68	40	

Figure 46. R/3 Database Performance Monitor - Tablespace Analysis

Figure 46 shows some of the same type of information that is displayed with the LIST TABLESPACES command. The name of each tablespace is displayed, its type, the current size in KB, free space in KB, the percentage used, how many containers are assigned to the tablespace and the tablespace state. Here, in R/3, it is called status. If you look at the column Current Size and Free. These columns are expressed in KB. The total of these two columns represents the usable pages displayed in the LIST TABLESPACES command. However, the output from the LIST TABLESPACES command is expressed in 4-KB pages.

One of the options you can perform from the screen shown in Figure 46 is to select a particular tablespace; here we again show SYSCATSPACE. Click on **SYSCATSPACE** and then on the **Analysis** button. The following screen appears:

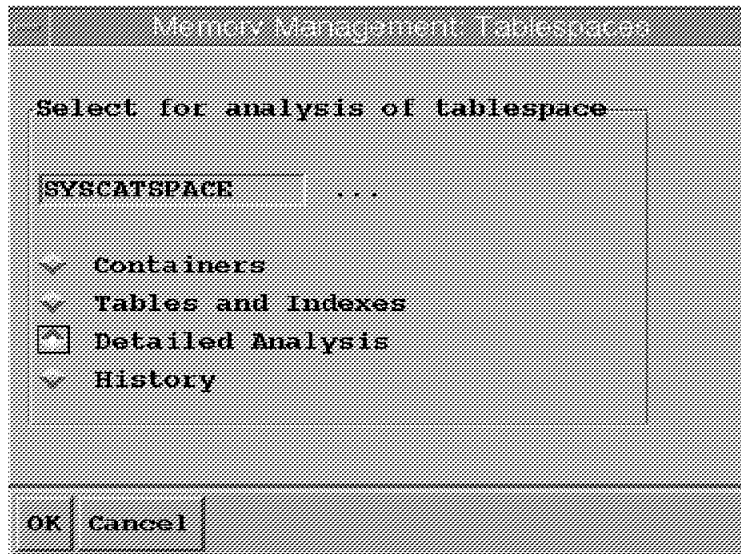


Figure 47. R/3 Database Performance Monitor - Tablespace Selection Pop-Up Menu

Select **Detailed Analysis** and click on **OK**; the following screen appears:

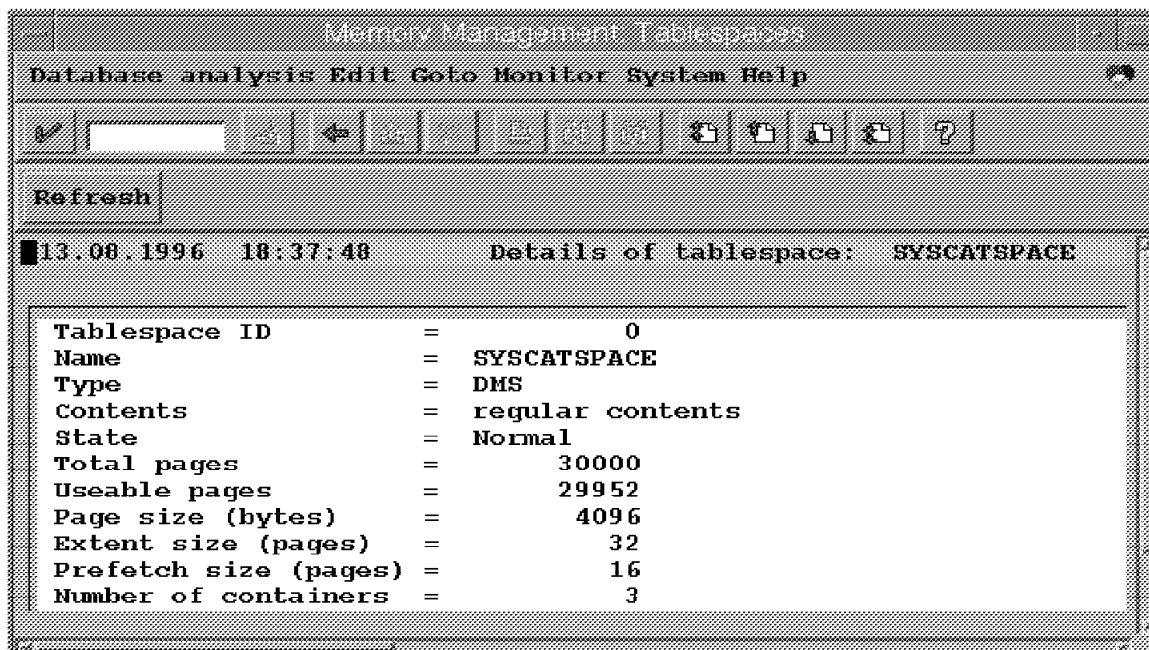


Figure 48. R/3 Database Performance Monitor - Detailed analysis

Figure 48 shows similar information about SYSCATSPACE.

4.2.4.1 Displaying Tablespace Information - Database Director

You also can obtain tablespace information from the Database Director. From the main Database Director screen, here are the steps to take you to the tablespace information: **Database Managers->Database->AUS**. Here, AUS is the <SAPSID> name for R/3 in our installation. It is, by default, our R/3 database name. You'll see Table space listed under the database information. Move your mouse so that Table space is highlighted. Then with the right-most

mouse button, activate the pull-down menu. Select **Open as->Details**. The following screen appears:

Table Spaces - Details View											
Table space	Database	View									Help
DB2AUS AUS											
Table space	Type of Data	Managed By	Status Information	Extent Size	Prefetch Size	Overhead	Transfer Rate	Allocated Size	Size Used	Comment	
PSAPBTAB	Regular	Database	No	16	16	24.1	0.9	50000	39680		
PSAPBTABI	Regular	Database	No	8	16	24.1	0.9	53000	39680		
PSAPCLUD	Regular	Database	No	64	16	24.1	0.9	27500	5184		
PSAPCLUI	Regular	Database	No	32	16	24.1	0.9	3500	2208		
PSAPDDICD	Regular	Database	No	16	16	24.1	0.9	12500	10432		
PSAPDDICI	Regular	Database	No	16	16	24.1	0.9	7000	3744		
PSAPDOCLD	Regular	Database	No	32	16	24.1	0.9	22500	17344		
PSAPDOCLI	Regular	Database	No	32	16	24.1	0.9	10000	6560		
PSAPEL30CD	Regular	Database	No	16	16	24.1	0.9	67500	36272		
PSAPEL30CI	Regular	Database	No	16	16	24.1	0.9	7500	1440		
PSAPES30CD	Regular	Database	No	16	16	24.1	0.9	450000	334486		
PSAPES30CI	Regular	Database	No	16	16	24.1	0.9	105000	78904		
PSAPLOADD	Regular	Database	No	64	16	24.1	0.9	17000	2240		
PSAPLOADI	Regular	Database	No	32	16	24.1	0.9	9750	1056		
PSAPPOOLD	Regular	Database	No	8	16	24.1	0.9	106000	99872		

Figure 49. Database Director - Table Spaces - Detailed View

From Figure 49, you can see all of the tablespaces within the R/3 system. The information that you can find is tablespace name, type of data (regular, long or index), whether the tablespace is DMS or SMS, the extent and prefetch sizes, overhead, transfer rate, how much space was allocated and used and a comment field. This is the same type of information displayed with the LIST TABLESPACES command. The total pages is the amount allocated to the containers when the tablespace is created or additional containers are added to the tablespace. There are three values to consider: the total amount of pages allocated, the amount of pages usable (after removing pages either in creating the tablespace or overhead in creating the objects within the tablespace) and the amount of pages that are available. You can obtain all three of these values with the LIST TABLESPACES SHOW DETAIL command. In Figure 49, the amount that is usable after overhead is removed and the amount that is available is displayed. The amount that was allocated is not displayed here.

4.2.5 Displaying Container Information

One of the main tasks that a SAP database administrator does on a daily basis is monitor the free space within the tablespaces. This becomes especially important in an R/3 production system because the R/3 installation consists of 26 tablespaces. Monitoring the amount of free space is crucial to a SAP DBA. The most heavily used tablespaces in R/3 will probably be PSAPBTAB, PSAPCLU and PSAPSTAB. Besides the LIST TABLESPACES command, the other DB2 command that displays storage information is the LIST TABLESPACE CONTAINERS command. We look at how these two commands can assist in tablespace management.

Depending on how you define containers, you also can obtain container information from AIX. If your containers for DMS tablespaces are files that reside in an AIX JFS file system, you can use the `df` command to obtain information about the amount of space available in your file systems. The following is a partial output from the command.

Filesystem	512-blocks	Free	%Used	Iused	%Iused	Mounted on
/dev/db2AUS	163840	25792	85%	402	2%	/db2/AUS
/dev/sapdata4AUS	4096000	2569904	38%	22	1%	/db2/AUS/sapdata4
/dev/sapdata6AUS	4096000	363744	92%	22	1%	/db2/AUS/sapdata6
/dev/sapdata1AUS	2048000	714320	66%	23	1%	/db2/AUS/sapdata1
/dev/sapdata3AUS	2048000	668256	68%	24	1%	/db2/AUS/sapdata3
/dev/logarchiveAUS	2048000	1980200	4%	53	1%	/db2/AUS/log_archive/AUS
/dev/ursapAUS	737280	690456	7%	103	1%	/usr/sap/AUS
/dev/ursaptrans	212992	192888	10%	42	1%	/usr/sap/trans
/dev/logdirAUS	458752	242576	48%	54	1%	/db2/AUS/log_dir
/dev/sapmntAUS	622592	278344	56%	921	2%	/sapmnt/AUS
/dev/tmpinstallAUS	49152	1344	98%	49	1%	/tmp/install
/dev/sapdata2AUS	2048000	584208	72%	22	1%	/db2/AUS/sapdata2
/dev/sapdata5AUS	2703360	410168	85%	25	1%	/db2/AUS/sapdata5
/dev/sapdata7aus	1638400	946224	43%	18	1%	/db2/AUS/sapdata7
/dev/lv00	1703936	1092720	36%	17	1%	/backup1

From the `df` command, you can see the logical volume name and the file system associated with the logical volume. The size of the file system is displayed in 512 KB blocks. You see the percentage of space that is free and the percentage used. This gives you a general idea about the space available in the file system. However, unless you have only one tablespace assigned to a file system, you will need to know the name of the tablespace whose containers are running out of space. The AIX `df` command does not monitor tablespace containers within DB2.

4.2.5.1 The LIST TABLESPACE CONTAINERS Command

The `list tablespace containers` command displays all the containers for a specific tablespace. It shows information about the name, type, and size (DMS only) of the containers in the tablespace. The syntax of the command is as follows:

```
LIST TABLESPACE CONTAINERS FOR TABLESPACE-ID [SHOW DETAIL]
```

As you can see from the command, you must supply a container ID as part of the command parameters. If we want to look at the containers for SYSCATSPACE, the command would be:

```
LIST TABLESPACE CONTAINERS FOR 0 SHOW DETAIL
```

The output is similar to the following:

Tablespace Containers for Tablespace 0

```
Container ID          = 0 1
Name                  = /db2/AJS/sapdata1/SYSCATSPACE.container001 2
Type                  = File
Total pages           = 10000 3
Useable pages         = 9984
Accessible            = Yes
Container ID          = 1
Name                  = /db2/AJS/sapdata1/SYSCATSPACE.container002
Type                  = File
Total pages           = 10000
Useable pages         = 9984
Accessible            = Yes
Container ID          = 2 4
Name                  = /db2/AJS/sapdata1/SYSCATSPACE.container003
Type                  = File
Total pages           = 10000
Useable pages         = 9984
Accessible            = Yes
```

The numbers from the output of the command are explained as follows:

1. Every container has a *unique ID* associated with it. Every container assigned to a tablespace has a unique ID associated with it. For tablespace 0, SYSCATSPACE, there are three containers whose IDs are 0, 1, and 2.
2. The type of R/3 container is a file. Remember that for the current implementation of R/3 within DB2 for AIX, all DMS tablespace containers are implemented as files. Those files are part of AIX file systems. The naming convention for container files is discussed in 2.3.5, "R/3 Container and Tablespace Names" on page 33. Here the container name is SYSCATSPACE.container001 for the first container.
3. The size allocated to each file is 10000 4-KB pages. If you recall the output of the LIST TABLESPACES command (see 4.2.2.1, "The LIST TABLESPACES Command" on page 87), the total allocated number of pages assigned to SYSCATSPACE was 30000. This is evenly distributed among the three containers. The usable pages is the amount available to the tablespace after the container is allocated. A page, depending on the extent size, is used for every container for the container tag. Remember this number; Usable pages, does not indicate how much space is free in the container.
4. Notice how the numbering of container IDs does not match the R/3 numbering of containers. DB2 gives each container a *unique ID* starting at 0 and incremented sequentially by 1. R/3 assigns *container file names* starting with container 001, also incremented sequentially by 1, for example 002, 003,

4.2.5.2 Displaying Container Information - DB2admin

The exact same information is displayed from DB2admin as with the LIST TABLESPACE CONTAINERS command. To access the List Tablespace Containers dialog screen, the path is **SMIT->Applications->Table Space Management->List Tablespace Containers**.

The database name is filled in because, at this time, only one database is allowed in R/3 DB2 installations. However, here you must supply the name of

the tablespace for which you want container information, not the tablespace ID. The file that is executed is in the db2admin subdirectory of the instance owner. The file name is sddb6ltc.ksh.

4.2.5.3 Displaying Container Information - R/3 Performance Monitor

To access the same container information using the R/3 Performance Monitor, we will start from the Tablespace Analysis screen that was displayed in Figure 46 on page 90. To get to the Tablespace Analysis screen from the Performance Monitor, click on the first **Detailed analysis** button. Next select the tablespace whose containers you want more information about. Double-click on **SYSCATSPACE**, and the following screen should appear:

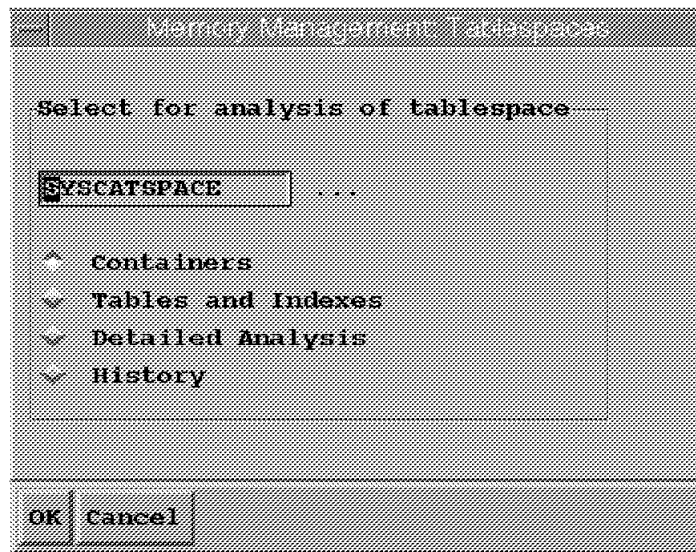


Figure 50. R/3 Database Performance Monitor - Obtaining Container Information

From Figure 50, we select **Containers** and press **OK**. The following screen appears:

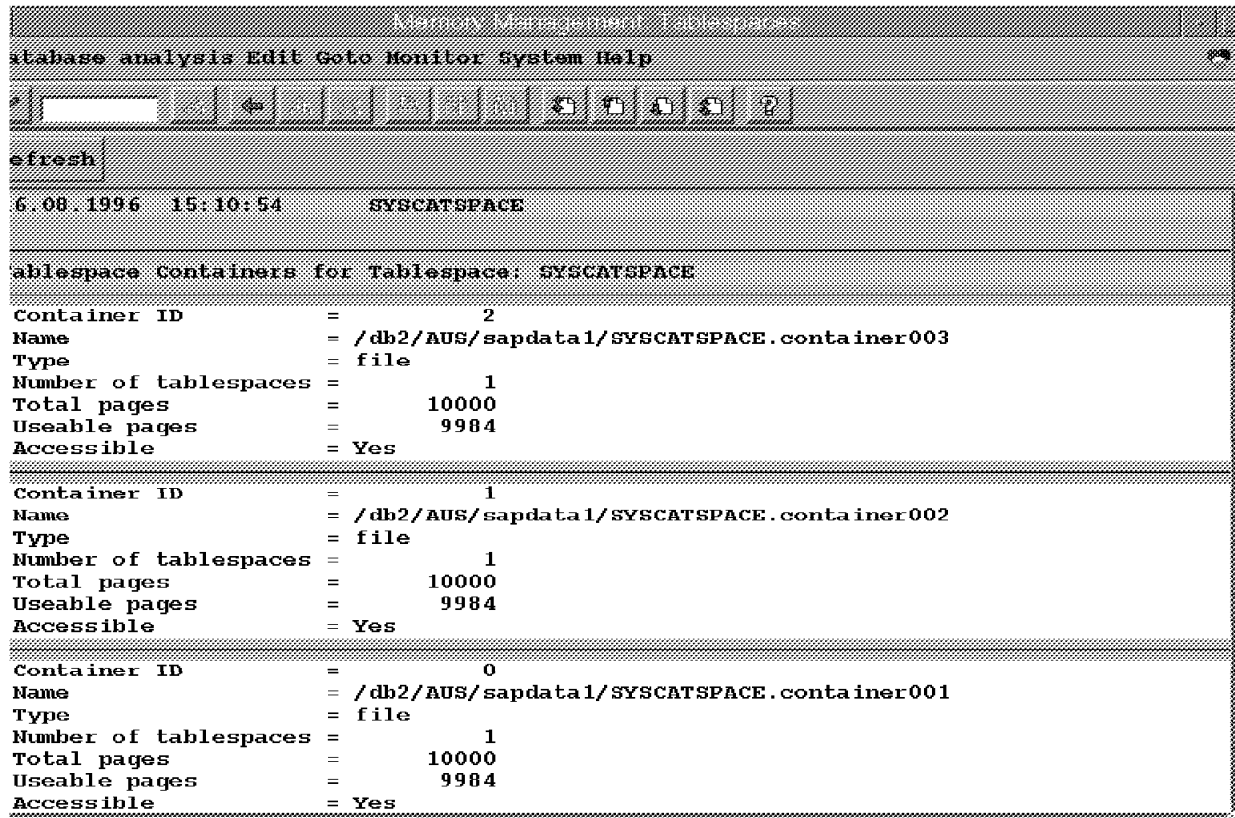


Figure 51. R/3 Performance Monitor - Tablespace Container Information

The information displayed is very similar to what we have seen in this section.

4.2.5.4 Displaying Container Information - Database Director

From the window displayed in Figure 49 on page 92, we can obtain container information about a specific tablespace. After selecting the **Tablespace -> Detail View** window, select the tablespace that you want container information on. We show SYSCATSPACE to be consistent with our example. Again, with the right mouse button, select **Open as -> Containers** from the menu. If you check the bottom of the current window, you should see a message indicating your next window to appear. It will be similar to the following window:

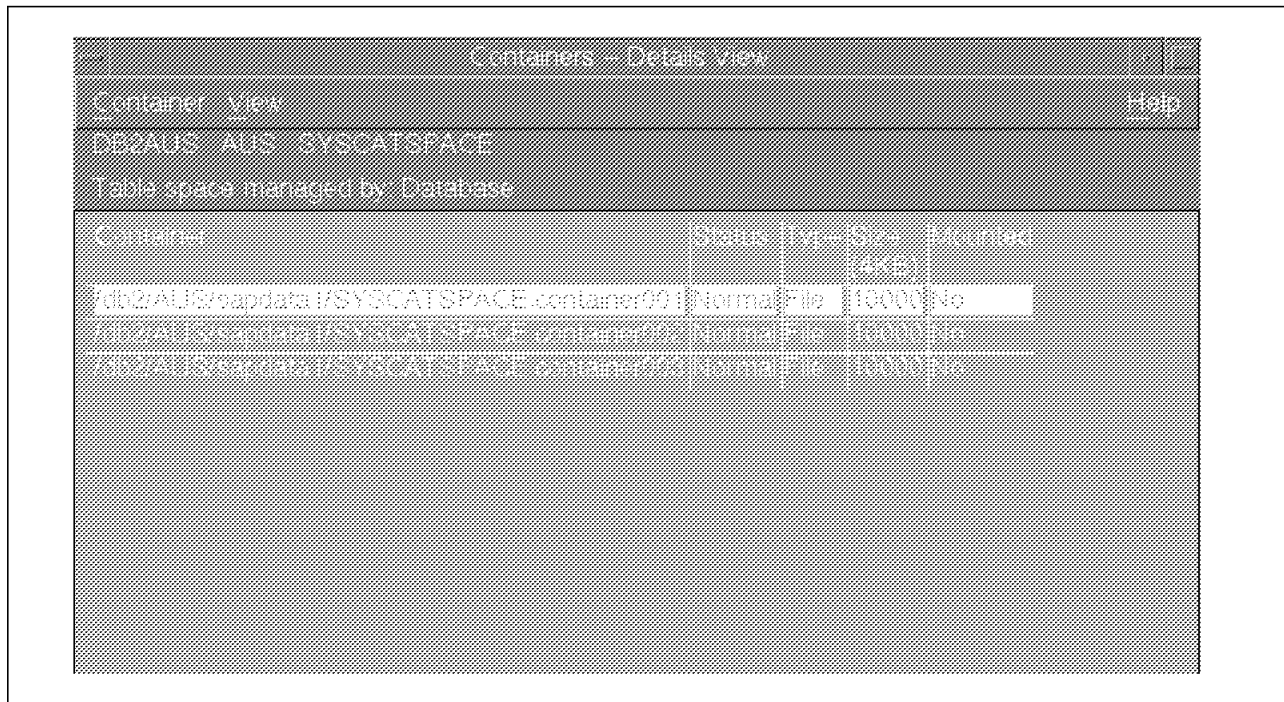


Figure 52. Database Director - Containers - Detailed View

One of the options that is available from the window shown in Figure 52 is the ability to add a container. Remember that only DMS tablespaces allow you to dynamically add a container to an existing tablespace.

4.2.5.5 Displaying Container Information - The R/3 Alert Monitor

The Alert Monitor shows critical parameter values for the database, several operating system values and network parameters. You can also use it to monitor tablespaces and containers within those tablespaces.

You need to first start the R/3 Alert Monitor. This was discussed in 4.1.2.1, "The R/3 Alert Monitor" on page 81. From the main or overview screen of the Alert Monitor, you can gather tablespace statistics. The last entry, shown in Figure 43 on page 82, contained an arrow pointing to the item "Space statistics checked on..." This brings up a screen similar to the following:

SAP Performance Monitor - DB2 for AIX			
Graphic Edit Goto Extras Help			
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
Fit On/Off			
AUS - Top 20 Space statistics			
Tablespace	Cur. Size(kb)	Free(%)	Chg/day(kb)
PSAPPOLED	399.488	6	13.167
PSAPDDICD	41.728	17	0
PSAPPOLE1	330.720	17	2.513
PSAPBTABD	266.752	17	561
SYSCATSSPACE	98.816	10	0
PSAPDDGUD	69.376	23	0
PSAPES30CI	319.616	24	10
PSAPBTABI	150.720	25	234
PSAPPROT1	10.368	26	0
PSAPES30CD	1.337.984	26	20
PSAPDDGHI	26.240	34	0
PSAPGLUI	8.832	37	0
PSAP50URGEI	7.680	45	0
PSAPDDICI	14.976	46	0
PSAPSTABI	181.952	46	549
PSAPEL30CD	145.024	46	561
PSAPSTABD	274.016	47	1.407
PSAP50URGED	15.872	72	0
PSAPPROTD	22.656	76	130
PSAPGLID	20.736	81	0

AUS 30C R/3 R/3

Figure 53. R/3 Alert Monitor - Monitoring Containers and Tablespaces

Figure 53 shows you the 20 most active tablespaces. The R/3 Alert Monitor gathers information from the DB2 Database Monitor and DB2 APIs. It collects not only the current size and free space but also the amount of change that has occurred on a daily basis. This can assist an administrator in anticipating storage needs in tablespaces.

The scheduling of the monitoring is controlled by different programs within R/3. The control of the collection schedule for the Performance Monitor is called COLLECTOR_FOR_PERFORMANCEMONITOR. There is a batch job within R/3 called RSORATDB. RSORATDB is scheduled in the TCOLL table. The default collection period is twice a day, seven days a week. You can change the schedule for collection of Performance Monitor data. To change the schedule for collecting this data, modify the TCOLL table in R/3.

There are two ways to update the data collected. One way is to schedule the collection job through the table TCOLL. The other way is to refresh the data online.

To refresh the data on-line, starting from Figure 46 on page 90, select **Database->Refresh**. You will receive a confirmation pop-up window warning you that the refresh may take a long time.

4.2.6 Monitoring Active Applications

This section looks at two of the commands within DB2 that a SAP database administrator can use to control users accessing the database. Use caution when monitoring commands through DB2. Make sure that you understand each process and job that you want to monitor. For more information on processes within DB2 and R/3, please see 3.3.4, “DB2 for AIX and R/3 Processes” on page 66. We now discuss the LIST APPLICATIONS and the FORCE commands.

4.2.6.1 The LIST APPLICATIONS Command

The LIST APPLICATIONS command displays information about the applications executing within the DB2 instance. The type of information you'll receive is the name of the application, the Auth ID, the Agent ID, the Application ID, and the database that the application is accessing.

The complete syntax of the command is shown below:

```
LIST APPLICATIONS [FOR DATABASE database-alias] {SHOW DETAIL}
```

The DB2 can use the output of this command as a tool for controlling users or problem determination. A common usage of the command is in conjunction with the FORCE command.

4.2.6.2 The FORCE Command

The FORCE command forces local or remote users or applications off the DB2 database. The syntax of the FORCE command is as follows:

```
FORCE APPLICATION {ALL | agent-id [{,agent-id} ...] ) } [MODE ASYNC]
```

A database administrator may use the LIST APPLICATIONS command to find a particular Agent ID and then force that user off the system. Figure 54 on page 100 shows these two commands.

```

db2 => list applications
Auth Id  Application Name  Agent Id  Application Id  DB Name
-----
DB2AUS  db2bp_32         24034    *LOCAL.db2aus.960820170925  AUS
DB2AUS  db2bp_32         28552    *LOCAL.db2aus.960820184620  AUS
DB2AUS  db2bp_32         34040    *LOCAL.db2aus.960820184846  AUS

db2 => force application ( 34040 )
DB20000I The FORCE APPLICATION command completed successfully.
DB21024I This command is asynchronous and may not be effective immediately.

db2 => list applications
Auth Id  Application Name  Agent Id  Application Id  DB Name
-----
DB2AUS  db2bp_32         24034    *LOCAL.db2aus.960820170925  AUS
DB2AUS  db2bp_32         28552    *LOCAL.db2aus.960820184620  AUS

```

Figure 54. Controlling Users within DB2

From Figure 54, you can see that there are three applications executing. All are from the same user username. They all access the same database, AUS. To remove the third application, the FORCE APPLICATION command specifying the Agent ID of the application to terminate is supplied with the command parameters.

4.2.7 Tablespace States

DB2 maintains information about the states of tablespaces and will not allow access to table spaces that are not in a "normal state". Tablespace states are expressed in hexadecimal numbers. Sometimes, a tablespace may have more than one state associated with it. This will result in a combined hexadecimal number. All of the states associated with tablespaces can be found in the sqlutil.h file. This file is located in /usr/lpp/db2_02_01/include directory.

To view the tablespace state, issue the LIST TABLESPACES command. Alternatively, you can use a utility called db2tbst that is shipped with the product. Typing db2tbst with the hexadecimal number from the command line will evaluate the tablespace state. The db2tbst utility can be found in the sqllib directory under the misc subdirectory of the instance owner.

A tablespace is placed in a non-normal state during load, backup and recovery operations, or if placed in a quiesced condition via the QUIESCE TABLESPACE command. Inconsistencies in data across tablespaces are avoided by restricting access to the tablespace. A list of some of the possible tablespace states are as follows:

Normal (0x0000)—Access to the tablespace is allowed.

Quiesce related states—A tablespace is in any of these states when a quiesce request is received by DB2. Any of these states must be explicitly reset.

(0x0001)—Quiesced share

(0x0002)—Quiesced update

(0x0004)—Quiesced exclusive

(0x0008) - Load Pending—A tablespace is in a Load Pending state when a table is being loaded or when the LOAD utility fails. After correcting the problem that caused the failure, invoking the LOAD utility with the RESTART option will continue the load process.

(0x0010) - Delete Pending—A tablespace is in a Delete Pending state when a table stored in the tablespace has been successfully loaded and the LOAD utility is processing the delete phase to delete rows with duplicate keys.

(0x0020) - Backup Pending—A tablespace will be in this state when a table contained in this tablespace has been successfully loaded using the LOAD utility. Access to the tablespace is not allowed until a backup of the database or tablespace is taken.

(0x0040) - Roll Forward in Progress—A roll forward of the log files is taking place.

(0x0080) - Roll Forward Pending—A tablespace or database is in a Roll Forward Pending state when as part of the recovery process, a backup has been restored. The backup can be an online database or online/off-line tablespace backup copy.

(0x01000) - Restore Pending—A tablespace will be in this state when you perform a restore of a tablespace or a database or a LOAD with the TERMINATE option. The tablespace or database will have to be recovered from a backup. If the backup being restored is a tablespace backup, the database will have to be rolled forward to the end of the log files.

4.2.8 System Catalogs in DB2

Information about tables and tablespaces is kept in the system catalogs. The database administrator uses DB2 commands to list, alter, or create tablespaces and tables, but this information can also be obtained by querying the system catalogs. These tables also contain information about the definitions of the database objects and security information about the type of access users have to these objects.

The system catalogs are known by a schema name. A schema name is a fully qualified table name in the form of “schemaname.tablename”. There are schema names that are reserved within DB2:

- *SYSIBM*
- *SYSCAT*
- *SYSSTAT*

The *SYSIBM* schema refers to the base tables that comprise the system catalog. You should not use a schema of *SYSIBM* for any of your user tables. In general, avoid using any schema with the prefix *sys*. Over time, these *SYSIBM* tables have evolved from release to release of the product. They include the base catalogs, built-in types, and built-in functions. Views of these tables have now been introduced to allow an end-user to query the contents of the catalog tables by using more meaningful names.

The database manager creates and maintains two sets of catalog views. All of the system catalog views are created when a database is created with the

CREATE DATABASE command. The catalog views cannot be explicitly created or dropped. The views are within the SYSCAT schema, and SELECT privilege on all views is granted to PUBLIC by default. A second set of views, formed from a subset of those within the SYSCAT schema, contain statistical information used by the optimizer. The following table shows the set of catalog views.

Catalog View	Description
SYSCAT.CHECKS	check constraints
SYSCAT.COLCHECKS	columns referenced by check constraints
SYSCAT.COLDIST	detailed column statistics
SYSCAT.COLUMNS	columns
SYSCAT.CONSTDEP	constraint dependencies
SYSCAT.DBAUTH	authorities on database
SYSCAT.EVENTS	events currently monitored
SYSCAT.EVENTMONITORS	Event Monitor definitions
SYSCAT.FUNCPARMS	function parameters
SYSCAT.FUNCTIONS	user-defined functions
SYSCAT.INDEXAUTH	index privileges
SYSCAT.INDEXES	indexes
SYSCAT.KEYCOLUSE	columns used in keys
SYSCAT.PACKAGEAUTH	package privileges
SYSCAT.PACKAGEDEP	package dependencies
SYSCAT.PACKAGES	packages
SYSCAT.REFERENCES	referential constraints
SYSCAT.STATEMENTS	statements in packages
SYSCAT.TBAUTH	table privileges
SYSCAT.TABCONST	table constraints
SYSCAT.TABLES	tables
SYSCAT.TABLESPACES	tables paces
SYSCAT.TRIGDEP	trigger dependencies
SYSCAT.TRIGGERS	triggers
SYSCAT.VIEWDEP	view dependencies
SYSCAT.VIEWS	views

Table 3. DB2 System CATALOG Views

The views within the SYSSTAT schema contain some columns that can be updated. This table shows the views in the system catalog tables that are updatable.

Catalog View	Description
SYSSTAT.COLDIST	detailed column statistics
SYSSTAT.COLUMNS	columns
SYSSTAT.FUNCTIONS	user-defined functions
SYSSTAT.INDEXES	indexes
SYSSTAT.TABLES	tables

Table 4. DB2 Updatable System CATALOG Views

There are three catalog view directly related to table spaces, tables and indexes:

- *SYSCAT.TABLESPACES*
- *SYSCAT.TABLES*
- *SYSCAT.INDEXES*

SYSCAT.TABLESPACES contains a row for each tablespace. Each row maintains information about the name of the tablespace, the tablespace ID, table space type, type of data this tablespace stores, extent size, and prefetch size.

4.2.9 Understanding and Changing the Size of DMS Tablespaces

This section looks at how to size a DMS tablespace and then how to relate that to the containers that you will be creating for the tablespace. We are only going to deal with DMS tablespaces here for two reasons:

1. R/3 tablespaces are only implemented as DMS tablespaces in the default R/3 installation.
2. DMS tablespaces are pre-allocated, whereas SMS tablespaces are not.

The sizing of tables within tablespaces whether DMS or SMS is the same. For more information on how to determine the size of your tables, consult the *DB2 Administration Guide*.

Using DMS tablespaces, you have two options that allow you to modify containers:

1. You can add a container to an existing DMS tablespace.
2. You can change the container definition. This is the only way that you can reduce the size of containers within a tablespace. You can also change the characteristics of containers.

4.2.10 Sizing Considerations for DMS Tablespaces

There is a certain amount of overhead required in a DMS tablespace. The space requirements break down as follows:

- One page per container - This is used for a container tag.
- Three extents per tablespace of overhead for creating the tablespace.
- One initial extent per object in the tablespace. One extent is allocated for each object that the table has defined in it. You can and should request more. These numbers are the minimum requirements. The other extent is required for what is referred to as the extent map. You need one extent for

the extent map for each object defined in the table. The extent map is used in locating data in the tablespace.

- If the tablespace will contain indexes, Long Field Data or LOBs (Large Binary Objects) the tablespace object requires extents for each type.

When a DMS tablespace is created, there must be enough space in the containers for five extents of the tablespace; otherwise the tablespace will not be created. Once DB2 removes the one page for the container tag, it will make the number of usable pages in a multiple of the extent size. So, for optimal space utilization, allocate:

$$(n * \text{extentsize}) + 1$$

pages in each container so that no space is wasted. (In the formula above, n is "some number"; so the size in the containers is a multiple of the extent size plus one.)

Let's go through an example that looks at the minimum amount of storage required to create a DMS tablespace. Suppose you have two containers allocated for the tablespace and the extent size is set to 16. The example calculates the minimum number of pages to create a DMS tablespace with three table objects in it. One of those objects is an index.

Calculating Minimum Number of Pages for a DMS Tablespace

2 pages (One page for each container tag) +
3 * 16 (tablespace creation overhead) +
2 * 16 * 3 (allowance for each initial object created in the tablespace) +
2 * 16 (Extra allowance for creation of index.)
Total of 177 pages

Let's look at a tablespace that we have created using the following command:
db2 "create tablespace dms01 managed by database using (file
'/home/db2aus/db/dms01' 100, file '/home/db2aus/db/dms02' 100) extentsize 16"

Now let's look at the space that was allocated for the tablespace containers:

Tablespace ID	= 3
Name	= DMS01
Type	= Database managed space
Contents	= Any data
State	= 0x0000
Detailed explanation:	
Normal	
Total pages	= 200
Useable pages	= 192
Used pages	= 48
Free pages	= 144
High water mark (pages)	= 48
Page size (bytes)	= 4096
Extent size (pages)	= 16
Prefetch size (pages)	= 32
Number of containers	= 2

Each container was allocated 100 pages. Two pages were used for container tags. A page is 4 KB. That leaves 192 usable pages. Three extents are used for overhead. That leaves 144 pages remaining. Next, let's create a table. The table has three objects in it; one of them is an index. You probably would place the indexes in a separate DMS tablespace. We are not showing the Data Definition Language (DDL) here; this is an exercise in how containers in DMS tablespace use pages.

If we issue the same command to show how pages are used (LIST TABLESPACES SHOW DETAIL), the output is the following:

Tablespace ID	= 3
Name	= DMS01
Type	= Database managed space
Contents	= Any data
State	= 0x0000
Detailed explanation:	
Normal	
Total pages	= 200
Useable pages	= 192
Used pages	= 112
Free pages	= 80
High water mark (pages)	= 112
Page size (bytes)	= 4096
Extent size (pages)	= 16
Prefetch size (pages)	= 32
Number of containers	= 2

Remember that after we created the tablespace and its containers, we had 144 pages available. As we can see, we have used 112. Let's break this down. Three extents were used for tablespace creation (48 pages). Then each object in that is created in the tablespace requires two initial extents. There was one object (32 pages). Finally, two extra extents are required if the tablespace will contain non-regular table data (32 pages). So, before we have any data in our tables in the tablespace, we have used 112 pages.

4.3 Altering Tablespaces

This section discusses the ways that you can change the containers in tablespaces. You can redefine the characteristics of tablespaces, including size and number of containers, location and type of containers, and characteristics such as the prefetch size, overhead, and transfer rate.

You can dynamically add a container to a DMS tablespace. This is only possible with containers in DMS tablespaces. The statement in DB2 to do so is ALTER TABLESPACE. There are different ways that you can execute this utility besides the command line in your DB2 R/3 environment. First, let's look at the concept.

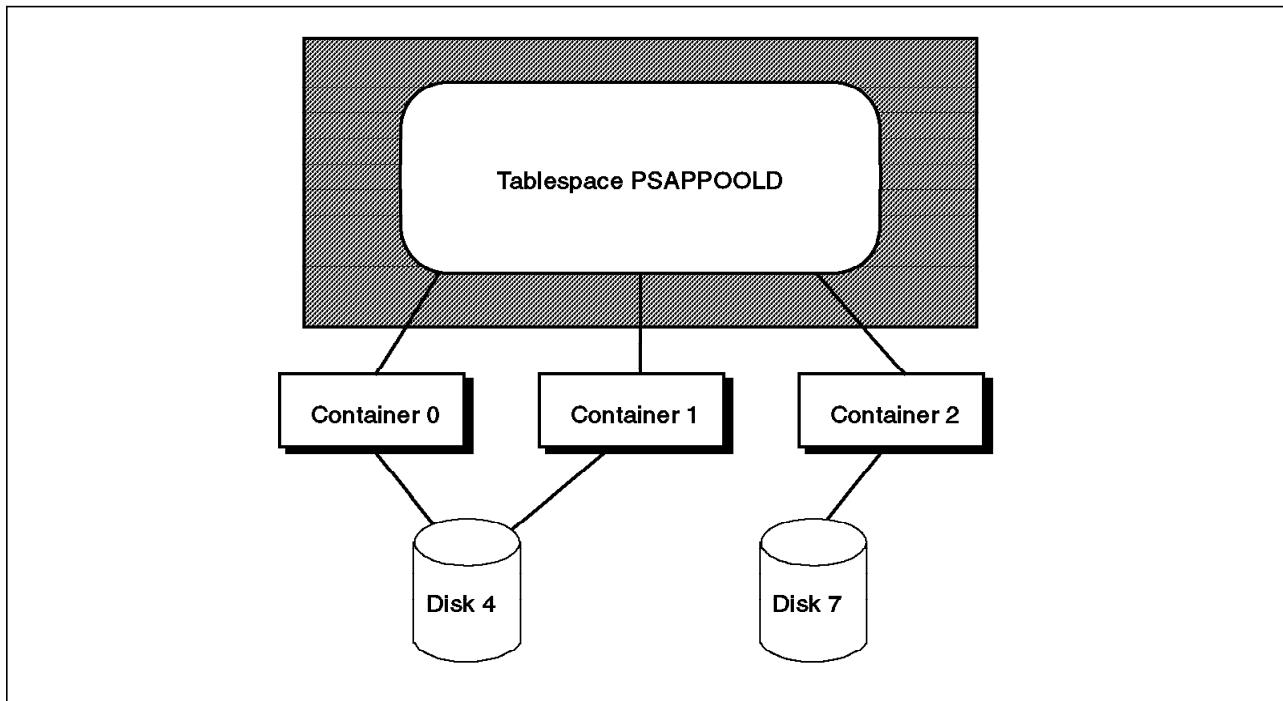


Figure 55. Adding a Container in a DMS Tablespace

You can increase the size of a DMS tablespace by adding one or more containers. Once the new container is added, the data will automatically be redistributed. As soon as the COMMIT operation is done for the transaction involving the adding of the container, the data is rebalanced among the containers. Figure 55 shows the tablespace PSAPPOOLD after a third container, Container 2, has been added.

You must be able to connect to the database to initiate the rebalancing process. There is a DB2 process created, db2rebal. You can check for the status of the process by entering the following command from the AIX command prompt:

```
ps -ef | grep db2rebal
```

While the rebalancing process is running, you are still able to access your R/3 system. However, the tablespace that is being rebalanced (in our example, PSAPBTABD) is unavailable until the transaction is completed.

There are a number of utilities from which you can invoke the ALTER TABLESPACE command:

1. DB2admin

2. Database Director
3. Command Line Processor

SAP recommends the DB2admin utility when possible. We now look at how this is done when using DB2admin.

4.3.1 Adding a Container to a DMS Tablespace

In this section, we add a container, PSAPPOOLD, to a DMS tablespace in R/3. There are a few preparation steps. Check the file systems in which the tablespace resides for freespace for the new container. Remember that the database performs best having containers of the same size within one tablespace.

If your tablespace containers will now be on different physical disks, you can take advantage of prefetching. This allows the database manager to access the data in parallel. It is one of several of the parameters that may be changed after the tablespace is created. You cannot change the extent size. SAP uses its own naming conventions for tablespaces and tablespace containers. Be sure that you are consistent. (See 2.3.5, "R/3 Container and Tablespace Names" on page 33 for more detail.)

First, we can use the R/3 Database Performance Tool to check the current condition of the containers in tablespace PSAPPOOLD. From the State Part screen (see Figure 42 on page 81) we can click on **Detailed Analysis** for a tablespace.

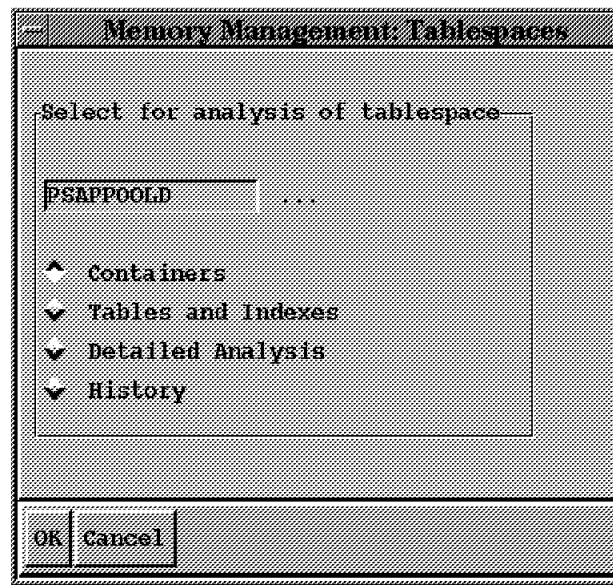


Figure 56. Checking Tablespace Containers Before Adding a Container

If we select **Containers** and click on **OK**, the following screen appears:

```

tput:
-----
Tablespace Containers for Tablespace 25
Container ID          = 0
Name                 = /db2/AUS/sapdata5/PSAPPOOLD.container001
Type                 = File
Total pages          = 53750
Usable pages         = 53744
Accessible           = Yes
Container ID          = 1
Name                 = /db2/AUS/sapdata5/PSAPPOOLD.container002
Type                 = File
Total pages          = 53750
Usable pages         = 53744
Accessible           = Yes

```

Figure 57. Container Details Before Adding Container

The Detailed Analysis option for tablespace containers is shown in Figure 57. There are two file containers located in /db2/AUS/sapdata5. Each file was allocated 75750 pages (approximately 215 MB). Unfortunately, there is not enough space in the file system to create another container (not shown here).

Although our R/3 system is running, we add the container. The best time to add a container is when there is little or no activity to the database. If R/3 is not running, we need to make sure that the database manager is running so we can get a connection to the database to perform the alter operation.

To alter a tablespace using DB2admin, use the following SMIT path:
Applications->DB2admin for R/3: IBM DB2 for AIX Administration
Utilities->Table Space Management->ALTER Table Space.

Select the R/3 Database and the tablespace to alter. In our example, we will add a container to the PSAPPOOLD tablespace:

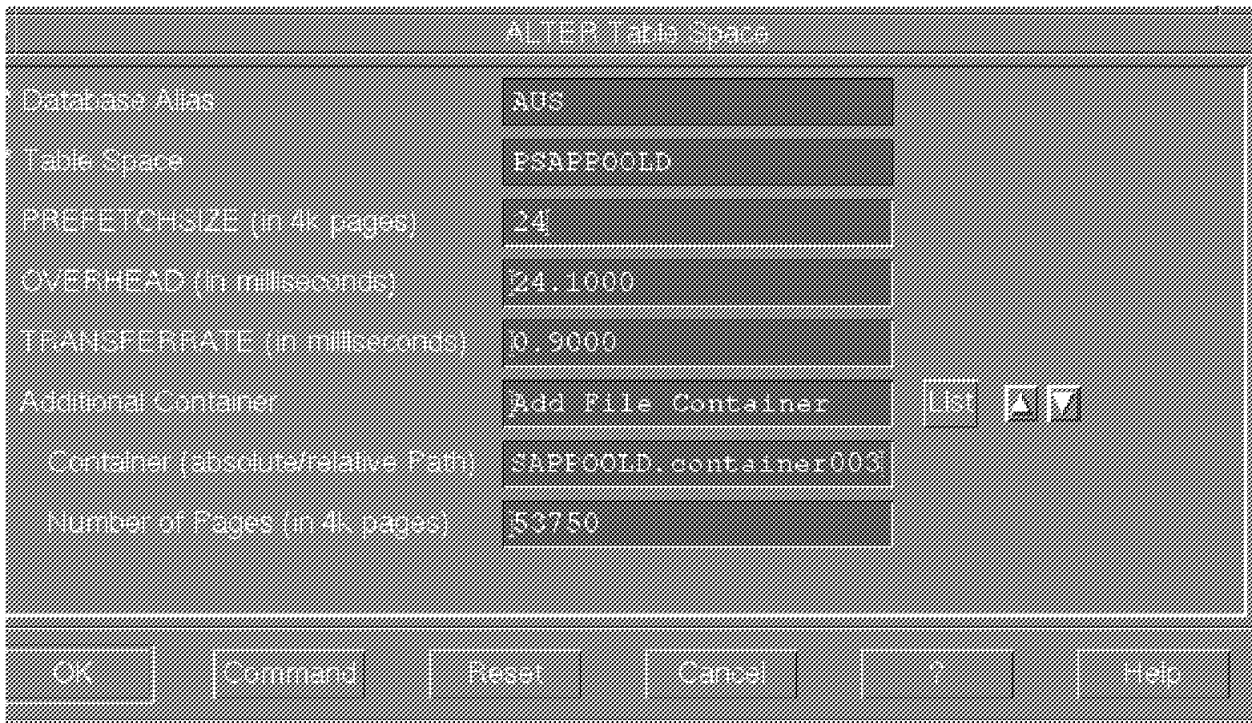


Figure 58. DB2admin Tool - Changing Prefetch Size when Adding Container

In the Additional Container field, you must change the option **Do not add a Container to Add File Container**, and fill in the path for the container. In our example, the path was /db2/AUS/sapdata3/PSAPPOOLD.container003. The size of the container is already filled in by the DB2admin tool. It is recommended to keep the containers the same size for performance reasons.

One of the parameters we can change is the prefetch size. The prefetch size was 16 4-KB pages. Since we are adding another container that happens to be on another disk, we'd like to take advantage of prefetching. Here we have increase the parameter to 24 4-KB pages. For more information about prefetching, see 2.6.2, "Prefetching in DB2" on page 55.

4.3.2 Redefining the Containers in a Tablespace (Redirected Restore)

You can redefine the containers in your tablespace. You can change the size and number of containers, the type of container, and its location. You can also change the prefetch size, overhead rate, and transfer rate. You CANNOT change the type of tablespace. When a tablespace is created as DMS or SMS, the only way to change the tablespace type, is by dropping and re-creating it.

In order to change characteristics (other than type of tablespace), you must use the Database Director. Specifically, you must perform a "Redirected Restore". This means that you must have a valid online/offline tablespace backup. Using this function of the Database Director, you are able to add a new container, change a container name, change its size and/or the path of existing containers or remove the container.

Here are some scenarios that show why you might want to consider using this option:

- You did a backup of your database or one or more tablespaces. However, you find that the file containers cannot be accessed after the backup.
- Your backup was taken from a development or test R/3 system, and you want to restore it in your production R/3 system.
- You created a tablespace for your R/3 system, and the tablespace is not going to be used any more. This can happen if you are working on a system that was migrated from an Oracle or Informix database to DB2 using the R3load utility (R3loadctl) for data structures of Version 3.0D. The tablespaces PSAPEL30CD/I and PSAPES30CD/I will not be used by the migrated system. However, the data is still needed and will be loaded into other tablespaces whose size had to be increased during the migration.

Here are tasks that you need to do before attempting the Redirected Restore:

1. If not yet done, perform a backup of the database or the tablespace that will be affected.
2. Check for freespace in the appropriate file systems if you are going to add container or increase the size of the containers.

We will cover the Redirected Restore option in more detail in 5.4, “Performing a Redirected Restore” on page 221.

4.4 Data Maintenance

The physical distribution of the data stored in a table may improve or decrease the performance of R/3. The way the data is stored in a table is affected by the update, insert and delete operations. For example, a delete operation will leave empty pages of data that may not be reused later. Also, updates done to variable-length columns may result in the new column value not fitting in the same data page. This can cause the row to be moved to a different location. This also will produce internal gaps or unused space in the tables. As a consequence, DB2 will read more physical pages to retrieve the same information stored in the tables.

These scenarios are almost unavoidable. However, as the SAP database administrator, you can use the data maintenance commands provided in DB2 to optimize the physical distribution of the data stored in your tables.

There are three related utilities or commands that help you organize the data in your tables. These commands are:

- REORGCHK
- REORG
- RUNSTATS

This section first introduces the DB2 utilities and commands. Then, we look at the utilities and tools within R/3 and DB2 that assist in checking for physical distribution and how to carry out a redistribution. As discussed throughout this chapter, we use the R/3 interfaces, the DB2admin utility, the DB2 commands and utilities, and the Database Director.

4.4.1 Analyzing Data's Physical Organization: REORGCHK

We have said that update, insert and delete operations may produce internal gaps in your tables. How can you know the physical organization of your tables or indexes? How can you know how much space is currently being used and how much is free? Questions like these are answered by the REORGCHK command. This utility is used to analyze the system catalog tables and gather information about the physical organization of your tables or indexes.

The REORGCHK command uses the RUNSTATS command to collect the statistics about the space allocation of your tables and indexes.

With the information collected from the system catalog tables, the REORGCHK command displays the space allocation characteristics of your tables and indexes. The command uses six formulas to help you decide if your tables and indexes require a physical reorganization.

These formulas are general recommendations that show the relationship between the allocated space and the space that is being used for the data in your tables. Three formulas are used for tables, and three are used for indexes.

It is recommended that you establish a data maintenance policy to ensure that your tables are correctly using disk space. If you don't, you may discover that your applications will start to suffer performance degradation. This may be caused by the poor physical organization of your data, so before this happens, it is better to do preventive maintenance on your tables.

When you perform many updates, deletes, and inserts on a table, your table and index data will increase, but not only with the new data. Not every space gained by a delete or update request can be used. Table and index data get fragmented. The following is an example of using the REORGCHK command on one of the R/3 tables:

```
db2 reorgchk on table sapr3.moni
```

We now use it to check the physical organization of the sapr3.moni table.

Doing RUNSTATS

Table statistics:

F1: $100 * \text{OVERFLOW} / \text{CARD} < 5$
F2: $100 * \text{TSIZE} / ((\text{FPAGES} - 1) * 4020) > 70$
F3: $100 * \text{NPAGES} / \text{FPAGES} > 80$

CREATOR	NAME	CARD	OV	NP	FP	TSIZE	F1	F2	F3	REORG
SAPR3	MONI	840	43	295	334	954240	5	71	88	*--

Index statistics:

F4: CLUSTERRATIO or normalized $\text{CLUSTERFACTOR} > 80$
F5: $100 * (\text{KEYS} * (\text{ISIZE} + 10) + (\text{CARD} - \text{KEYS}) * 4) / (\text{NLEAF} * 4096) > 50$
F6: $90 * (4000 / (\text{ISIZE} + 10) ** (\text{NLEVELS} - 2)) * 4096 / (\text{KEYS} * (\text{ISIZE} + 10) + (\text{CARD} - \text{KEYS}) * 4) < 100$

CREATOR	NAME	CARD	LEAF	LVLS	ISIZE	KEYS	F4	F5	F6	REORG
Table: SAPR3.MONI										
SAPR3	MONI__0	840	13	2	36	840	89	72	9	---

CLUSTERRATIO or normalized CLUSTERFACTOR (F4) will indicate REORG is necessary for indexes that are not in the same sequence as the base table. When multiple indexes are defined on a table, one or more indexes may be flagged as needing REORG. Specify the most important index for REORG sequencing.

Figure 59. Results of Executing the REORCHK Utility

By default, REORGCHK calls the RUNSTATS command before it executes. The output of REORGCHK is divided into two sections. The first section shows you the table statistics and formulas. The second section displays information about the table's indexes.

4.4.2 Interpreting the Output from REORGCHK

The REORGCHK command uses six formulas that may help you decide whether or not a table requires reorganization. In this section, we look at the table-related formulas in more detail.

Let's first explain the elements that are used to calculate those formulas. These elements are shown in the output of the REORGCHK on table `sapr3.moni` command.

```

Table statistics:
F1: 100*OVERFLOW/CARD < 5
F2: 100*TSIZE / ((FPAGES-1) * 4020) > 70
F3: 100*NPAGES/FPAGES > 80

CREATOR  NAME                CARD  OV   NP   FP   TSIZE  F1  F2  F3  REORG
-----
SAPR3    MONI                    840  43  295  334  954240  5  71  88  *--
-----

```

Figure 60. REORGCHK Table Information

- CREATOR—This column indicates the schema to which the table belongs. Remember that the creator of an object is used as the default schema.
- NAME—This column indicates the name of the table for which the REORGCHK utility has been run. REORGCHK can check a set of tables at one time.
- CARD—This column indicates the number of data rows in the table.
- OV—This column is the overflow indicator. It indicates the number of overflow rows. An overflow may occur when a new column is added to a table or when a variable-length value increases its size.
- FP—This column indicates the total number of pages that had been allocated to the table.
- NP—This column indicates the total number of pages that contain data.
- TSIZE—This column indicates the table size in bytes. This value is calculated from the result of multiplying the number of rows in the table times the average row length allowing for an overhead of eight bytes for each row.
- REORG—This column has a separate indicator for each one of the first three formulas. A hyphen (-) indicates that reorganization is not recommended. An asterisk (*) indicates that reorganization is recommended.

The formulas F1, F2, and F3 provide guidelines for the table reorganization. The formulas are shown in the REORGCHK output.

F1 works with the overflow rows. It recommends a table reorganization if more than 5 percent of the total number of rows are overflow rows.

F2 works with the free or unused space. It recommends a table reorganization if the table size (TSIZE) is less than 70 percent the size of the total space allocated to the table. In other words, it recommends to reorganize a table when more than 30% of the space allocated is unused.

F3 works with free pages. It recommends a table reorganization when more than 20 percent of the pages in a table are free. A page is considered free when it has no rows in it.

Whenever the formulas find that table reorganization is needed, it will be shown with an asterisk in the REORG column of the output.

For example, if the overflow rows of a table exceed the recommended value, the REORG column will look like

REORG

*--

Figure 61. REORGCHK Output Showing Exceeded Value for Overflowed Rows

Remember, these values are only general guidelines. You may use your own thresholds. However, most of time, these values will be adequate for your environment.

4.4.3 Interpreting the Index Output from REORGCHK

Another part of REORGCHK involves interpreting the output gathered from indexes. To do this, we need some information about the structure of indexes in DB2.

Indexes in DB2 are created using a B+ tree structure. These data structures provide an efficient search method to locate the entry values of an index. The logical structure of a DB2 index is shown in Figure 62.

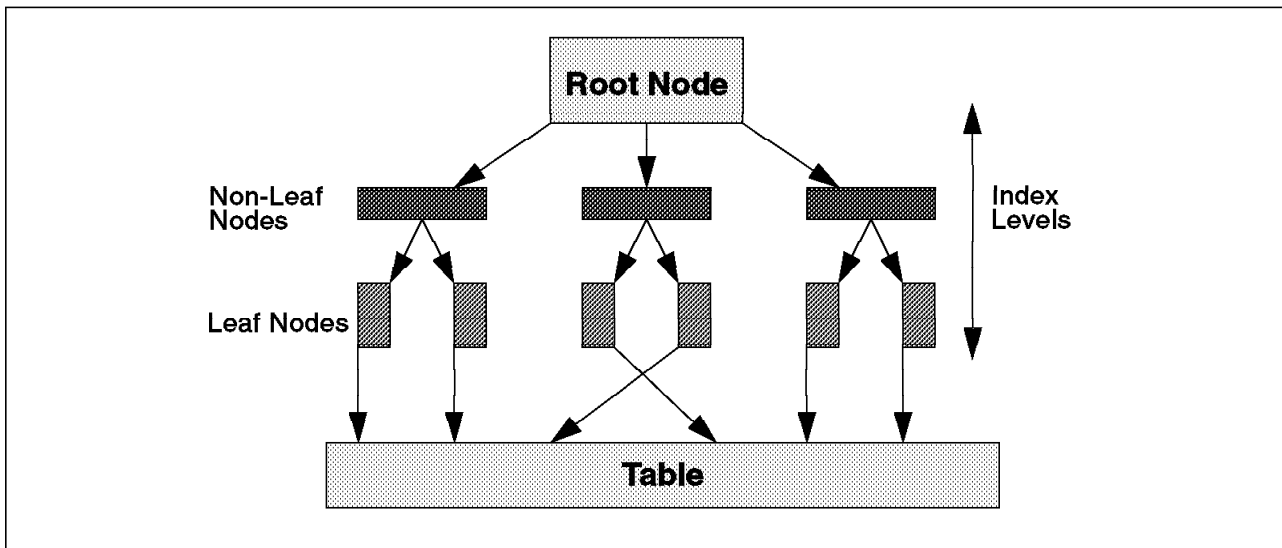


Figure 62. Logical Structure of an Index in DB2

An index can have several levels. The fewer levels an index has, the more quickly DB2 can access the table or data pages. The index shown in Figure 62 has two levels. Now let's review the REORGCHK index information.

```

Index statistics:

F4: CLUSTERRATIO or normalized CLUSTERFACTOR > 80
F5: 100*(KEYS*(ISIZE+10)+(CARD-KEYS)*4) / (NLEAF*4096) > 50
F6: 90*(4000/(ISIZE+10)**(NLEVELS-2))*4096/ (KEYS*(ISIZE+10)+(CARD-KEYS)*4)<100

CREATOR  NAME          CARD  LEAF  LVLS  ISIZE   KEYS   F4    F5    F6  REORG
-----
Table: SAPR3.MONI
SAPR3   MONI__0          840   13    2    36    840   89    72    9  ---
-----

```

Figure 63. REORGCHK Index Information

- CREATOR—This column indicates the schema to which the index belongs. Remember that the creator of an object is used as the default schema.
- NAME—This column indicates the name of the index. REORGCHK is only specified at table level. It will show statistics about all the indexes of a table. The information is also collected for system defined indexes, such as primary key indexes.
- CARD—This column indicates the number of rows in the associated table.
- LEAF—This column indicates the number of LEAF nodes of the index.
- LVLS—This column indicates the total number of levels of the index.
- ISIZE—This column indicates the average column length of the key columns. This column DOES NOT give space allocation information.
- KEYS—This column indicates the number of different key values in the index. In a UNIQUE index, the key and card columns should have the same value.
- REORG—This column has a separate indicator for each one of the index formulas. A hyphen indicates that the reorganization is not recommended, and an asterisk indicates that reorganization is recommended.

The formulas F4, F5, and F6 provide guidelines for the index reorganization. The formulas are shown in the REORGCHK output.

F4 indicates the CLUSTERRATIO or normalized CLUSTERFACTOR. This ratio shows the percent of data rows that are stored in the same physical sequence as the index.

F5 calculates used space. It says that less than 50 percent of the space allocated for the index should be empty.

F6 measures the usage of the indexes pages. It indicates that the number of index page and it should be more than 90 percent of the total entries that NLEVELS can handle.

Whenever the formulas find that table reorganization is needed, this is shown with an asterisk in the REORG column of the output.

For example, if the CLUSTERRATIO of an index is below the recommended level, the REORG column appears as follows:

REORG ----- *--

The REORGCHK command is an analysis tool provided with DB2. To reorganize your tables, you must use the REORG command.

4.4.3.1 REORGCHK Options

You can avoid using the RUNSTATS command by using the CURRENT STATISTICS option of the REORGCHK command. For example, to analyze the current statistics of table sapr3.moni

```
db2 reorgchk current statistics on table sapr3.moni
```

To review the current statistics of all the tables in a database, including the system catalog and user tables:

```
db2 reorgchk current statistics on table all
```

The all option of REORGCHK is not recommended in R/3.

You can also verify the organization of the system catalog tables by using the SYSTEM option. You can select all the tables under the current user schema name by specifying the USER keyword.

```
db2 reorgchk current statistics on table system
```

If you don't specify the CURRENT STATISTICS parameter, REORGCHK will call the RUNSTATS utility.

4.4.4 Using R/3 Performance Monitor to Check for Reorganization

Using the R/3 Database Performance Monitor, you can get a list of all tables and indexes and all statistics available in the DB2 system catalogs. You can use the Performance Monitor to display the REORGCHK output.

Go to the screen shown in Figure 64 on page 117. From the State Part, press the **Detailed Analysis** button in the Tables_and_indexes section. (Note: This is transaction DB02 and was shown in Figure 42 on page 81.)

Analysis of tables and indexes								
Database analysis Edit Goto Monitor System Help								
<input type="text"/> <input type="button" value="Home"/> <input type="button" value="Back"/> <input type="button" value="Forward"/> <input type="button" value="Print"/> <input type="button" value="Refresh"/> <input type="button" value="Help"/>								
Failed Analysis Sort								
1.00.1996 12:10:41								
Table Reorg Recommendation Index								
Owner	Name	Tablespace	Table Reorg Recommendation				Table Size (KB)	Datetime
			Overflow Rows	Freespace Problems	Empty Pages	Reorg		
IPR3	DOKCLU	PSAPES30CD	-	-	-	-	188.477	1996-08-20-11.52.46 >=REAL
IPR3	D342L	PSAPES30CD	-	-	-	-	92.000	1996-08-20-11.52.46 >=REAL
IPR3	ATAB	PSAPPOOLD	-	-	-	-	85.199	1996-08-20-07.09.51.000534
IPR3	DOKHL	PSAPES30CD	-	-	-	-	56.904	1996-08-20-11.52.46 >=REAL
IPR3	DD03L	PSAPES30CD	-	-	-	-	53.930	1996-08-20-11.52.46 >=REAL
IPR3	FHMCT	PSAPES30CD	-	-	-	-	40.428	1996-08-20-11.52.46 >=REAL
IPR3	D010SINF	PSAPES30CD	-	-	-	-	39.086	1996-08-20-11.52.46 >=REAL
IPR3	D301T	PSAPES30CD	-	-	-	-	28.980	1996-08-20-11.52.46 >=REAL
IPR3	TOD1R	PSAPES30CD	-	-	-	-	26.770	1996-08-20-11.52.46 >=REAL
IPR3	TAD1R	PSAPSTABD	-	-	-	-	19.436	1996-08-20-11.52.46 >=REAL
IPR3	EUDB	PSAPES30CD	-	-	-	-	18.391	1996-08-20-11.52.46 >=REAL
IPR3	TSKT4	PSAPSTABD	-	-	-	-	18.200	1996-08-20-11.52.46 >=REAL
IPR3	DOKIL	PSAPES30CD	-	-	-	-	17.077	1996-08-20-11.52.46 >=REAL
SIENH	SYSCOLUMNS	SYSCATSPACE	-	-	-	-	16.574	1996-08-18-12.00.45.000707
IPR3	DSYAT	PSAPSTABD	-	-	-	-	16.141	1996-08-20-11.52.46 >=REAL
IPR3	D345T	PSAPES30CD	-	-	-	-	15.510	1996-08-20-11.52.46 >=REAL
IPR3	DD04T	PSAPES30CD	-	-	-	-	14.369	1996-08-20-11.52.46 >=REAL
IPR3	CROSS	PSAPSTABD	-	-	-	-	14.180	1996-08-20-11.52.46 >=REAL
IPR3	TBD1R	PSAPES30CD	-	-	-	-	14.157	1996-08-20-11.52.46 >=REAL
IPR3	D010S	PSAPES30CD	-	-	-	-	12.617	1996-08-20-11.52.46 >=REAL
IPR3	T100	PSAPES30CD	-	-	-	-	12.559	1996-08-20-11.52.46 >=REAL

Figure 64. Detailed Analysis of Tables and Indexes

You can see each table, its owner, and in which tablespace you will find the table. The other columns are described as follows:

- Table Reorg Recommendation - Overflow Rows
This is formula F1 that is displayed in the REORGCHK output.
- Table Reorg Recommendation - Freespace Problems
This is formula F2 of the REORGCHK output.
- Table Reorg Recommendation - Empty Pages
This is the F3 formula from REORGCHK.
- Index Reorg Recommendation
If reorganization of an index is recommended, this column will have an asterisk (*).
- Table Size (KB)
The REORGCHK command calculates the table size in bytes. This is the same value that is displayed with the TSIZE column from the REORGCHK command.
- Datetime
This is the timestamp from the last execution of the RUNSTATS command. If you find a timestamp such as 1996-08-20-11.52.46 >=REAL, this indicates that the exact timestamp of the last RUNSTATS execution *cannot* be evaluated. You

can only assume that the last RUNSTATS execution was before the date and time that is indicated in the Datetime column shown.

The output shown in Figure 64 on page 117 is sorted by number in the Table Size column. You sort on other columns as well. For example, suppose you want to sort on one of the recommended Reorg columns. You will get output similar to the following:

The screenshot shows a window titled "Analysis of tables and indexes" with a menu bar (Database analysis, Edit, Goto, Monitor, System, Help) and a toolbar. Below the toolbar, there are tabs for "Detailed Analysis" and "Sort". The main area displays a table titled "Table Reorg Recommendation Index" with the following columns: Owner, Name, Tablespace, Overflow Rows, Freespace Problems, Empty Pages, Reorg Recommend, and Table Size(kB). The data is sorted by the "Overflow Rows" column in descending order.

Owner	Name	Tablespace	Overflow Rows	Freespace Problems	Empty Pages	Reorg Recommend	Table Size(kB)
SYSIBM	SYSDINDEXES	SYSCATSPACE	*	-	*	-	3.423
SAPR3	MONI	PSAPPROTD	*	-	*	*	806
SAPR3	CDCLS	PSAPCLUD	*	-	-	-	348
SAPR3	DB6IREORG	PSAPBTABD	*	*	*	-	43
SYSIBM	SYSSMT	SYSCATSPACE	-	*	*	-	16
SYSIBM	SYSSSECTION	SYSCATSPACE	-	*	*	-	108
SAPR3	SAPWLSFIDX	PSAPBTABD	-	*	*	-	13
SAPR3	DB6TREORG	PSAPBTABD	-	*	*	-	23
SAPR3	TB20	PSAPPOOLD	-	*	-	-	3
SAPR3	TKCS	PSAPPOOLD	-	*	-	-	3
SAPR3	DBDIFF	PSAPES30CD	-	*	-	-	3
SAPR3	TAQTS	PSAPSTABD	-	*	-	-	262
SAPR3	T513H	PSAPPOOLD	-	*	-	-	3
SAPR3	TBCO	PSAPSTABD	-	*	-	-	3

At the bottom right of the window, the text "AUS (1) f01n07" and "F/3" are visible.

Figure 65. Detailed Analysis of Tables and Indexes, Sorted by Overflow Rows

Figure 65 shows the output sorted by overflow rows (F1 of REORGCHK). This can be useful in that you can bring these tables to the top of the screen. Remember that there are approximately 7500 tables in the current release of R/3. The redisplaying of the screen will take some time. But, it will be helpful because tables that you need to see will be displayed first.

We can click on **Detailed Analysis** and enter the table name, SAPR3.MONI, to see information just for that table.

Then we can take a detailed look at the statistics for a table and its indexes.

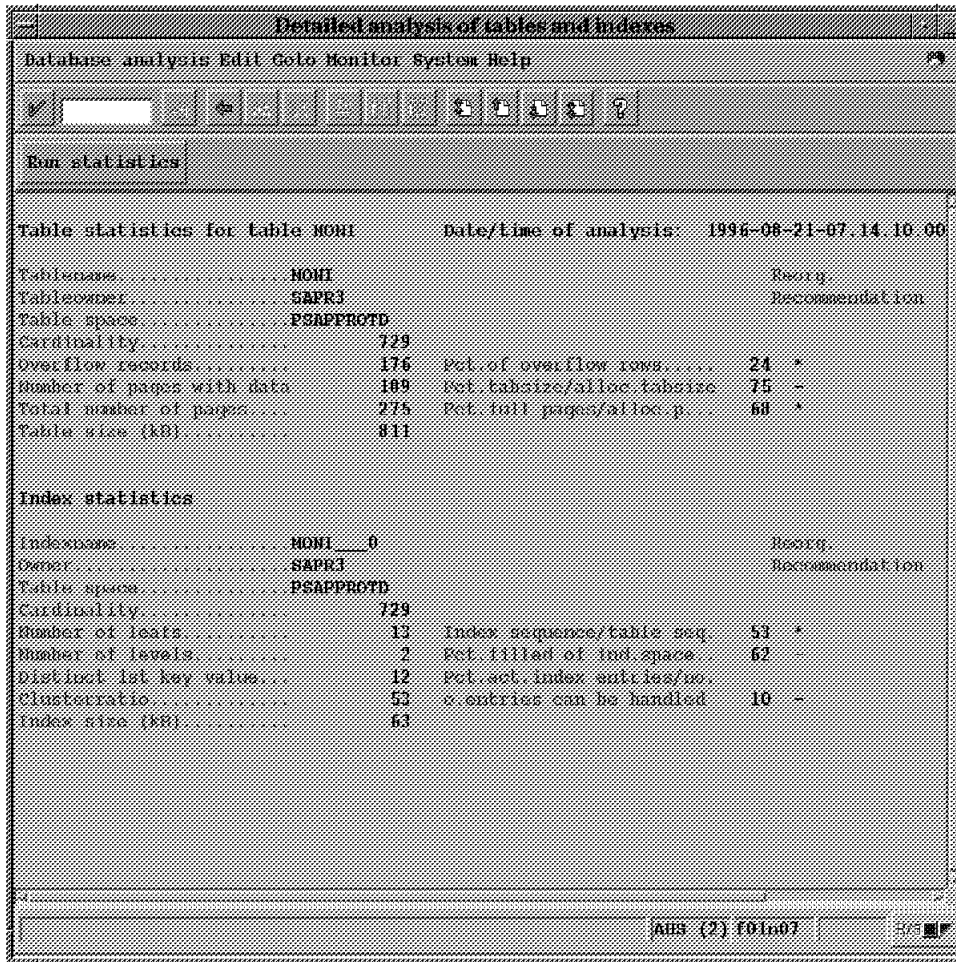


Figure 66. Detailed Analysis of Table and Its Index

4.4.5 Table Reorganization - REORG

After using the REORGCHK command, you may find the physical reorganization of a table necessary. The REORG command will delete all the unused space and write the table data in contiguous pages. With the help of an index, it can also be used to place the data rows in the same physical sequence as the index. These actions can be used to increase the CLUSTERRATIO of the selected index.

Let's say that after running the REORGCHK command, you find that it is necessary to reorganize the SAP3.MONI table:

```
db2 reorg table sapr3.moni
```

This command reorganizes the SAP3.MONI table and all of its indexes.

When using REORG, it is mandatory to use the qualified name of the table. Also consider that the application accessing the tables that are being reorganized cannot be run while the REORG is taking place. In other words, R/3 cannot be running. However, if the size of the table is small, R/3 can still be running. Small is a relative term. In our example, SAP3.MONI is a small table.

4.4.5.1 Using an Index to Reorganize a Table

With the help of an index, REORG will put the data in the same physical order as the selected index. This operation can help improve the response time in the execution of your applications. This is because the data pages of the table will be placed in sequential order according to an index key. This will help DB2 find the data in contiguous space and in the desired order, reducing the seek time needed to read the data. If DB2 finds an index with a very high cluster ratio, it may use it to avoid a sort, thus improving the performance of applications that require sort operations.

When your tables have only one index, it is recommended to reorganize the table using that index. If your table has more than one index defined on it, you should select the most frequently used index for that table. The table and index data will have the same logical and physical order after reorganization, and this will reduce the number of access of the database manager.

As an example, we will assume that the table SAP3.MONI has an index called *sapr3.moni_0*. We will also assume that most of the queries that use the table are grouped by country. Therefore, you might want to reorganize the SAP3.MONI table using the *moni_0* index. The REORG command is as follows:

```
db2 reorg table sapr3.moni index moni_0
```

The INDEX option tells the REORG command to use an index to reorganize a table. After the REORG command has completed, the physical organization of the table should match the order of the selected index. This way, the key columns will be found sequentially in the table.

In Figure 67, the HIGH CLUSTER RATIO INDEX was used to REORG the table SAP3.MONI. The LOW CLUSTER RATIO INDEX is shown to emphasize the difference between using or not using an index to perform the reorganization.

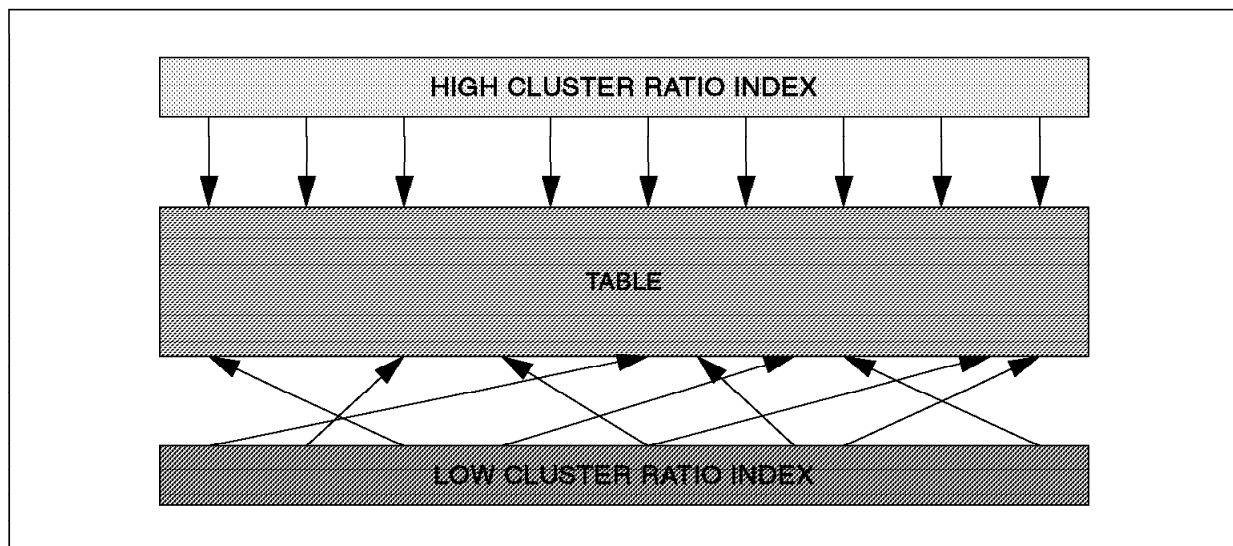


Figure 67. Cluster Ratio Index

The arrows in Figure 67 illustrate the position of the index keys in the table. In a HIGH CLUSTER RATIO INDEX, the keys are in sequential order. In a LOW CLUSTER RATIO INDEX, the keys are distributed along the table.

After reorganizing a table using the index option, DB2 will not force the subsequent inserts or updates to match the physical organization of the table. You may wish to run the REORG command on a regular basis. It is recommended to use the REORGCHK command before attempting a reorganization of a table.

Let's look at an example of what the REORGCHK command evaluated and how the REORG command reorganizes the data in our tables. We look at the SYSIBM.SYSINDEXES table by using the R/3 Database Performance Monitor. If from the State Part screen, which is transaction DB02, we select **Detailed analysis on tables and indexes** for the table, SYSIBM.SYSINDEXES, the following screen appears:

The screenshot shows a window titled "Detailed analysis of tables and indexes" with a menu bar (Database analysis, Edit, Goto, Monitor, System, Help) and a toolbar. The main content is divided into three sections: "Table statistics", "Index statistics", and another "Index statistics" section.

Table statistics for table SYSINDEXES Date/time of analysis: 1996-08-10-12 00.55.000321

Field	Value	Field	Value	Recommendation
tablename	SYSINDEXES	Reorg.		
tableowner	SYSIBM	Recommendation		
table space	SYSCATSPACE			
cardinality	8.719			
overflow records	5.805	Pct. of overflow rows	66	*
number of pages with data	327	Pct. tabsize/alloc. tabsize	94	-
total number of pages	925	Pct. full pages/alloc. p...	35	*
table size (kB)	3.423			

Index statistics

Field	Value	Field	Value	Recommendation
indexname	IBN02	Reorg.		
owner	SYSIBM	Recommendation		
table space	SYSCATSPACE			
cardinality	8.719			
number of leaves	99	Index sequence/table seq.	93	-
number of levels	2	Pct. filled of ind space	60	-
distinct 1st key value	2	Pct. act. index entries/no.		
clusterratio	93	o. entries can be handled	1	-
index size (kB)	511			

Index statistics

Field	Value	Field	Value	Recommendation
indexname	IBN03	Reorg.		
owner	SYSIBM	Recommendation		
table space	SYSCATSPACE			
cardinality	8.719			
number of leaves	87	Index sequence/table seq.	93	-
number of levels	2	Pct. filled of ind space	70	-
distinct 1st key value	2	Pct. act. index entries/no.		
clusterratio	93	o. entries can be handled	1	-
index size (kB)	460			

At the bottom right, the status bar shows "AUS (1) f01n07" and "R/3".

Figure 68. Closer Look at Table and Index Statistics

From Figure 68, we can see that there is a large number of pages allocated for the table, but approximately a third of that number is used to store data. Next we will use DB2admin to help us with the database reorganization.

To get to the appropriate screen we use the following path in SMIT:

SMIT->Applications->DB2admin->Database Reorganization->Reorganize Single Table using an Index.

The following screen appears:

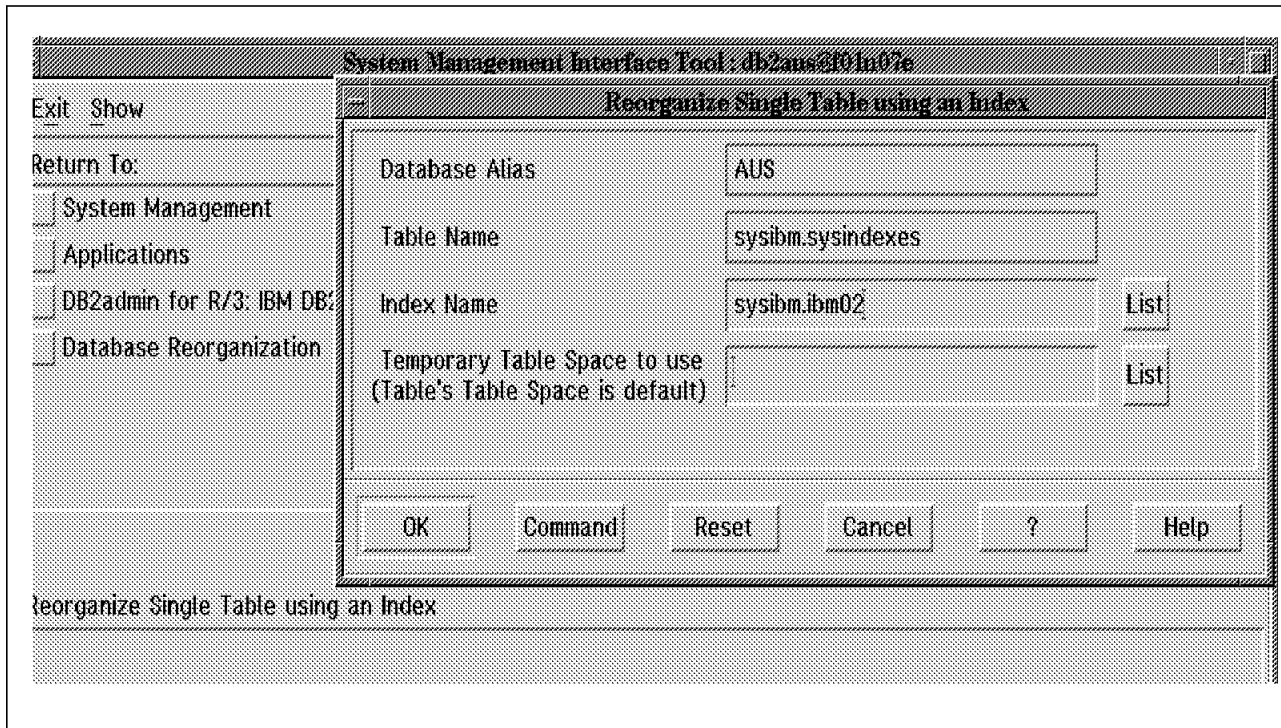


Figure 69. Reorganizing Table Using DB2admin

After filling in the required information, the REORG command executes. (In Figure 69, we showed the graphical form of DB2admin.)

Upon completion, we can use the R/3 Database Performance Monitor to execute the RUNSTATS command and view the output. We need to press the **Run statistics to execute RUNSTATS** button.

In Figure 70 on page 123, we see that we have different results:



Figure 70. Tables and Index Statistics after a REORG

Though this example only shows that we have save 30 pages of storage, the exercise is one that can be used for larger tables in R/3.

4.4.5.2 Using a Temporary Tablespace for Reorganization

While REORG is executing, it creates temporary files in the table space where the table resides. However, you can specify a different table space to create these files. If you want to have control over the location of the files, you need to specify the USE option.

In our example, we will place the temporary tables created during the processing of the REORG command in the temporary table space, PSAPTEMP.

```
db2 reorg table sapr3.moni index index_name use psaptemp
```

In the event of a failure, **DO NOT DELETE** the temporary files created by REORG. DB2 will use them for recovery purposes. Be sure that you have enough space for the temporary table that REORG uses (the temp. table can be much larger than the table itself!), either in the temporary tablespace or in the same tablespace the table resides. For performance of the REORG command, it's recommended to use the same tablespace; using the DMS temporary tablespace takes a lot more time.

Do not delete any temporary file while REORG is running. In case of any error during the reorganization, these files will be used for a rollback.

4.4.5.3 Recommended Actions After Reorganizing a Table

Table reorganization may help to improve the performance of your applications. There are some additional steps that you should follow. These actions allow your existing applications to benefit from the table reorganization.

Here is a list of recommended actions:

1. Use the RUNSTATS utility on your table and its indexes. This provides the optimizer with the information about the new physical organization of the table and its indexes.
2. Use the REBIND utility on the packages that access the reorganized table. This allows your applications to take advantage of any changes selected by the optimizer to improve the access strategy of your SQL statements. In R/3 there are only two packages that need re-binding. Most of R/3 is done using CLI.

4.4.6 RUNSTATS

The system catalog tables contain different information about columns, tables and indexes. They contain information such as the number of rows in a table, the use of the space for a table or index, and the number of different values in a column.

However, this information is not actualized automatically. It has to be collected by a special command called RUNSTATS.

The statistics collected by the RUNSTATS command can be used in two ways: to show the physical organization of the data and to provide information that the DB2 optimizer needs in selecting the best access path for executing SQL statements.

We have seen that the REORGCHK command calls the RUNSTATS command to review the information about the physical organization of tables and indexes. We'll talk about some of the options of the RUNSTATS command and how RUNSTATS is involved in query optimization.

```
db2 runstats on table sysibm.syscolumns
```

the RUNSTATS command does not produce any output. You can only see its results by querying the system catalog tables. More importantly, the performance of your applications may be improved. As we said before, after executing the RUNSTATS command, the system catalog tables are updated with the information gathered by the command.

The partial output below shows some of the data updated by RUNSTATS. This output was obtained by selecting all the columns of the `SYSSTAT.TABLES` catalog view.

TABSHEMA	TABNAME	CARD	NPAGES	FPAGES	OVERFLOW
SYSIBM	SYSTABLES	8842	651	651	0
SYSIBM	SYSCOLUMNS	130672	4116	4276	4588
SYSIBM	SYSINDEXES	8719	904	904	0
SYSIBM	SYSVIEWS	1521	294	294	0
SYSIBM	SYSVIEWDEP	2718	42	42	0
SYSIBM	SYSPLAN	29	2	2	0
SYSIBM	SYSPLANDEP	17	1	1	0
SYSIBM	SYSSECTION	57	34	66	0
SYSIBM	SYSSTMT	56	5	9	0
SYSIBM	SYSDBAUTH	3	1	1	0
SYSIBM	SYSPLANAUTH	53	1	1	0
SYSIBM	SYSTABAUTH	8846	126	126	0
SYSIBM	SYSINDEXAUTH	8664	109	109	0
SYSIBM	SYSRELS	0	0	1	0
SYSIBM	SYSFUNCTIONS	112	9	9	0
SYSIBM	SYSFUNCPARMS	270	7	7	0
...					
SYSIBM	SYSTABCONST	7286	101	101	0
SYSIBM	SYSKEYCOLUSE	27514	363	363	0

If you find a -1 (minus one) in one of the columns, this value indicates that there are no statistics available for that object. The columns of the `SYSSTAT.TABLES` table have the same meaning as those of the `REORGCHK` command.

- **CARD**—This column indicates the number of data rows in the table.
- **NPAGES**—This column indicates the total number of pages that contain data.
- **FPAGES**—This column indicates the total number of pages that have been allocated to the table.
- **OVERFLOW**—This column indicates the number of overflow rows. An overflow may occur when a new column is added to a table or when a variable length value increases its size.

4.4.7 Identifying Updated Statistics

It is not difficult to identify the absence of available statistics for a specific object. The -1 (minus one) value in the statistical information columns indicates this state. However, it is also important to identify the time of the last update of the object's statistics.

When you perform `RUNSTATS` on an object, the utility records the timestamp of its execution in the system catalog tables. Depending on the type of object, you can find the timestamp information in `SYSCAT.TABLES` or `SYSCAT.INDEXES`. In both views, the information is stored in the `STATS_TIME` column.

The `WITH DISTRIBUTION` option is used to instruct DB2 to collect data about the distribution of values for the columns in a table. This option is related to three database configuration parameters: `NUM_FREQVALUES`, `NUM_QUANTILES`, and `STAT_HEAP_SZ`. These parameters limit the action of the `WITH DISTRIBUTION` option.

- **NUM_FREQVALUES**—This indicates the number of most frequent values that DB2 will look for. For example, if it is set to 10, only information for the 10 most frequent values will be obtained.
- **NUM_QUANTILES**—This indicates the number of quantiles that DB2 will look for.
- **STAT_HEAP_SZ**—This indicates how much memory DB2 uses for collecting these statistics. It is recommended to increase the size of this parameter if you have large tables.

This procedure is demanding, and it is not recommended for all your tables. Only tables presenting a high volume of non-uniform values are candidates for this option.

4.4.8 Collecting Detailed Information about Indexes

It is also possible to collect data that will give more information about an index. Index data can become even more fragmented than regular table data. This fragmentation has performance costs for your SQL statements.

This information is used to model the number of I/O operations required to read data pages into the buffer pools. Now, let's say that you want to collect detailed information about the indexes of the table, SAPR3.MONI.

```
runstats on table sapr3.moni for detailed indexes all:
```

The DETAILED option is used to gather this information. The statistics collected are stored in the CLUSTERFACTOR and PAGE_FETCH_PAIRS columns of the SYSIBM.SYSINDEXES system catalog table.

This option gives similar information to that of the CLUSTERRATIO. However, CLUSTERRATIO and PAGE_FETCH_PAIRS provide a more accurate measurement of the relationship between the index and the data pages. These two values can give the optimizer a more detailed way to model the I/O operations and select a better access path for a SQL statement. The DETAILED option is also affected by the STAT_HEAP_SZ database parameter.

4.4.8.1 Recommended Actions after RUNSTATS

Now that the system catalog tables have been updated, you should perform the following procedures. This will give your applications the benefit of the recently collected statistics about your tables.

1. Do a REBIND on the packages that access the reorganized table. This will provide your applications with any changes selected by the DB2 optimizer to improve the access strategy for your SQL statements.
2. Dynamic SQL statements will experience immediate benefits from the execution of the RUNSTATS utility. Packages in this situation do not need to be rebound.

4.4.8.2 The REBIND Utility

The REBIND utility provides you the ability to re-create a package with the information available in the system catalog tables. This allows your embedded SQL applications to use a different access path as selected by the optimizer.

The REBIND command is recommended after doing REORG or RUNSTATS. The DB2 optimizer will use the new organization and recently collected statistics to

generate an access path. This access path will be better suited for the new physical organization of your data.

For example, suppose you have an application called app1 that uses the table DSAPR3.MONI. You have just created a new index for the table, and you need to use REBIND so that the app1 application uses the new index for data access. The REBIND command does not have any options.

```
rebind app1
```

4.4.8.3 Data Maintenance Process

As shown in Figure 71, the data maintenance process starts with the REORGCHK command.

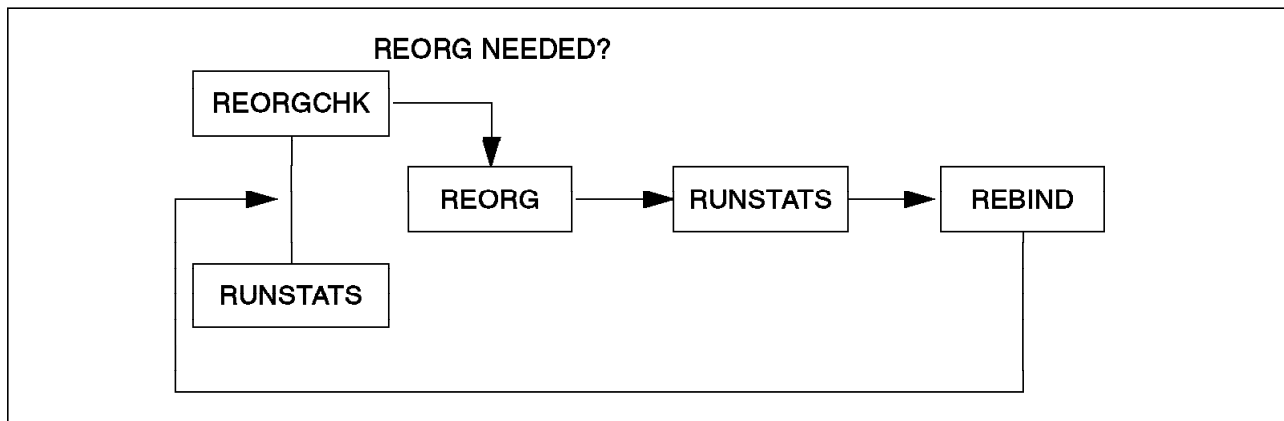


Figure 71. The Data Maintenance Process

The REORGCHK command can execute the RUNSTATS command at the same time. However, it is recommended to call RUNSTATS separately so that you can control the command's options to customize for your environment. After performing the RUNSTATS command, the REORGCHK command reviews the statistics collected, applies six formulas, and then gives you the recommendations about REORG.

If reorganization is recommended, you will use the REORG command on the selected objects and then do RUNSTATS and REBIND. You also must perform a REBIND on any packages affected by the above operations, so that they can take advantage of the benefits of the new physical organization and updated statistics of your data. After new update, insert, and delete operations, as part of the data maintenance process, repeat by first executing the REORGCHK command.

You may want to establish a routine for doing the RUNSTATS and the REBIND processes. Updated statistics will give you the precise information about your database state.

4.4.9 Modelling a Production Environment

You can use the SYSSTAT schema to update system catalog statistical information. You can enter values for statistics such as table cardinality, column distribution, or index cluster ratio. By doing this, you will be able to model different volumes of data on smaller databases.

This is useful when you want to be sure that the access plan your applications are using in your production environment are the same as those in your

development environment. The access paths selected by the optimizer in each environment may differ because of the different data volumes.

By updating the *SYSSTAT* views, you will be able to provide the optimizer the same environment that you will find in a different database. The update procedure is made by using the SQL UPDATE statement.

There are five updatable views under the *SYSSTAT* schema. These views are:

- *SYSSTAT.TABLES*
- *SYSSTAT.COLUMNS*
- *SYSSTAT.INDEXES*
- *SYSSTAT.COLDDIST*
- *SYSSTAT.FUNCTIONS*

These catalog views can also be used to provide a “what-if” analysis for your database applications. For example, you can increase the cardinality of a table or the cluster ratio of an index.

You can use the REBIND command to model the behavior of your static SQL applications in the “what-if” analysis process. You can create or drop an index and then use the REBIND command to see how the access plan is affected with it.

As a general recommendation, only update the *SYSSTAT* schema view in a development environment.

4.4.10 Using DB2admin to Reorganize Data

So far, we have discussed how to use R/3 Database Performance Monitor to display output from the RUNSTATS and REORGCHK commands. The DB2admin tool also provides not only with information about whether a reorganization of a table would be beneficial for your R/3 application, but it provides the interfaces to perform the reorganization.

From the main DB2admin screen when you select **Database Reorganization**, you will have a choice of the following selections:

1. Update Table And Index Statistics (RUNSTATS)
2. Check if Database Reorganization is Required (REORGCHK)
3. Reorganize Single Table using an Index
4. Reorganize Several Tables

We will go into the first and second choices in this chapter and demonstrate a reorganization in the next section.

One of the differences in using DB2admin is that the operations for update statistics and the reorganization check can be done on only one table at a time. Recall that from the R/3 Database Performance Monitor you will get a list of all tables displayed. RUNSTATS is possible to do for one table and index at a time, but within R/3, there are several batch jobs that allow you to update statistics for a number of tables.

The DB2admin tool gives you the possibility of updating statistics for a table with a number of options:

- Table only
- Index Only (Basic)
- Index Only (Detailed)
- Table+Index (Basic)
- Table+Index (Detailed)
- All Possible
- Table only+Distribution
- Table+Index+Distribution

Here we step through the DB2admin utility showing an example where we execute RUNSTATS on a table and index. Then based on the output displayed from REORGCHK, we will REORG the table. We will do this by using DB2admin.

4.4.10.1 Using DB2admin to Update Statistics on a Table (RUNSTATS)

Let's say that we know the SAPR3.MONI table experiences a lot of activity. We also have noticed a slower-than-usual response time when accessing that table. First, we execute a RUNSTATS through DB2admin.

Figure 72 on page 130 is the screen we use. To get there from the basic DB2admin screen, use the following path:

Update Table and Index Statistics

That takes you to the update screen. You need to supply the table name. If you don't know the table, you can select the list option. You will see the names of the 7500 R/3 tables appear.

The default level of statistics is table and one index. There are a list of other possible options. Next you can select the access permitted to other users while RUNSTATS is executing. The default is for read-only for other users.

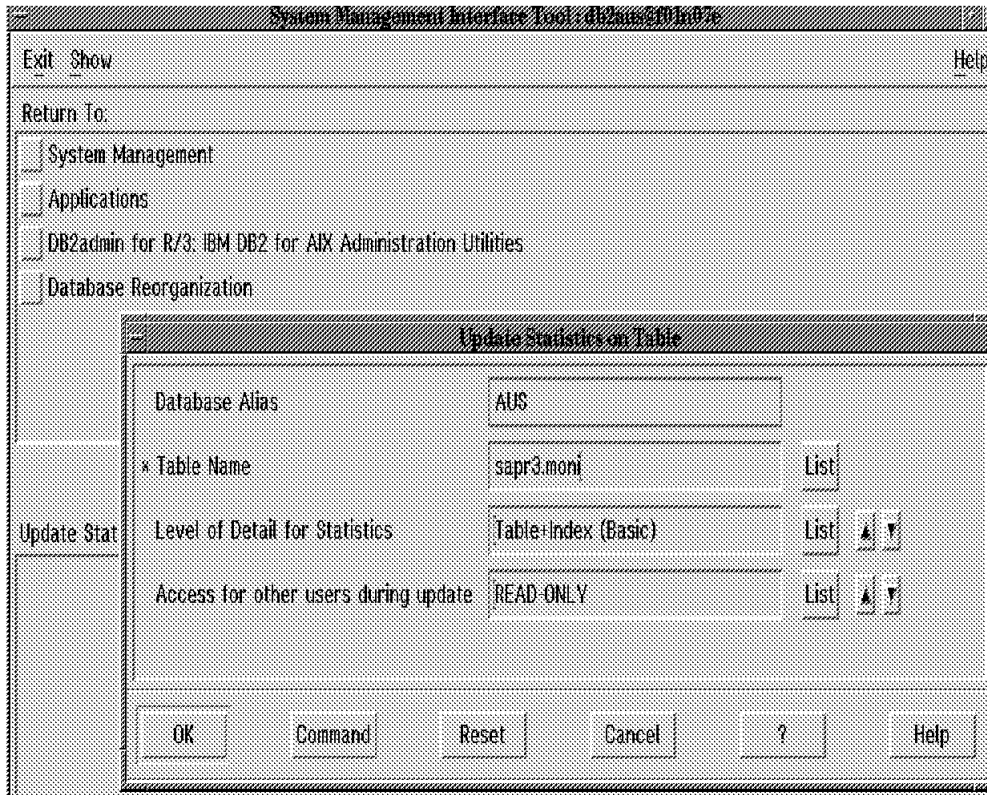


Figure 72. DB2admin - Update Statistics on Table with REORGCHK

Next, we'd like to see the outcome of the RUNSTATS by executing the REORGCHK command. From the Reorganization menu, select **Check if Reorganization is Required** and press **Enter**. We enter the table name, **sapr3.moni**, and select **CURRENT** statistics. This displays all the information from the REORGCHK command. The results are displayed in Figure 74 on page 131.

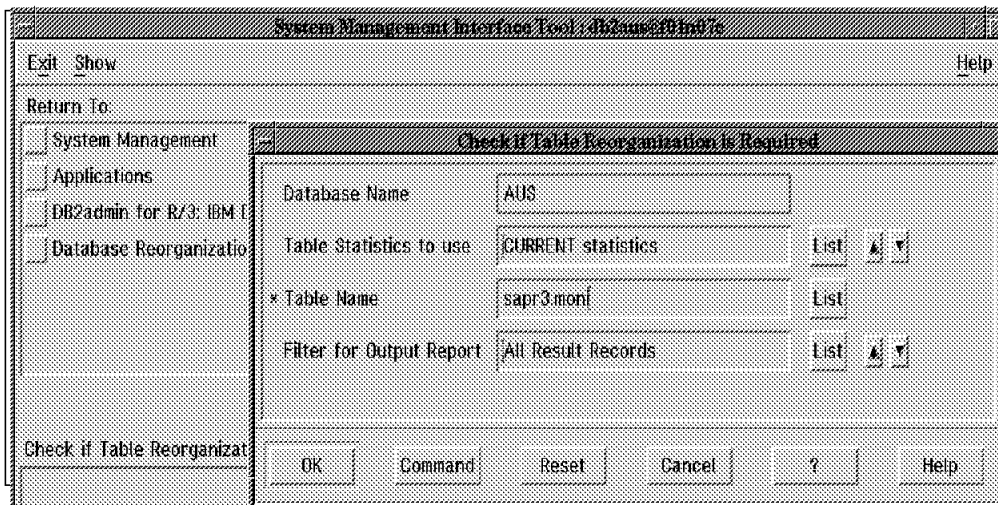


Figure 73. DB2admin - Using Current Statistics

The result is done in a few seconds:

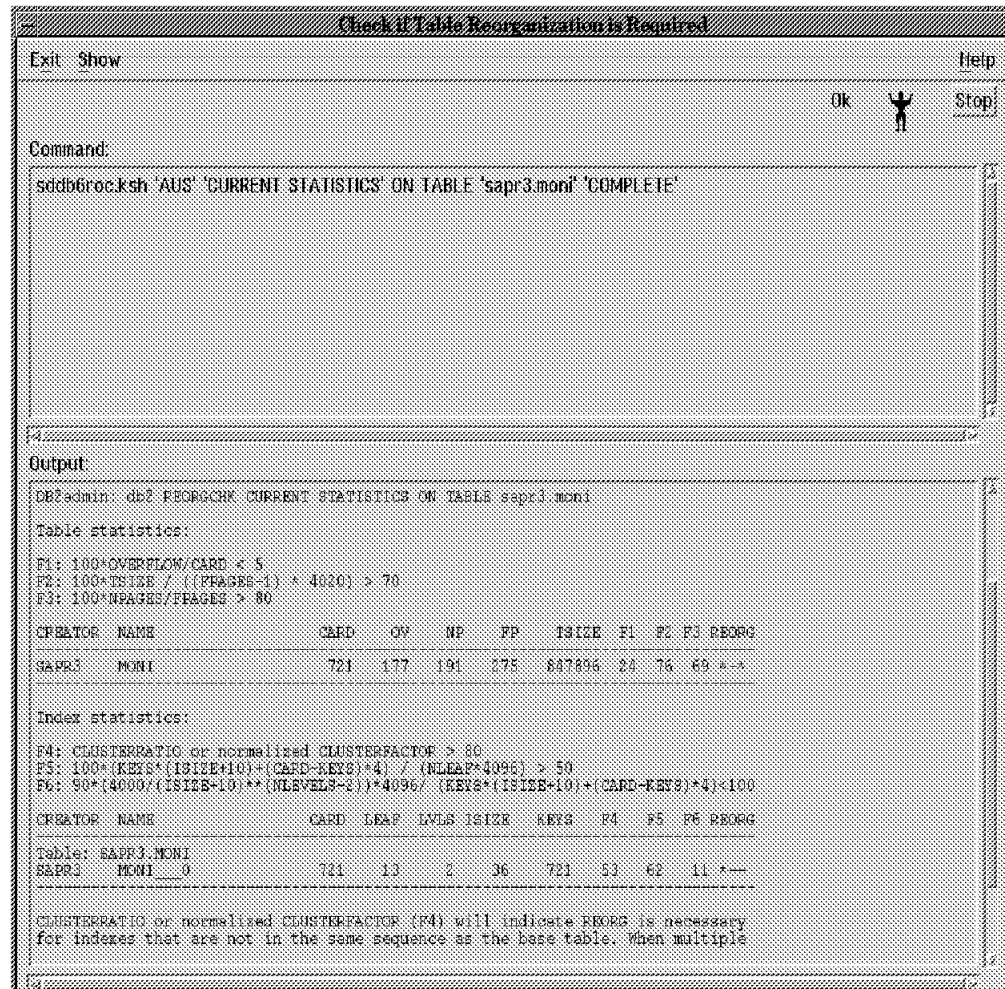


Figure 74. DB2admin - Displaying Results of REORGCHK before REORG

The REORGCHK command recommends a reorganization. This is indicated by the asterisk in the REORG column. Next, we do the reorganization and check the result by using DB2admin.

4.4.10.2 Using DB2admin to Reorganize Data -REORG

From the Database Reorganization screen within DB2admin, select **Reorganize Single Table using an Index** or **Reorganize Several Tables**. Here we reorganize only one table, SAPR3.MON1.

We have to supply the table and index name as screen input. You also have an option as to where the temporary tables will be stored during the execution of the REORG command. The default is the tablespace in which the table is located. Make sure before you execute, that there is enough space available.

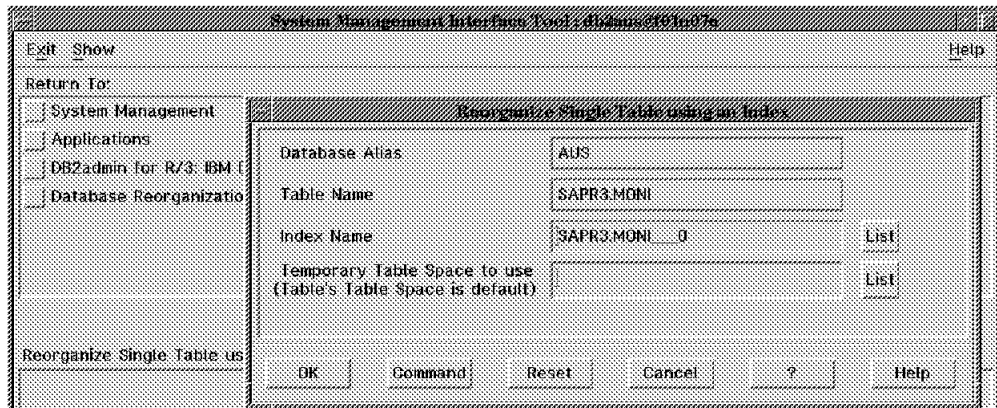


Figure 75. DB2admin - Update Statistics on Table, REORGCHK

Before executing the REORG, SAP recommends performing a backup of the data being reorganized. This is not done automatically by the DB2admin utility. Another recommendation is to monitor those tables you want to reorganize and perform them together since backups can be time consuming. In our example, the reorganization was successful because we had enough space in the tablespace, and no other applications were running in our R/3 system. The following protocol was written to the display screen:

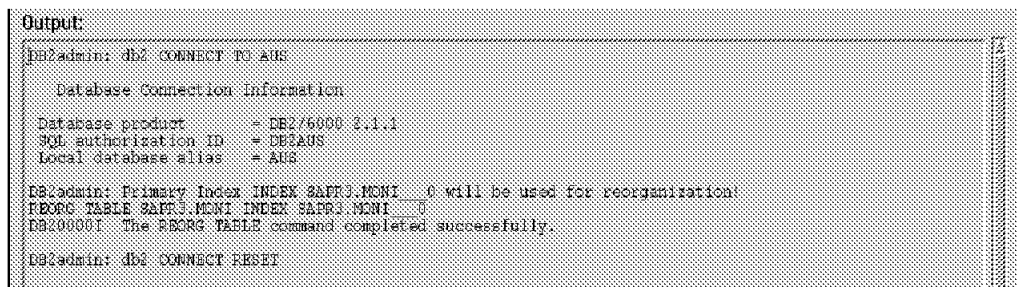


Figure 76. DB2admin - Displaying Results of REORGCHK after REORG

The R/3 installation maintains a set of protocol files that are used to log the backup/recovery activities that are done through the DB2admin interface. For a complete description of the protocols and profiles, see the SAP Database Administration Guide: *DB2 for AIX* that is shipped with the R/3 product.

We can go back to check the result of the REORG using the RUNSTATS and REORGCHK commands. We can use the option **Check if Table Reorganization is Required** and do the UPDATE statistics with the REORGCHK:

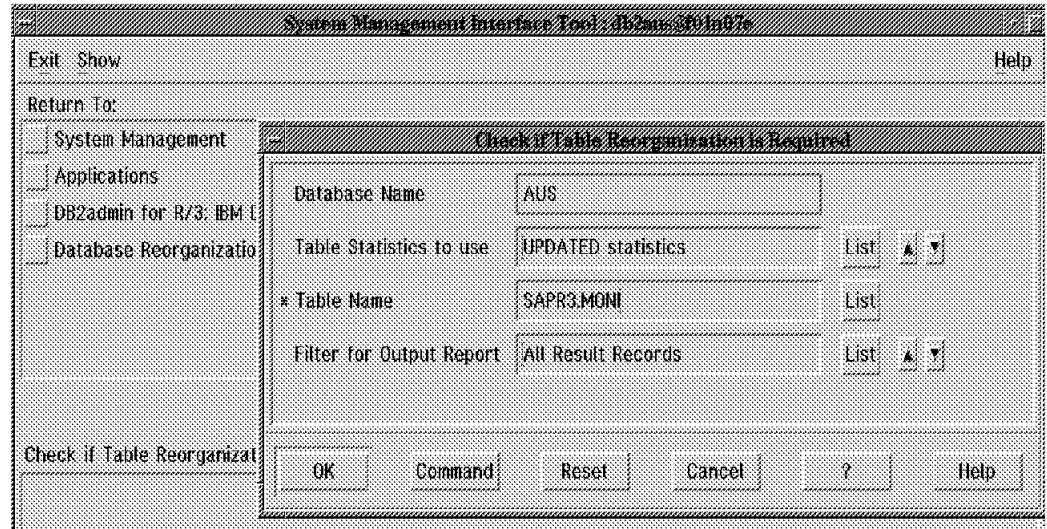


Figure 77. DB2admin - Displaying Results

Here is the result. You can compare it to the REORGCHK output before the reorganization.

```

Output:
Doing RUNSTATS ....

Table statistics:
F1: 100*OVERFLOW/CARD < 5
F2: 100*TSIZE / ((PPAGES-1) * 4020) > 70
F3: 100*NPAGES/PPAGES > 80

CREATOR  NAME                CARD   CV   NP   FP   TSIZE  F1  F2  F3  REORG
-----  -
SAPR3    MONI                    747    0   281  281  852327  0  75 100  ---

Index statistics:
F4: CLUSTERATIO or normalized CLUSTERFACTOR > 80
F5: 100*(KEYS*(ISIZE+10)+(CARD-KEYS)*4) / (NLEAF*4096) > 50
F6: 90*(4000/(ISIZE+10)**(NLEVELS-2))*4096 / (KEYS*(ISIZE+10)+(CARD-KEYS)*4)<100

CREATOR  NAME                CARD  LEAF  LVL5  ISIZE   KEYS   F4  F5  F6  REORG
-----  -
Table: SAPR3.MONI
SAPR3    MONI_____0          747    9     2    36     747  100  93  10  ---

CLUSTERATIO or normalized CLUSTERFACTOR (F4) will indicate REORG is necessary

```

Figure 78. DB2admin - Displaying Results of REORGCHK after REORG

4.4.11 Summary

The DB2 tool that does the reorganization is the DB2 REORG command. It can be executed from the DB2 command line or from db2<SAPSID> with the DB2admin tool.

REORG takes time and has a performance impact for large tables. You must ensure that there is enough temporary space during the reorganization. It's recommended to evaluate the necessity of a REORG by running statistics first. The DB2 RUNSTATS command collects statistics for tables and indexes and updates them in the DB2 system catalog tables. The DB2 REORGCHK command performs

calculations on the statistics and uses formulas to indicate whether a REORG would be beneficial.

RUNSTATS allows to collect varying levels of statistics; you can collect basic-level statistics and distribution statistics. For indexes you can also collect basic-level statistics and detailed statistics. The distribution statistics collects statistics for the most frequently occurring column values. They are most useful for dynamic and static SQL statements that do not use host variables. For SAP R/3, we can focus on basic statistics because the calls are all done via CLI. The detailed statistics do a more detailed evaluation of the clustering factor for indexes.

The following factors may help to identify the necessity of a reorganization:

- The table is frequently used by the application (many updates/inserts/deletes).
- The current size of the table is large, but is predicted to be much smaller after a reorganization.

A reorganization of the table and index can help in two ways:

- Table and index data are copied to a temporary table. Data is rearranged, and copied back. Empty space is removed. Data is stored more compactly.
- The degree of fragmentation has an impact on the performance of your application because data from tables and indexes is more distributed over physical disk. The database manager must perform additional reads to access the data. After a reorganization, the data can be accessed more efficiently.

4.5 Database Monitoring

The data reorganization utilities provide table information, but they do not provide detailed resource usage information. REORGCHK and RUNSTATS do not provide information regarding locking, connections, buffer pool usage, tablespace usage, and memory usage. To gather detailed resource usage information, the DB2 Database Monitor must be used. The monitor interface is invoked using CLP commands, graphical Performance Monitors, or the provided monitoring APIs. The R/3 Database Performance Monitor and the R/3 Alert Monitor also display some of this information. We look at the R/3 and the DB2 graphical interfaces, as well as the DB2 commands, in this section. There are two types of monitors: Snapshot Monitors and Event Monitors. The DB2 Performance Monitor displays information obtained from the DB2 Snapshot Monitor. You can also see information obtained from the DB2 Snapshot Monitor displayed by the R/3 Database Performance Monitor and the R/3 Alert Monitor.

Snapshot Monitoring provides information regarding database activity for a specific point in time. It is a picture of the current state of the DB2 activity. The amount of data returned to the end-user when a snapshot is taken is determined using the monitor switches. These switches may be set in the DBM configuration file or at an application level.

Event Monitoring records the occurrence of specific milestones of DB2 events. This can allow you to collect information about transient events, including: deadlocks, connections, SQL statements.

The performance impact of monitoring is based on the frequency of monitored events and the amount of data captured for each event. Each Snapshot Monitor is invoked immediately and Event Monitors are invoked by DB2 according to the Event Monitor definition.

If excessive amounts of monitoring are performed, the performance of the DB2 database and its applications, including R/3, will be negatively impacted.

4.5.1 Snapshot Monitoring

The Snapshot Monitor provides cumulative information in the form of counters. These counters may be reset using an API. The snapshot information is provided in special data structures which may be examined by the application issuing the snapshot. The R/3 Database Performance Monitor is an example of such an application. The amount of data returned in the monitoring data structures is set according to the switches defined in the table below. The switches may be turned on and off at the instance (DBM configuration) level or at the application level. At the application level you can use the Command Line Processor (CLP) UPDATE MONITOR SWITCHES command. Listed below is a summary of the information provided when a snapshot is performed. There is also base information provided by the Snapshot Monitor.

Group	Information Provided	Monitor Switch	DBM Configuration Parameter
Sorts	number of heaps user, overflows, sort performance	SORT	DFT_MON_SORT
Locks	number of locks held, number of deadlocks	LOCK	DFT_MON_LOCK
Tables	measure activity (rows read, rows written)	TABLE	DFT_MON_TABLE
Buffer pool	number of reads and writes, time taken	BUFFERPOOL	DFT_MON_BUFPOOL
Unit of work	start times, end times, completion status	UOW	DFT_MON_UOW
SQL statements	start time, stop time, statement identification	STATEMENT	DFT_MON_STMT

Table 5. DB2 Monitor Switches

The R/3 installation sets all the monitor switches on so that information can be obtained at any time from the R/3 Database Performance Monitor and the Alert Monitor.

Let's examine the necessary command to turn on the monitor switch to capture all of the SQL statements for each application at the DB2 instance level.

```
db2 update dbm configuration using dft_mon_stmt on
db2 update monitor switches using statement on
```

The first command modifies the DBM configuration; therefore statement information is captured for all applications accessing all database within the instance. The second command only captures SQL statements for the application that activated the switch (in this case the application is the Command Line Processor). This means if the UPDATE MONITOR SWITCHES command is used, only the subsequent SQL statements are considered within the CLP interface.

Most of the data values returned during a database monitor snapshot are based on counters. These counters may be used to determine the amount of database activity. The counters are initialized or reset when one of the following activities occur:

- When the application connects to the database gathering of information is dependent on the level defined for taking snapshots:
 - At the application level, this is when the application connected.
 - At the database level, this is when the first application connected.
 - At the table level, this is when the table was accessed.
 - At the table space level, this is when the table space was accessed.
- Last counter reset
- Responsible monitor group turned on

If you turn on a switch, the data is collected in DB2 until a snapshot is taken. There is a performance impact of gathering monitor data; so ensure that only the appropriate switches have been activated.

4.5.2 Using R/3 to Display Monitor Information

This section looks at the R/3 Database Performance Monitor and the type of information that can be monitored and displayed. There are a number of different types of information that are displayed when using the R/3 utility:

- Table activity—The 50 tables with the largest number of read/write and overflow access can be displayed.
- SAP client—This is a list of database activity by the Application Server.
- DB2 applications—This gives you a list of the database activity per DB2 agent as it is allocated to a SAP Work Process.
- Exclusive lock waits—This situation can occur when a process accesses a table or a tablespace that is exclusively protected by another user.
- Database message log—You can display the DB2 error messages or warnings that are recorded into the AIX system log.
- State on disk—You can obtain storage information and statistical analysis of tablespaces, tables, and indexes.
- Snapshot structures—You can display data from the DB2 system monitor. This data is used in the SAP Database Monitor.
- Parameter changes—You can obtain a history of the changes for the configuration parameters at the DB2 instance level.
- Performance database—This gives you a daily analysis of the database activity.

With the R/3 Database Performance Monitor, you can look at the overview of the database activities. To get there from the entry screen (Figure 41 on page 80),

select **Database->Activity**. This will bring up the overview screen that is referred to as the activity part of the Performance Monitor (transaction ST04).

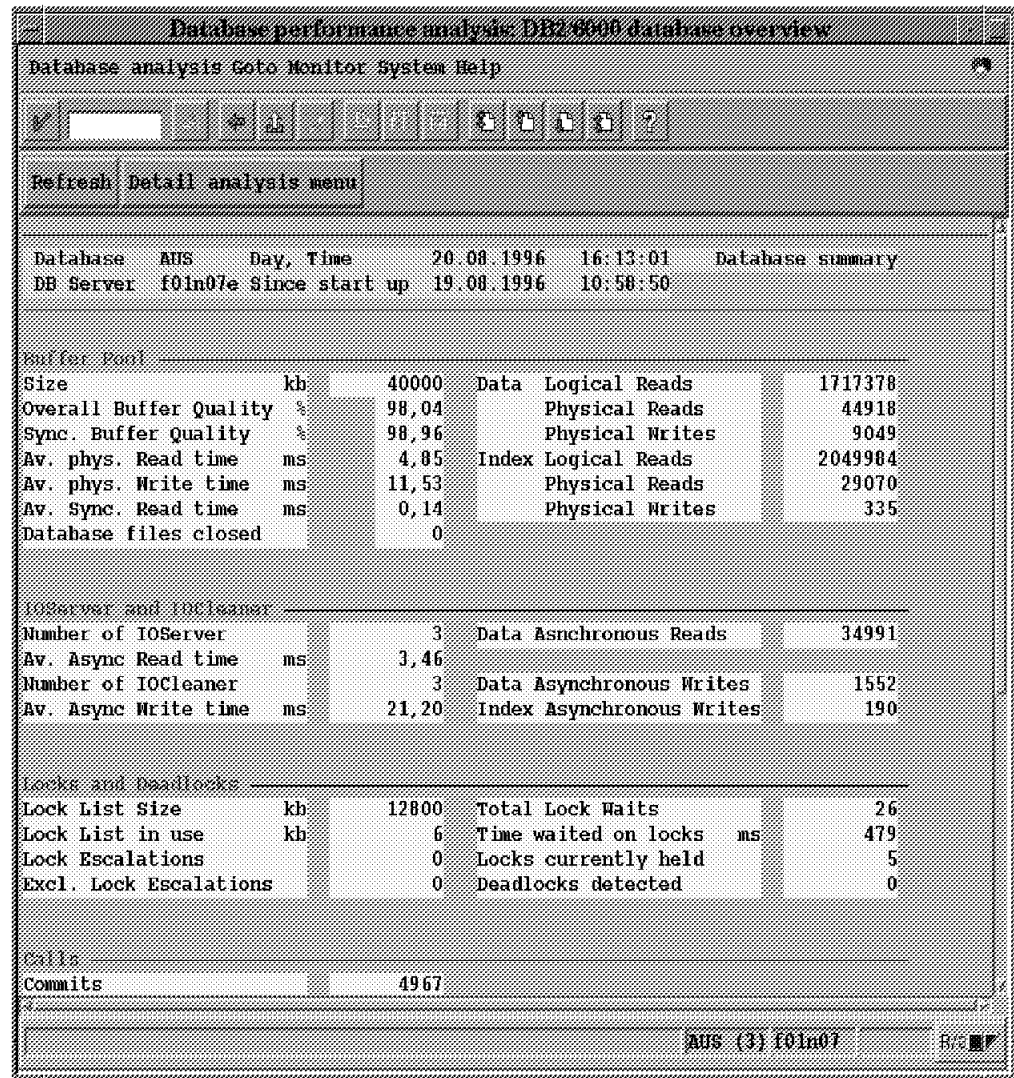


Figure 79. Database Summary - Part One

This following type of information can be found here: (Please see Figure 79 and Figure 80 on page 138.)

- **Buffer Pool**
The buffer pool is the area of storage into which database pages are read and changed. Buffer pool pages are written to disk either by the database manager agents (synchronous writes) or by the I/O Cleaner (asynchronous writes.) There is an entry called "Overall Buffer Quality". This indicates the number of requests that have been read that did not need I/O to transfer data and index pages to the buffer pool. The value here includes the number of reads that were performed synchronously by the DB2 agents and I/O Servers.
- **I/O Server and I/O Cleaner**
Buffer pool pages are read from disk synchronously by database manager agents or are prefetched asynchronously by I/O Servers. Too many prefetching operations can generate unnecessary asynchronous I/O. You

should monitor hit ratios along with buffer pool size, I/O Servers, and I/O Cleaners.

- Locks and Deadlocks
You will be able to see the size of the lock list and how much of the lock is in use. Locks currently held and changes to lock status if the number of locks held increases or decreases. Any deadlocks that are detected can be shown.
- Calls
This is a summary of the total number of commits, rollbacks, inserts, updates, deletes and number of selected and changed rows.
- Sorts
The "sort overflows" field indicates a problem with application performance. When data from sort overflows is written to disk, more disk I/O activity is performed, and thus, application performance is affected.
- Caching
The Catalog Cache Size indicates the maximum amount of database heap space that the CATALOG cache can use. The CATALOG cache is referenced whenever a table, view, or alias name is processed during the compilation of an SQL statement. Remember that the Package Cache Size is used for caching a package that consists of static and dynamic SQL. The quality value tells you whether or not the PACKAGE cache is being used effectively. If the hit ratio of the PACKAGE or CATALOG cache is greater than 80 percent, the cache performs well.

Database performance analysis: DB2/6000 database overview					
Database analysis Goto Monitor System Help					
Refresh Detail analysis menu					
Calls					
Commits		14125			
Rollbacks		8			
Select SQL statements		33571	Rows selected		82493
Upd/Ins/Del statements		4432	Rows modified		4411
Sorts					
Sort Heap Size	kb	2048			
Sort Heap allocated	kb	0			
Total Sorts		835	Sort Overflows		4
Total Sort Time	ms	1491	Active Sorts		0
Caching					
Catalog Cache Size	kb	256	Catalog Cache Lookups		318
Quality	%	65.72	Catalog Cache Inserts		109
			Catalog Cache Overflows		0

Figure 80. Database Summary - Part Two

When we click on the **Detail analysis menu**, the following screen appears:

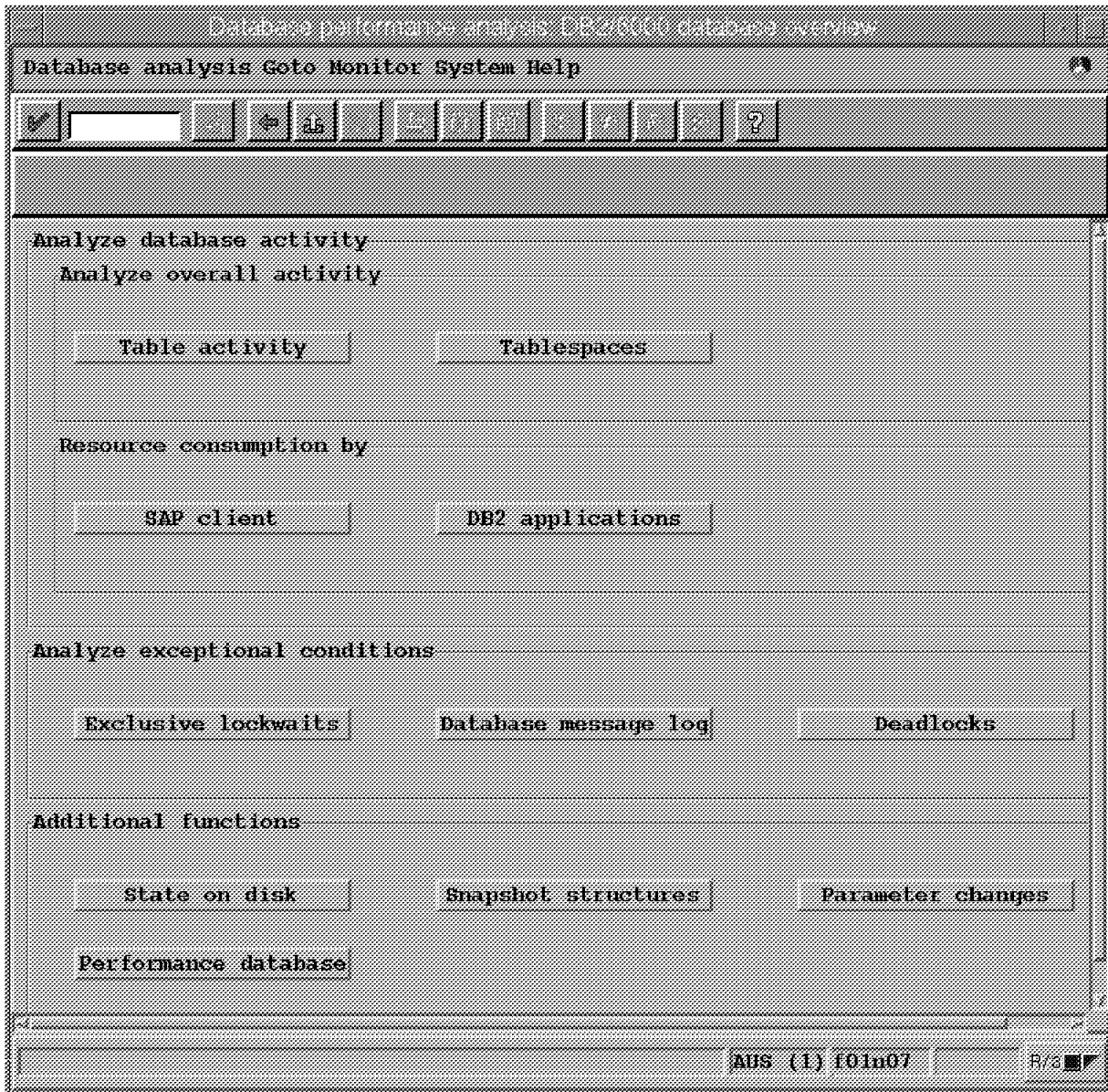


Figure 81. Detailed Database Performance Analysis

This screen gives an overview of critical database performance indicators. The following functions are displayed from here:

- Table activity - This indicates the 50 tables with the highest number of rows read, the 50 tables with the highest number of rows written and the 50 tablespaces with the highest number of access (reads and writes) to overflowed rows.
- Tablespaces - For each tablespace defined for the R/3 database, buffer pool activities and direct access information is displayed.
- SAP client - Analysis of the database activity according to Application Server. This is to monitor Application Servers whose activity is out of a normal range.
- DB2 applications - The session monitor shows performance data for every DB2 application, which is represented by a DB2 agent. The DB2 agent is allocated to SAP Work Processes. General information is displayed as well

as data about the buffer pool activities, the current lock situations, the SQL statements, and caching activities.

- Exclusive lockwaits—This function is used to search for exclusive lockwaits indicating that at least one process is locked by the lock on another process. The process causing the lock and the waiting process are displayed.
- Database message log—Currently the information that is displayed is from the AIX syslog. However, it is intended to display the information that is found in the db2diag.log file.
- Deadlocks—The Event Monitor from DB2 for AIX records all deadlocks. Information on the SAP Work Process involved in the deadlock, the database manager agent process, and detailed lock information on the object is provided.
- State on Disk—This part of the Performance Monitor watches the physical state of the database. We have discussed it in previous chapters. It is often referred to as the State Part. The following functions are provided:
 - Storage management
 - Reorganization statistics and analysis
- Snapshot structures—This function displays all data provided by the DB2 Snapshot Monitor. It is this information that is used in the SAP database monitor, Database Snapshot, Application Snapshot, Lock Snapshot, Table Snapshot, and Tablespace Snapshot.
- Parameter Changes—This displays current and previous setting of the database and database manager configuration parameters within DB2, with respect to the date of change.
- Performance database—This is a day-by-day trend analysis of the database activity. You can emphasize peak periods and display delta value. They are displayed graphically according to the time period you select.

Database performance Table activity statistic					
Database analysis Edit Goto Monitor System Help					
<input type="text"/> <input type="button" value="←"/> <input type="button" value="→"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="↺"/> <input type="button" value="↻"/> <input type="button" value="⌂"/> <input type="button" value="⌂"/> <input type="button" value="⌂"/> <input type="button" value="⌂"/> <input type="button" value="⌂"/> <input type="button" value="⌂"/>					
<input type="button" value="Print"/> <input type="button" value="Reset"/> <input type="button" value="Since reset"/> <input type="button" value="Since DB start"/> <input type="button" value="Graphics"/>					
7.08.1996 14:59:19 Statistics of table activities - top 50's					
Table schema	Table Name	Type	Read	Written	Overflow Accesses
APR3	MONI	User	612	130	15
YSIBM	INTERNAL	User	66	0	0
YSIBM	INTERNAL	User	22	0	0
APR3	D010TINF	User	76	0	0
APR3	D010T	User	75	0	0
APR3	D010Y	User	265	0	0
APR3	D010L	User	317	0	0
APR3	D010LINF	User	264	0	0
YSIBM	INTERNAL	User	10	0	0
YSIBM	INTERNAL	User	30	0	0
APR3	D020LINF	User	57	12	0
YSIBM	INTERNAL	User	39	0	0
YSIBM	INTERNAL	User	13	0	0
APR3	BTCSEV	User	1	0	0
APR3	TSP03	User	1	0	0
APR3	TSP03C	User	1	0	0
APR3	D020L	User	45	8	0
APR3	D342L	User	51	0	0
APR3	TFDIR	User	91	0	0
APR3	D345T	User	42	1	0

Figure 82. Table Activity Statistics

Let's display some of the functions. Let's start with Table Activity Statistics. This is shown in Figure 82. To get to this screen, select **Detail analysis menu** from the Database Overview Screen and then select **Table Activity** in the detail menu.

The Overflow Accesses field shows how often the database accessed records that were overflowed. This is specified by the 50 tables with the highest number of accesses to the overflow record.

The row of the database will overflow if it is updated and no longer fits in the data page where it was originally written. Overflow rows indicate that data fragmentation has occurred. If the number of overflow accesses is high, you may be able to improve table performance by reorganizing the table.

Next, we look at tablespace information.

Database Performance Statistics by Client						
Database analysis Edit Goto Monitor System Help						
<input type="text"/> <input type="button" value="Home"/> <input type="button" value="Back"/> <input type="button" value="Forward"/> <input type="button" value="Print"/> <input type="button" value="Refresh"/> <input type="button" value="Help"/>						
Report	More information	Reset	Since reset	Since DB start		
7.00.1996 15:02:51 Tablespaces - buffer pool						
Tablespace	Log reads	Phy reads	Asy reads	Writes	Asy writes	Ix log rea
SYSCATSPACE	34,027	870	727	0	0	35,653
PSAPSTABI	99	20	0	0	0	20,765
PSAPES3OCI	177	23	0	0	0	20,012
PSAPPROTI	147	21	0	0	0	7,670
PSAPBTABI	95	19	0	0	0	7,425
PSAPPOOLI	86	19	0	0	0	3,621
PSAPEL3OCI	153	19	0	0	0	2,354
PSAPLOADI	263	35	0	0	0	26
PSAPPROTD	1,726	136	0	0	0	0
PSAPLOADD	536	69	0	0	0	0
PSAPSOURCEI	258	33	0	0	0	0
PSAPUSERID	130	17	0	0	0	0
PSAPPOOLD	1,761	207	124	0	0	0
PSAPDOCUD	258	33	0	0	0	0
PSAPCLUD	514	65	0	0	0	0
PSAPUSERI1	66	9	0	0	0	0
PSAPDDICD	61,591	1,019	888	0	0	0
PSAPBTABD	1,784	59	0	0	0	0
PSAPTEMP	95	1	0	11	0	0

Figure 83. Tablespaces

The tablespace information received by the Database Snapshot Monitor shows detailed information on buffer pool usage for the tablespace.

Figure 83 shows only some of the information that is available on this screen. You can obtain detailed information about buffer pool usage on a tablespace basis. The following is a list of information you can obtain:

- Log reads—This indicates the number of logical read requests for data pages that have gone through the buffer pool.
- Physical reads (Phy reads)—This is the number of read requests that required I/O to get data pages into the buffer pool.
- Asynchronous reads (Asy reads)—This is the number of pages read asynchronously into the buffer pool.
- Writes—This indicates the number of times a buffer pool data page was physically written to disk.

- Asynchronous writes (Asy writes)—This is the number of times a buffer pool data page was physically written to disk by an asynchronous page cleaner.
- Index log read (Ix log read)—This indicates the number of logical read requests for index pages that have gone through the buffer pool.
- Index physical read (Ix phy read)—This indicates the number of times a buffer pool index page was physically written to disk.
- Index writes (Ix writes)—This indicates the number of times a buffer pool index page was physically written to disk.
- Index asynchronous write (Ix asy write)—This tells you the number of times a buffer pool index pages was physically written to disk by an asynchronous page cleaner.
- Read time—This provides the total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
- Write time—This provides the total amount of time spent physically writing data or index pages from the buffer pool to disk.
- Asynchronous read time (Asy r.time)—This tells you the total elapsed read time for I/O prefetchers.
- Asynchronous write time (Asy w.time)—This is the total elapsed time spent writing data or index pages from the buffer pool to disk by the I/O page cleaners.
- Asynchronous read requests (Asy r.reqs)—This is the number of asynchronous read requests.

apl	DB2 agent	DB2 system	Cint proc.	Cint system	Status	User CPU (s)
1	24920	f01n07e	18932	f01n07e	UOW Waiting	14,8300
2	15396	f01n07e	25634	f01n07e	UOW Waiting	0,3000
3	40978	f01n07e	45846	f01n07e	UOW Waiting	5,3900
4	12118	f01n07e	20522	f01n07e	UOW Waiting	0,0700
5	20826	f01n07e	34866	f01n07e	UOW Waiting	0,0700
6	18204	f01n07e	32026	f01n07e	UOW Waiting	0,1300
7	22878	f01n07e	44838	f01n07e	UOW Waiting	0,1000
8	4840	f01n07e	53046	f01n07e	UOW Waiting	0,6300
9	19554	f01n07e	47122	f01n07e	UOW Waiting	19,7500
10	17766	f01n07e	21038	f01n07e	UOW Waiting	0,0800
11	28692	f01n07e	31034	f01n07e	UOW Waiting	0,0600
12	54098	f01n07e	19742	f01n07e	UOW Waiting	0,0500
13	21662	f01n07e	52636	f01n07e	UOW Waiting	1,2100

Figure 84. Active DB2 Applications

You can analyze the work load of the database per Application Server in the SAP system. By means of this analysis, you can find out which Application Server is placing the greatest load on the database server.

Some of the information you can obtain is the following:

- Application (Appl)—This is the number of the application.
- DB2 agent—This is the process ID of the DB2 database process belonging to a SAP Work Process.
- DB2 system—This will be the host name of the database server.
- Client process (CInt proc)—This is the process ID of the SAP work process.
- Client system (CInt system)—This is the host name of the application server where the Work Process is executing.
- Status—This is the current status of the application. Possible values for this field are:
 - Database Connect Pending
 - Database Connect Complete
 - Unit of Work Executing
 - Unit of Work Waiting
 - Lock Wait
 - Commit Active
 - Rollback Active
 - Recombining
 - Compiling
 - Request Interrupted
 - Database Disconnect Pending
 - Transaction Prepared
 - Transaction Heuristically Rolled Back
 - Transaction Ended
 - Creating Database
- User CPU time, System CPU time - The CPU values provide the time (in seconds) used by the database manager agent (DB2agent) process. This can help you understand the level of activity within the application and may help to identify applications that require further tuning. The values include CPU time for both SQL and non-SQL statements executed by the application.
- Operation - This is the database operation that is currently being processed or the one most recently processed. The operation may be one of the following:
 - PREPARE
 - EXECUTE
 - EXECUTE IMMEDIATE
 - OPEN
 - FETCH
 - CLOSE
 - DESCRIBE
 - COMMIT (static SQL)
 - ROLLBACK (static SQL)
- SQL statement - You can see the SQL statement that is being processed when the snapshot was taken. It can also be the SQL statement most recently processed, depending on when the snapshot was taken.

Parameter Changes in Database		
Database analysis Edit Goto System Help		
<input type="text"/> <input type="button" value="←"/> <input type="button" value="↓"/> <input type="button" value="→"/> <input type="button" value="↑"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="↸"/> <input type="button" value="⌂"/> <input type="button" value="⋮"/> <input type="button" value="⌘"/>		
Table	Select Period	History of file Active parameters
7.08.1996	15:06:20	List of Active Parameters
Parameter		Value
iftdbpath	db mgr	/db2/AUS
liaglevel	db mgr	3
liagpath	db mgr	/./subsys/database/
lir_cache	db mgr	no
lir_obj_name	db	
lir_obj_name	db mgr	
lir_path_name	db mgr	/./subsys/database/
lir_type	db mgr	NONE
llchktime (ms)	db	300000
lrda_heap_sz (4kb)	db mgr	128
fileserver	db mgr	
indexrec	db	system
indexrec	db mgr	restart
indexsort	db	1
ipx_socket	db mgr	879E
ceepdari	db mgr	yes
locklist (4kb)	db	3200
locktimeout (sec)	db	-1
log_retain_status	db	on
logbufsz (4kb)	db	64
logfilsiz (4kb)	db	5000
loghead	db	S0000041.LOG
logpath	db	/db2/AUS/db2aus/SQL00001/SQLLOGDIR/
logprimary	db	9
logretain	db	enabled
logsecond	db	1

Figure 85. Parameter Changes

This function, if selected, displays current and previous settings of the DB2 for AIX database and database manager parameters, along with the respective date of change. (For more information about these parameters, please see 2.5, “DB2 Configuration Parameters” on page 44.)

If you compare the database history with the parameter changes, you can recognize the effects of the parameter changes.

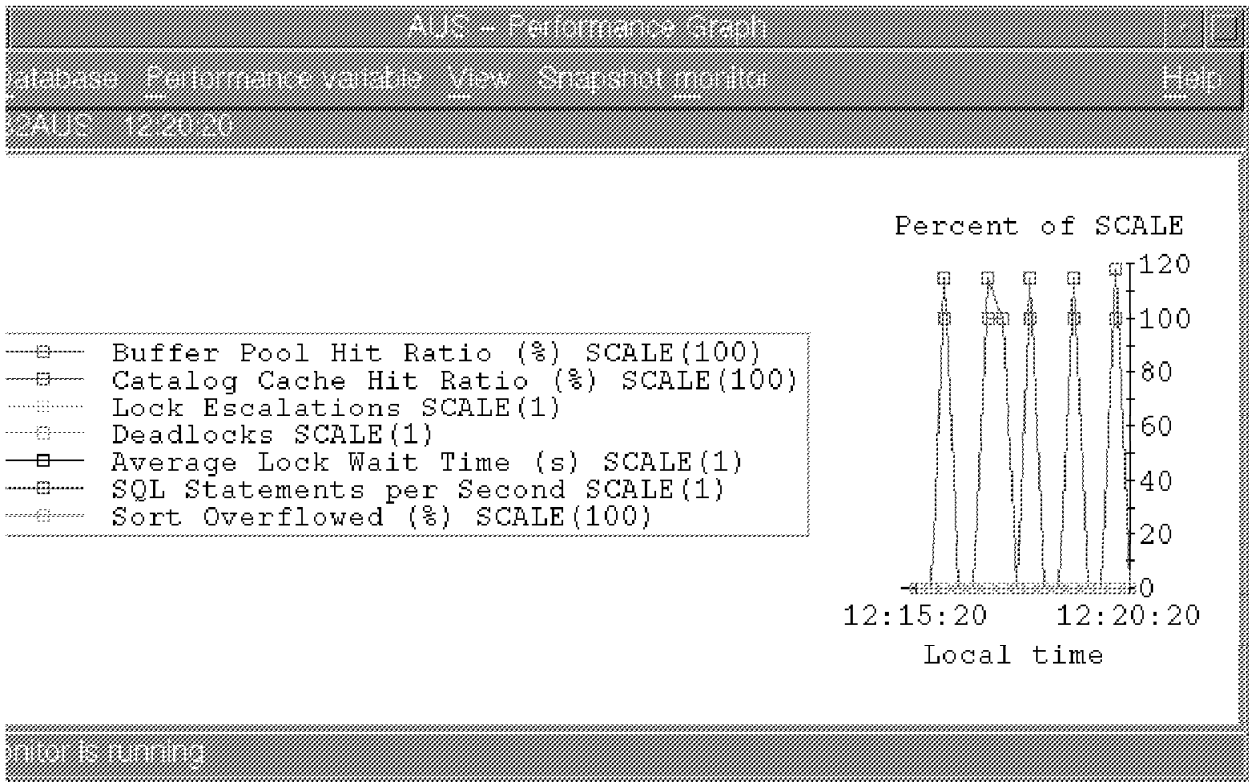


Figure 87. Performance Graph for a Database

You may decide to monitor different values and display only the values you are interested in. Some useful performance variables include the following:

- Buffer Pool Hit Ratio—This value indicates how often DB2 could read buffer pool data directly from memory instead of reading the page from disk. The greater the value of buffer pool hit ratio, the less amount of disk access time required to satisfy the query.
- Catalog Cache Hit Ratio—This value indicates if the system catalog information describing the reference objects was found in memory (CATALOGCACHE_SZ database parameter). The higher the ratio, the less amount of disk activity was required.
- Lock Escalation—This value indicates how many lock escalations have occurred. The amount of lock escalation activity depends on the MAXLOCKS and LOCKLIST parameters. If lock escalation occurs, the amount of database concurrency is reduced. Usually, escalations occur as locks are escalated from row-level locks to table-level locks.
- Deadlocks—This value indicates the total number of deadlocks that have occurred since the monitor switches have been set.
- Average Lock Wait Time—This value indicates how long (average) an application requesting a lock had to wait because the resource was being used by another application. If the average wait time is high, end-user query response time will be adversely affected.
- SQL Statements per Second—This value indicates how many SQL statements are processed per second. A low value may indicate complex query processing or concurrency problems.

- Sort Overflowed—This value indicates the total number of sorts that ran out of sort heap and may have required disk space for temporary storage.

In the monitor graph shown in Figure 87 on page 147, there were no lock escalations, no deadlocks, no sort overflows, and the buffer pool and CATALOG cache hit ration was nearly 100 percent.

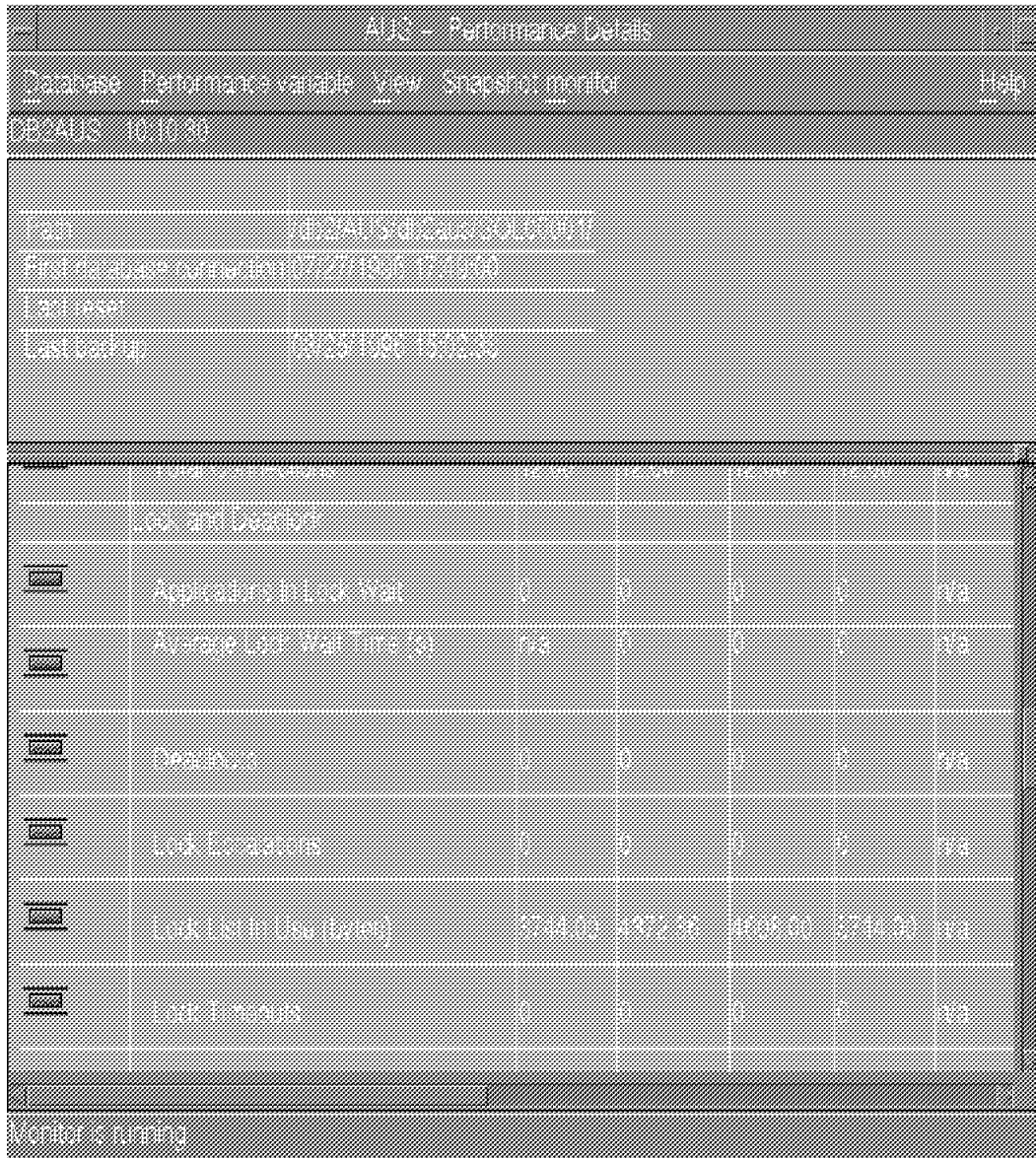


Figure 88. Example of Performance Details Display

Detailed information about the current, average, maximum, and minimum values for the monitored performance variables is shown in Figure 88. From the data that is displayed (this is a partial display), we notice that the current size of the LOCKLIST memory parameter is 3744 bytes. The maximum amount used was 4608 bytes. This information is valuable for tuning database configuration parameters.

4.5.4 Event Monitoring

While Snapshot Monitoring records the state of database activity when the snapshot is taken, an Event Monitor records the database activity when an event or transition occurs. Some database activities that need to be monitored, cannot be easily captured using the Snapshot Monitor. An example of these activities include deadlock scenarios. When a deadlock occurs, DB2 resolves the deadlock by issuing a ROLLBACK for one of the transactions. Information regarding the deadlock event cannot be easily captured using the Snapshot Monitor because the deadlock has likely been resolved before a snapshot can be taken.

Event monitors are similar to other database objects because they are created using the SQL DDL (Data Definition Language). The main difference is that an Event Monitor may be turned on or off, much like the Snapshot Monitor switches. The only R/3 user that can create an Event Monitor is db2<SAPSID>.

When the Event Monitor is created the type of event to be monitored must be stated. The event records are recorded when the following events occur:

- DATABASE—records an event record when the last application disconnects from the database
- TABLES—records an event record for each active table when the last application disconnects from the database
- DEADLOCKS—records an event record for each deadlock event
- TABLESPACES—records an event record for each active table space when the last application disconnects from the database
- CONNECTIONS—records an event record for each database connection event when an application disconnects from a database
- STATEMENTS—records an event record for every SQL statement issued by and application (dynamic and static)
- TRANSACTIONS—records an event record for every transaction when it completes (COMMIT or ROLLBACK)

The output of an Event Monitor is stored in a directory or in a named pipe. The existence of the pipe or the file will be verified when the Event Monitor is activated. If the target location for an Event Monitor is a named pipe, it is the responsibility of the application to read the data promptly from the pipe. If the target for an Event Monitor is a directory, the stream of data will be written to a series of files. The files are sequentially number and have a file extension of evt (such as 00000000.evt, 00000001.evt and so on). The maximum size and number of Event Monitor files is specified when the monitor is defined.

An Event Monitor will turn itself off if the defined file space has been exceeded.

Let's create an Event Monitor that stores its event records in a directory. In the following sample, an Event Monitor called evmon1 has been created. The Event Monitor is not active at this time because it has not been turned on. It has been defined to store the event information in the /db2/AUS/db2event directory.

Sample Event Monitor Definition

```
create event monitor mgrpay for database, statements  
  
where APPL_NAME = 'PAYROLL' and AUTH_ID = 'SAPR3'  
  
to file '/db2/AUS/db2event' maxfiles10 maxfilesize 1000 nonblocked append
```

The Event Monitor output directory (here, db2event) will NOT be created by DB2; it must be created by the R/3 database administrator.

The monitor in the example is defined to allocate up to 10 files each 4 MB in size, for a total monitor storage area of 40 MB. It will append to the file. Other Event Monitor options include: specifying the size of the write buffer, synchronous (BLOCKED) writes, asynchronous (UNBLOCKED) writes, APPEND the Event Monitor data to existing records, or REPLACE the Event Monitor data in the directory when the monitor is activated. The Event Monitor may be defined to be automatically started when the database is active, but the default is manual start.

Two system catalog tables are used to store Event Monitor definitions:

- SYSCAT.EVENTMONITORS - Contains a record for each Event Monitor including the current state of the Event Monitor.
- SYSCAT.EVENTS - Contains a record for each event being monitored. A single Event Monitor may be defined to monitor multiple events (for example, DEADLOCKS and STATEMENTS).

Event monitors may be defined to monitor many different types of database activities. A condition may also be specified for an Event Monitor. The condition can be based on the APPL_ID, AUTH_ID, or APPL_NAME, as in the example shown. There is no limitation in the number of defined Event Monitors, but there is a limitation of 32 active Event Monitors.

Once the Event Monitor has been defined, using the CREATE EVENT MONITOR statement, it must be activated. An Event Monitor can be started automatically each time the database is started, or it can be started using the following statement:

```
db2 set event monitor mgrpay
```

When the Event Monitor has been activated, Event Monitor records are written to the files contained in the defined directory, the /db2/AUS/db2event directory in our example. The Event Monitor files cannot be analyzed directly; an application must be used. There are a few application alternatives provided by DB2 and one by R/3 that we will discuss, but let's first examine some of the Event Monitor records.

To ensure that all of the event records have been written to disk (some may be buffered), turn the Event Monitor off.

The Event Monitor is similar to tracing because each event is recorded as it occurs, and it is appended to the event record log files (to be analyzed later). An Event Monitor file contains a number of event records. The following table shows all of the event record types and when they are used.

Record Type	Collected for Event Type
Event Monitor Log Header	All
Event Monitor Start	All
Database Header	All
Database Event	Database
Tablespace Event	Tablespace
Table Event	Table
Connection Header	Connection
Connection Event	Connection
Connection Header	Transaction
Transaction (Unit of Work) Event	Transaction
Connection Header	Statement
Statement Event	Statement
Dynamic Statement Event	Statement
Connection Header	Deadlock
Deadlock Event	Deadlock
Deadlock Connection Event	Deadlock
Event Monitor Overflow	All (if any)

Table 6. DB2 Event Types

Some event records are created when any application disconnects from the database, and others are only created when the last application disconnects from the database.

If the Event Monitor is monitoring database, tablespace, or table events, it will only write event records when the last application using the database disconnects. If database, tablespace, or table monitoring data is required before the last application disconnects, use the Snapshot Monitor.

To flush all of the event records turn the monitor off by using the command:

```
db2 "set event monitor mgrpay state = 0"
```

The Event Monitor is still defined, in the system catalog tables, but it is not actively recording event information. To determine if an Event Monitor is active or inactive, use the SQL function `EVENT_MON_STATE`. An example SQL statement is shown.

```
db2 "select evmonname, event_mon_state(evmonname) from syscat.  
eventmonitors where evmonname = 'mgrpay'"
```

It can be used to determine which Event Monitors are active. A value of 1 indicates that the Event Monitor is active, and a value of 0 indicates that the monitor is inactive.

To remove the definition of an Event Monitor, the DROP statement must be used. Removing the definition would remove the associated rows in the SYSCAT.EVENTMONITORS and SYSCAT.EVENTS system catalog tables. An example of removing the evmon1 Event Monitor is shown:

```
db2 drop event monitor mgrpay
```

4.5.4.1 Analyzing Event Monitor Output

This section discusses how to display Event Monitor output in DB2 and in R/3.

R/3 will create two Event Monitors as part of the default installation. These monitors are used to monitor deadlocks. The R/3 Event Monitors are started automatically with the start of the DB2 instance. There are two monitors located in /db2/AUS/db2event/edlmon1 and /db2/AUS/db2event/edlmon2. One writes data into files and the other performs continuous deadlock monitoring. You can see the result of the R/3 monitoring displayed in the R/3 Database Performance Monitor. From the screen shown in Figure 81 on page 139, when you click on **Deadlocks**, the second Event Monitor will be started, and the first one will be stopped. Data is written into a file that eventually updates the database table, sapr3.db6edlck. If a deadlock has been detected, it will be displayed.

There are two utilities available in DB2 to analyze Event Monitor data. The db2evmon utility is located in the misc subdirectory of the sqllib directory. It is a text-based tool which will read the event records and generate a report.

The output of the db2evmon utility is displayed on the screen by default. It is recommended to redirect the output to a file for analysis. Let's examine a portion of the R/3 Event Monitor output for a DEADLOCK-based Event Monitor. You can use one of the following two DB2 commands to display the output of the information that was written into the edlmon1 R/3 Event Monitor files:

```
db2evmon -path /db2/AUS/db2event/edlmon1
db2evmon -db AUS -evm edlmon1
```

The following is a portion of the output:

```
First connection timestamp: 08-24-1996 18:18:04.692906
```

```
2) Event Monitor Start Event ...
```

```
Start time: 08-27-1996 14:56:50.973608
```

```
3) Connection Header Event ...
```

```
Application Id: *LOCAL.db2aus.960824231804
```

```
Sequence number: 0001
```

```
DRDA AS Correlation Token: *LOCAL.db2aus.960824231804
```

```
Authorization Id: SAPR3
```

```
Execution Id: ausadm
```

```
Application Program Name: disp+work
```

```
Client NNAME:
```

```
Client product Id: SQL02011
```

```
Client Database Alias: AUS
```

```
Client Process Id: 18932
```

```
Agent Id: 29920
```

```
Application codepage id: 819
```

```
Application country code: 1
```

```
Application client platform: AIX
```

```
Application client comms protocol: Local
```

```
Connection timestamp: 08-24-1996 18:18:04.692906
```

This example did not detect a deadlock situation. You can tell when the Event Monitor was started. Some of the information we are able to obtain is the application ID, the authorization ID, codepage of the application and connection, timestamp.

Using an Event Monitor to capture deadlock information is just one use of Event Monitors. For example, you may monitor every SQL statement that is issued against a database.

4.5.4.2 Event Analyzer

If you have gathered Event Monitor data as described in the previous section, you may also use the DB2 Event Analyzer to analyze the data. The Event Analyzer is an alternative to the db2evmon tool.

The Event Analyzer displays the Event Monitor records that have been previously collected. To invoke the Event Analyzer, issue the following command:

```
db2eva -evm event_name -db database_name
```

If you do not include the database name and monitor name, you will be prompted for the path or location of the event records. The Event Analyzer will only display previously captured data. Therefore, there is no overhead in using the Event Analyzer as there is when using the Snapshot Monitor. The Event Analyzer may not contain the desired information as it can only display the event records that have been previously captured.

Chapter 5. Backup and Recovery

This chapter looks at how the BACKUP and RESTORE utilities are used to protect the data in your R/3 DB2 database in the event of a failure. With any relational database system, the possibility of error exists. This error could result in the loss of data or even the loss of the database. If you encounter such an error, you need to be able to recover. That is, you need to be able to bring the database back to a state of consistency so that applications and users can continue to work. The goal of recovery is to place the database into a state of consistency prior to any system error so that users and applications can continue with a minimum amount of time loss and so that the integrity of the data is protected.

In order to discuss backup and recovery, we need to introduce the concept of logging in a relational database system. Specifically, we examine how logging is implemented in an R/3 database. We also look at how log files are used in a recovery situation. There are two database configuration parameters that we discuss, LOGRETAIN and USER EXIT. In a typical R/3 production environment, you have many gigabytes of storage to protect in your DB2 database. The log files that are used in backup and recovery operations need to be managed by the R/3 database administrator. More than likely, you will use another product to assist in this task, such as ADSM. We therefore discuss how ADSM can be configured to be used in an R/3 DB2 environment to assist with the management of log files.

This chapter is outlined as follows:

- Logging considerations
- Configuring ADSM
- DB2 BACKUP and RESTORE utilities

5.1 Logging Considerations

It would be impractical, from a performance point of view, for a database system to write out changes in data for each user whenever they were made. Most, if not all, relational database systems batch these changes so that data modified by several users only has to be written once. This allows the database manager to determine the most efficient time to write the changes to permanent storage. The problem with this implementation is how to recover from a system crash. Since the changes were not written out immediately, there is no way to recover them. This is the concept behind logging. Logs are files that are used by DB2 to ensure the integrity of your database even when the system crashes due to some unforeseen problem, such as a power failure.

DB2 has implemented a write-ahead logging scheme to ensure the integrity of your data. The basis for write-ahead logging is that when a transaction is made which deletes, inserts, or updates any data in the database, the changes are first written to the log files. When the transaction performs a commit, DB2 ensures that all log files required for replay are written to disk. In case of a mishap such as a power failure, for example, the log files are used to bring the database back to a consistent state. All committed transactions are redone, and all uncommitted transactions are rolled back.

All databases have log files associated with them. Log files have a predefined length. Therefore, when one log file gets filled, logging continues in another log file.

5.1.1 Types of Logging

There are two types of logging in DB2:

- Circular logging
- Archival logging

5.1.1.1 Circular Logging

With this type of logging, log files are used in sequence. They can be reused when all units of work contained within them are committed or rolled back. The committed changes are reflected on the disks supporting the database. Circular logging uses two types of log files:

- Primary log files
- Secondary log files

Primary log files are pre-allocated, but secondary log files are only allocated when necessary. If the database manager requests the next log in the sequence and it is not available for reuse, a secondary log file is allocated. Secondary log files will be allocated until the primary log files becomes available for reuse or the number of secondary logs permitted for allocation is exceeded. Secondary log files are de-allocated once the database manager determines that they are no longer needed. Primary log files are allocated when the database becomes active.

The number of primary and secondary log files is determined by the database parameters LOGPRIMARY and LOGSECOND.

Databases configured with circular logging are recoverable to the point at which the last backup was taken. All work done on the database since the last backup was taken is lost when the database is restored.

Post-Installation Step in R/3

As a post-installation step in R/3, the logging type is changed from circular logging to archival logging. You must perform a full database backup whenever you switch from one logging type to another. It is recommended that you perform this switch prior to any users connecting to the database on your R/3 system. Circular logging is not recommended in a production R/3 environment.

5.1.1.2 Archival logging

Archival logging is the log-file management technique where log files are archived when they become inactive.

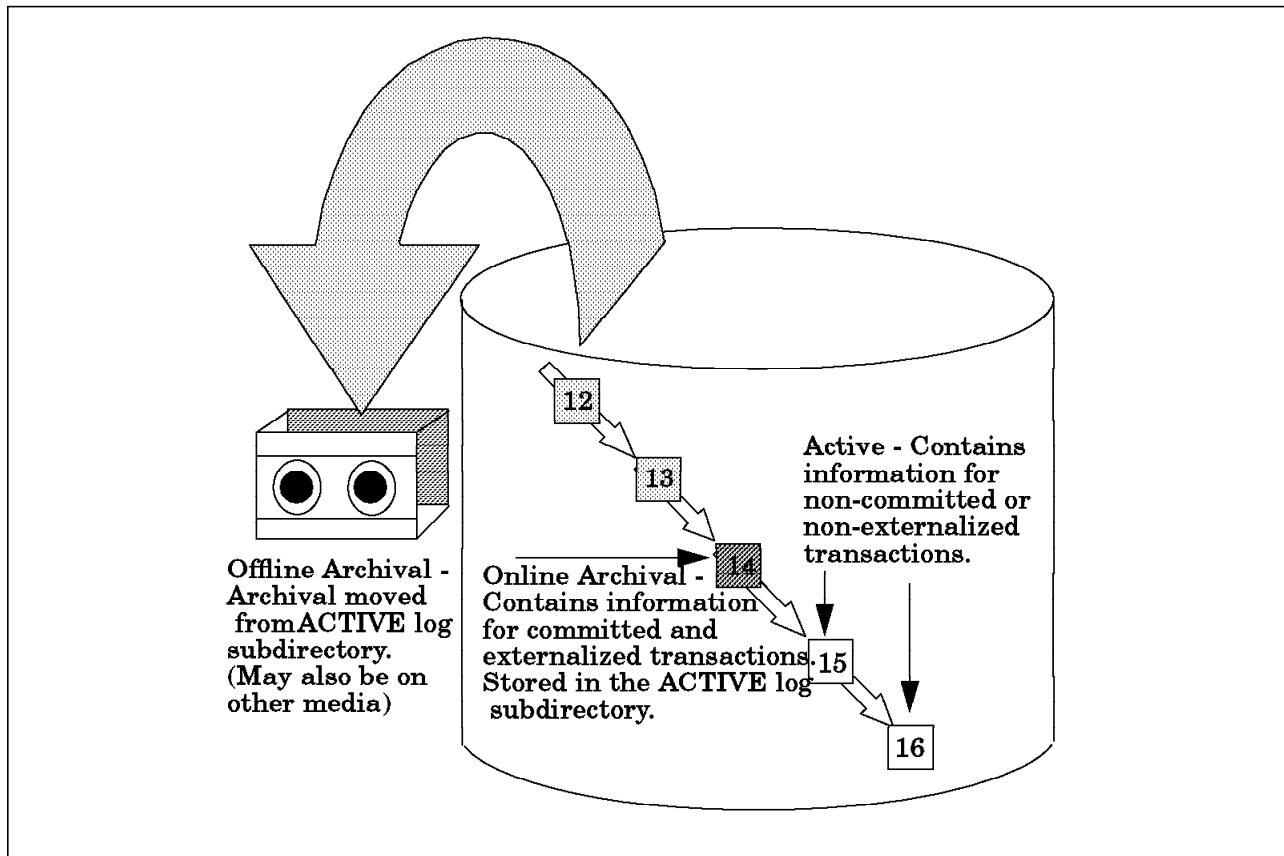


Figure 89. Archival Logging

There are three types of log files associated with this method. They are:

1. Active (Numbers 15 and 16 in Figure 89)

These files contain information related to transactions that have not yet committed or rolled back. They also contain information about transactions that have been committed but whose changes have not yet been written to the database files. In the R/3 default installation, these files are written to `/db2/<SAPSID>/log_dir`.

2. Online Archival (Number 14 in Figure 89)

These files contain information related to complete transactions no longer required for crash recovery protection. They are called "online" because they reside in the same subdirectory as the active log files. These files are written to `/db2/<SAPSID>/log_dir` in the default R/3 installation.

3. Offline Archival (Numbers 12 and 13 in Figure 89)

These files have been moved from the active log file directory. In the SAP R/3 system environment, two options are provided for offline archival: disk and tape. As a post installation step, you select whether the offline archival logs are sent to disk or tape by choosing the user exit program that your system uses. The file `/db2/<SAPSID>/.db2uexit` is modified to indicate whether offline archives are moved to tape or to disk.

Two configuration parameters allow you to configure a database for archival logging:

- LOGRETAIN

- USEREXIT

When the LOGRETAIN database configuration parameter is enabled, log files are not deleted when they become inactive. When the USEREXIT database configuration parameter is enabled, the database manager calls a program named /db2/<SAPSID>/db2uexit each time a log file is no longer needed for log writes. The database name and path of the log files are passed to the program.

User exits allow you to use the R/3 USER EXIT program to interact with storage devices that are not directly supported by the operating system, such as ADSM.

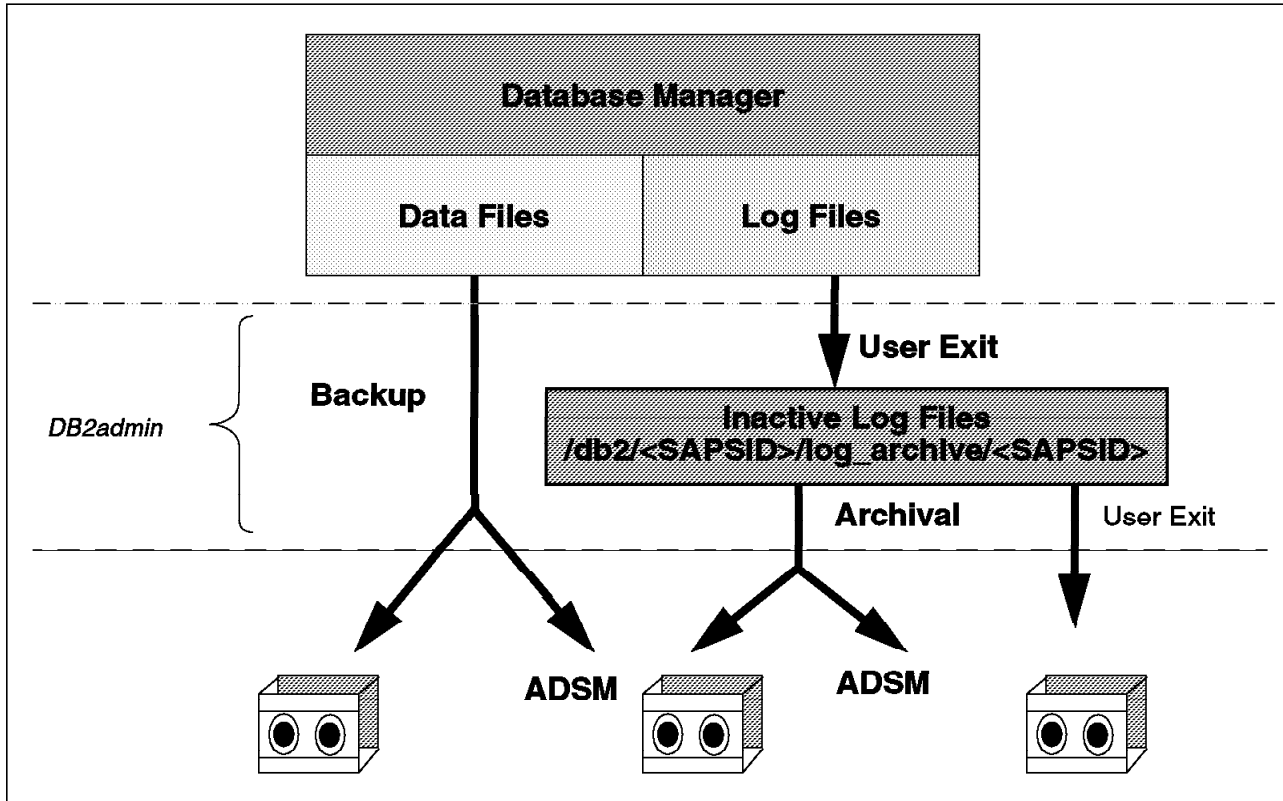


Figure 90. Overview of Database Backup and User Exit

Figure 90 gives an overview of database backup with USER EXIT enabled. Data files (which are database and tablespace or a selection of tablespace files) are collected by the database manager. Here, the backup is invoked from the DB2admin tool. These data files are then written to tape or to an ADSM Server.

If a log file is full, it will be deactivated. (This is known as an online archive log file.). Then, the user exit program for R/3, db2uexit, is called. This program copies the files to a save directory. The directory path is /db2/<SAPSID>/log_archive/<SAPSID>. The R/3 database administrator uses the archive function found in the DB2admin tool to copy the files to tape or to an ADSM Server.

Archival logging is not the default logging method. It is enabled as a post-installation step. It is the only method that allows you to perform roll forward recovery. SAP requires that you enable archival logging for a production system.

The R/3 implementation provides one additional level for log files. If your offline archive files are saved to disk, you can move these files to an additional backup media: tape or ADSM. Once these files have been saved to tape or to ADSM, you can delete them from the offline archival directory. Please see the section 5.5, "Managing Archive Log Files" on page 230 for more information on saving offline archive logs.

5.1.2 Log Management Configuration Parameters

There are a series of parameters that are used by the logging process with DB2. If the values of any of these parameters are changed, you must first disconnect all users from the database before any changes take place at the database level. However, the database manager (DB2 instance) does not have to be stopped. These parameters are set during the installation of SAP R/3. The parameters are:

- LOGBUFSZ

The log buffer size parameter determines how much database shared memory will be allocated to buffer log records before they are written out to disk. The value for this parameter indicates the number of 4 KB pages up to a maximum of 128 4-KB pages. At installation of R/3 with DB2, this value is set to 64 4-KB pages.

- LOGFILSIZ

This value determines the number of pages to be allocated when a log files is requested. Combined with LOGPRIMARY and LOGSECOND, this value determines the disk space required to support logging. It is measure in units of 4 KB pages. The value set at R/3 installation is 5000.

- LOGPRIMARY

This value represents the number of primary log files that are allocated to support database logging. Each one is LOGFILSIZ in size. This value is set to nine (9) in the R/3 installation.

- LOGSECOND

This parameter specifies the maximum number of secondary log files that can be created when needed by the system. When the primary log files become full, the secondary log files of size LOGFILSIZ are allocated one at a time as needed. The default number of secondary log files for R/3 is one.

- NEWLOGPATH

The default subdirectory for the log files is defined in SQLLOGDIR, which is a subdirectory of the database directory. This parameter identifies a new path for placement of log files. In R/3, the default path is /db2/<SAPSID>/db2<SAPSID>/SQL0001/SQLOGDIR/. This is a link to /db2/<SAPSID>/log_dir.

- SOFTMAX

When a database is restarted, the log control file is used to determine which records from the log file need to be applied to the database. This value should be set to a percentage of the value of LOGFILSIZ. SOFTMAX determines if a log control file should be written more frequently than the default to the disk. The default is that the control file is always written to disk when it is full. This is known as a hard checkpoint. The R/3 installation sets this value to 100.

- LOGRETAIN

This indicates if the log retention or archive logging is to be used. During the installation of R/3, this value is set to NO. One of the post-installation tasks is to change this value to YES and execute an offline database backup.

- **USEREXIT**

Setting this value enables archive logging and roll forward recovery. At R/3 installations this value is set to NO. A post installation task is to set this value to YES and perform a database offline backup.

- **MINCOMMIT**

This value indicates that a grouping of commits issues by multiple applications is to be attempted if the value is set greater than 1. This parameter is set to 1 at R/3 installation. However, the recommended value for an R/3 database is 3.

- **OVERFLOWLOGPATH**

This is an alternative log path to search for archived logs.

- **NUM_IOCLEANERS**

The number of asynchronous page cleaners for the database. The page cleaners examine the buffer pool to look for pages that need to be written to disk. In this way, the regular database agents can more likely find empty space in the buffer pool. This parameter is set to 3 in the standard R/3 installation.

One way of checking the parameters for database configuration is with the DB2 GET DATABASE CONFIGURATION command.

```
db2 get database configuration for aus
```

If we issue the GET DATABASE CONFIGURATION command for the aus database, the output shows some of the parameters that have been set with the R/3 installation that relate to backup and recovery. The following is only a part of the output:

```

Backup pending                               = NO
Database is consistent                       = NO
Rollforward pending                         = NO

Log retain for recovery status               = YES
User exit for logging status                 = YES
...

Log file size (4KB)                         (LOGFILSIZ) = 5000
Number of primary log files                  (LOGPRIMARY) = 9
Number of secondary log files                (LOGSECOND) = 1
Changed path to log files                    (NEWLOGPATH) =
Path to log files                            = /db2/AUS/db2aus/SQL00001/SQLLOGDIR/
Next active log file                         = S0000058.LOG
First active log file                        = S0000051.LOG

Group commit count                          (MINCOMMIT) = 1
Percent log file reclaimed before soft ckcpt (SOFTMAX) = 100
Log retain for recovery enabled               (LOGRETAIN) = ON
User exit for logging enabled                 (USEREXIT) = ON

Auto restart enabled                         (AUTORESTART) = ON
Index re-creation time                       (INDEXREC) = SYSTEM (RESTART)
Default number of loadrec sessions           (DFT_LOADREC_SES) = 1
Recovery history retention (days)           (REC_HIS_RETENTN) = 366

```

5.1.3 Terminology in DB2 and R/3

This section discusses some of the terminology used by DB2 and R/3 when discussing logging and backup/recovery activities.

As discussed previously in this chapter, DB2 uses log files as a way of protecting data. The R/3 DB2admin tool also uses the concept of log files. They are called history log files. These history log files are written by the DB2admin tool to store return codes from operations that the R/3 DB2 database administrator has executed. These history log files for the database administrator are also referred to as protocol files. Using the DB2admin tool, an R/3 DBA can edit what are called protocols in R/3. There are protocol files that are used to log DB2admin backup and recovery activities. There are five different files that collect information about various backup and restore activities. These protocol files store return codes that the R/3 database administrator can examine after executing specific backup and restore operations. They are discussed in more detail in 5.3.15.3, "R/3 Protocol Files" on page 218.

5.1.3.1 R/3 User Exit and Archive Profiles

Another option you have in the DB2admin tool is to edit profiles. You must be in the graphical version of DB2admin to edit. There are two profiles within R/3 that you can edit: the user exit profile and the archive profile. You can also directly edit these profiles from an AIX shell by using an editor. To edit from the graphic mode of SMIT, make sure that the **EDITOR** variable is set in the shell environment.

The two profiles that are available to you are the user exit profile and the archive profile. The user exit profile contains information that is passed to the User Exit program. For example, it specifies some paths that will be used, whether the User Exit program saves to disk or tape and the tape device name.

The R/3 user exit profile is named `.db2uext` and is located in the database administrator's home directory. An example of `.db2uexit` is as follows:

```
# general parameters for user exit
#-----
export UEXIT_PROG=brdb6u2d      # user exit program to use: brdb6u2d = to disk
                                #                               brdb6u2t = to tape
export AUDIT_ACTIVE=1          # enable audit trail logging
export ERROR_ACTIVE=1         # enable error trail logging
export AUDIT_ERROR_PATH=$INSHOME/errors
                                # path of directory for audit/error log files
                                # ATTENTION: directory must (!) exist
export AUDIT_ERROR_ATTR=a      # append to text file

# parameters for UEXIT_PROG=brdb6u2t (to tape)
#-----
export TAPE_DEV=/dev/rmt0      # tape device name
export TAPE_PERM=0666         # tape permissions
export TEMP_DIR=/tmp/UserExits # intermediate dir to store a LOG to be archived
                                # ATTENTION: directory must (!) exist

# parameters for UEXIT_PROG=brdb6u2d (to disk)
#-----
export ARCHIVE_PATH=$INSHOME/log_archive
                                # Log_Archive for archive
export RETRIEVE_PATH=$INSHOME/log_archive
                                # Log_Archive for retrieve
export ARCHIVE_MODE=C          # C : store in compressed format (*.Z)
                                # N : store in non-compressed format
```

In our example, the User Exit program that R/3 uses is `brdb6u2d`. This means that files will be saved to disk, not tape. Logging of the activity, both normal operations and errors will be logged. The path for archiving and retrieving archived log files is set. Also, the archived log files will be stored in a compressed format (*.Z) to save disk space.

The other profile that you can edit (either with the graphical mode of DB2admin or from an AIX shell) is the archive profile. The archive profile contains several values for the archiving of offline log file. Information such as the tape device name, the volume_archive ID, the expiration period for written tapes, and the number of time a tape may be written is stored here. The file is found in the R/3 database administrator's home directory under `db2`. The file is called `init<SAPSID>.sap`.

5.1.4 Enabling Archive Logging

A full, offline backup of the database must be performed after the database parameters for archival have been set to off. If archiving is turned on before the complete offline backup has been taken, you will not be able to connect to the database and R/3 will not start. After the initial installation, archiving must be turned on as a post-installation step.

Note that you must be logged into AIX as the database administrative user ID, `db2<SAPSID>`. The SMIT screen for this entry works differently than most SMIT screens. The values for `LOGRETAIN` and `USEREXIT` should be set to ON for archiving. To turn these parameters on, enter the value in the New value of `LOGRETAIN` field and in the New value of `USEREXIT` field. You will not be able to toggle the values in the `LOGRETAIN` status field and the `USEREXIT` status field.

To configure the database for archival, select the following from the SMIT screen:

- **Applications**

- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Manage Log File**
- **Configure Database Parameters for Archival**
- Select the database by clicking on the entry in the SMIT pop-up screen.
- Change **New value of LOGRETAIN** and **New value of USEREXIT** to **ON** for archiving. As shown in Figure 91.
- Enter **OK** in the confirmation pop-up.

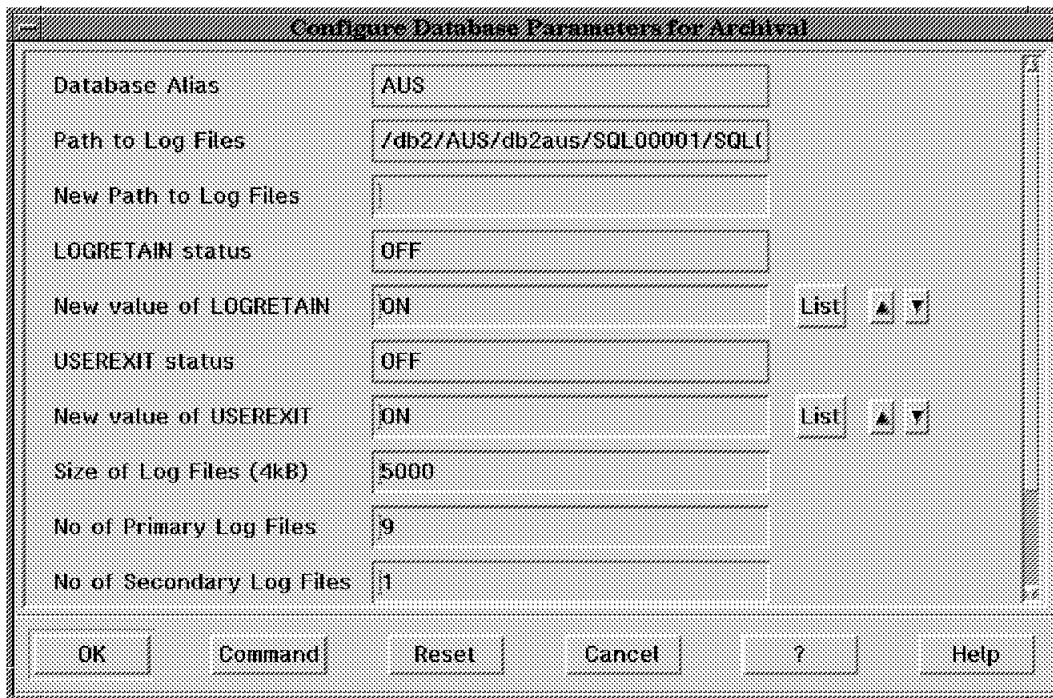


Figure 91. Enable Archive Logging

The only parameter we have changed in Figure 91 is LOGRETAIN. The path of the log files that is shown here is /db2/AUS/db2aus/SQL00001/SQLLOGDIR. This is a link to the file system where the log files are stored, /db2/AUS/log_dir. Remember that AUS is our <SAPSID>.

You must perform a full, offline backup whenever LOGRETAIN is enabled. This is described in more detail in 5.3.3, "Performing an Offline Backup" on page 193.

5.1.5 Log File Information

DB2 uses a control file to determine the status of log files. The control file identifies the active log file with the "lowest" name.

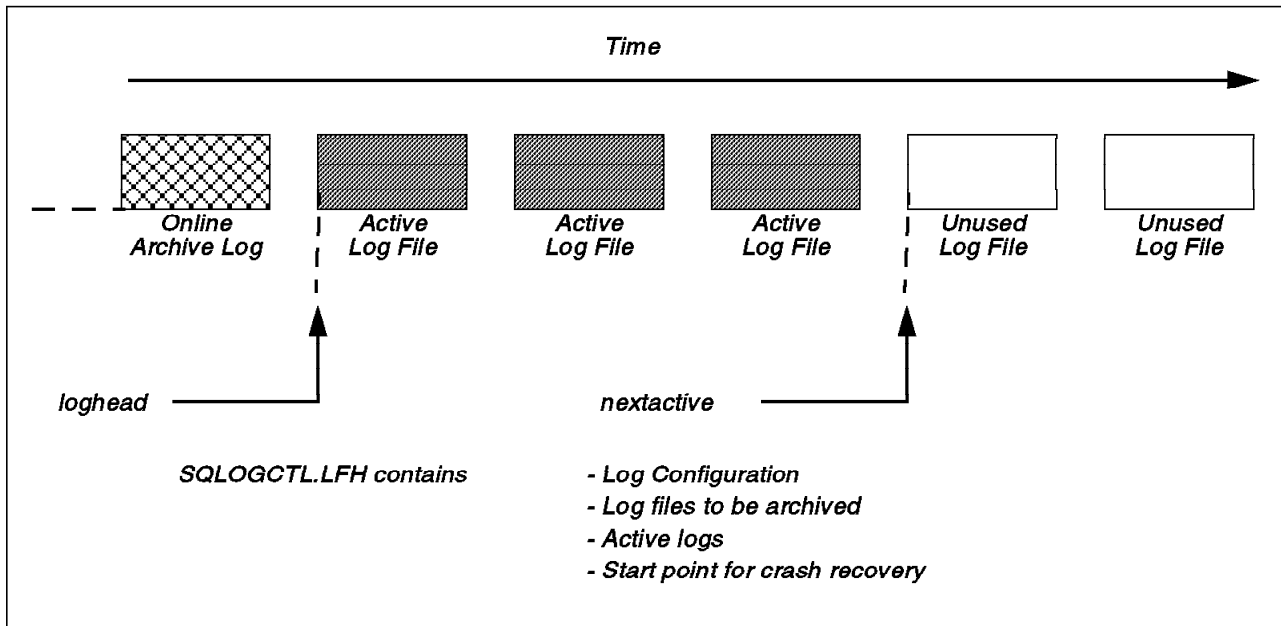


Figure 92. Log File Information

Figure 92 shows the log file information over time. The control file also identifies the name of the next log file to be used. This file is called the nextactive. When we issued the get database configuration command for the R/3 database, we saw that the nextactive log file was S00000058.LOG. The values for loghead and nextactive can be valuable to the R/3 database administrator. (The control file is identified for information purposes only. It is not in a readable format and should not be edited.)

Log files with names that are "less" than the loghead are archive files. They are not required for crash recovery and could be moved to a different media. Crash recovery is discussed in 5.1.7, "Log File Usage" on page 165.

The nextactive value, used in conjunction with loghead, can be used to determine the number of active logs currently allocated. If the number allocated becomes abnormally large, an application may not be committing in a timely basis.

5.1.6 Naming of Log Files in R/3

This section discusses the naming of log files in R/3. The naming convention for log files is as follows:

Sxxxxxxx.LOG

where xxxxxxx is a number starting from 0000000 to 9999999. The numbers are assigned by the database manager. By default, log files are located in SQLLOGDIR, which is a subdirectory of the database directory. In R/3, the directory for active log files should be a separate file system, preferably on a disk that is separate from the database files. In our configuration, the file system was /db2/AUS/log_dir. The archived log files were placed on a disk separate from the online log files and database files. The file system was /db2/AUS/log_archive/AUS.

5.1.7 Log File Usage

The log files are used for the following situations:

1. Rollback

The SQL ROLLBACK statement uses log files to terminate a unit of work and back out database changes made by that unit of work

2. Crash Recovery

If your system experiences a disk crash, power outage, or other kind of failure, crash recovery is needed to bring the database back to a consistent usable state. Crash recovery consists of two phases. The first phase reapplies all transactions to the database regardless of whether they were committed or not. This phase completes when the end of the active log files is reached.

The second phase is to roll back all uncommitted transactions. The database configuration parameter that sets the crash recovery is AUTORESTART. In the R/3 installation, this parameter is set to ON.

3. Roll Forward Recovery

Roll forward recovery applies transactions recorded in the database log files. The ROLLFORWARD DATABASE command is invoked after an online database or online/offline tablespace backup has been restored or if any tablespaces have been taken offline by the database manager due to a media error.

If an I/O error is encountered while trying to read from or write to disk, a tablespace in which the page resides is disabled and placed in a "Roll Forward Pending" state. You should be able to clear the Roll Forward Pending state by rolling forward the log files.

Restore is the first phase of a complete roll forward recovery of a database or tablespace. After a successful database restore, a database that was configured for roll forward recovery at the time a backup was taken enters a Roll Forward Pending state. It is not usable until the ROLLFORWARD DATABASE command has been executed successfully. If the restore used is a tablespace-level backup, the tablespace restored enters a Roll Forward Pending state.

When the ROLLFORWARD DATABASE command is issued, if the database is in a Roll Forward Pending state, the database is rolled forward. If the database is not in a Roll Forward Pending state, all tablespaces in the database that are in the roll pending state are processed.

Another database restore is not allowed when the roll forward process is executing. Note that if you restore from a full, offline database backup image, you can bypass the Roll Forward Pending state during the recovery process. The RESTORE DATABASE procedure gives you the option to use the restored database immediately without rolling the database forward.

You cannot bypass the roll forward phase when recovering at the tablespace level or if you restore from a backup image that was created using the ONLINE option of the backup operation.

5.1.8 DB2 R/3 Logging Summary

This section talked about logging in DB2 and R/3. Some of the terminology is the same between DB2 and R/3; however, the meaning is slightly different. The two database parameters that were discussed in detail were LOGRETAIN and USEREXIT. In a production R/3 system, LOGRETAIN must be set to ON.

Disabling LOGRETAIN

The only time you may consider disabling LOGRETAIN is when performing an upgrade to your R/3 system. Disabling LOGRETAIN may save time during the upgrade procedure. As soon as the upgrade process completes, you should immediately switch LOGRETAIN to ON and perform a full, offline database backup.

USEREXIT should also be set to ON. USEREXIT is the parameter that calls a program that assists you in managing the archived log files. These archived log files are necessary for any database or tablespace recovery procedure. The archived logs have a variable size. The size of the archived logs are dependent on the database parameter LOGFILSIZ and the compression rate. In general, you need sufficient disk space to store the active log files and archived log files. Remember that these two sets of files should be in separate file systems, preferably on separate physical disks. Archived log files can be managed and moved to tape or ADSM. See 5.5, “Managing Archive Log Files” on page 230 for more information.

5.2 Configuring ADSM for DB2

ADSM is IBM's solution to enterprisewide distributed storage management. It provides 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. It addresses the need for customer asset protection and data availability for distributed environments. The main functions of ADSM are backup and restore.

5.2.1 Main Components

The main components of ADSM are the backup/archive client, the administrative client, the server, and the application client.

5.2.1.1 Backup/Archive Client

The backup/archive client runs on the workstation and, depending on the platform, provides both a graphical user interface and command line interface. Although all clients are similar, they each have the look and feel of the platform on which they are running.

5.2.1.2 Administrative Client

The 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.

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, personal computer, or

mainframe. An administrative client passes commands through an administrative command line, or in some cases, a graphical user interface.

5.2.1.3 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.

The key components of the ADSM Server are the storage pools where 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. The storage pools contain the client files that have been backed up or archived.

The ADSM 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 completed successfully
- The activity log that contains the messages generated by the server
- Information about ADSM volumes
- Information used to locate files that reside in storage pools

The recovery log is used to help maintain the integrity of the ADSM database. It keeps track of all changes made to the database.

5.2.1.4 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 object from an ADSM Server.

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.

5.2.2 Installing ADSM

Both the ADSM Server and ADSM client must be installed to use ADSM for back ups. Please see *ADSM/6000 Installing the Server and Administrative Client* (SH35-0136-01) for instructions on installing ADSM.

The ADSM Server requires approximately 40 MB in /usr and 40 MB in /var or 80 MB in /usr. Some customizing must be done to be able to use ADSM for backups and restores. During R/3 installation, some of the client customizing is done for you.

5.2.3 Customizing the ADSM Server

The server should automatically be configured for you during the basic installation described in Chapter 3, "Basic Installation and Configuration" in *ADSM/6000 Installing the Server and Administrative Client*. If there was a problem during installation, see Chapter 4, "Manual Configuration for the ADSM Server on AIX" in *ADSM/6000 Installing the Server and Administrative Client* (SH35-0136-1). In most cases you will not need to configure ADSM manually.

Updating dsmserv.opt

Increase the COMMITimeout value in /usr/lpp/admserv/bin/dsmserv.opt. If you do not change this value, the Redirected Restore (discussed later in this chapter) will not complete successfully and will leave the database in an inconsistent state. The COMMITimeout value in dsmserv.opt specifies the communication time-out value in seconds. We increased it from 60 to 6000.

5.2.3.1 Starting the ADSM Server

The ADSM Server is started automatically when the system is booted. To start the server in the background, change to the directory /usr/lpp/admserv/bin and enter the following command at the AIX command line:

```
nohup dsmserv quiet &
```

To start the server from the ADSM GUI (graphical user interface):

- If you are not already at the ADSM screen, shown in the figure, type `adsm` in any xterm window.
- From the main panel, double-click on the ADSM Server icon.
- Press **Begin Session**.

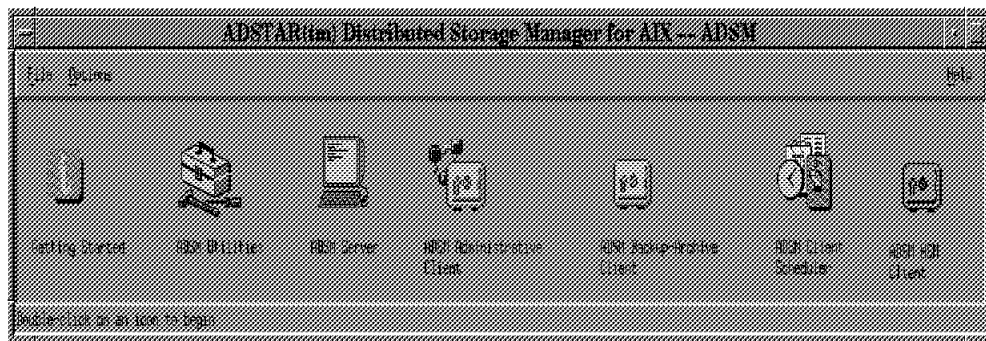


Figure 93. ADSM Graphical Interface

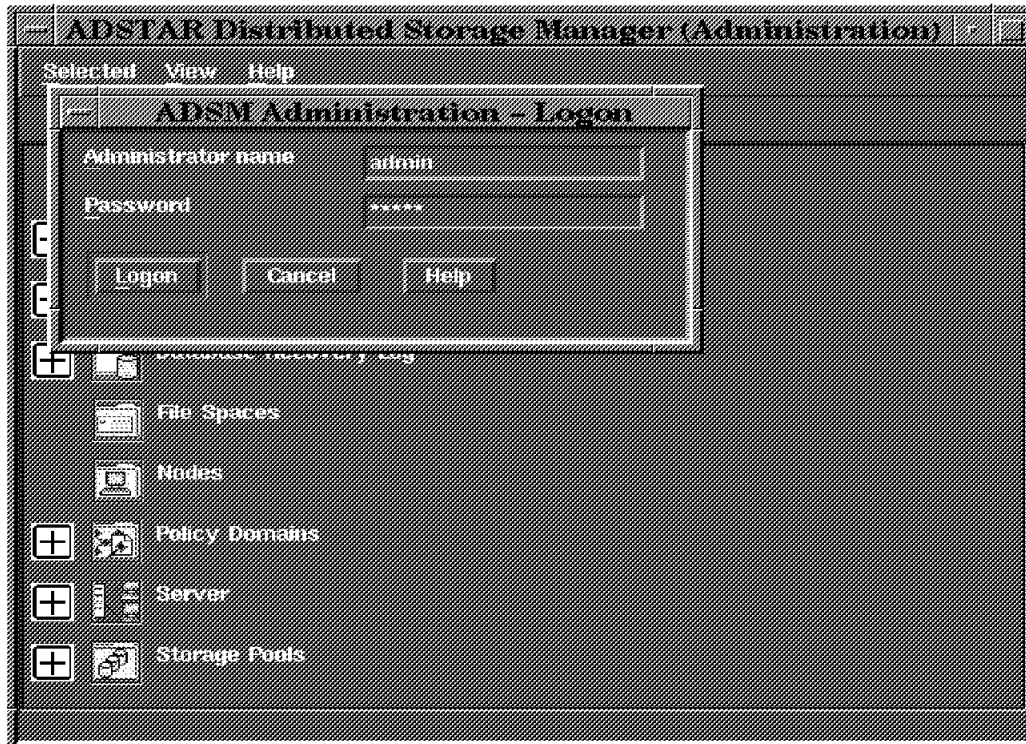


Figure 94. ADSM Administrative Client GUI Login Screen

5.2.3.2 Using the Administrative Client

There are two ways you can start the administrative client:

1. Start the administrative client's graphical interface by using the command:
`dsmadm &`

You can also start the administrator's graphical interface by double-clicking on the **ADSM Administrative Client** icon on the ADSM graphical interface shown in Figure 93 on page 168. Enter your administrator user ID and password. The default administrator user ID after installation is `admin`. The password is also `admin`.

2. Start the Administrative Client command line interface by entering the following command at the AIX prompt:
`dsmadm`

Enter your administrator user ID and password. The default administrator user ID after installation is `admin`. The password is also `admin`. This is displayed in Figure 95 on page 170.

```
Korn Shell
root@f01n07e:/ [188] dsmadm
ADSTAR Distributed Storage Manager
Command Line Administrative Interface - Version 2, Release 1, Level 0.0
(C) Copyright IBM Corporation, 1990, 1995, All Rights Reserved.

Enter your user id: admin
Enter your password:
ANS5100I Session established with server R3SERV: AIX-RS/6000

adsm> █
```

Figure 95. ADSM Administrative Client Command Line Interface

5.2.3.3 Registering an Administrator

When the server is installed, an administrative client is automatically set up with the name admin and password of admin. An administrator user ID should be set up for the person who will be setting up and doing the database and R/3 backups.

- To register and authorize an administrator from the command line interface, enter the following commands:

1. Register an administrator with the REGISTER ADMIN command. For example, to register an administrator with ID Laura and password sapr3, enter:

```
adsm> register admin laura sapr3
```

2. Grant system privilege to administrator Laura by entering:

```
adsm> grant authority laura classes=system
```

Now Laura has unrestricted privilege for the ADSM Server. To revoke unrestricted privilege from the default administrative userids, enter the following commands:

```
adsm> revoke authority admin classes=system
```

```
adsm> revoke authority server_console classes=system
```

- To register an administrator using the graphical interface:
 1. Start the ADSM Administrative Client by double-clicking on the **ADSM** icon on the ADSM GUI or by entering **dsmadm &** from AIX. Enter the administrator's ID and password.

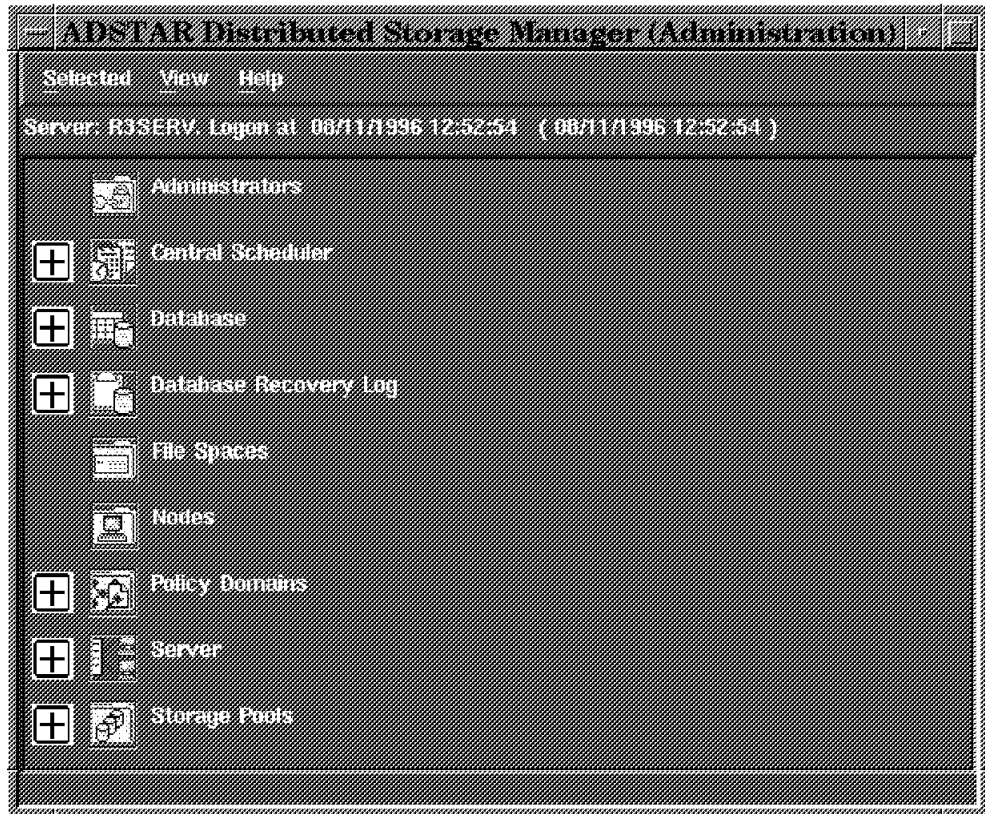


Figure 96. ADSM Administrative Client Graphical Interface

2. Double-click on the **Administrators** icon.



Figure 97. Defined ADSM Administrators

3. Select **Edit** from the menu bar.
4. Select **Add** from the pull-down menu.

5. Enter the User ID in the Administrator name field.
6. Press the **Privilege** tab.
7. Select **System**.

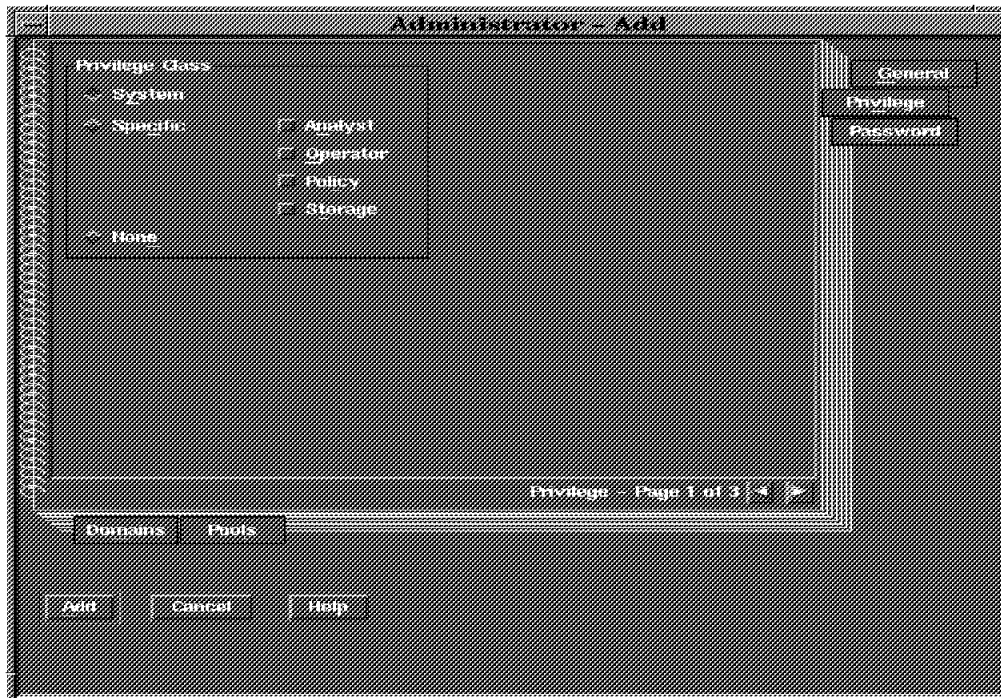


Figure 98. Adding an Administrator

8. Press the **Password** tab.
9. Enter the administrator password.
10. Press the **Add** button at the bottom of the screen.

To revoke unrestricted privilege from the default administrative user IDs from the graphical interface:

1. You must log in to an administrative User ID other than the one you wish to restrict. For example, if you wish to restrict or delete the default admin user ID, you must log in as an administrative user other than admin.
2. Double-click on the **Administrators** icon.
3. Select the User ID you wish to restrict by double-clicking on the icon.
4. Press the **Privilege** tab.
5. Select the desired privilege class for this User ID.
6. Press the **Apply** button at the bottom of the screen.

See Chapter 4, "Manual Configuration for the ADSM Server on AIX" in *ADSM/6000 Installing the Server and Administrative Client (SH35-0136-01)* for more information on registering an administrator and reducing administrative privilege.

5.2.3.4 Naming the Server

By default, the server name is ADSM. You can change the name of the server with the SET SERVERNAME command. For example, to change the name of the ADSM Server to r3serv, enter the following command on the command line interface:

```
adsm> set servername r3serv
```

The new name will be displayed to any client that accesses that server. The dsm.opt and dsm.sys files on the client should be changed to reflect the new server name. You will modify these files when you configure the client.

5.2.3.5 Registering a Backup/Archive Client

You must register a backup/archive client with the ADSM Server to allow the client to backup and restore files to the ADSM Server. To register a client node or workstation with the command line interface, use the REGISTER NODE command. Register a backup-archive client node with the following command:

```
adsm> register node <nodename> <password>
```

For example, to register a client with node name alpha and password client, enter:

```
adsm> register node alpha client
```

To query the nodes already defined, enter the following command at the command line interface:

```
adsm> query node
```

To register a node using the graphical interface:

- Double-click on the **Nodes** icon.
- Select **Edit** from the menu bar.
- Select **Add** from the pull down menu.
- Fill in the node name.

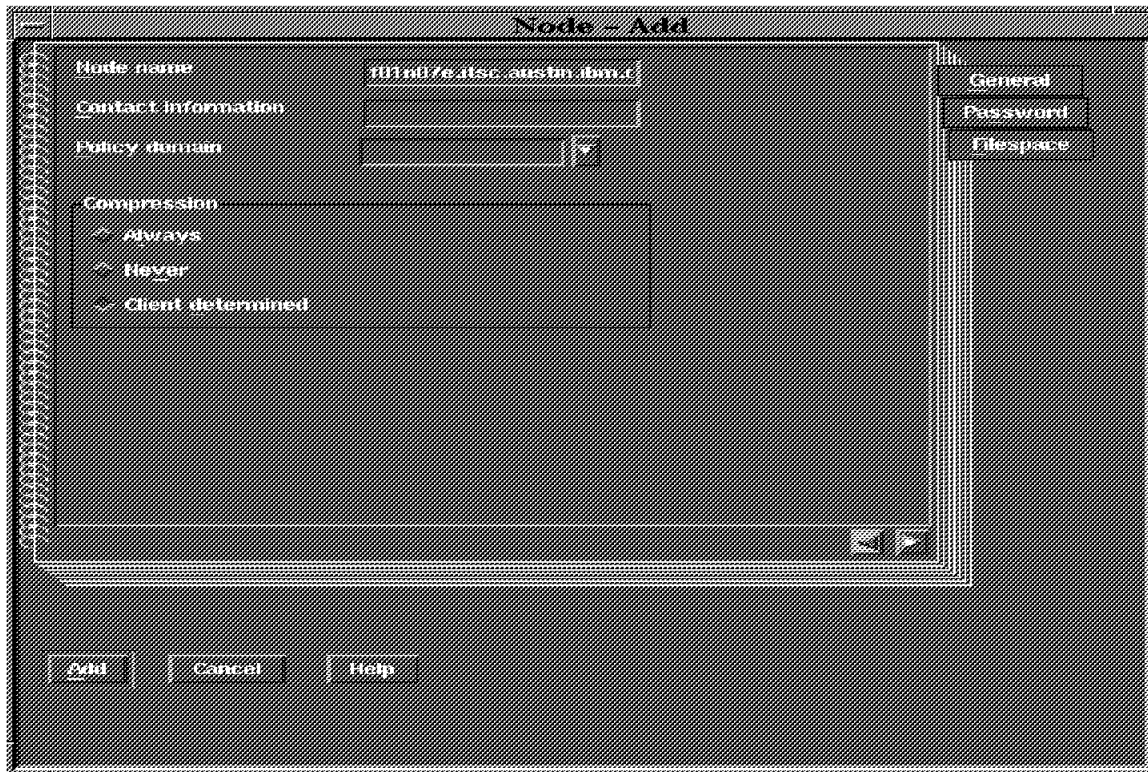


Figure 99. Registering a Client Node

- Press the **Password** tab.
- The screen shown in Figure 100 on page 175 appears and prompts you for a password.

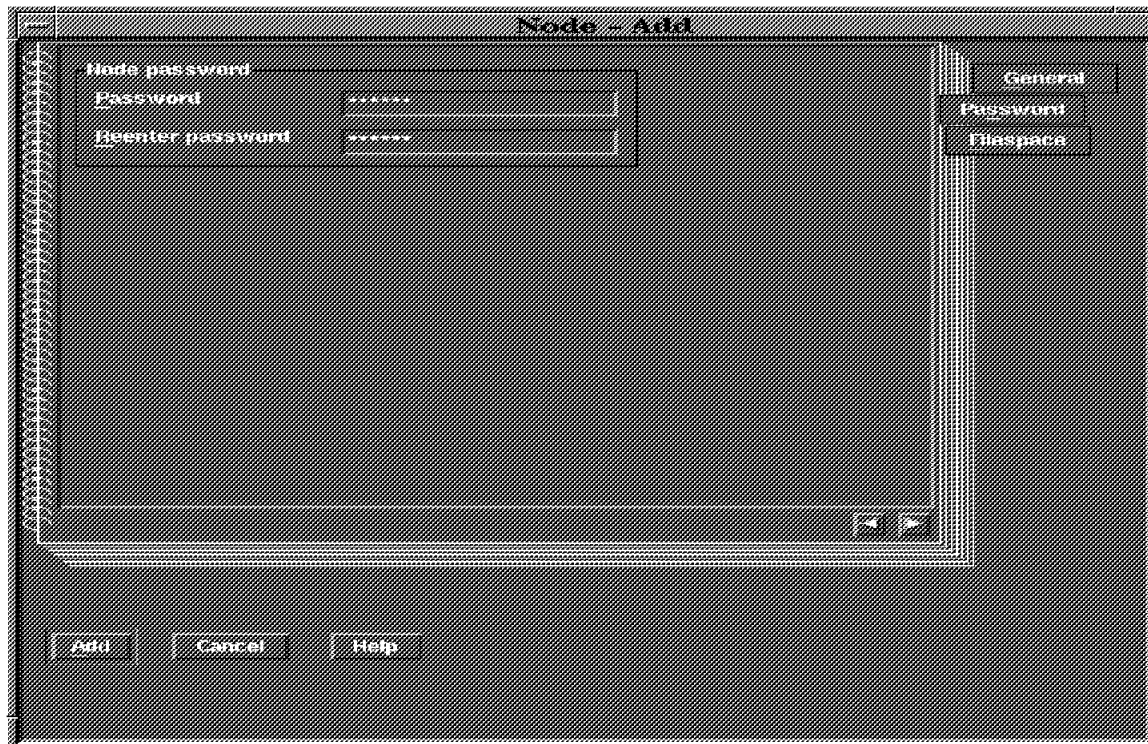


Figure 100. Setting the Password for a Client Node

- Fill in the password.
- Press the **Add** button at the bottom of the screen.

5.2.4 Customizing the ADSM Backup/Archive Client for R/3 with DB2

After the ADSM client is installed, some customization must be done to be able to use ADSM to back up the R/3 database. The customization required is described in this section.

1. The ADSM files are found in /usr/lpp/adsm/api/bin. Make sure you have the following files in the /usr/lpp/adsm/bin directory:
 - libApiDS.a - API Library
 - dscameng.txt - message file
 - dsgameng.txt - message file
 - dsmapitca - API trusted agent

We found that all the files except libApiDS.a were already in /usr/lpp/adsm/bin. We added a link from /usr/lpp/adsm/api/bin/libApiDS.a to /usr/lpp/adsm/bin with the following command:

```
In -s /usr/lpp/adsm/api/bin/libApiDS.a /usr/lpp/adsm/bin
```

2. Check if /usr/lib/libApiDS.a exists. If it does not, add a link from /usr/lpp/adsm/api/bin to /usr/lib/libApiDS.a
3. Check the environment variables for the db2<SAPSID> userid. Verify that DSMI_DIR, DSMI_CONFIG, and DSMI_LOG have been set. If they have not been set, see "Setting up an ADSTAR Distributed Storage Manager Client for AIX" in the *Database 2 Administration Guide for Common Servers, Version 2*, (S20H-4580-01).

4. Copy dsgameng.txt, dscameng.txt, and dsmapitca from /usr/lpp/adsm/bin to the directory path specified in DSMI_DIR.
5. Copy /usr/lpp/adsm/bin/dsm.opt.smp to dsm.opt in the directory path specified in DSMI_CONFIG. Modify the file as directed in the comments.
6. Copy /usr/lpp/adsm/bin/dsm.sys.smp to /usr/lpp/adsm/bin/dsm.sys. Modify dsm.sys as directed in the comments. Add the following line to dsm.sys:

Passwordaccess Generate

This will enable backups through the SMIT screens. You will need to set the password as described in the next step to be able to use this option. For more information on the options available in dsm.sys, see the file /usr/lpp/adsm/bin/options.doc.

7. Set the ADSM password. See "Setting up an ADSTAR Distributed Storage Manager Client for AIX" in *Database 2 Administration Guide for Common Servers, Version 2*, (S20H-4580-01) for instructions on establishing the ADSM password. You will need to do this to be able to run backups from the SMIT screens as db2<SAPSID>. To set the password, run the following command: /db2/<SAPSID>/sql1lib/adsm/dsmapipw

5.2.5 Setting up ADSM to Backup to Disk

In our R/3 environment, we used two nodes on an SP. Node 1 was an Application Server. Node 7 was the Database Server and the R/3 Central instance. R/3 was installed on external SSA disk. We used another RS/6000 with a large disk bank for the ADSM Server (Figure 102 on page 178).

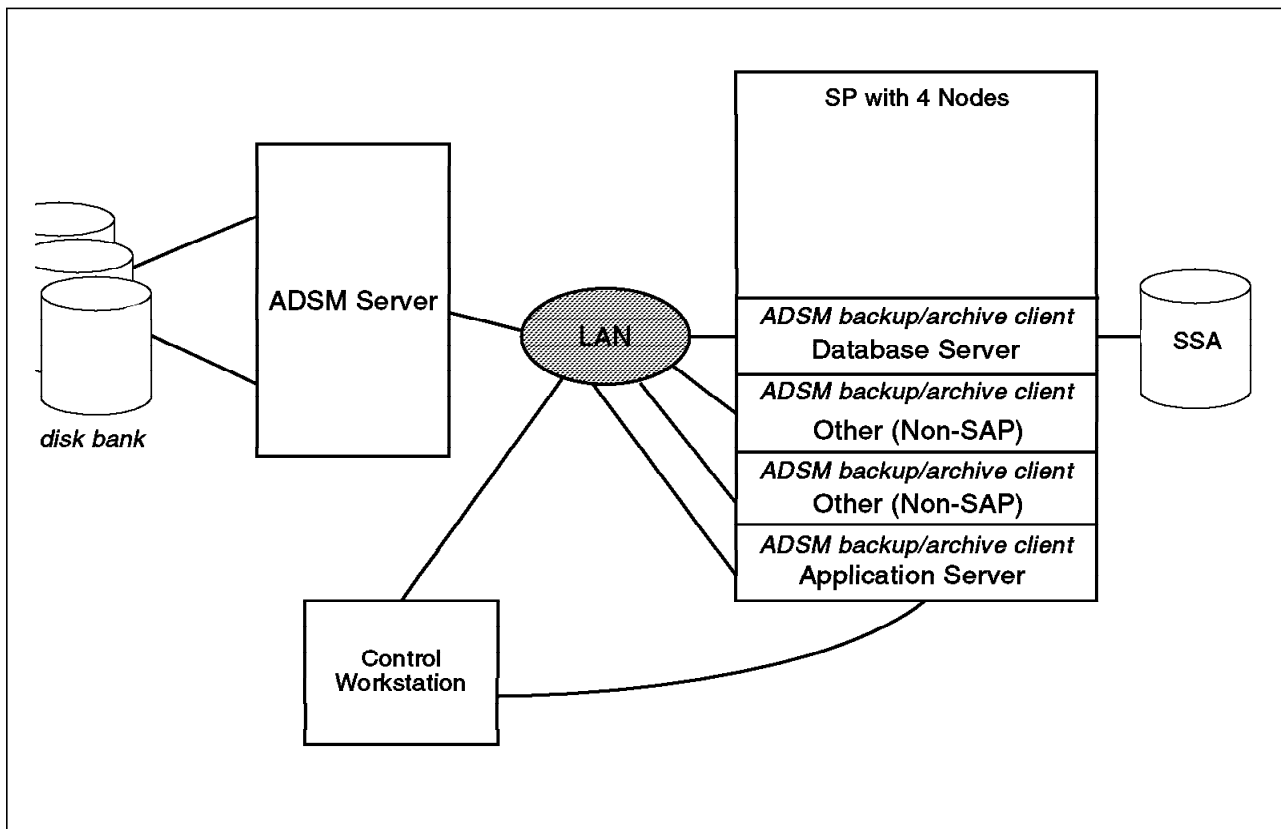


Figure 101. Sample ADSM Server and Client in an R/3 DB2 Environment

If you wish to set up your ADSM Server to save backups to disk, follow the steps in this section. For information on directing ADSM backups to other media, see *ADSM/6000 Installing the Server and Administrative Client*, (SH35-0136-01).

5.2.5.1 Setting up AIX Logical Volumes

Create AIX logical volumes on the ADSM Server and define them as ADSM volumes for use in the disk storage pools. You must be logged in as root to create AIX logical volumes. Use the SMIT command and select:

- **System Storage Management (Physical & Logical Storage)**
- **Logical Volume Manager**
- **Logical Volumes**
- **Add a Logical Volume**
 1. Fill in the volume group name field.
 2. Specify a name in the logical volume name field. For example: backup1.
 3. Provide number of partitions.

5.2.5.2 Defining the Logical Volume to ADSM

There are two ways to define the volumes in ADSM/6000:

- Using the command line interface. To set up a storage pool called DISKPOOL, enter the command:

```
adsm> define stgpool diskpool disk
```

- Using the graphical interface

Double-click on the **Storage Pools** icon.

Select the **Edit** option.

Select the **Add Primary Storage Pool** from the pull-down menu.

Enter the name of your storage pool as shown in Figure 102 on page 178.

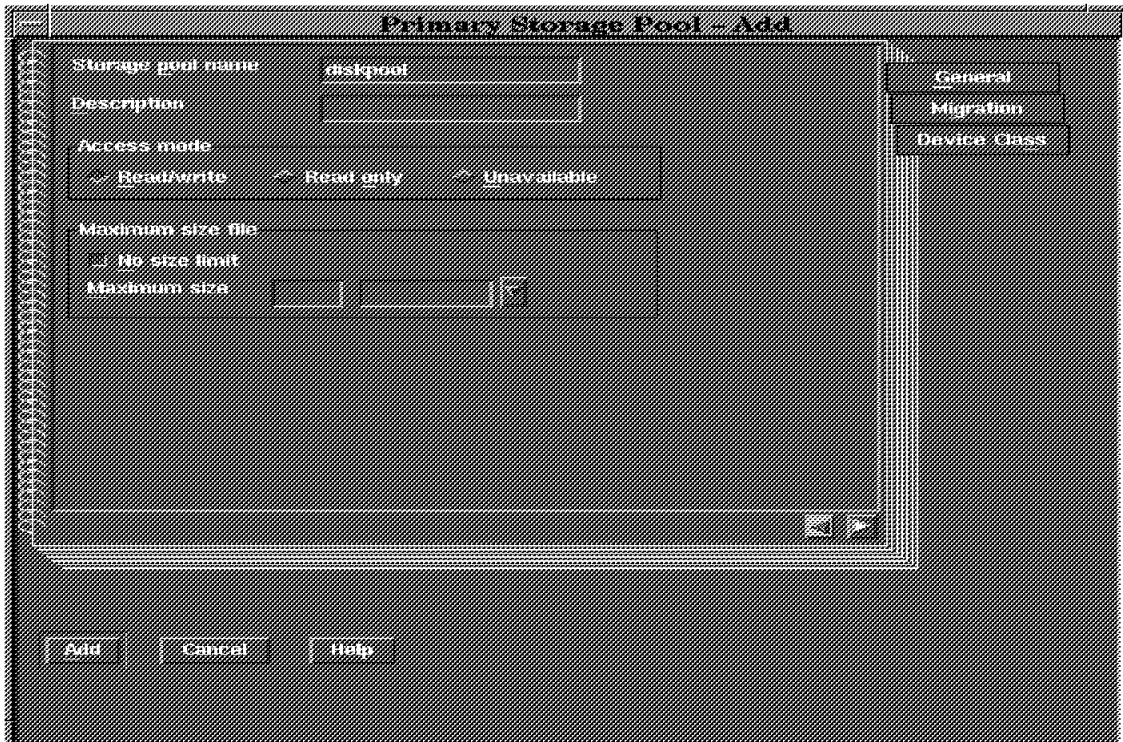


Figure 102. Adding a Storage Pool for Backup

Select the tab labeled **Device Class**.

The following screen appears.

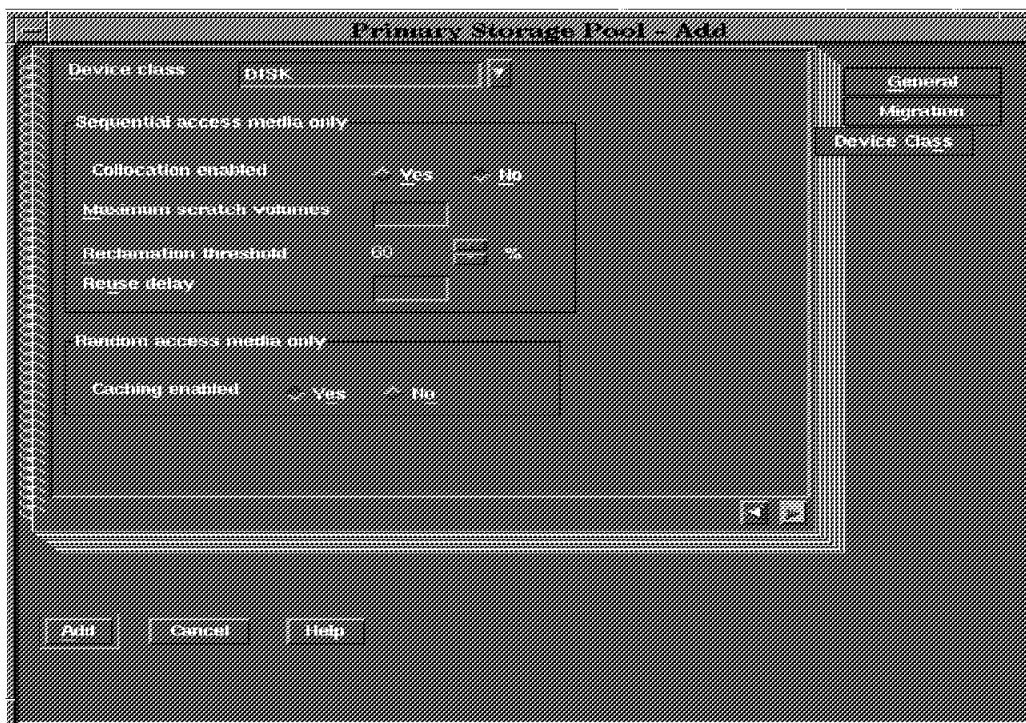


Figure 103. Adding a Storage Pool for Backup

Select the device class corresponding to the storage pool type you are creating. In this case, the storage pool type is DISK.

Press the **Add** button to add the storage pool.

5.2.5.3 Assigning a Logical Volume to a Storage Pool

You must allocate the logical volumes you have created to a storage pool. Logical volumes can be assigned to ADSM storage pools by using the command line interface or with the graphical administration interface.

- Using the command line interface. To assign logical volume backup1 to the storage pool diskpool, enter the following command:

```
adsm> define volume diskpool /dev/rbackup1
```
- Using the graphical administration interface, perform the following steps:
 - Expand the Storage Pools icon by clicking on the + sign in front of the icon.
 - Double-click on the **Storage Pool Volumes** icon.
 - Select **Edit** on the option bar.
 - Select **Add** in the pull-down menu.
 - Enter a storage pool name in the Storage Pool field.
 - Specify a previously defined AIX logical volume as shown in Figure 104, and press the **Add** button.

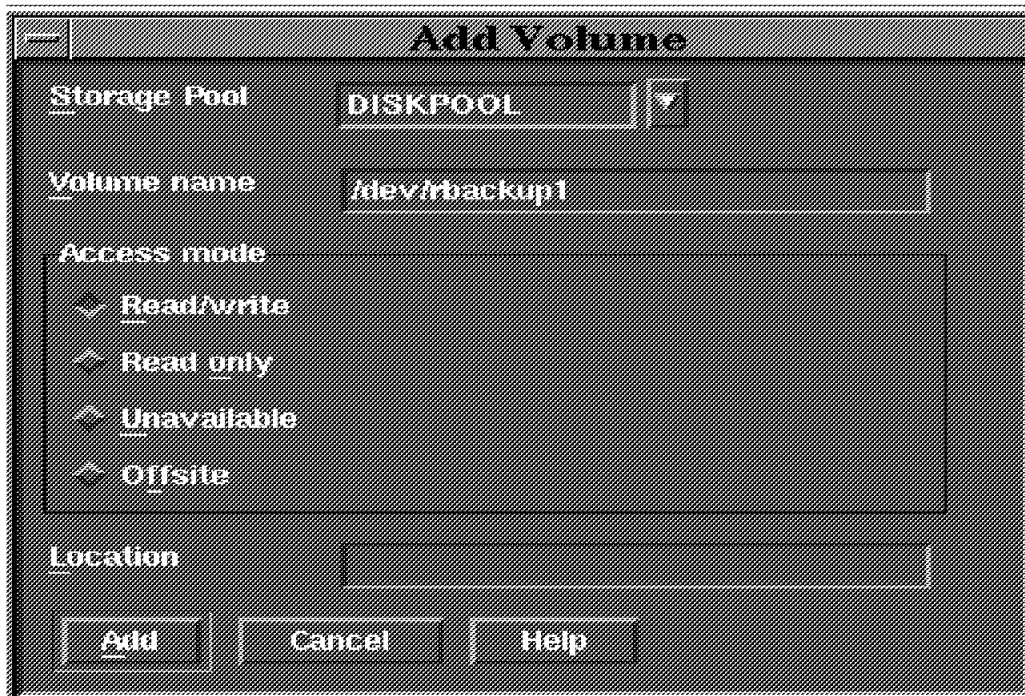


Figure 104. Adding a Volume to a Storage Pool

5.2.5.4 Defining Storage Pool Destination

If you want to test the disk storage pool, change the destination of your backup in the backup copy group.

- Using the command line interface, enter:
adsm> update copygroup standard standard standard type=backup destination=diskpool
Activate the policy with the following command:
adsm> activate policysset standard standard
- Using the graphical administration interface:
 1. Expand the Policy Domains icon (Figure 94 on page 169) by single-clicking on the plus sign (+) in front of the icon. The screen should be similar to the following:

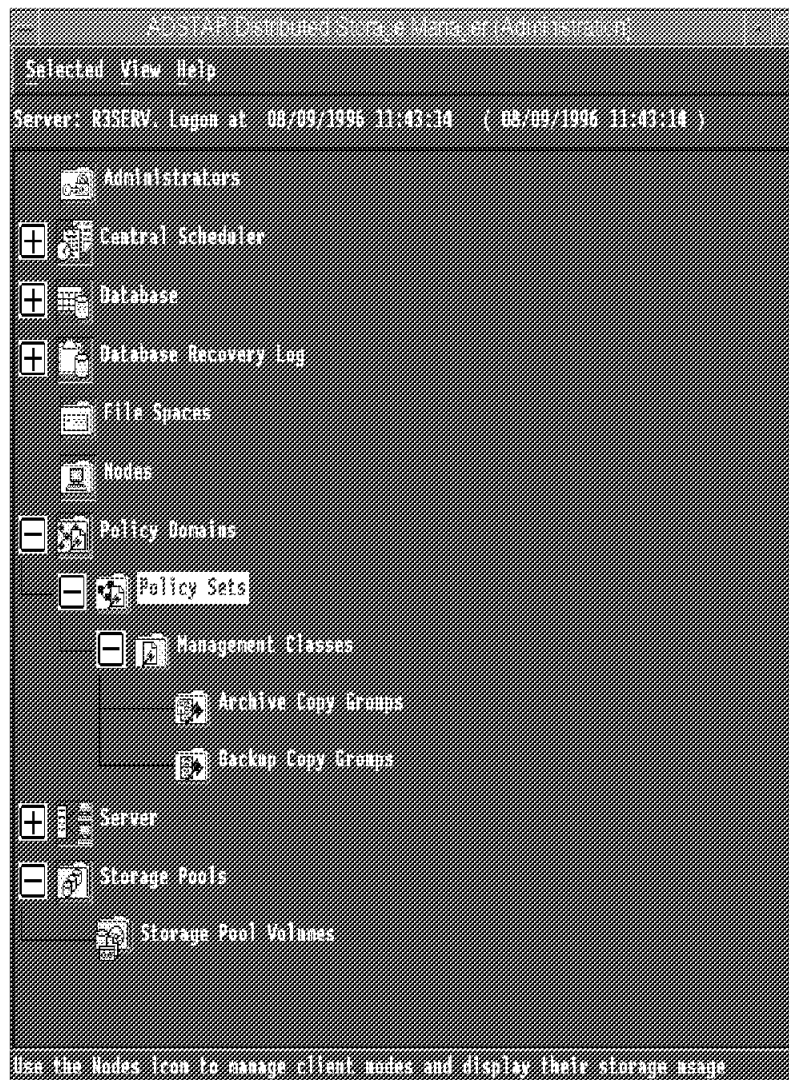


Figure 105. ADSM Administration Client GUI

2. Under Management Classes, double-click on the **Backup Copy Groups** icon. The following screen appears:

Backup Copy Groups - Details							
Copy groups Selected Edit View Help							
Server: R3SERV As of 09/06/1996 13:11:19 (09/06/1996 13:11:19)							
Policy Domain Name	Policy Set Name	Management Class Name	Copy Group Name	Versions Data Exists	Versions Data Deleted	Retain Extra Versions	Retain Only Version
STANDARD	ACTIVE	STANDARD	STANDARD	2	1	30	64
STANDARD	STANDARD	STANDARD	STANDARD	2	1	30	64

Each object in this view represents one backup copy group

Figure 106. Backup Copy Groups - Details Screen

3. Select the **STANDARD** policy domain name, policy set name, and management class name as shown in Figure 106.
4. From the menu, choose **Selected**.
5. Then from the pull-down menu, select **Open** as properties.
6. Select the **Copy** control tab.
7. Go to the second page of the Copy control section.
8. The screen should be similar to the following one.

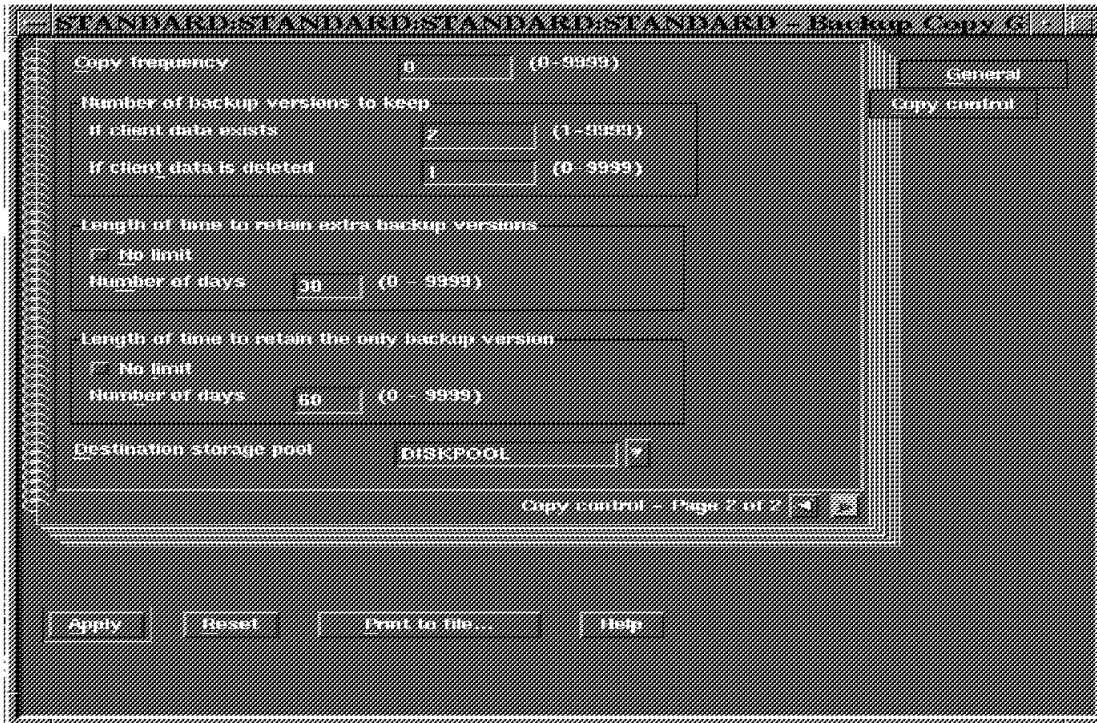


Figure 107. Defining Storage Pool Destination for Backup

9. Change the Destination storage pool to the disk storage pool. In our example, we used DISKPOOL.
10. Press the **Apply** button.
11. Return to the main menu (Figure 105 on page 180) and double-click on the **Policy Sets** icon. The following screen should appear.

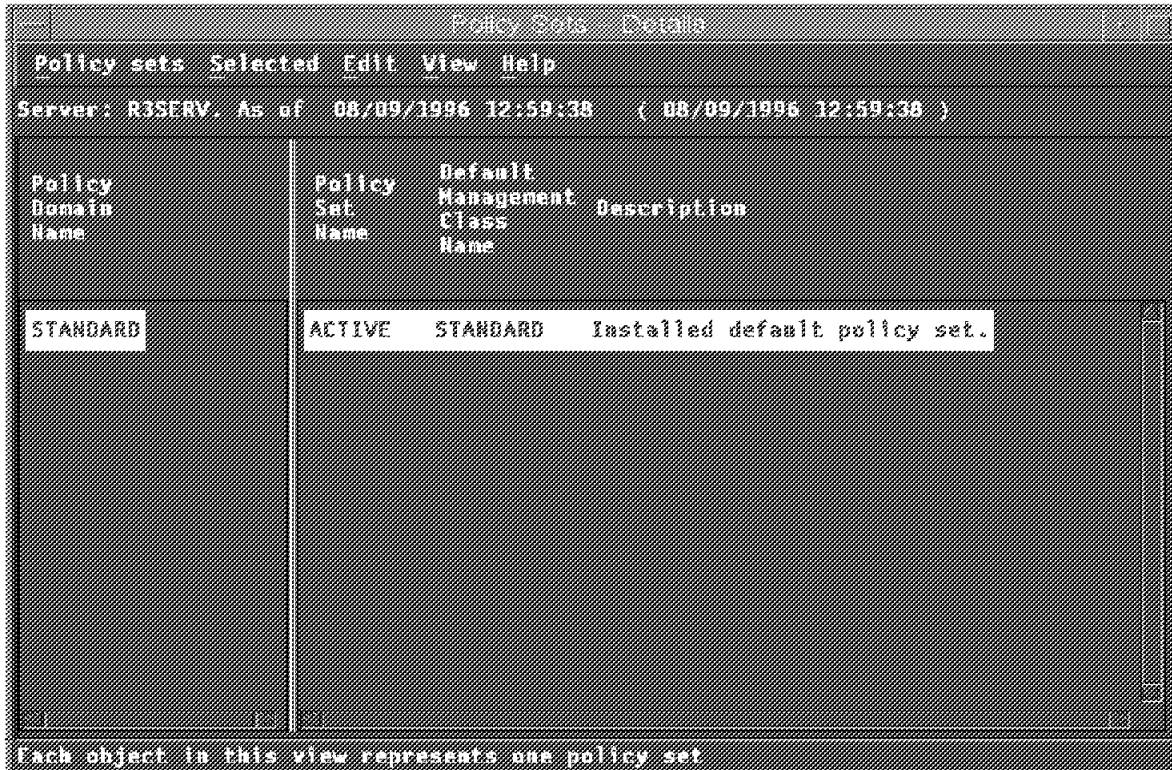


Figure 108. Policy Sets - Details Screen

12. Select the **STANDARD** policy domain name, policy set name, and management class name.
13. Chose **Selected** from the menu bar.
14. From the pull-down menu, select **Activate**.



Figure 109. Policy Sets - Activate Screen

15. Press the **Activate** button in the pop-up window.
16. Press the **OK** button in the message pop-up window. This updates the active policy set with your last modification. If you run a backup, it should go to this disk pool.

5.2.5.5 Executing a non-Database Backup from the Client

You can use ADSM to execute a backup to the disk area just defined using the graphical administration interface.

1. Start the graphical administration interface from the client by entering `dsm &` on the AIX command line of the client. The screen is similar to the following:

ADSTAR Distributed Storage Manager			
Backup	Restore	Archive	Retrieve
Utilities	View	Help	
File System Information			
File Systems for Backup/Archive	Type	Used	Capacity
mp	JFS	5 %	16,777,216
ome	JFS	18 %	8,388,608
b2/AUS	JFS	92 %	83,886,080
b2/AUS/sapdata4	JFS	37 %	2,097,152,000
b2/AUS/sapdata6	JFS	91 %	2,097,152,000
b2/AUS/sapdata1	JFS	65 %	1,048,576,000
b2/AUS/sapdata3	JFS	67 %	1,048,576,000
b2/AUS/log_archive/AUS	JFS	3 %	1,048,576,000
sr/sap/AUS	JFS	5 %	377,487,360
sr/sap/trans	JFS	9 %	109,051,904
b2/AUS/log_dir	JFS	73 %	234,881,024
apmnt/AUS	JFS	55 %	318,767,104
mp/install	JFS	97 %	25,165,824
b2/AUS/sandata2	JFS	71 %	1,048,576,000

File Systems for Restore/Retrieve	Type	Last Incremental Backup
US	DB2/6000	

Figure 110. ADSM Client - Main Screen

2. Select **Backup** in the action bar.
3. Select by **Directory Tree** in the pull-down menu. The next screen appears.

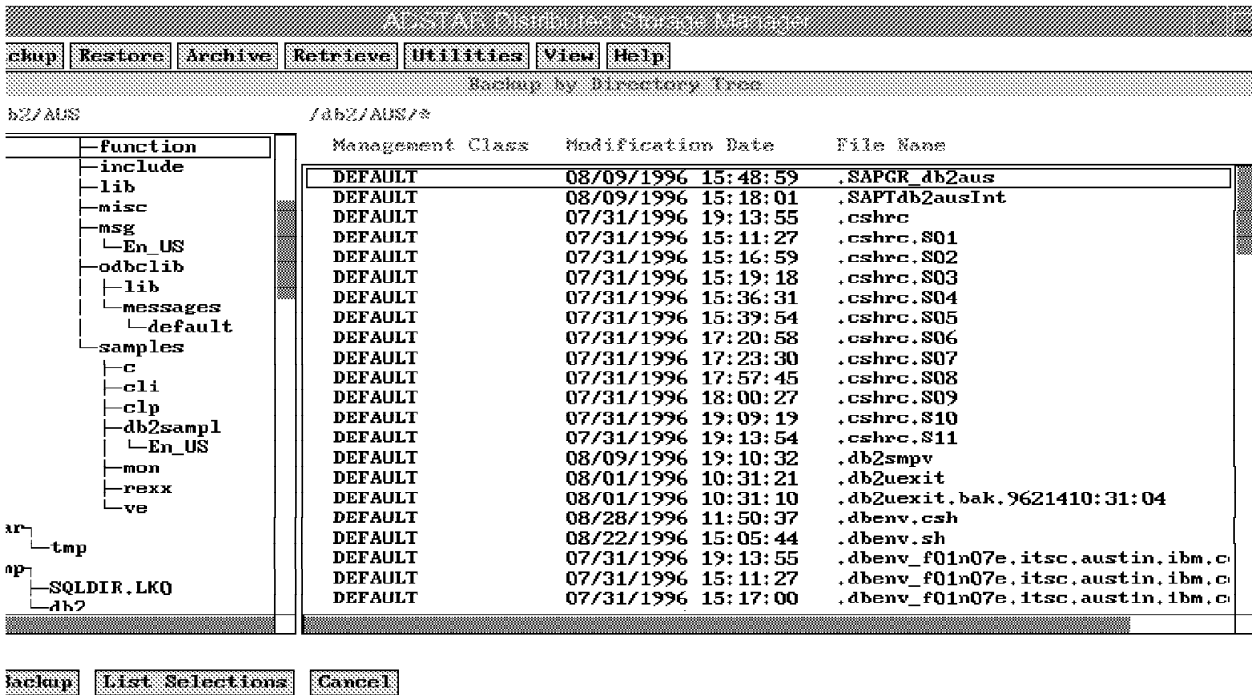


Figure 111. Using ADSM to Backup

4. Click on the directory or the file you want to backup.
5. Execute the backup by clicking on the **Backup** push-button.

5.2.5.6 Verifying the Backup

The backup can be verified from the graphical administration interface:

- On the administration interface (Figure 94 on page 169), click on the plus sign (+) in front of the Storage Pools icon.
- Double-click on the **Storage Pool Volumes** icon.
- On the next panel, select the volume you want to verify. In this example, select **/dev/rbackup1**.

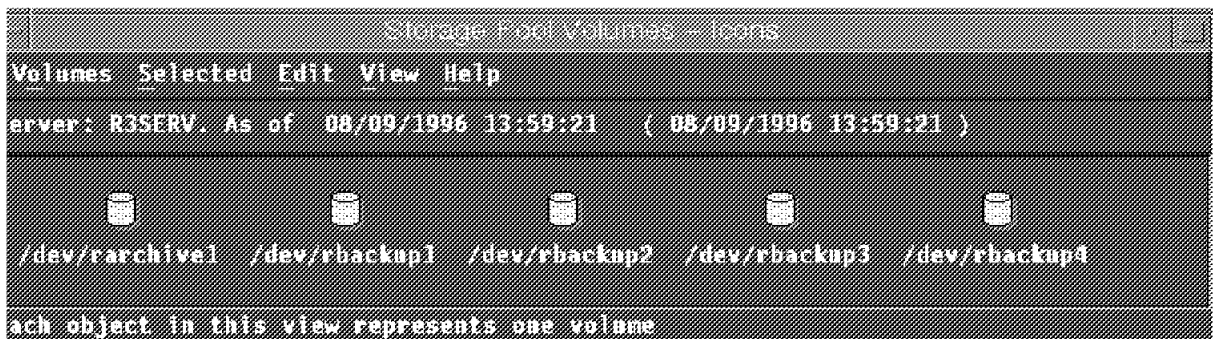


Figure 112. Storage Pool Volumes - Icons Panel

- Choose the **Selected** option in the action bar.
- Using the pull-down menu, select **List Volume Contents to File**.
- Fill in the file name you want to write the contents to.

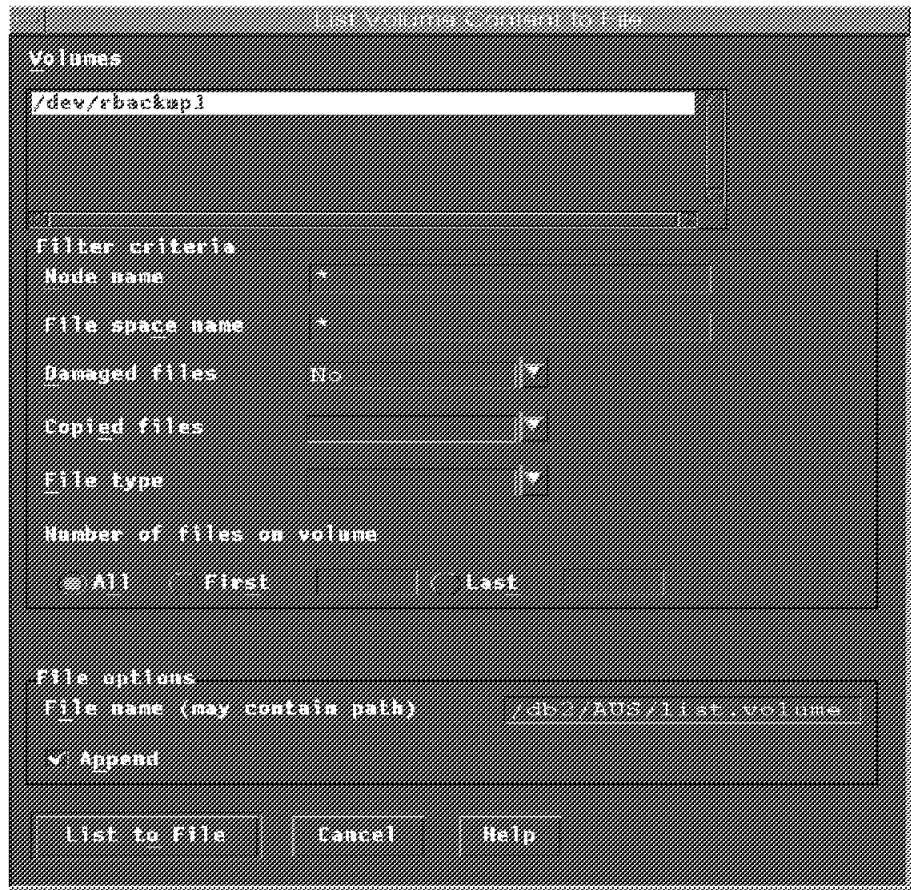


Figure 113. List Volume Content to File Panel

- Press the **List to File** button at the bottom of the screen.
- View the file generated on AIX.

5.2.6 Setting up ADSM to Save Archived Logs to Disk

Create AIX logical volumes on the ADSM Server and define them as ADSM volumes for use in the disk storage pools. You must be logged in as root to create AIX logical volumes. Use the SMIT command and select:

- **System Storage Management (Physical & Logical Storage)**
- **Logical Volume Manager**
- **Logical Volumes**
- **Add a Logical Volume**
 1. Fill in the volume group name field
 2. Specify a name in the logical volume name field. For example: archive1.
 3. Provide number of partitions.

There are two ways to define the volumes in ADSM/6000:

- Using the command line interface. To set up a storage pool called logarchive, enter the command:

```
adsm> define stgpool logarchive disk
```
- Using the graphical administration interface
 Double-click on the **Storage Pools** icon.

Select the **Edit** option.

Select the **Add Primary Storage Pool** from the pull-down menu.

Enter the name of your storage pool as shown in Figure 114.

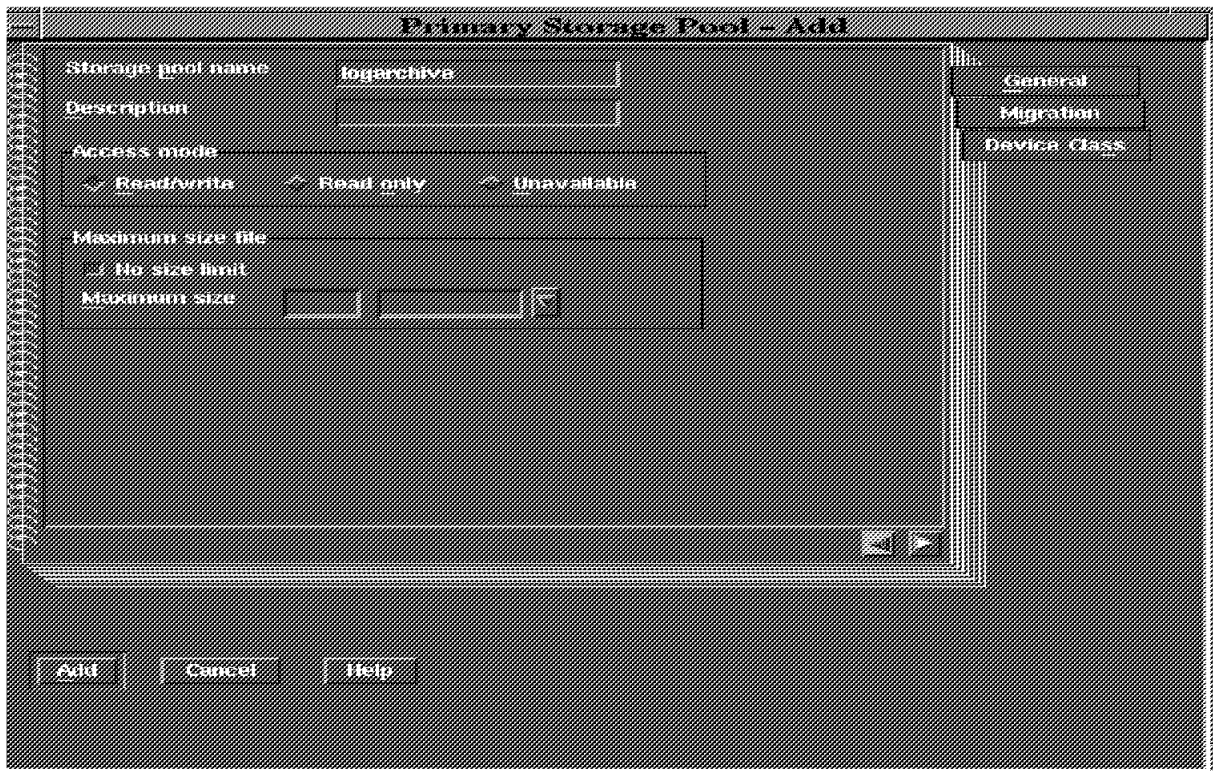


Figure 114. Defining Storage Pools for the Archived Log Files

- Press the tab labeled **Device Class**.
- Select the device class corresponding to the storage pool type you are creating. In this case, the storage pool type is DISK.
- Press the **Add** button to add the storage pool.

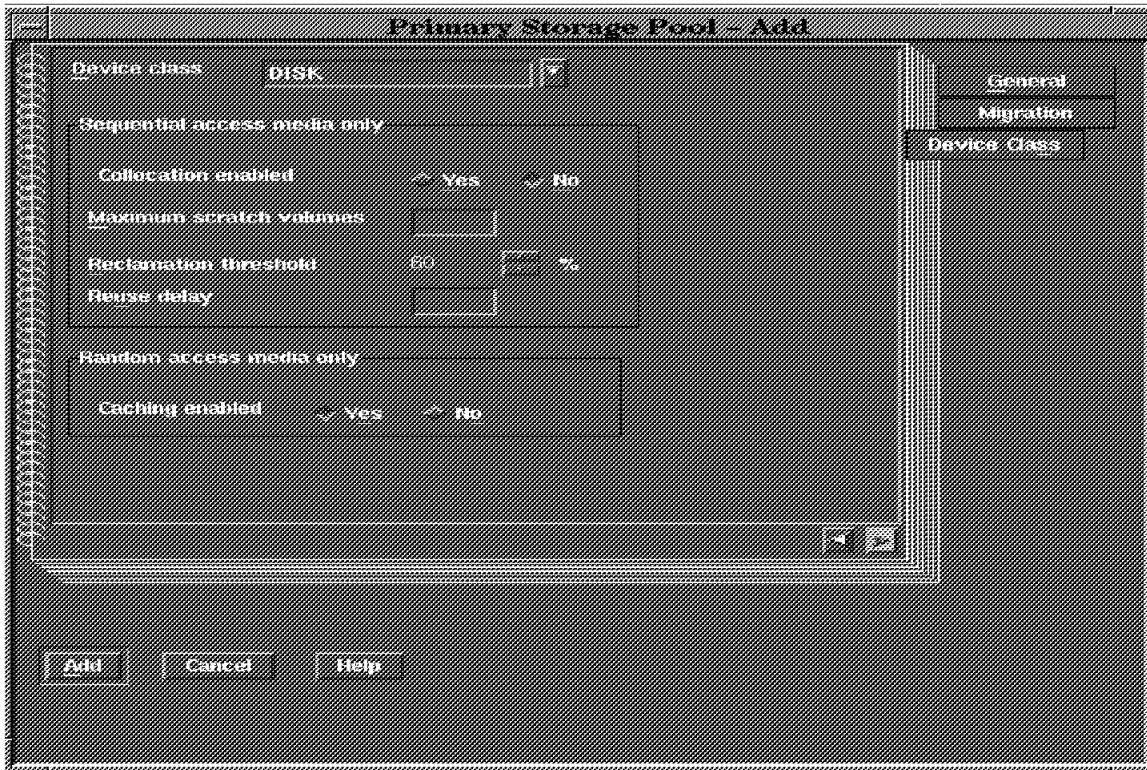


Figure 115. Adding a Storage Pool for the Archived Log Files

5.2.6.1 Assigning a Logical Volume to a Storage Pool

You must allocate the logical volumes you have created to a storage pool. Logical volumes can be assigned to ADSM storage pools using the command interface or with the graphical interface.

- Using the command interface. To assign logical volume archive1 to the storage pool logarchive, enter the following command:


```
adsm> define volume logarchive /dev/rarchive1
```
- Using the graphical administrator interface, perform the following steps:
 - Expand the Storage Pools icon by clicking on the + sign in front of the icon. The screen displayed in Figure 116 on page 189 appears.

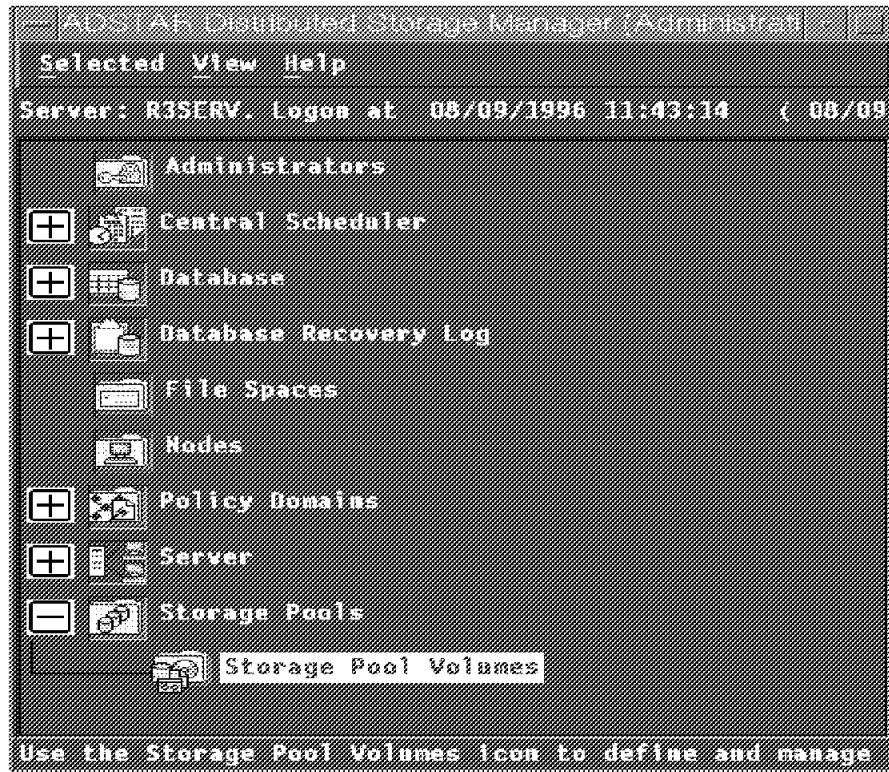


Figure 116. Using Storage Pool Volumes

- Double-click on the **Storage Pool Volumes** icon.
- Select **Edit** on the option bar.
- Select **Add** in the pull-down menu. The following screen appears:

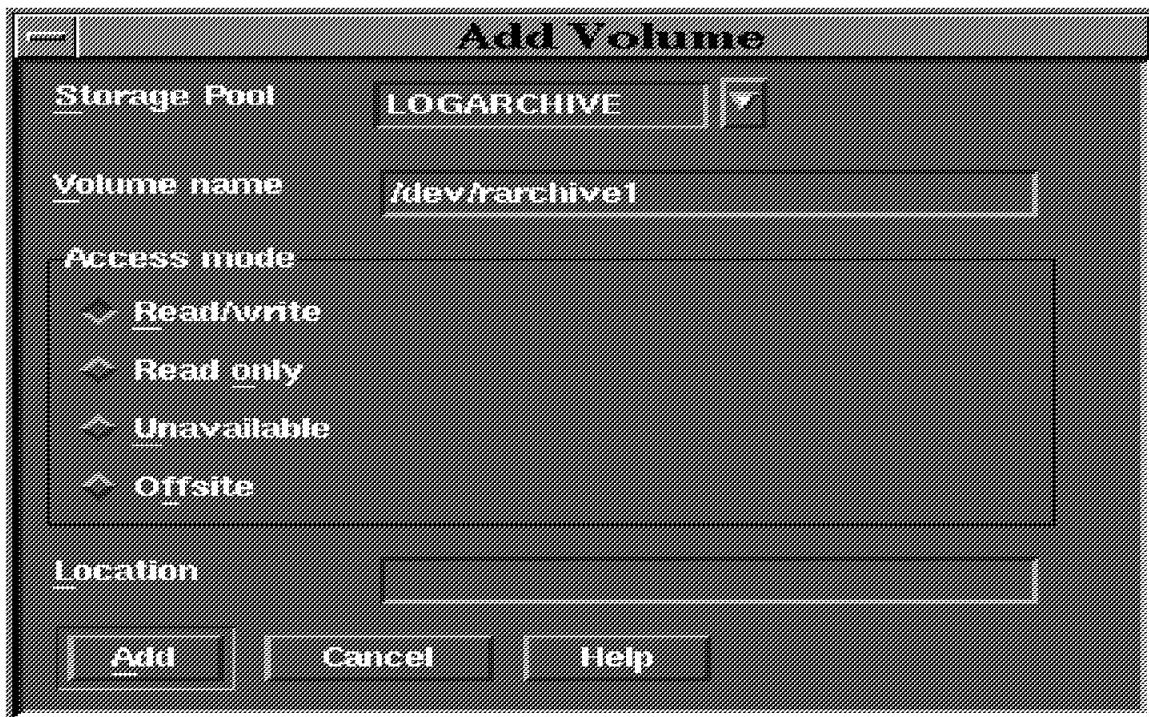


Figure 117. Assigning a Logical Volume to a Storage Pool

- Specify the Storage pool and a previously defined AIX logical volume as shown in Figure 117.
- Press the **Add** button.

5.2.6.2 Defining Storage Pool Destination

Change the destination of your archive to the archive copy group.

- Using the command line interface, enter:


```
adsm> update copygroup standard standard standard
type=archive destination=saparchivepool
```

Activate the policy with the following command:

```
adsm> activate policysset standard standard
```
- Using the graphical administration interface:
 1. Expand the Policy Domains icon from the main administration screen (by single-clicking on the plus sign (+) in front of the icon).
 2. Double-click on the **Archive Copy Groups** icon.
 3. Select the **STANDARD** policy domain name, policy set name, and management class name.

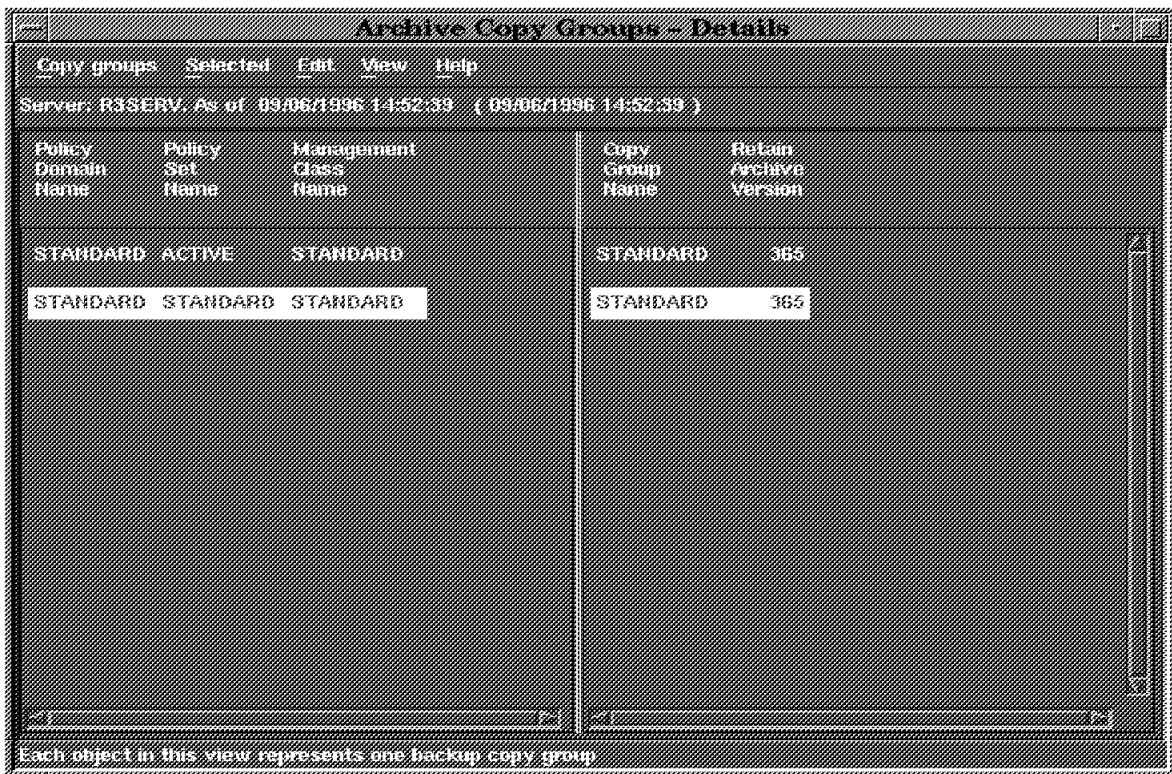


Figure 118. Setting the Storage Pool Destination

4. Choose the **Selected** option.
5. In the pull-down menu, select **Open** as properties.
6. Select the **Copy** control tab.
7. Go to the second page of the Copy control section.
8. Change the **Destination storage pool** to the **archive storage pool**.

9. Press the **Apply** button.

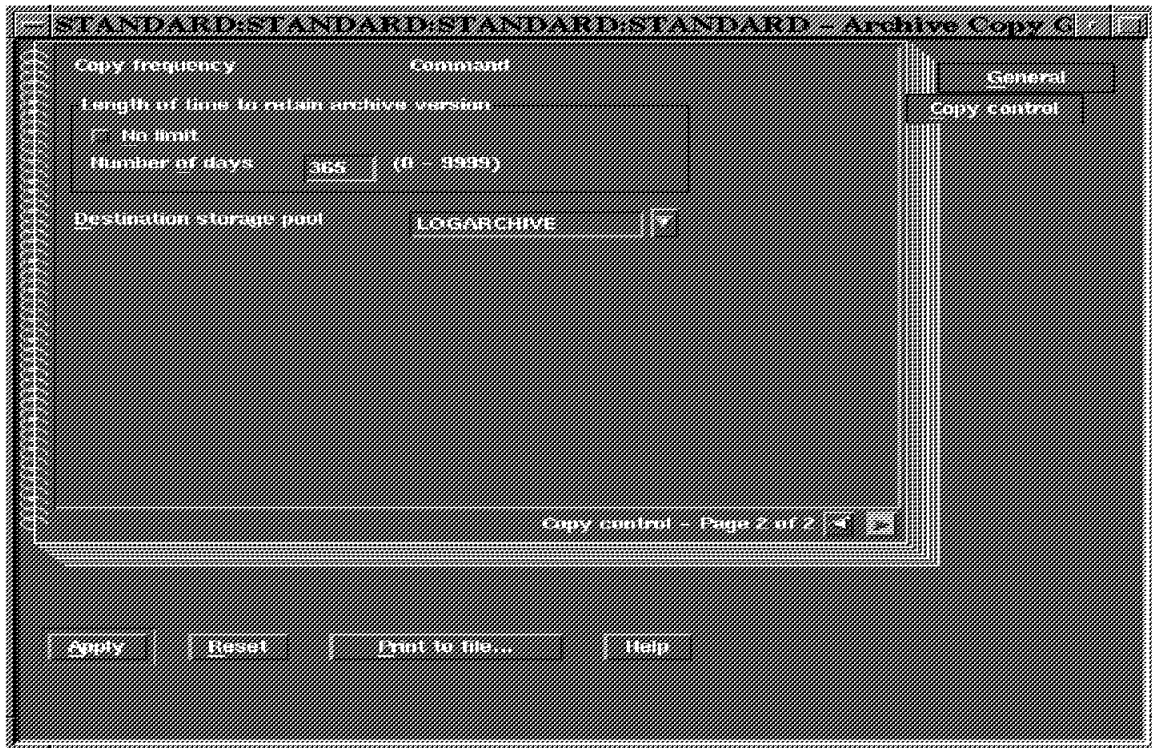


Figure 119. Setting the Storage Pool Destination

10. Return to the main menu and double-click on the **Policy Sets** icon.
11. Select the **STANDARD** policy domain name, policy set name, and management class name.
12. Chose the **Selected** option.
13. In the pull-down menu, select **Activate**.
14. Press the **Activate** button in the pop-up window.
15. Press the **OK** button in the message pop-up window. This will update the active policy set with your last modification. If you archive the log files, they should go to this storage pool.

5.3 R/3 DB2 Backup and Restore Utilities

This section talks about backup for a DB2 database in a R/3 system. For more information about backup in general, see the *DB2 Administration Guide*. We discuss what you need to do in an R/3 to perform a backup/restore. There are also different levels of backup and restore. This section is outlined as follows:

- Backup and restore considerations
- Different ways to perform a DB2 R/3 backup
- Performing and verifying an offline backup to both disk and ADSM
- Performing an online backup
- Performing various types of recovery, including offline and online
- Managing your backups and restores

- Backup and restore considerations and recommendations

5.3.1 Backup and Restore Considerations

There are two basic types of backups:

- OFFLINE - The database is not in use. The only connection to the database is the backup operation.
- ONLINE - Access to the database/tablespace is permitted. Users can continue to work normally during the backup.

A full, offline database backup provides you with a complete snapshot of the data at a fixed point in time. This level of backup is a requirement for disaster recovery and should be an essential part of any backup/restore strategy.

There are different levels of backup/restore. You can backup (and restore) at either the tablespace or database level.

Using ADSM for Backup and Restore

If you are using ADSM for backup and restore, you must increase the COMMITimeout value in /usr/lpp/admserv/bin/dsmserv.opt and restart ADSM. If you do not change this value, the Redirected Restore will not complete successfully and will leave the database in an inconsistent state. The COMMITimeout value in dsmserv.opt specifies the communication time-out value in seconds. The default value for this parameter is 60 seconds. We increased it to 6000 seconds.

5.3.2 How to Perform a DB2 Backup in an R/3 Environment

There are three basic ways to perform a backup (restore) in a DB2 R/3 environment. You can:

1. Use the R/3 DB2admin tool via the SMIT interface
2. Use DB2 Database Director
3. Issue the DB2 BACKUP DATABASE command from the AIX command line

There are certain considerations with each. The preferred method is the R/3 DB2admin tool. We show several examples using the DB2admin tool. The only person who can perform a backup (restore) using the DB2admin tool is the R/3 database administrator (db2<SAPSID>). The exception to this is if you are executing the backup as a batch job. In this case, you must be the AIX root user. The reason for this is that the batch backup job is executed as a cron job. Certain values, such as time and frequency, must be registered in the crontab operating system table. Only the AIX root user can update this crontab table or execute a cron job.

The <SAPSID>adm user can issue the DB2 BACKUP DATABASE command or issue a backup through the Database Director. The DB2admin tool actually calls an R/3 program that checks that the caller of the program is the R/3 database administrator. So, in the DB2admin tool, any user other than db2<SAPSID> will not be able to perform a backup or restore operation. The reason that the <SAPSID>adm user can initiate a backup/restore from the command line or the Database Director has to do with the group authorities assigned by DB2. Recall that <SAPSID>adm is placed in the SYSCTRL group of DB2. For more

information about users and group authorities within DB2, please see 3.3.6, “DB2 Users” on page 69.

5.3.3 Performing an Offline Backup

5.3.3.1 Doing an Offline Backup to Disk

Create the directories to be used for the backup. Make sure the directory owner is the database administration user (db2<SAPSID>) and, the group is sysadm. The AIX operating system has a two-gigabyte limitation on file size; so you must create several directories for a full database offline backup. The backup utility creates one file in each directory when the backup is run.

R/3 must be shut down when an offline backup is run.

Log in to AIX as db2<SAPSID>. Enter smit on the command line. Select the following menu path:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Back up Database**
- **Execute Back Up**
- Select the **database ID** for the back up you wish to run.
- Select **Device**.
- For a full, offline backup, leave **Tablespaces to be backed up** blank. For a partial database backup, specify the tablespace(s) you want to back up separated by blanks. If you specify more than one tablespace, you must separate the tablespace names with blanks.
- Specify the full path name of the files for the backup in **Target area**. If multiple files are used, separate by commas.
- Press **OK** to begin the backup.

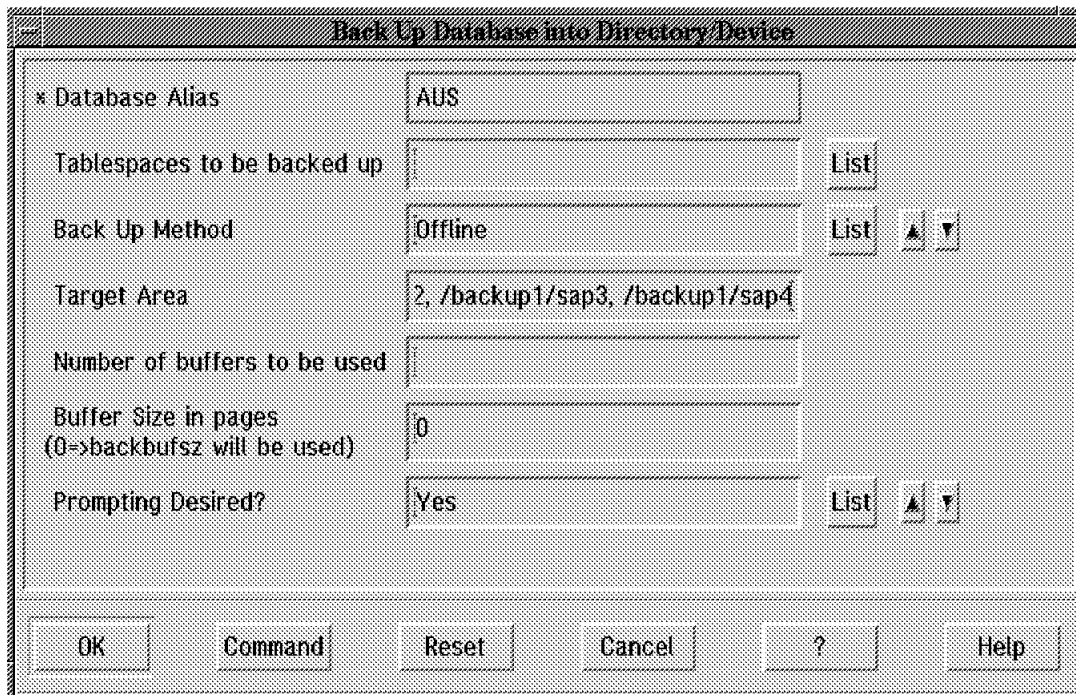


Figure 120. Offline Database Backup to Disk

5.3.3.2 Doing an Offline Backup with ADSM

The offline backup can be done using SMIT. Log in to AIX as the database administrator (db2<SAPSID>) and enter smit. Select the following options:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Back Up Database**
- **Execute Back Up**
- Select the database by clicking on the correct line in the pop-up window.
- Select **ADSM** for the Back Up Target.
- Enter the tablespaces you wish to back up in the field Tablespaces to be backed up. If you want to back up all tablespaces, leave this field blank. If you want to back up multiple tablespaces, separate with blanks.
- Press the **OK** button.

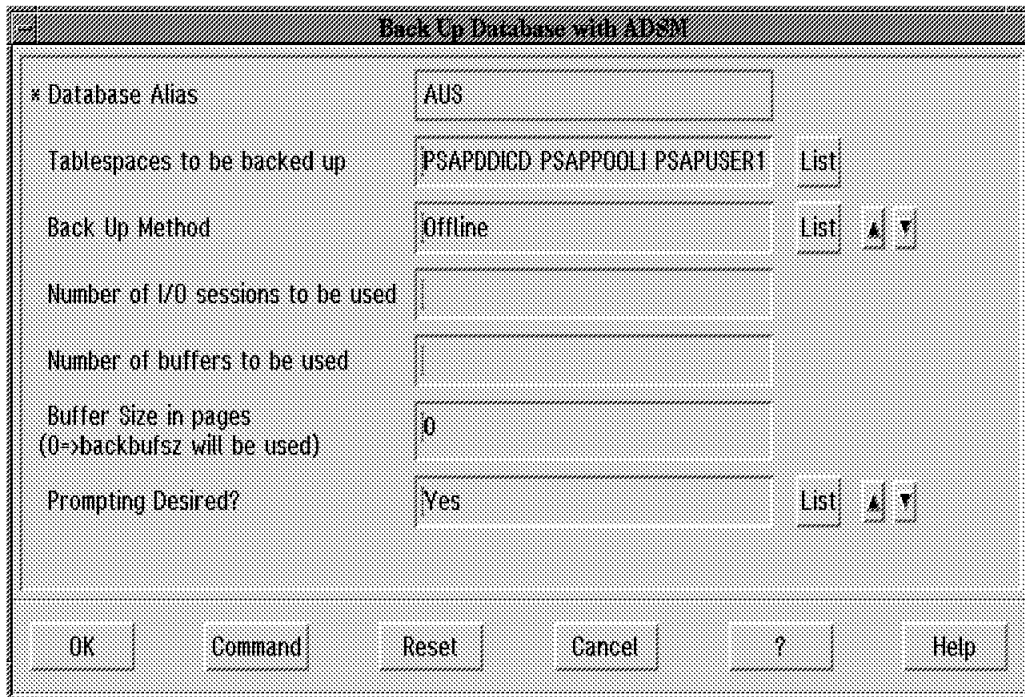


Figure 121. Offline Backup to ADSM

5.3.3.3 Verifying the ADSM Backup

From the ADSM command line interface, you can verify the backup with the following command:

```
adsm> query filespace
```

From the graphical administration interface:

- Expand the Storage Pools icon by clicking on the plus sign (+).
- Double-click on the **Storage Pool Volumes** icon.
- Select the storage pool you want to view, and select **Selected** from the option.
- Select **List volume content to file** from the pull-down menu.
- Enter the file name in the file name field.
- Press the **List to file** button.
- The file may be viewed on AIX.

5.3.4 Performing an Online Backup

In this section, we show how to do an online backup to disk and an online backup using ADSM.

5.3.4.1 Online Backup to Disk

Create the directories to be used for the online back up. These directories can be the same directories used for an offline back up, but you may want to keep the online and offline backups in separate directories to make management of these files easier.

R/3 can be running with active users during an online backup. The backup will impact system performance; so it should not be scheduled during peak hours.

To start an online backup to disk, log in to the database administration user (db2<SAPSID>) and enter smit. Select:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Back Up Database**
- **Execute Back Up.**
- Select the database to be backed up from the pop-up screen.
- Select **Device** for the back up target.
- If you want to do an online backup of the complete system, leave **Tablespaces to be backed up** blank. If you want to back up only selected tablespaces, enter the tablespaces, separated by blanks, in **Tablespaces to be backed up**.
- Change **Back up Method** to online.
- Change the **Target Area** to the directories you created for online backups, separated by commas.

Back Up Database into Directory/Device	
* Database Alias	AUS
Tablespaces to be backed up	PSAPBTABI PSAPUSER1D PSAPDDIC List
Back Up Method	Online List ▲ ▼
Target Area	/backup1/online1, /backup1/onlin
Number of buffers to be used	
Buffer Size in pages (0=>backbufsz will be used)	0
Prompting Desired?	Yes List ▲ ▼
OK Command Reset Cancel ? Help	

Figure 122. Online Backup to Disk

5.3.4.2 Performing an Online Backup with ADSM

You must be logged on as the db2<SAPSID> user ID to start a backup. Log in as db2<SAPSID>, set the display variable if necessary, and enter smit at the command line. Select the following options from the SMIT menu:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Back up Database**
- **Execute Back Up**
- Select the database to be backed up by clicking on it in the pop-up window.
- Select **ADSM** for the Back Up Target.
- Enter the tablespace(s) you want to back up or leave blank to back up all tablespaces. If you enter more than one tablespace, separate by blanks.
- Change **Back Up Method** to online.
- Press the **OK** button.

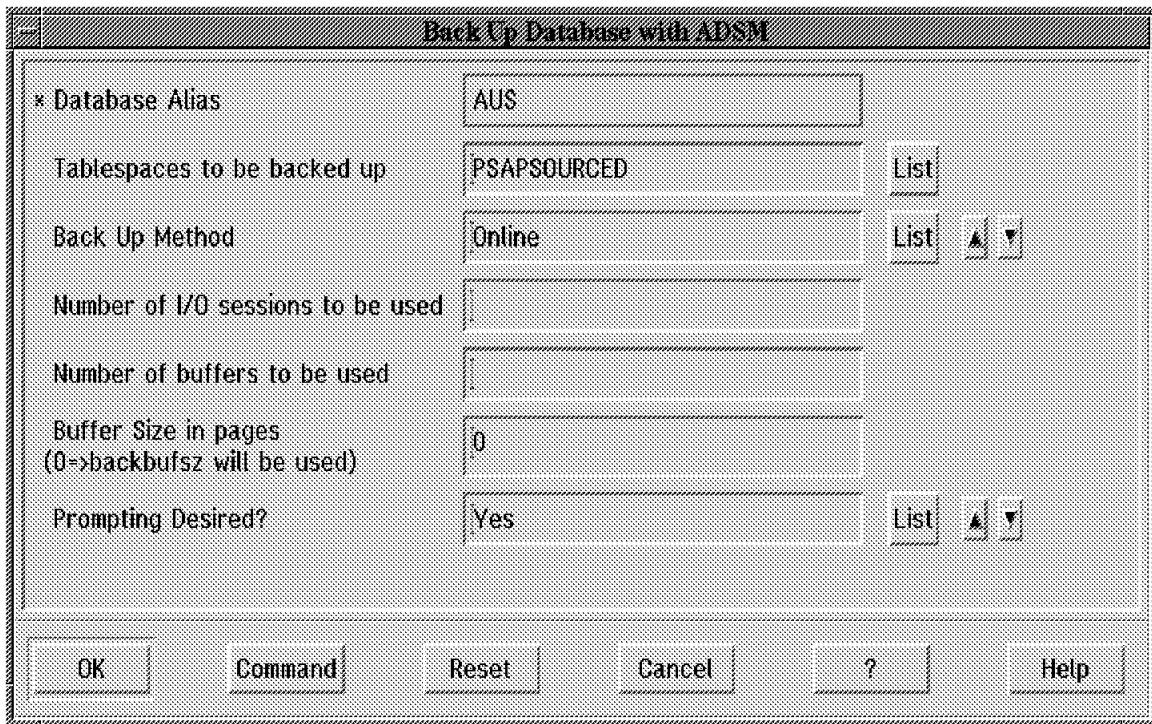


Figure 123. Online Backup to ADSM

5.3.5 Methods of Recovery

You need to know the strategies available to you to help when there are problems with the database. Typically, you will deal with media and storage problems and application failures.

You need to know that you can back up your database, or individual tablespaces, and then rebuild them should they be damaged or corrupted. The rebuilding of these objects is called recovery. DB2 provides the capability to recover from a variety of data-processing problems.

The degree of recovery is dependent on the type of recovery required. For example:

- Recovery from a program logic flaw can only be made to a point in time prior to when the erroneous program began executing.
- Recovery from a power failure can restore the database to a consistent state up to the last committed unit or work.
- In order to support recovery from all the potential problems, the R/3 database administrator needs to take full advantage of all the recovery features in DB2.

There are basically three types or methods of recovery (Figure 124):

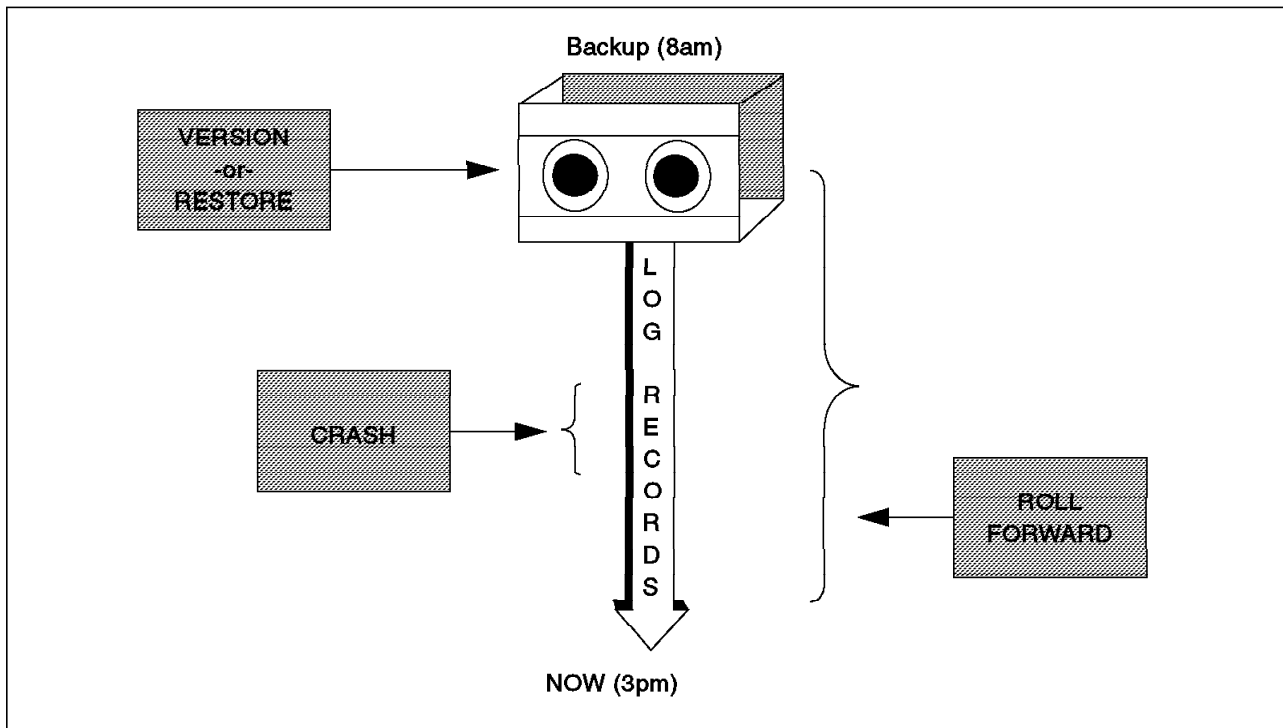


Figure 124. Three Methods of Recovery

1. Crash Recovery

This method entails using the log files to recovery from power interrupts or application abends. Crash recovery can be initiated by entering a `RESTART DATABASE` command. However, it is more common to use the automatic restart enable (`AUTORESTART`) database configuration parameter. This is set to `ON` in the R/3 installation.

`AUTORESTART` enabled will cause crash recovery to occur automatically at the first connect after a power failure. The default for this configuration parameter is that the `DB2 RESTART DATABASE` command will be started every time it is needed.

2. Restore or Version (No Roll Forward)

This method entails using a backup copy of the database to replace the current data in the database. This type provides recovery from media, hardware, operational, and software problems, but only to the point that the backup copy was made. Therefore, the more frequent the backups, the more current the recovered database will be.

You must schedule and perform a full backup of the database on a regular basis. The longer the time between backups, the greater the number of units or work that may be lost. The loss of units of work must be acceptable within your business operation, or you must have a way to reapply the missing units of work against the restored database.

Tablespace recovery is not supported by the restore or version strategy.

3. Roll Forward Recovery

Entails using a backup copy of the database or tablespace to replace the current data and then applying log records to recovery changes made after the backup image was created. This type provides recovery from media, hardware, operational, and software problems. The recovery can be to a point in time or the last committed unit of work. Note, that if tablespace-level recovery is being used, the only option is to apply all of the log records. Point-in-time recovery is not supported for tablespaces in this version of DB2. Roll forward recovery using a recent backup will require less log record application than a roll forward using an older backup; so the frequency of backups will still be a consideration for the R/3 database administrator.

To use the database roll forward recovery method, you must ensure that log archiving is enabled and that the log files created since the backup are available for the roll forward process.

5.3.6 Recovery Using an Offline Backup from Disk without Roll Forward

1. Log in to the database administrator user ID on AIX, db2<SAPSID>.
2. Enter smit on the AIX command line.
3. From the SMIT screen, select
 - **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Recover Database**
 - **Restore Database**
 - Select the **database name** by clicking on the appropriate line in the pop-up window.
 - Select the method used for the back up (for example, **Device** or **ADSM**).
 - Fill in the **Source Directory/Device** names. If you want to restore a backup done to disk device, enter the filesystem names as shown in Figure 125 on page 200.
 - If there is more than one backup on the device or in the filesystem, specify the time of the backup you want to restore in the format `yyyymmddhhmmss`.
 - If you are restoring from a full database backup image that was created using the offline backup option and you do not wish to roll-forward the database after the restore is complete, then change **Place Database in Roll Forward Pending State** to **No**.

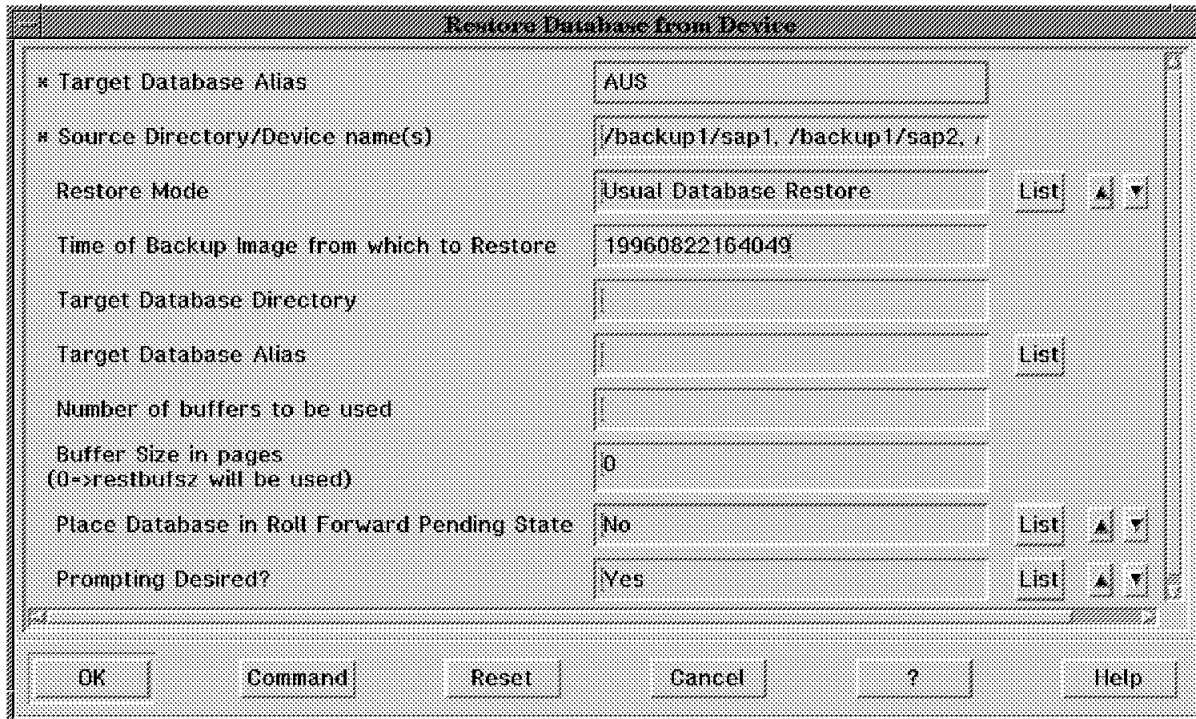


Figure 125. Recovery from Offline Backup without Roll Forward

- Press the **OK** button.
- Press the **OK** button in the confirmation pop-up window.
- You will get the following warning on the SMIT screen, enter **Y** if you want to continue:

```

DE2admin: Start Database Manager
DE2admin: db2 restore database AUS FROM /backup1/sap1, /backup1/sap2, /backup1/sap3
DE2admin: start of restore at Fri Aug 23 11:45:08 CDT 1996
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)

```

Figure 126. SMIT Warning Message

5.3.7 Recovery Using an Offline Backup from ADSM (No Roll Forward)

1. Log in to the database administrator user ID on AIX, db2<SAPSID>.
2. Enter smit on the AIX command line.
3. From the SMIT screen, select:
 - **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Recover Database**

- **Restore Database**
- Select the **database name** by clicking on the appropriate line in the pop-up window.
- Select **ADSM** for the Backup Image Location.
- If there is more than one backup on the device or in the filesystem, specify the time of the backup you wish to restore in the format `yyyymmddhhmmss`.
- Change **Place Database in Roll Forward Pending State** to **No**.

Restore Database with ADSM	
* Database Alias	AJS
Restore Mode	Usual Database Restore
Time of Backup Image from which to Restore	19960811192502
Target Database Directory	
Target Database Alias	
Number of I/O sessions to be used	
Number of buffers to be used	
Buffer Size in pages (0->restbufsz will be used)	0
Place Database in Roll Forward Pending State	No
Prompting Desired?	Yes

Buttons: OK, Command, Reset, Cancel, ?, Help

Figure 127. Recovery from Offline Backup to ADSM without Roll Forward

- Press the **OK** button.
- Press the **OK** button in the confirmation pop-up window.
- Respond **Y** to the SMIT warning message if you want to continue.

5.3.8 Recovery Using an Offline Backup from Device with Roll Forward

1. Log in to the database administrator user ID on AIX, `db2<SAPSID>`.
 2. Enter `smit` on the AIX command line.
 3. From the SMIT screen, select
 - **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Recover Database**
 - **Restore Database**
- Select the **database name** by clicking on the appropriate line in the pop-up window.

- Select **Device**.
- Fill in the **Source Directory/Device** names. If you want to restore a backup done to disk device, enter the filesystem names as shown in Figure 128.
- If there is more than one backup on the device or in the filesystem, specify the time of the backup you wish to restore in the format `yyyymmddhhmmss`.
- Verify that "Place Database in Roll Forward Pending State" is set to **YES**.

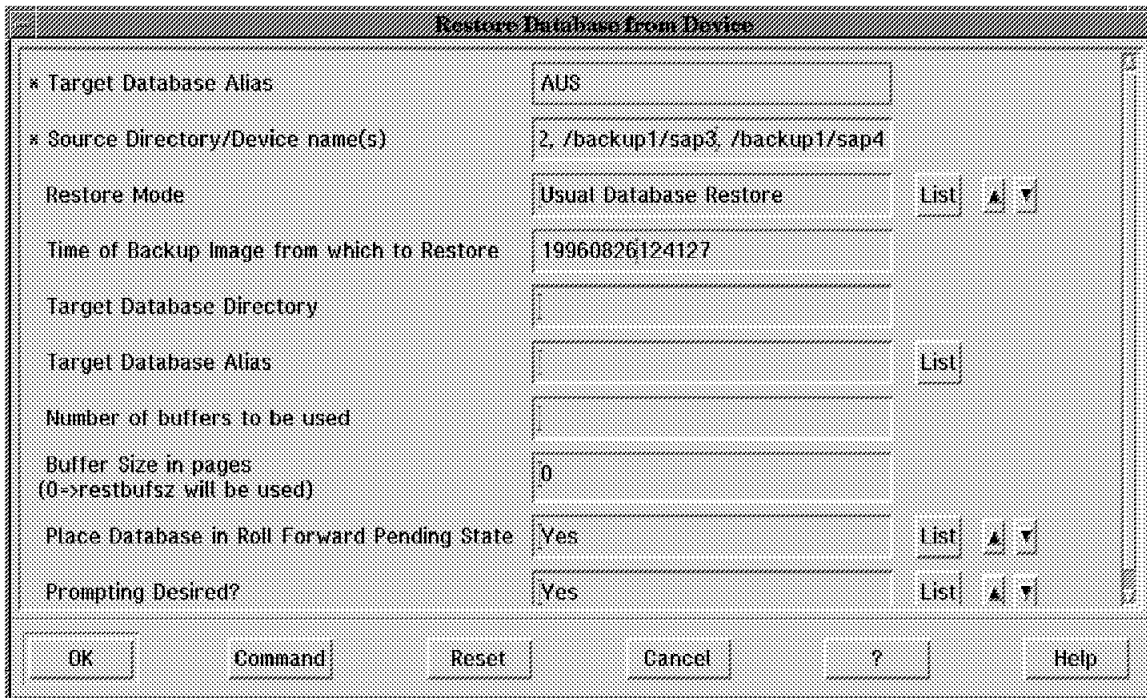


Figure 128. Recovery from Offline Backup to Disk with Roll Forward

- Press the **OK** button.
- Press the **OK** button in the confirmation pop-up window.
- You will get a warning message on the SMIT screen, enter **Y** if you want to continue.

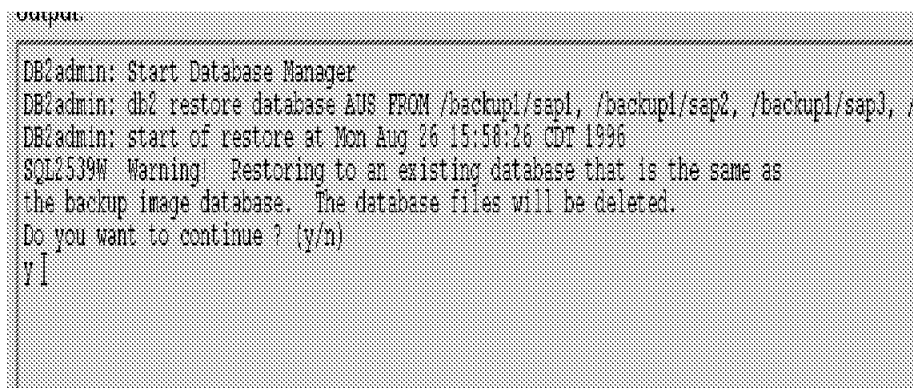


Figure 129. SMIT Warning Message

When the restore is complete, the database will be placed in Roll Forward Pending state. You will not be able to start R/3 until you have completed the roll-forward. After the restore is complete, back up to the Recover Database SMIT screen and select:

- **Roll Forward Database**
- Select the Roll Forward Function you want. Press the **List** button for a list of options. The options are detailed in Chapter 10, "Recovering the Database", in *BC SAP Database Administration Guide: DB2 for AIX*. For this example, we reapply all committed transactions from all the online archived log files residing in the current database log directory and rollback any incomplete transactions. The Roll Forward Pending state of the database will be turned off allowing access to the database again. Note that timestamps for committed transactions are given in Coordinated Universal Time.

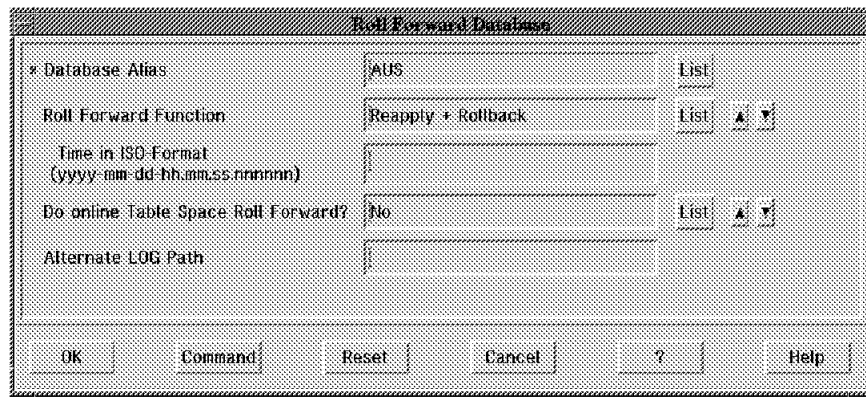


Figure 130. Roll Forward Database after Recovery from Disk

5.3.9 Recovery Using an Offline Backup from ADSM with Roll Forward

1. Log in to the database administrator user ID on AIX, db2<SAPSID>.
2. Enter smit on the AIX command line.
3. From the SMIT screen, select:
 - **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Recover Database**
 - **Restore Database**
 - Select the **database name** by clicking on the appropriate line in the pop-up window.
 - Select **ADSM**.
 - Fill in the time of the back up that you wish to restore from in the format `yyyymmddhhmmss`.
 - Make sure that Place Database in Roll Forward Pending State is set to **YES**.
 - Press the **OK** button.
 - If you wish to continue with the restore, press **OK** in the confirmation pop-up window

- If you wish to continue, answer **Y** to the SMIT warning message.
4. When the restore completes, you will need to roll forward the database before R/3 or the database can be started. To roll forward the database, select the following options from the SMIT menu:
- **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Recover Database**
 - **Roll Forward Database**
 - Select the option you wish for Roll Forward Function. Press the **List** button to see all options available. We chose Reapply & Rollback as shown in Figure 130 on page 203.

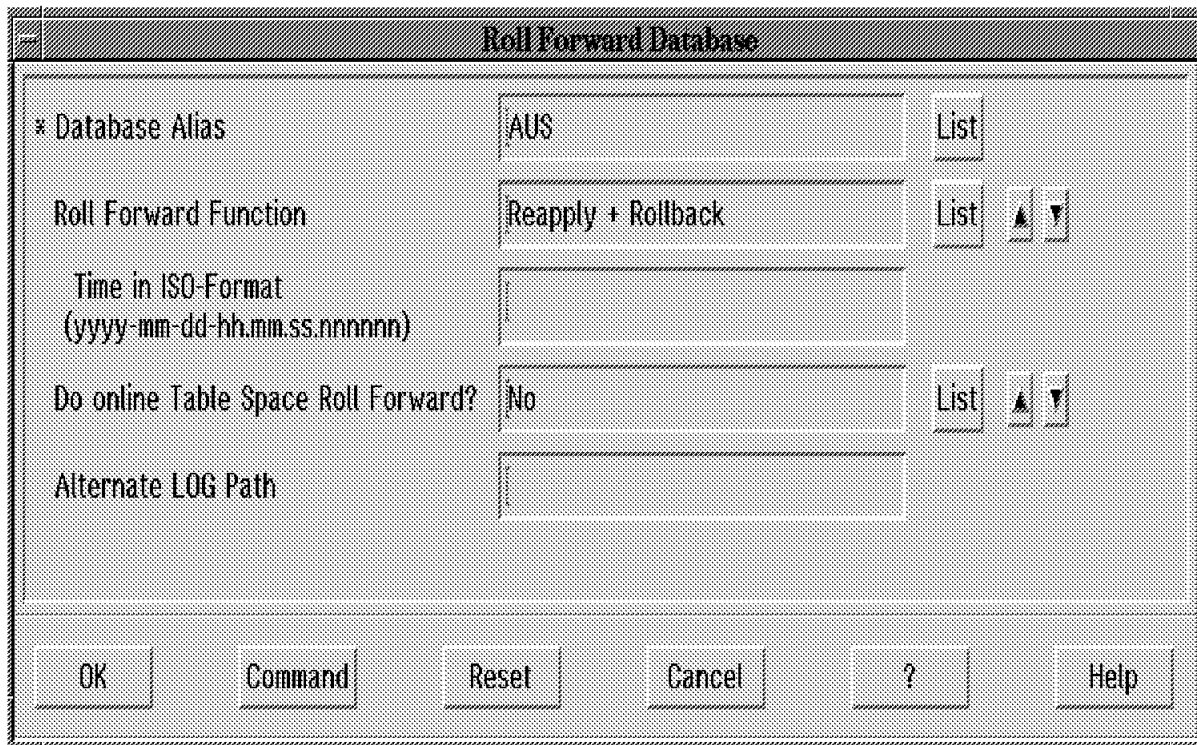


Figure 131. Roll Forward Database after Recovery from ADSM

- Press the **OK** button.

5.3.10 Recovery Using an Online Backup from Disk

Log in to AIX as the database administrator (db2<SAPSID>). Start SMIT and select the following menu path:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Recover Database**
- **Restore Database**
- Select the **database name** by clicking on the database name.
- Select **Device**.

- Enter the filenames for the backup in the field Source Directory/Device name(s). If there is more than one filesystem, separate with commas.
- If there is more than one backup in the directory, enter the date and time stamp in the form `yyyymmddhhmmss` in the field Time of Backup Image from which to Restore.
- The field Place Database in Roll Forward Pending State should be set to **YES**. YES is the default. If it is set to NO, you get the following error message when you attempt to restore:

```

DB2admin: Start Database Manager
DB2admin: db2 restore database AUS FROM /backup1/online1, /backup1/online2, /backup1/online3
DB2admin: start of restore at Tue Aug 27 10:37:18 CDT 1996
SQL537N Roll forward is required following the Restore.
DB2admin: finish of restore at Tue Aug 27 10:37:19 CDT 1996

DB2admin: Restore failed. Error Code 4

      Be sure that the following conditions are fulfilled:
      - no other applications are connected to the database
      if restore from tape:
      - you have chosen the right tape device
      - the tape device is mounted
      - the media surface of the tape is not damaged
        (check with an explicit tar-command)

Failure: Restore UNSUCCESSFUL (4)

```

Figure 132. Error When Recovering from Online Backup (No Roll Forward)

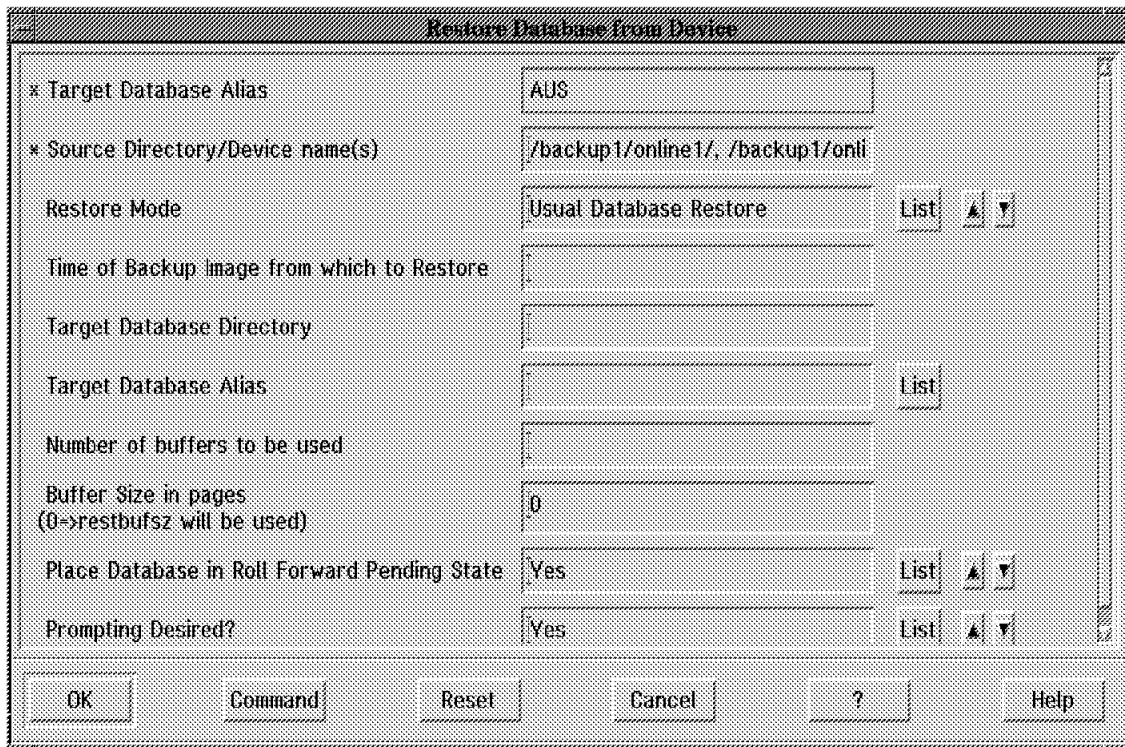


Figure 133. Restore Database from Device Panel

- Press the **OK** button.
- Press the **OK** button in the confirmation pop-up window.

- You will receive a warning message in SMIT. If you want to continue, enter **Y** and press **Enter**.
- When the restore has completed successfully, press the **Done** button to return to the previous menu.
- Press the **Cancel** button from the Restore Database from Device screen to return to the Recover Database screen.
- Select **Roll Forward Database**.
- Press the **List** button for Roll Forward Function to get a list of options available. See Chapter 10, "Recovering the Database" in *BC SAP Database Administration Guide: DB2 for AIX* for a description of the options. In this example, we used the option Reapply + Rollback to roll forward all available logs and rollback any uncommitted transactions. This will return the database to a consistent state and allow R/3 to start.
- If you get an error message similar to the message in Figure 134, check that the archived log file is available. You may need to restore it from your offline storage. Follow the instructions for restoring offline archive log files.

```

Output:
db2 rollforward database AUS to end of logs and stop
SQL1276N The database cannot be brought out of rollforward pending state
until Roll Forward has passed a point in time greater than or equal to
"1996-09-03-22.51.46.000000".

Rollforward Status for Database: AUS

Next log file to be read      = S0000035.LOG
Log files processed          = -
Last committed transaction   = 1996-09-03-22.34.08.000000

```

Figure 134. Roll Forward Process Error Message: Archived Log File Unavailable

5.3.11 Recovery Using an Online Backup from ADSM

Log in to AIX as the database administrator (db2<SAPSID>). Start SMIT and select the following menu path:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Recover Database**
- **Restore Database**
- Select the **database name** by clicking on the database name.

- Select **ADSM**.
- If there is more than one backup in the directory, enter the date and time stamp in the form `yyyymmddhhmmss` in the field Time of Backup Image from which to Restore.
- The Place Database in Roll Forward Pending State field should be set to **YES**. YES is the default. If it is set to NO, you get an error message when you attempt to restore. A recovery from an online restore must roll forward the logs.

After the restore is complete, roll forward the database. Select the following SMIT options to roll forward the database:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Recover Database**
- **Roll Forward Database**
- Select the Roll Forward Function you want. Press the **List** button for a list of available options. In this example, we will use Reapply + Rollback to roll forward to the end of the log files and roll back any uncommitted transactions.

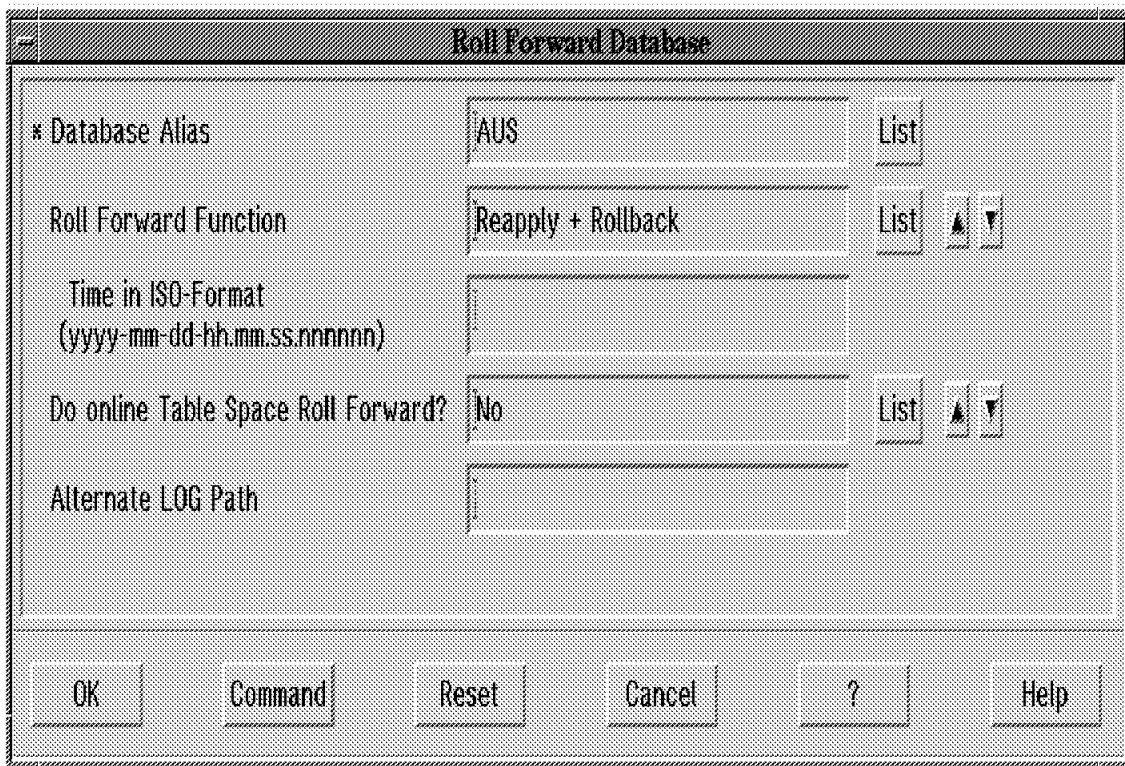


Figure 135. Roll Forward Database Panel

- Press the **OK** button.
- For this example, we backed up the archive log files to ADSM and did not restore them before attempting the roll forward. The following error message resulted from the roll forward attempt:

Output:

```
db2 rollforward database AUS to end of logs and stop
SQL1276N The database cannot be brought out of rollforward pending state
until Roll Forward has passed a point in time greater than or equal to
"1996-08-12-21.30.55.000000".
```

Rollforward Status for Database: AUS

```
Next log file to be read      = S0000001.LOG
Log files processed           = -
Last committed transaction    = 1996-08-12-21.01.30.000000
```

Figure 136. Error Message During Roll Forward Process

- Before the database can be rolled forward, the logs must be restored from ADSM. See the section on restoring offline archive log files in this chapter. Restore the log files and repeat the roll forward.
- After the log files were restored from ADSM, we ran the roll forward again. This time the roll forward was successful as shown in the following SMIT screen:

Output:

```
db2 rollforward database AUS to end of logs and stop

Rollforward Status for Database: AUS

Next log file to be read      =
Log files processed          = S0000001.LOG - S0000002.LOG
Last committed transaction   = 1996-08-13-18.17.05.000000

DB20000I The ROLLFORWARD command completed successfully.

Rollforward Status for Database: AUS

Next log file to be read      =
Log files processed          = S0000001.LOG - S0000002.LOG
Last committed transaction   = 1996-08-13-18.17.05.000000
```

Figure 137. Successful Roll Forward Process

- After the roll forward is complete, you may want to remove the log files recovered from ADSM. See the section on removing used log files in this chapter.

5.3.12 Roll Forward Pending

Roll Forward Pending is a state used by DB2 to protect the integrity of the database. This state indicates that a roll forward process is necessary to ensure consistency of the data.

Roll Forward Pending is set as a result of one of the following:

- You have done a restore of an offline database backup but omitted the command option `WITHOUT ROLLING FORWARD`.
- You have done a restore of an online database backup. This is roll forward recovery that was discussed in 5.3.5, “Methods of Recovery” on page 197.
- You have done a restore of a tablespace-level backup. DB2 requires that log files be applied when a tablespace-level backup is used in recovery.
- The DB2 database manager has detected a media failure and has isolated it at the tablespace level.

The scope of the Roll Forward Pending state depends on the level, either tablespace or database. If the Roll Forward Pending is at the database level, DB2 will not permit any activity against the database. If the Roll Forward Pending is at the tablespace level, access is permitted to tablespaces not participating in the Roll Forward Pending situation. For example, media error can be isolated at the tablespace level. This allows remaining tablespaces in the database to remain accessible for use.

If an I/O error is encountered while trying to read from or write to disk, the tablespace in which the page resides is disabled and placed in Roll Forward Pending state. It is possible that a roll forward to the end of the log files will clear the state. If the Roll Forward Pending state cannot be cleared with just a roll forward of the tablespace, a restore followed by a roll forward is required.

Two exception situations concerning tablespace I/O error exist:

1. If the tablespace in error contains the system catalog tables, you are not able to connect to the database. Recovery of the tablespace can be done by restoring a backup of the tablespace or by doing a database restore. Roll forward could then be applied to the restored tablespace.
2. If the tablespace in error is the last temporary tablespace in the database (PSAPTEMP), the database will be shut down. Restarting the database will re-initialize the temporary tablespace.

5.3.13 Roll Forward Database

The DB2 ROLLFORWARD DATABASE command can be issued from the command line, through the DB2 Database Director, or through the R/3 DB2admin utility.

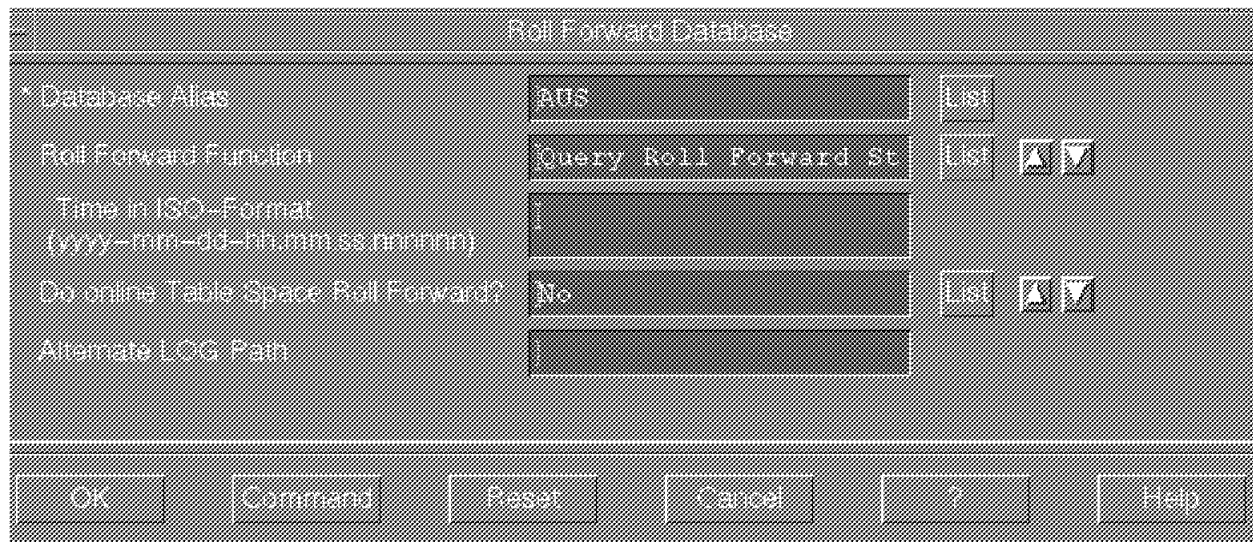


Figure 138. Roll Forward Database Screen in DB2admin

Figure 138 shows the roll forward database screen found in DB2admin. The SMIT path is **SMIT->Applications->DB2admin for R/3: IBM DB/2 for AIX Administration Utilities->Recover Database->Roll Forward Database.**

Roll forward applies transactions recorded in the database log files. The command is invoked after a online database or any tablespace backup has been restored or if tablespaces have been taken offline by the database manager due to a media error.

Restore is the first phase of a complete roll forward recovery of a database or tablespace. After a successful database restore, a database that was configured for roll forward recovery at the time the backup was taken enters a Roll Forward Pending state. It is not usable until the ROLLFORWARD DATABASE command has been executed successfully. If the restore used a tablespace level backup, the tablespace(s) restored enter a Roll Forward Pending state.

When the ROLLFORWARD DATABASE command is issued, if the database is in a Roll Forward Pending state, the database is rolled forward. If the database is not in a Roll Forward Pending state, all tablespaces in the database that are in a Roll Forward Pending state are processed. Note that you cannot perform another database restore operation while the ROLLFORWARD command is executing.

If you restored from a full offline database backup image, you can bypass the Roll Forward Pending state during the recovery process. The RESTORE DATABASE command gives you the option to use the restored database immediately without rolling forward the database.

You cannot bypass the roll forward phase when recovering at the tablespace level or if you restore from a backup image that was created using the ONLINE option of the BACKUP DATABASE command.

The following are some of the command parameters that you will find in the DB2admin utility (Figure 138 on page 210):

- Database alias - This is the alias name of the database to roll forward.
- Rollforward Function

The following roll forward functions are possible:

- Query Rollforward Status

This will list the log files that the database manager has rolled forward, the next archive file required and the timestamp of the last commit transaction since roll forward processing began in Coordinated Universal Time.

- Reapply to End of Logs

Reapplies all committed transactions from all online archive log files residing in the current database log directory. This directory is defined in the database configuration parameter LOGPATH. If additional log files in other directories/devices exist, they can be used in an additional ROLLFORWARD DATABASE command.

- Reapply + Rollback

Reapplies all committed transactions from all online archive log files residing in the current database log directory (defined in the database configuration parameter LOGPATH) and completes the roll forward recovery process by rolling back any incomplete transactions and turning off the Roll Forward Pending state of the database. This will allow access to the database again.

- Reapply to Point in Time

This option executes a roll forward recovery that rolls forward all committed transactions to a specified point in time. If additional log files exist, they can be used in an additional ROLLFORWARD DATABASE command.

- Point in Time + Rollback

This option executes a roll forward recovery that rolls forward all committed transactions to a specified point in time and completes the roll forward recovery process by rolling back any incomplete transactions and turning off the Roll Forward Pending state of the database. This allows access to the database.

- Complete by Rollback

Completes a pending roll forward recovery process by rolling back any incomplete transactions and allow access to the database.

- Time in ISO-Format

You need to enter a timestamp (seven-part character string) that identifies a combined date and time. The format is yyyy-mm-dd-hh.mm.ss.nnnnnn (year-month-day-hour.minutes.seconds.microseconds), which is the Coordinated Universal Time.

- Do online Table Space Rollforward?

Specifies if the tablespace level roll forward is to execute while online. This means that other applications and users are allowed to connect while the roll forward recovery is in progress. However, you cannot access the tablespace in which the roll forward is processing.

- Alternate LOG Path

Specify an alternate LOG path to be searched for archived logs during recovery.

5.3.13.1 How Far to Roll Forward

The R/3 database administrator can clear a Roll Forward Pending condition by issuing the ROLLFORWARD DATABASE command. The point in time to which the roll forward stage proceeds is also controllable by the administrator.

The integrity of the database must be protected; therefore, the earliest point in time at which the roll forward stage can end is the end of the online backup image.

The ISOTIME parameter (Time in ISO-Format in the DB2admin utility) can be used to identify a particular point in time up to which the logs should be applied. This time is specified as the ISO format of the Coordinated Universal Time (CUT).

Coordinated Universal Time (CUT)

The Coordinated Universal Time is used in log records so that the database manager does not have a recovery dependency regarding daylight savings time or other local time anomalies. However, this introduces a degree of complexity for the R/3 database administrator. While backup images are identified via timestamps reflecting local time, rolling forward (since it applies to logs) must designate time in CUT format.

If in your R/3 environment you have various Application Servers running in different time zones, you need to use the CUT format as a way to coordinate the log files that are applied against the database during recovery.

You need to set the same time zone for the db2<SAPSID> user by setting the environment variable TZ = GMT.

An online backup requires roll forward past the end of the backup to ensure integrity. The DB2 recovery history files and the R/3 protocol files can be helpful in this situation. For further details, see 5.3.15, "Managing Backup and Restore Operations" on page 214. The end of logs is a typical stopping point in roll

forward. End means the end of the current log path. Be aware that other logs not in the current log path may need to be moved in the path. This is required for any tablespace recovery strategies.

If you are performing the ROLLFORWARD DATABASE command through the DB2 Database Director or the command line, check the *DB2 Command Reference* for details on the command parameters. The options that you see in the DB2admin have been re-worded to assist you in the roll forward procedure. If you are using the Database Director or command line, you will need to understand the command parameter AND STOP. The AND STOP parameter is a necessary parameter to permit the database manager to rollback any transactions that are not complete after applying the log records to the indicated point. This is true even if END OF LOGS is utilized. Otherwise, the database will remain in Roll Forward Pending status. In the DB2admin utility, the options that allow access to the database are performing the AND STOP option for you. Make sure you understand the option that you are selecting.

5.3.14 Point-in-Time Recovery

After restoring the database from an online or an offline backup, you may want to restore the database to a specific point in time. For example, if the database was corrupted by a user error or programmer error, you may want to recover to just before the point when the database was corrupted. The steps for a point-in-time recovery are as follows:

- Restore the database from an offline or an online backup. Set **Place Database in Roll Forward Pending State** to **YES**
- When the restore completes, you will receive a message stating the last committed transaction and the next log file to be read. Note that all times given are in Coordinated Universal Time. The message will be similar to the message displayed in Figure 139.

```
Y
DB20000I The RESTORE DATABASE command completed successfully.
DB2admin: finish of restore at Wed Sep 4 11:52:47 CDT 1996
Restore SUCCESSFUL (0) !!
DB2admin: db2 rollforward database AUS query status

Rollforward Status for Database: AUS

Next log file to be read      = S0000035.LOG
Log files processed          = -
Last committed transaction   = 1996-09-03-21.59.20.000000 !
```

Figure 139. Output Message from the ROLLFORWARD DATABASE Command

- In this example, we will roll forward until 20:00 on September 3, 1996.
- From the SMIT menu, return to the Recover Database menu.

- Select **Roll Forward Database**.
- Change **Roll Forward Function** to **Point-in-Time + Rollback**.
- Enter the **time** in ISO format for Coordinated Universal Time.
- Press the **OK** button.

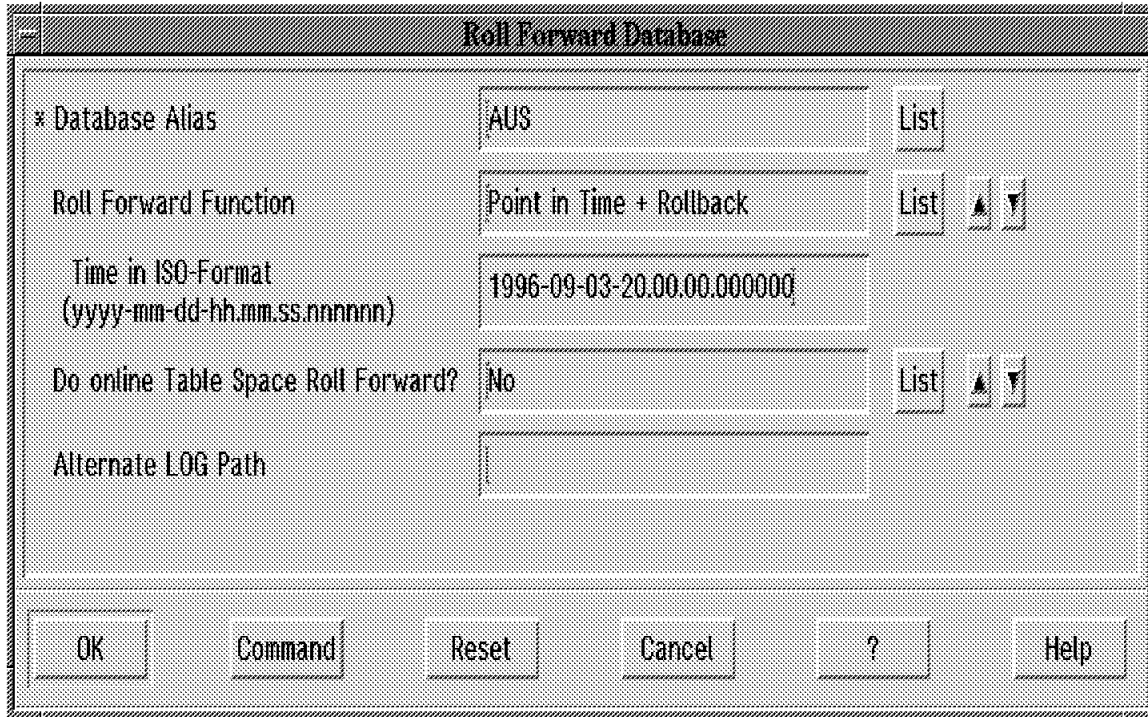


Figure 140. Roll Forward Database Screen in DB2admin

5.3.15 Managing Backup and Restore Operations

The ability to provide to recover from an error to your database depends on your backup and restore operations. The frequency with which you perform backup and restore will affect your ability to recover in a timely manner. To assist in this task, there are facilities available to the R/3 database administrator that will help to keep track of backup and restore activities. We have briefly mentioned those facilities that are available in the R/3 DB2admin tool. These are the protocol files that were discussed in 5.1.3, “Terminology in DB2 and R/3” on page 161. Also, there is a recovery history file provided with the DB2 product. We discuss this file first.

5.3.15.1 Recovery History File

The recovery history file in DB2 contains historical information for a database. The file is maintained for the entire database and resides in the same directory as the database configuration file. If the database is dropped, the history file is lost.

The recovery history file is updated when any one of the following operations is performed:

- A backup of the full database or tablespace(s)
- A restore of the full database or tablespaces(s)

- A load of a table

The last option, (load of a table) is through the DB2 LOAD command. This utility, while available for use, is not usually used in the R/3 environment.

The recovery history file can be viewed through the DB2 Database Director, the R/3 DB2admin tool or by the DB2 LIST HISTORY command. The recovery history file provides a summary of backup information useful in the event that a database or tablespace must be restored. The backup information includes:

- The part of the database that has been copied by a backup, load, or copy from a load
- When the database was copied
- Time of the last restore

For example, to list all the backups, restores, and loads that have been done for our R/3 database (aus), the command would be:

```
db2 list history all for aus
```

An option on the DB2 RESTORE command permits only the history file in a backup image to be restored. This is useful in recovery situations where the history file currently associated with the database is not accessible and information contained in the history file is needed to plan the recovery strategy.

All insertions to the recovery history file are done automatically whenever a backup, restore, or load is performed. A warning will be given if the history file cannot be inserted into or updated due to either a full file system, a damaged history file, or an I/O error. If the file system is full, the current entry will be lost, since insertions are made as the backup, restore, or load is being done.

The recovery history file has its own format. The table below lists the columns contained in the recovery history file and their description:

Column Name	Type	Description
OPERATION	CHAR(1)	Type of operation performed, B=Backup, R=Restore, L=Load
OBJECT	Char(1)	Granularity of operation, D=Full Database, P=Tablespace, T=Table
OBJECT_PART	Char(17)	First 14 characters=timestamp=yyyymmddhhnnss. Next three characters=Sequence Number. Backup can save to multiple files/tapes. Restore/Load always '001'
OPTYPE	Char(1)	Additional operation qualification, F=Offline backup, N=Online backup, R=Load Replace, A=Load Append, C=Load Copy, Blank for other operations
DEVICE_TYPE	Char(1)	D=Disk, K=Diskette, T=Tape, A=ADSM, U=UserExit, O=Other Vendor Device Support, L=NULL
FIRST_LOG	Char(12)	Earliest Log File ID (S0000000 to S9999999). Required for roll forward recovery after full database/tablespace backup, empty on restore
LAST_LOG	Char(12)	Latest Log File ID, also empty on restore
BACKUP_ID	Char(14)	Timestamp 'yyyymmddhhmmss' that references one or more file lines representing backup operation. For full database restore, references full database backup that was restored. For tablespace, references tablespace backup or full database backup used to restore specified tablespaces. Blank for other operations
SCHEMA	Char(8)	Tablename qualifier for load
TABLE_NAME	Char(18)	Name of Loaded Table
NUM_TABLESPACES	Char(3)	Number of tablespaces involved in backup/restore. If non-zero, next lines in fill contain one line for each tablespace.
LOCATION	Char(255)	Where data is saved for backups/load copy. For restore/loads where first part of data was saved. Refers to DEVICE_TYPE. If D or K, then fully qualified file name. If T, then tape volume label. If A, then ADSM Server name. If U or O, then free form text.
COMMENT	Char(30)	Free from text

Table 7. Format of DB2 History File

If the current database is unusable or not available, and the associated recovery history file is damaged or deleted, an option on the DB2 RESTORE command allows only the recovery history file to be restored. The recovery history file can then be reviewed to provide information on which backup image to use to

restore the database. This restored history file will contain all entries up to, but not including the backup used for restoration.

Let's look at a sample of the output.

```

List Recovery History File for HISTORY

Number of matching file entries = 37

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| R  D 19960812182939000  F   A S0000001.LOG S0000001.LOG 19960811192502 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Contains 25 tablespace(s):

00001 SYSCATSPACE
00002 PSAPBTABD
00003 PSAPDDICD
00004 PSAPPOOLI
00005 PSAPBTABI
00006 PSAPUSERII
00007 PSAPCLUD
00008 PSAPSOURCED
00009 PSAPEL30CD
00010 PSAPLOADI
00011 PSAPCLUI
00012 PSAPDOCUI
00013 PSAPSTABD
00014 PSAPPROTI
00015 PSAPEL30CI
00016 PSAPPROTD
00017 PSAPLOADD
00018 PSAPSOURCEI
00019 PSAPUSERID
00020 PSAPES30CI
00021 PSAPSTABI
00022 PSAPDOCUD
00023 PSAPPOOLD
00024 PSAPDDICI
00025 PSAPES30CD

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Comment: DB2
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
00001 Location: adsm/libadsm.a
Standard input

```

From the partial output of the recovery history file, we can see that the entry was a full, offline database restore. The log file information is not relevant. We can see that there were 25 tablespaces that made in the backup image.

5.3.15.2 Managing the Recovery History File

You will want to manage the recovery history file. This can be performed via the DB2 PRUNE HISTORY command. This command is available through the R/3 DB2admin tool, the Database Director or the command line.

The DB2 PRUNE HISTORY command may be used to delete entries from the recovery history file. The syntax for the command is as follows:

```
db2 prune history timestamp with force option
```

The timestamp is used to identify a range of entries in the recovery history file. A complete timestamp (in the format `yyyymmddhhmmss`) or an initial prefix (at a minimum, `yyyy`) may be specified. All entries with timestamps equal to or less than the timestamp provided are deleted from the recovery history file.

The `WITH FORCE OPTION` specifies that the entries will be pruned according to the timestamp specified, even if some entries from the most recent restore set are deleted from the file.

The following are examples of the `PRUNE HISTORY` command:

```
db2 prune history 19960818041900
db2 prune history 19960815 with force option
```

You must be the R/3 `<SAPSID>` user to use this command.

There will be a recovery history file for every database. The size of the file depends on the `REC_HIS_RETENTN` database configuration parameter and the frequency of backups, restores, and table loads. `REC_HIS_RETENTN` is used to set the retention period of the history file; the default is 366 days. If the recovery history file is not needed to keep track of backups, restores, or loads, `REC_HIS_RETENTN` can be set to a small value.

If you want to keep the recovery history indefinitely, set the `REC_HIS_RETENTN` value to `-1`. In this case, the user must explicitly prune the recovery history file. By default, the recovery history file is automatically pruned after recording a full database backup.

No matter how small the retention period, the most recent full database backup plus its restore set will always be kept unless the `PRUNE` with `FORCE OPTION` is used.

5.3.15.3 R/3 Protocol Files

The protocol files are used to log backup/recovery activities. You can view the logs of information generated by the different parts of backup and restore. You also can edit the log archive and user exit profiles as was discussed in 5.1.3, "Terminology in DB2 and R/3" on page 161. Unlike the DB2 recovery history file, you can use an editor, such as `vi`, to view the protocol files. More than likely, you will view them via the R/3 `DB2admin` tool. The following screen is from the `DB2admin` tool:

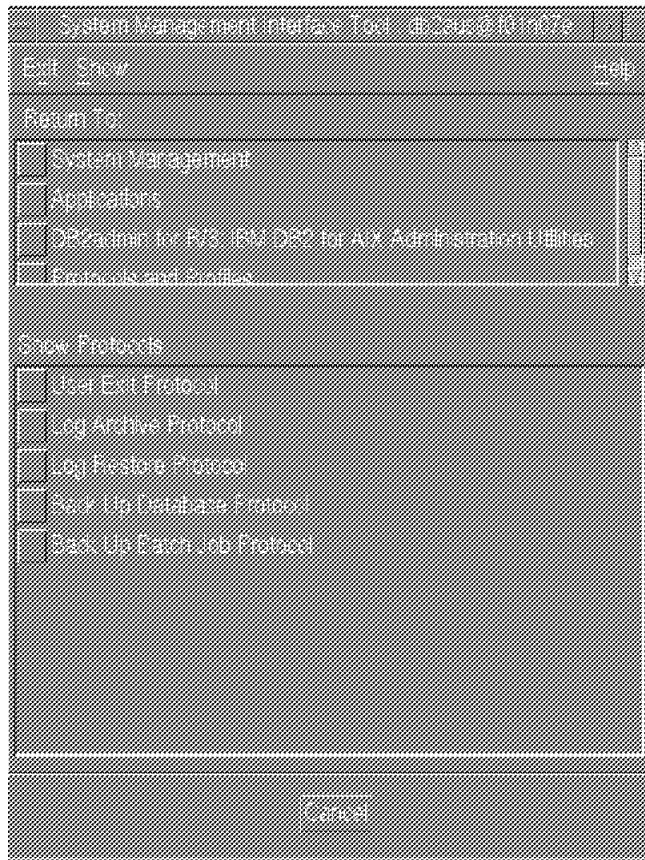


Figure 141. Show Protocol Screen in DB2admin Tool

The path to this screen is **SMIT->Applications->DB2admin for R/3->Protocols and Profiles->Show Protocols**.

They are described as follows:

- User Exit Protocol

If the database parameter USEREXIT is enabled, you will be using the R/3 User Exit program called db2uexit. This protocol file is updated during log archiving or retrieval or if a database backup is done through the DB2admin utility. The path for this files is set in the User Exit profile, .db2uexit. This profile is found in the DB2 instance owner's home directory. In our environment, the path was set to \$INSTHOME/errors, where \$INSTHOME is the DB2 instance owner's home directory. The errors subdirectory must exist before writing begins or an error will be generated.

- Log Archive Protocol

When LOG RETAIN is enabled, return codes about the archiving process are logged into a file. Each archiving process will write messages to a different file. These files are in the DB2 instance owner's home directory under saparch.

- Log Restore Protocol

When logs are used in the restore process, the different stages of the restore are logged in a special file. These files are in a subdirectory called sapbackup under the instance owner.

- Back Up Database Protocol

This option will check the progress of the backups that were performed using DB2admin. Included in the information is timestamps, warnings, and error messages. The protocol file is also located in the same directory as the User Exit protocols. The file is called DB2_BACK_REST.LOG

- Back Up Batch Job Protocol

This file records the information when a backup batch job is performed. The file is located in the batch subdirectory of the instance owner. The file name is brdb6ucron.log.

5.3.15.4 Using the Backup Database Protocol File

This section shows an example usage of the backup database protocol file. The following is an excerpt from the file:

```
*****
=====
=====
Actual date: Thu Aug 22 14:18:01 CDT 1996
db2 backup database AUS TABLESPACE PSAPUSER1I ONLINE USE ADSM
DB2admin: start of backup at Thu Aug 22 14:18:01 CDT 1996

Backup successful. The timestamp for this backup image is : 19960822141802

DB2admin: finish of backup at Thu Aug 22 14:18:22 CDT 1996
*****
=====
=====
Actual date: Thu Aug 22 14:38:09 CDT 1996
```

From the information in the backup database protocol file, you can determine the timestamp of the backup. This information can be used in a recovery situation. You can tell that this was an online tablespace backup of PSAPUSER1I that was done with ADSM. The second line shows the DB2 command that was executed for the backup. You can calculate the difference between the timestamps that indicate the start of the backup and the finish of the backup. This will give you an indication of the duration needed to backup this database object.

Managing Protocol Files

The R/3 protocol files mentioned in this section that are used to store information regarding backup and restore are appended to. They have no predetermined size. Therefore, as part of the R/3 database administrator's tasks, these files should be monitored for growth and periodically deleted.

5.3.16 Backup and Restore Considerations and Recommendations

Your backup strategy should be one that allows you to reproduce the database to a point of consistency that is acceptable to your operation with a minimal amount of system downtime. An offline database backup will provide more consistency. However, in a production system, this may not always be scheduled as often as the R/3 database administrator would like. An online database backup requires all logs to not only be consistent but also available for the recovery. The online backup itself must also recover a consistent database. If any of the logs that are required are not available, recovery is not possible. This requirement of consistent, accurate log files is assisted with different kinds

of archiving media, such as tape, disk, or ADSM. Any archiving done must also be checked for correctness.

The frequency of the full database backup will depend on the level of activity in the database and the time required for database recovery. A high level of activity increases the number of logs that are created between complete backups. Therefore, the recovery time for the database will increase.

All R/3 production databases should be configured for archival logging.

For performance reasons, the log files should not reside on the same physical disk as the archive log files or other database files.

Set `.db2uexit` to move inactive log files to disk. Use the file paths recommended by SAP in the installation documentation. Be sure to monitor the directory that in which the archived log files are stored to ensure that space is always available. If logging is not possible in a DB2 database due to a file system full, the database will crash until action is taken to correct the situation. Use the options in SMIT to move old archived log files to tape or to ADSM. See the section 5.5, "Managing Archive Log Files" on page 230 later in this chapter for more information.

If you do not have sufficient disk space to save archive log files to disk, use tape. Keep in mind that the tape drive must be dedicated to saving archive log files. If you use tape for archiving log files, you will need another tape or another device to use for backups.

5.4 Performing a Redirected Restore

You can redefine the containers in your tablespace with an option in the DB2 Database Director. It is done as a part of the recovery process. This special option is called the Redirected Restore option and is only supported through the DB2 Database Director. It allows you to add a new container, change a container name, change the size of a container, change the path of an existing container, and remove a container. You cannot change the type of tablespace. For example, you cannot change a DMS tablespace to a SMS tablespace or vice versa. Before performing the Redirected Restore, you must have a valid backup of the tablespace or database. The backup can be offline or online.

Some of the situations where you may need to use the Redirected Restore option are:

- The file containers cannot be accessed.
- You want to restore a backup to a different system.
- You created a tablespace for your R/3 system that is no longer going to be used. This can happen if you are working on a system that was migrated from an Oracle or Informix database to DB2, using the R3load utility (`R3loadctl`) for data structures of Version 3.0D. The tablespaces `PSAPEL30CD/I` and `PSAPES30CD/I` are not used by the migrated system.
- You want to move a tablespace to a different file system to make it easier to manage a growing system.

Before you can perform a Redirected Restore, check the following:

1. You have a backup of the tablespace(s) or database you wish to restore.

2. Make sure there is enough space in the file system(s) that you will be using. You may need to increase the size of the target file system. If the file system does not yet exist, you will need to create it.
3. Make sure the file system(s) to be used for the new containers have the appropriate owner and permissions.

5.4.1 Changing Containers in a Tablespace

This section illustrates the Redirected Restore option in the DB2 Database Director. We also discuss the DB2 ROLLFORWARD DATABASE command. Note that the DB2 ROLLFORWARD DATABASE command is used in other restore operations as well as the Redirected Restore.

In the following example, we move the PSAPBTABD tablespace from /db2/AUS/sapdata1 to /db2/AUS/sapdata7. We first need to make sure we have a valid backup for the tablespace. A filesystem for the new container must be defined to AIX. We use the Redirected Restore to restore the tablespace to the new filesystem and then roll forward the tablespace to the end of the logs.

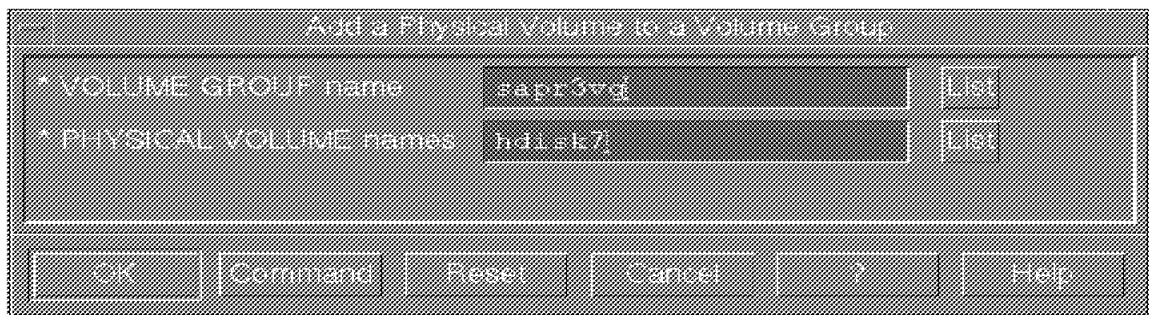
5.4.1.1 Verify a Valid Backup Exists

Log in to AIX as the database administrator, db2<SAPSID>, and select the following SMIT menu options:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- You can select either **Backup the Database** or **Recover Database**.
- If you are using ADSM, select **Show Existing ADSM BACKups**. You may also select **List Back Up / Recovery history file**.
- If you do not have a backup of the tablespace(s) you want to change or if you are not sure the backup exists, do another backup!

5.4.1.2 Define Location for New Containers

1. Add a physical volume to the volume group by selecting:
 - **System Storage Management (Physical & Logical Storage)**
 - **Logical Volume Manager**
 - **Volume Groups**
 - **Set Characteristics of a Volume Group**
 - **Add a Physical Volume to a Volume Group**



2. Create the logical volume by selecting:

- System Storage Management (Physical & Logical Storage)
- Logical Volume Manager
- Logical Volumes
- Add a Logical Volume
- Select the Volume Group name.

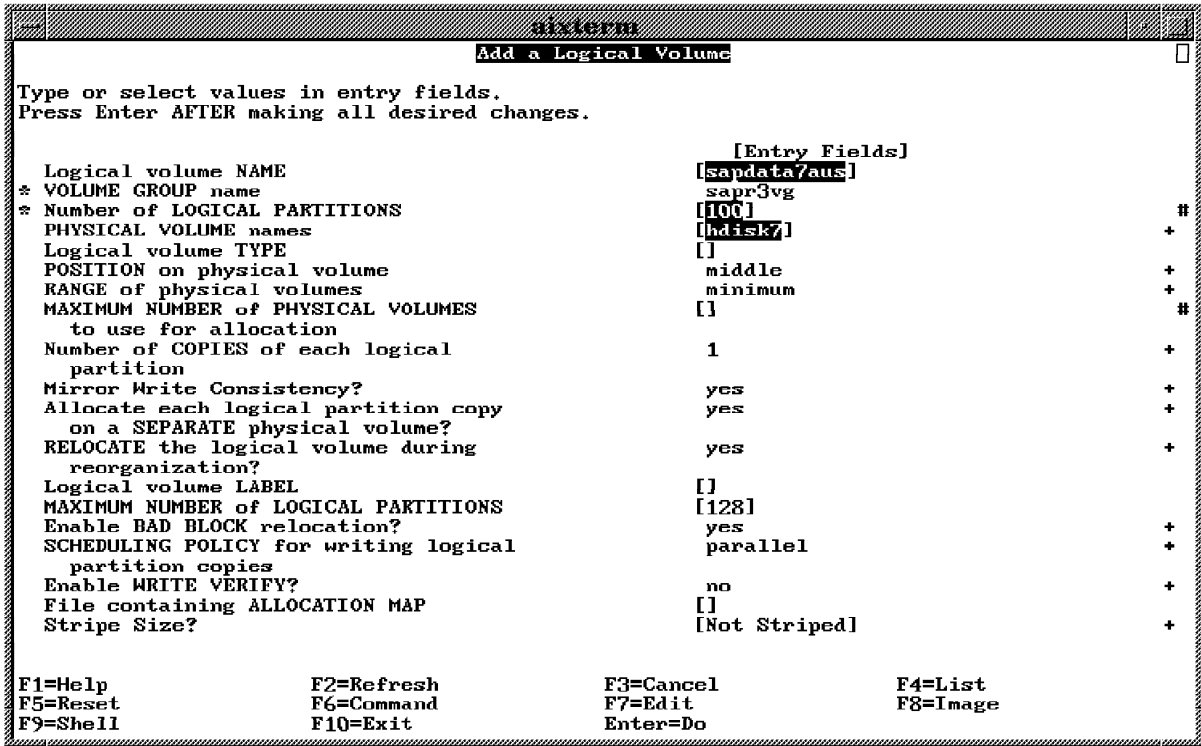


Figure 142. Using SMIT to Add a Logical Volume

3. Create JFS filesystem by selecting:
 - System Storage Management (Physical & Logical Storage)
 - File Systems
 - Add /Change/Show/Delete File Systems
 - Journaled File Systems
 - Add a Journaled File System on a Previously Defined Logical Volume

aixterm															
Add a Journalled File System on a Previously Defined Logical Volume															
Type or select values in entry fields. Press Enter AFTER making all desired changes.															
		[Entry Fields]													
* LOGICAL VOLUME name		sapdata7aus	+												
* MOUNT POINT		[/db2/AUS/sapdata7]	+												
Mount AUTOMATICALLY at system restart?		yes	+												
PERMISSIONS		read/write	+												
Mount OPTIONS		[]	+												
Start Disk Accounting?		no	+												
Fragment Size (bytes)		4096	+												
Number of bytes per inode		4096	+												
Compression algorithm		no	+												
<table border="0"> <tr> <td>F1=Help</td> <td>F2=Refresh</td> <td>F3=Cancel</td> <td>F4=List</td> </tr> <tr> <td>F5=Reset</td> <td>F6=Command</td> <td>F7=Edit</td> <td>F8=Image</td> </tr> <tr> <td>F9=Shell</td> <td>F10=Exit</td> <td>Enter=Do</td> <td></td> </tr> </table>				F1=Help	F2=Refresh	F3=Cancel	F4=List	F5=Reset	F6=Command	F7=Edit	F8=Image	F9=Shell	F10=Exit	Enter=Do	
F1=Help	F2=Refresh	F3=Cancel	F4=List												
F5=Reset	F6=Command	F7=Edit	F8=Image												
F9=Shell	F10=Exit	Enter=Do													

4. Mount the new filesystem by entering the MOUNT command:

```
mount /db2/AUS/sapdata7
chown db2aus.sysadm /db2/AUS/sapdata7
chmod 700 /db2/AUS/sapdata7
```

5.4.1.3 Restore Tablespace to New File System

1. Log in to AIX as the database administrator, db2<SAPSID>.
2. Start the Database Director. See 1.3.6, "Database Director" on page 10 for more information on starting the Database Director.
 - Click on **instance** (DB2AUS).
 - Click on **Databases**
 - Hold down the right mouse button on the database icon on the pull-down menu. You may also select the database icon by clicking on it and selecting **Selection** from the menu bar.
 - Select **Recover**.
 - Select **Pause to redefine table space containers** in the Recover screen as shown in Figure 143 on page 225.

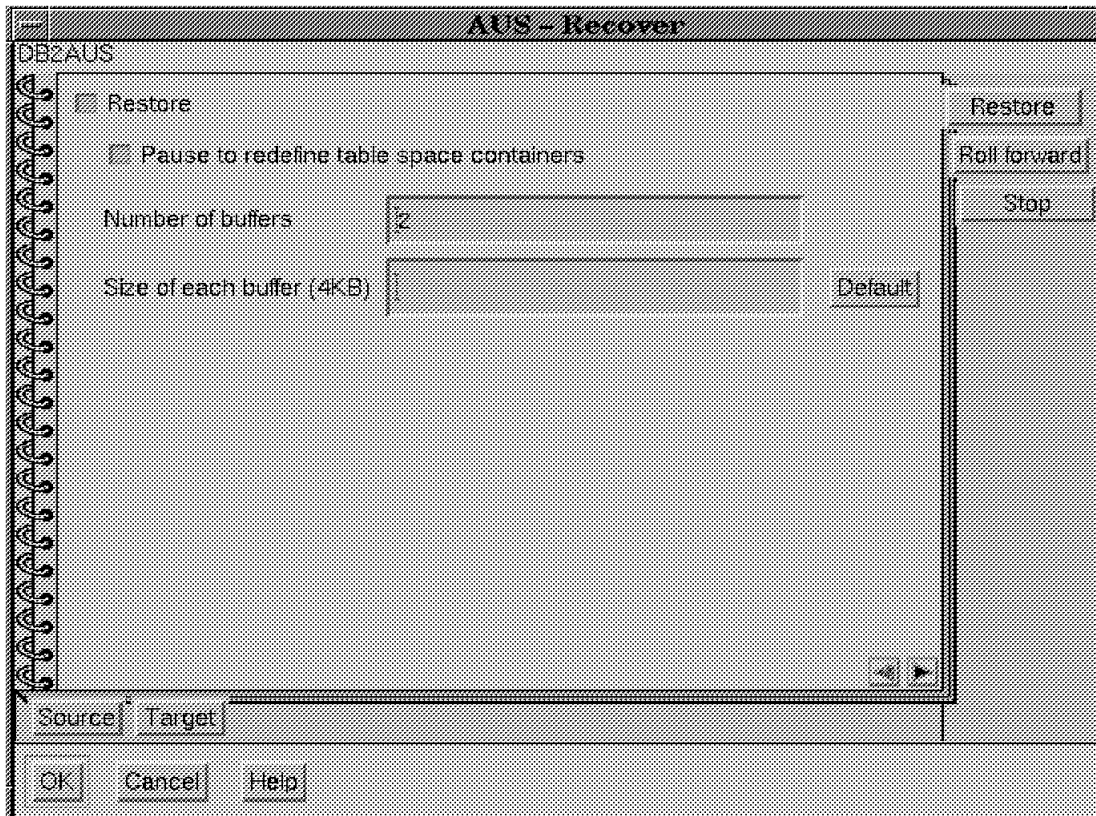


Figure 143. Recover Screen in Database Director (Redirected Restore Option)

- Press the **Source** button near the bottom of the screen.
- Select the **backup method** you used to backup the tablespace. If you backed up the tablespace to a directory, you will need to specify the directory path in the Paths input field on this screen. In this example, we used ADSM.
- Enter the **date and time of the backup** you want to restore.

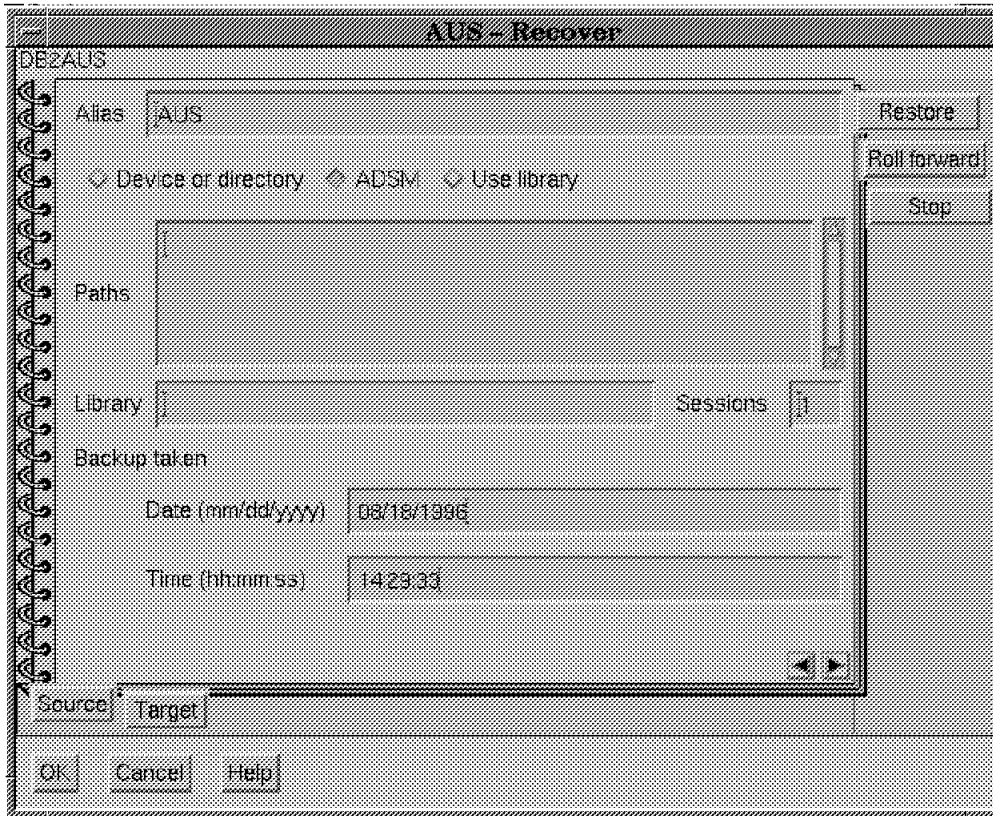


Figure 144. Redirected Restore using the Database Director

- Press the **OK** button to start the restore
- If you get the error message displayed in Figure 145, check Chapter 6, “Monitoring and Troubleshooting” on page 237 for more information.

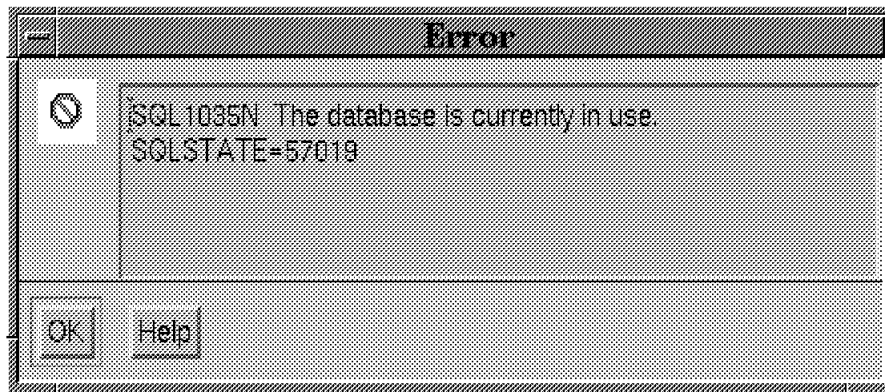


Figure 145. Error Message During Redirected Restore

- Question DBA2231W is displayed. Press the **Redefine** button.

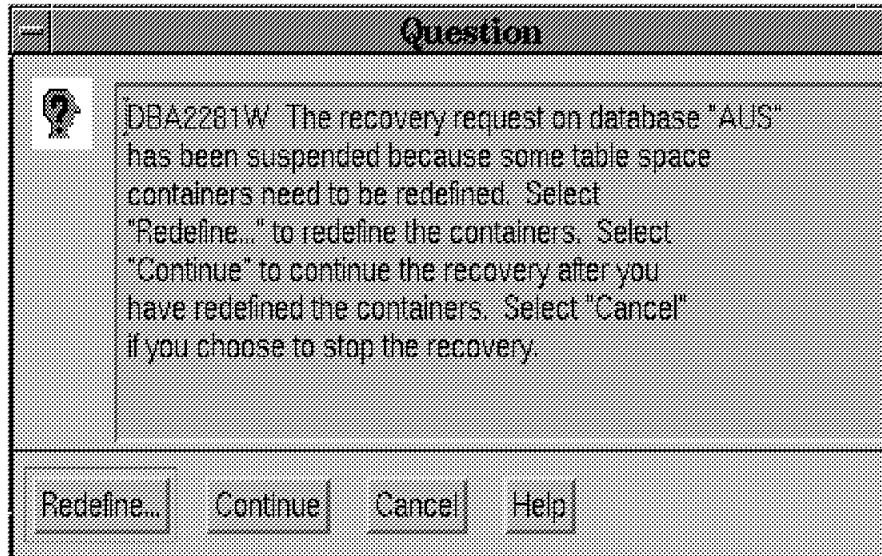


Figure 146. Pausing to Redefine Containers

- The current container definitions are displayed in the Redefine Table Space Containers screen.

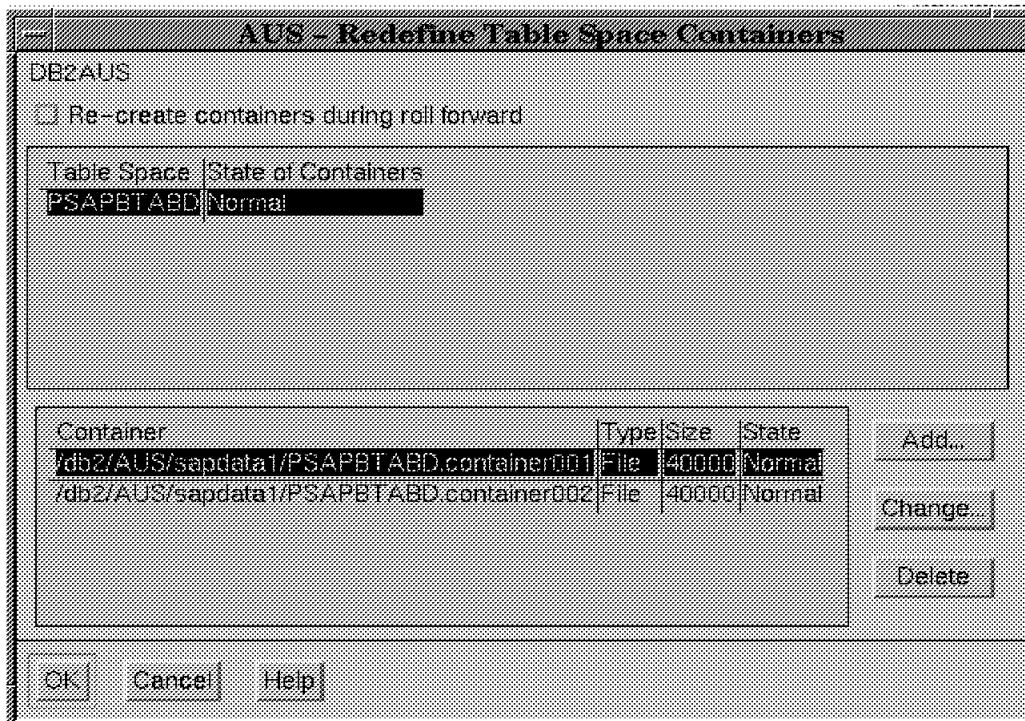
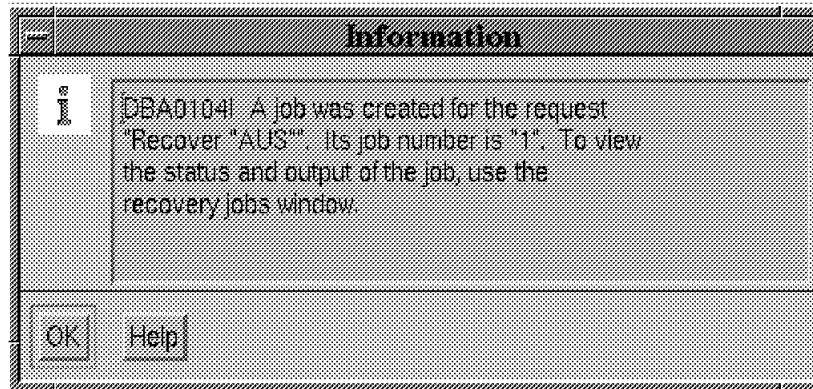


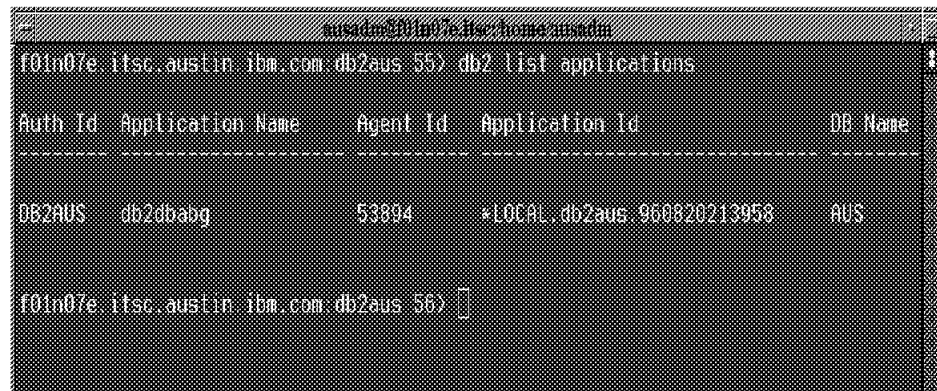
Figure 147. Container Specification in Redirected Restore

- Press the **Change** button to change the container definition for each container that is assigned to the table space as shown in Figure 147.
- Press the **OK** button.
- The DBA2281W question reappears. This time, press the **Continue** button.

- The following information message is displayed.



- Press the **OK** button. The message states that you can view the status of the restore job with the recovery jobs window. This did not work in our testing. We used SAP Release 3.0C.
- To check the status of the recover job, login to db2<SAPSID> and enter db2 list applications. Look for the application db2dbabg.



- If the db2dbabg job is no longer running, check the status of the database. Enter db2 get db cfg for <SAPSID>. If "Rollforward pending" is equal to TABLESPACE, then you must roll forward the database before you can start R/3. If all logs were available during the Redirected Restore, then the rollforward would have been done automatically. If a needed log file had already been archived, then the rollforward does not complete.
- You can also list the status of the tablespaces. Enter the db2 list tablespaces command. Look for the tablespace you are restoring.

```
Tablespace ID          = 4
Name                   = PSAPBTABD
Type                   = Database managed space
Contents               = Any data
State                  = 0x0080
Detailed explanation:
Roll forward pending
```

5.4.2 Make Inactive Log Files Available for Roll Forward

This step is only needed if the roll forward did not complete after the Redirected Restore completed. The roll forward will complete automatically if the log files are available in the log_dir directory. If the log files are not available, it will leave the tablespace in Roll Forward Pending state.

- Log in to the database administrator, db2<SAPSID>
- Select the following SMIT options
 - **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Manage Log Files**
 - **Make Available Inactive Log Files**
 - Select the database by clicking on the **database name**.
 - Enter the **log_archive directory file path** in the Inactive LOG files location field.
 - Select the **log files needed for recovery**.
 - Press **OK**.

5.4.3 Roll Forward Database

This step is only needed if the roll forward did not complete after the Redirected Restore completed. The roll forward will complete automatically if the log files are available in the log_dir directory. If the log files are not available, the tablespace is left in the Roll Forward Pending state.

- Select the following SMIT options:
 - **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Recover Database**
 - **Roll Forward Database**
 - Enter **Reapply + Rollback** in the Roll Forward Function field.
 - Press the **OK** button.
 - When the command completes, exit SMIT.

- Check that the roll forward completed by entering the following command

```
db2 get db cfg for <SAPSID>
```

 Check that Roll Forward Pending is set to **NO**.

5.4.4 Removing Log Files

This step is only needed if the roll forward did not complete after the Redirected Restore completed. The roll forward completes automatically if the log files are available in the log_dir directory. If the log files are not available, the tablespace is left in the Roll Forward Pending state.

- Log in to the database administrator, db2<SAPSID>
- Select the following SMIT options:
 - **Applications**
 - **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
 - **Manage Log Files**
 - **Remove Used Log Files**
 - Select the correct database by clicking on the **database name**.
 - Press **OK** for the confirmation message.
 - The log files that were uncompressed and copied from the log_archive to the log_dir directory are removed (cleaned up).

5.5 Managing Archive Log Files

This section discusses the management of archive log files. We look at how to archive inactive log files and remove them from the log_archive file system by using ADSM. We also restore archive log files from ADSM.

5.5.1 Backing up the Archive Log Files to ADSM

Log in to AIX as the database administrator, and enter smit on the AIX command line. Select the following options from the SMIT menu:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Manage Log Files**
- **Interactive Archival of Log Files**
- **Archive inactive Log Files**
- Select an archive function by clicking on the option you want. The options are as follows:
 - s: Save archived logs
 - sc: Second copy of archived logs
 - ss: double save archived logs
 - sd: save and delete archived logs
 - scd: second save and delete archived logs
 - ssd: double save and delete
 - ds: delete saved logs

- dc: delete double saved logs
- **Change Archive Target to ADSM**
- **Change Archive Mode to Execute**

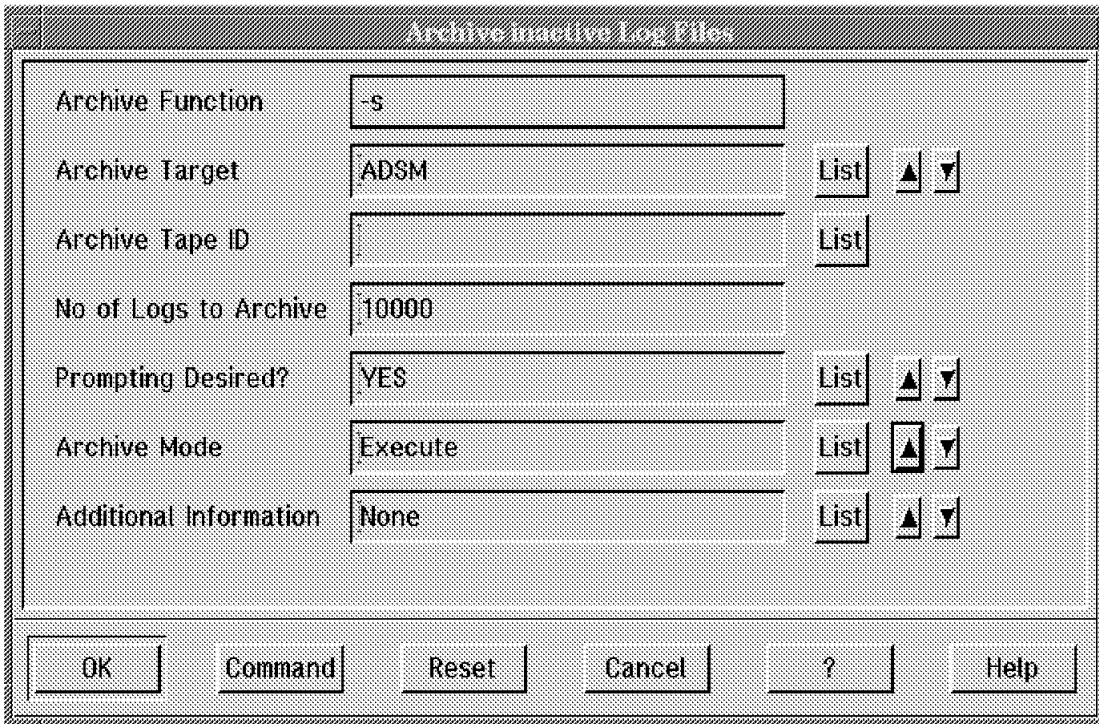


Figure 148. Saving Inactive Archive Log Files via ADSM

- Press the **OK** button.
- Enter cont when prompted with the message shown in Figure 149 on page 232.

Output:

```
BR002I BRARCHIVE 3.0C  
  
BR008I Offline redo log processing for database: AUS  
BR009I BRARCHIVE action ID: acswcdsz  
BR010I BRARCHIVE function ID: sve  
BR011I 1 offline redo log file found for processing, size 0.016 MB.  
BR130I Backup device type: adsm  
BR109I Files will be saved by backup utility at file level.  
BR256I Please enter 'cont' to continue, 'stop' to cancel the program:  
cont ]
```

Figure 149. Prompt to Continue Message

5.5.2 Removing Files from log_archive File System with ADSM

If the log_archive file system runs out of space, the log files will continue to be written to the log_dir directory until the log_dir file system runs out of space. If the log_dir file system does run out of storage, the database will crash and a crash recovery will be required to restart the database and R/3. To avoid this problem, the log_archive file system should be monitored carefully and cleared out periodically. The frequency at which the file system needs to have files removed, will depend on the activity of your system. These archive log files are essential for restoring the database; so they should be carefully backed up. To clear out the files which have already been saved, you can use one of the archive functions that will save and delete files, or you can delete the files after they have already been saved.

These log files should be kept in your backup pool until you are certain that they will not be needed again for recovery. If you at some later point in time have an online or offline backup that is a valid recovery point, you can get rid of the older log files permanently. This should be part of your backup and recovery strategy.

To remove archive log files that have already been saved, log in to AIX as the database administrator, db2<SAPSID>. Set your display variable. From the SMIT menu, select the following options:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Manage Log Files**
- **Interactive Archival of Log Files**
- **Archive inactive Log Files**
- **Select -ds: Delete Saved Logs.**

- Change the Archive Target to **ADSM**.
- Change Archive Mode to **Execute**.
- Press the **OK** button.

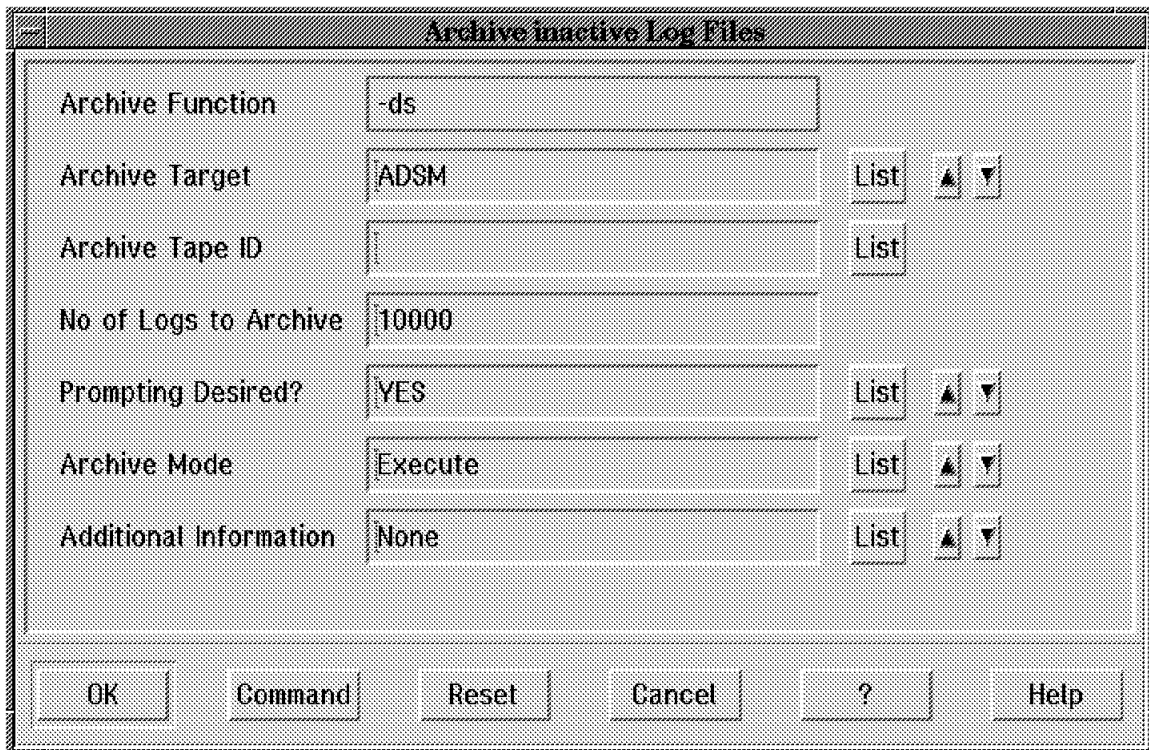


Figure 150. Deleting Inactive Archive Log Files

- Enter cont when prompted in the SMIT message.

5.5.3 Restoring Offline Archive Log Files

If you need to restore archive log files that you have already moved to offline storage, log in to AIX as the db2<SAPSID> user ID. Set your display variable; if necessary, select the following SMIT options:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Manage Log Files**
- **Restore Log Files**
- **Change Restore Log Source to ADSM**
- Enter the first log you want to restore in the First Log No to Restore field. In this example, we want to restore log number 35 (S0000035.LOG.Z).
- Enter the last log you want to restore in the Last Log No to Restore field.
- **Change Restore Mode to Execute.**
- Press the **OK** button.

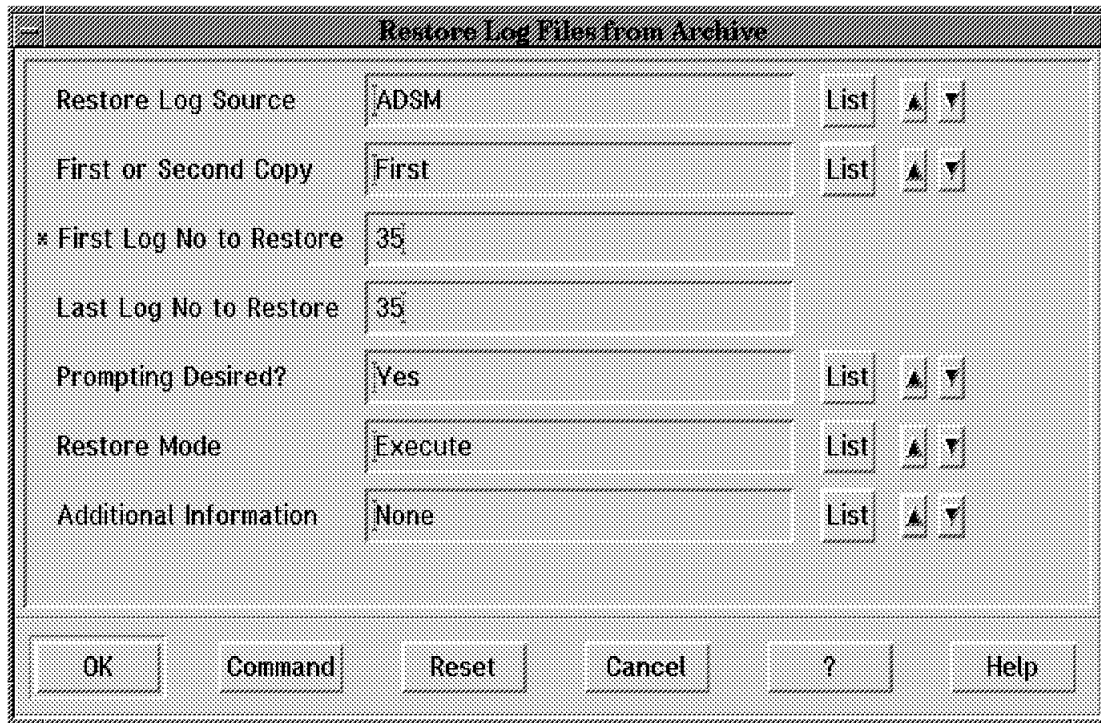


Figure 151. Restoring Archived Log Files with ADSM

- Enter cont when prompted by SMIT.
- After you have finished with the log files, you may want to remove them from the archive_log filesystem. See 5.5.5, “Removing Used Log Files” for more information.

5.5.4 Determining Backed Up Log Files

To check which log files have been backed up, log in to the database administrator and enter the following from SMIT:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Manage Log Files**
- **Interactive Archival of Log Files**
- **Show Archive List**
- Select the appropriate **backup method**. For example, if you use ADSM, select **Log Files Archived with ADSM**.

5.5.5 Removing Used Log Files

After rolling forward the database, you may have log files in the archive log directory that are no longer needed. You can remove these files after the roll forward is complete to free up the disk space. You should not remove the log files unless you are sure that you have backups of the log files or that you have a complete offline backup taken after the log files were created. If you want to remove old log files that have already been used for a roll forward, log in to AIX as the database administrator, db2<SAPSID>, and enter the following SMIT options:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Manage Log Files**
- **Remove Used Log Files**
- Select the database by clicking on the **database name**.
- Press the **OK** button in the confirmation pop-up window.

Chapter 6. Monitoring and Troubleshooting

This chapter provides some tips and techniques to assist you in monitoring your R/3 DB2 system, both to prevent problems from occurring and to help if you should experience problems. This is not a conclusive guide. Rather, it is a guide based on practical experience. Should you experience a problem, you will need to isolate where the problem occurred. A methodology to help you monitor your R/3 system is desirable.

6.1 Monitoring

The best way to avoid having to use the next section on troubleshooting is to periodically monitor your system and respond to problems before they happen. The most difficult part of monitoring the system is knowing what to monitor. As part of your ongoing activities, you should be checking for conditions that, depending on the number of users and amount of activity, will fill certain files and/or log files in your system.

The following section describes monitoring activities that the R/3 administrator, database administrator, or AIX administrator should perform. The SAP* user ID on R/3 has all of the required authorizations to perform R/3 transactions. Any user ID that has the required authorizations can perform these R/3 transactions.

In addition to the monitoring activities, there are housekeeping jobs that should be run periodically to clean up old spool jobs, old batch jobs and old log files. See OSS note 16083 for more information and for recommended frequency to run these jobs.

6.1.1 Alerts

This section goes through some of the AIX and R/3 alerts.

6.1.1.1 SAP Internal Event Alerts

Log in to R/3 and select the following menu path:

Tools -> Administration -> Monitoring -> Performance -> Alerts -> Global -> SAP system (transaction AL01)

Click on the button under the column "ALERT - text" to display alert categories.

Click on the category to see the alert details.

Respond to any yellow or red detail entries.

6.1.1.2 AIX File Systems

Select the following menu path:

Tools -> Administration -> Monitoring -> Performance -> Alerts -> Local -> Filesystem (AL18)

Check for any file systems that are getting full. You also need to find out which tablespaces, in particular which containers in those tablespaces, are affected. Note, that you can fill up containers in tablespaces and still have space available in the AIX file systems. Monitoring only AIX file systems does not always indicate a tablespace full situation.

6.1.1.3 SAP Events

R/3 monitors the state of internal events such as response times, ABAP errors, update errors, and system log errors.

Select the following menu path:

Tools -> Administration -> Computing Center -> Management System -> Control -> Control panel (RZ03)

Select a server and press the **Choose** button.

Select **Monitoring -> Alerts per server -> Alert details**

Double-click on any line of the display to see detailed entries.

6.1.2 System Logs

There are several logs that should be monitored on a daily basis. These are described below.

6.1.2.1 R/3 System Log

The R/3 system log contains any errors that are recognized by a R/3 Work Process and logged through R/3.

Select the following menu path:

Tools -> Administration -> Monitoring -> System log (SM21)

Enter the time and date of first entry in the system log you want to look at and press the **Refresh Syslog** button. Scroll through the system log and look for abnormal events such as ABAP errors, users locked out, buffers that are full, and other unexpected activities.

6.1.2.2 DB2DIAG.LOG and DB2ALERT.LOG

Diagnostic information is provided by DB2 in different forms depending on the target audience. The main audiences are the database administrator and IBM service personnel. The information required by a database administrator should be in their own language, and the information should describe the reason for failure. A detailed description of the error should help the R/3 database administrator fix the problem, for example, a resource limitation such as memory or disk. The IBM service personnel need the descriptive information that is contained in the DB2DIAG.LOG file. They also need more detailed information, such as database control blocks. The database control block information will be dumped into separate files, and therefore, it will not be contained in the DB2DIAG.LOG file.

There are two files that the R/3 DBA can monitor for DB2 problems. Both files are located in the <SAPSID> subdirectory, sqllib/db2dump. The files are DB2DIAG.LOG and DB2ALERT.LOG. There is an DB2 environment variable that you can set at the instance level to obtain different levels of information being recorded into these files. The environment variable is DIAGLEVEL, and it has the following settings:

- 0 - No error logging
- 1 - Log severe errors
- 2 - Log severe and non-severe errors

- 3 - Log severe, non-severe and warning errors
- 4 - Log severe, non-severe, warning and informational errors

The default setting is DIAGLEVEL 3. You can change the level with the update database manager configuration command. For example, to change the DIAGLEVEL to 4:

```
db2 "update database manager configuration using diaglevel 4"
```

Remember that for any changes made at the instance level of DB2 to take effect, the instance must be stopped and restarted. This means either issuing a db2stop, stopsap db, or stopsap all.

When an error is logged and the DIAGLEVEL is set at the proper level, the DB2DIAG.LOG file is updated with information about the error. If an error is determined to be an alert, then an entry to the DB2ALERT.LOG file is made and an entry in the AIX syslog is also made. You may notice other files in the sqllib/db2dump directory. These files may either have a .trp extension to the file name (trap files) or a .dump extension (dump files). You cannot read these files. They are intended for IBM service personnel. If the problem is severe, you may need to submit these trace or dump files to service personnel.

When you experience a DB2 error, the following steps should be performed:

- Check the online error message text by issuing the following command:
db2 ? pppnnn (where ppp is SQL, DBa, or DBI and nnn is the error number)
- If the actions in the message text did not resolve the problem, examine the DB2DIAG.LOG error log. You can use any editor to view the file.
- Errors are appended to DB2DIAG.LOG as they occur. So, you would likely want to examine the end of the log file for the most recent errors. You may find SQLCA information and/or diagnostic messages. These diagnostic messages are used to aid the description of the error. Note, these messages are translated and not available in the online help. The diagnostic message should aid you in resolving the problem. If any network alerts (SNA or SNMP) were generated, they will be recorded in the DB2DIAG.LOG file and the DB2ALERT.LOG file. Also, dump files may have been created when the error occurred. These dump files are only to be examined by IBM.
- If the problem still exists, you should seek assistance from other support people in your organization or IBM.

Let's look at the contents of a sample DB2DIAG.LOG file.

```

Sun Aug 11 17:12:44 1996
db2aus pid(41402) process (db2agent (AUS))
buffer_pool_services sqlbAllocateExtent Probe:830 Database (AUS)

Tablespace 25(PSAPP00LD) is full

Sun Aug 11 17:12:45 1996
db2aus pid(41402) process (db2agent (AUS))
buffer_pool_services sqlbObtainDataExtent Probe:800 Database (AUS)
DIA9999E A internal return code occurred. Report the following : "FFFFD121".

Sun Aug 11 17:12:45 1996
db2aus pid(41402) process (db2agent (AUS))
buffer_pool_services sqlbObtainDataExtent Probe:800 Database (AUS)

Obj={pool:25;obj:34315;type:0} State=x27 Parent={25;34315}, EM=879441, PP0=879521

Data Title :SQLB_OBJECT_DESC pid(41402)
0019 860b 0019 860b 0000 0000 0000 0000 .....
0000 5e4d 82ab 0000 0001 5788 0000 0000 ..-M.....W.....
0001 5790 0000 0002 0000 0027 0000 0000 ..W.....'.....
0000 0000 0000 0000 .....

```

The partial output from the DB2DIAG.LOG shows some of the following regarding the problem:

- Instance name - The instance name here is db2aus.
- Time of the error - Here the time is Sun Aug 11 17:12:44 1996.
- The component is buffer_pool_services.
- The actual function name is sqlbAllocateExtent.
- The database alias name is AUS.
- The next line is the most informative. It gives a description of the error: Tablespace 25 (PSAPP00LD) is full.

As the R/3 DBA, you may want to remove old error logs and dump files. The files used for problem determination that can be erased are the following:

- DB2DIAG.LOG
- DB2ALERT.LOG
- ####.DMP(s)
- ####.TRP(s)

These files can be removed at any time. New ones will be created as needed.

6.1.2.3 Developer Traces

R/3 trace files are located in the /usr/sap/<SAPSID>/DVEBMGS<nn>/work directory, where <nn> is the instance number. The <SAPSID>adm user ID on AIX is set up to change to the /usr/sap/<SAPSID>/DVEBMGS<nn> directory with the command cd. From this directory, change to the work directory.

These trace files can also be viewed from R/3. Select **Tools -> Administration -> Monitoring -> Traces -> Developer traces (ST11)**.

The most recently modified files are at the top of the list. Double-click on a file to display the contents.

6.1.2.4 AIX Error Log

The AIX error log is useful for monitoring hardware problems or software problems at the AIX level.

Log in to AIX as root and enter `errpt` on the AIX command line. The time stamp format is MMDDHHmmYY, where MM is month, and mm is minutes.

`errpt -a` displays a detailed analysis of the entire error log.

`errpt -d H` displays all hardware errors.

6.1.2.5 ABAP/4 Dumps

When a program running in a work process abends, R/3 will dump the program. The dumps are stored in the database in the SNAP table.

To view dumps, select **Tools -> Administration -> Monitoring -> Dump analysis (ST22)**. Press the **selection** button. Enter the date range or time range you want to search. Double-click on the dump file you want to review.

6.1.3 Database Monitoring

See 4.5, "Database Monitoring" on page 134 for more information on monitoring the database.

6.1.3.1 Free Space in Tablespaces

Select **Tools -> Administration -> Monitoring -> Performance -> Database -> Tables/Indexes (DB02)**.

Press the **Detailed Analysis** button in the Tablespaces section of the screen. Check the free space in each tablespace. See Chapter 4, "Database Administration" for details on how to add additional space to a tablespace if necessary.

6.1.3.2 Missing Indexes

Select **Tools -> Administration -> Monitoring -> Performance -> Database -> Tables/Indexes (DB02)**.

Press the **Missing indexes** button in the Tables and indexes section of the screen. Missing indexes should be reported to SAP.

6.1.3.3 Checking for Reorganization of Data

Check whether any tables in the database need to be reorganized. See 4.4, "Data Maintenance" on page 110 for information on monitoring and reorganization.

6.1.4 Database Alerts

Select **Tools -> Administration -> Monitoring -> Performance -> Alerts -> Global -> Database system (AL02)**.

Investigate any red or yellow alerts.

6.1.5 R/3 Buffer Pools

R/3 buffers data and objects (such as screens, programs, and CUAs) so that they can be shared among the processes for better performance. If the buffers fill, then performance may suffer because of the overhead required to flush and reload the buffers and because of the additional time required to read from disk rather than read from memory.

To monitor the buffers, select **Tools -> Administration -> Monitoring -> Performance -> Setup/Buffers -> Buffers (ST02)**.

The "Hitratio" for all of the buffers except "single record" should be greater than 90 percent. "Single record" hitratio will be lower. If R/3 has just been started, the hitratio may be lower than 90 percent. It will take some time for the buffers to be filled after a system start up.

Check the number of swaps for each buffer. The swaps are counted from system start time. If the system has been running for a long time, you may see higher number of swaps. If the system has not been running for a long time and you see swaps, you will probably need to increase the size of the buffer.

6.2 Troubleshooting

If you are having trouble starting R/3, try starting R/3 in phases. Try issuing "startsap db" and see if the database will start up ok. If that works, then try issuing "startsap R3". This may help you narrow down whether the problem is with the database or with R/3.

Check the OSS system to see if someone else has already reported the same problem.

Check all the logs described in the 6.1.2, "System Logs" on page 238 of this chapter.

Check the areas listed in the 6.1.3, "Database Monitoring" on page 241 of this chapter for any unusual conditions.

6.2.1 Where to Look

This section describes some of the locations where error messages are recorded.

6.2.1.1 System Log

If R/3 is running, check the system log for any error messages. To check the log, login to R/3 select the following menu path:

Tools -> Administration -> Monitoring -> System Log (SM21)

6.2.1.2 R/3 and Database Start and Stop Logs

Log in to AIX as the <SAPSID>adm user ID. Look in the home directory at the startsap_<nodename>.log, startdb.log, stopsap_<nodename>.log, and stopdb.log files for any error messages.

6.2.1.3 Check the AIX Processes

Log in to AIX as db2<SAPSID> or <SAPSID>adm, and check the processes that are running. Look for DB2 processes and R/3 processes. Check DB2 processes with the following command:

```
ps -ef | grep db2
```

Check R/3 processes with the command:

```
ps -ef | grep sap
```

6.2.1.4 Check Space in File Systems

Check whether any of the filesystems are full or missing. Log in to AIX and enter df. Make sure all the expected filesystems are mounted and have space. If there is not enough space, use SMIT from the root user ID to add additional space, or ask the AIX system administrator to add more space to the filesystem.

6.2.1.5 Check Space in the Tablespaces

See 4.2, “Managing Tablespaces” on page 86 for more information on monitoring and altering tablespaces.

6.2.1.6 Look at the SAP R/3 Developer Trace Files

R/3 trace files are located in the /usr/sap/<SAPSID>/DVEBMGS<nn>/work directory, where <nn> is the instance number. The <SAPSID>adm user ID on AIX is set up to change to directory /usr/sap/<SAPSID>/DVEBMGS<nn> with the cd command. From this directory, change to the work directory.

These trace files can also be viewed from R/3. See 6.1.3, “Database Monitoring” on page 241 in this chapter for more information on how to view these files through R/3.

6.2.1.7 Check the DB2 Log Files

The DB2 log files are located in /db2/<SAPSID>/sqllib/db2dump. Check log file DB2DIAG.LOG for error messages.

6.2.1.8 DB2 Error Logs

There are additional log files located in /db2/<SAPSID>/errors. Check these error logs for any unusual conditions.

6.2.1.9 Check the Hardware Error Log

Log in to AIX as root and enter the following command at the AIX command prompt:

```
errpt -a
```

6.2.1.10 Check the Database Configuration

Log in to AIX as the database administrator, db2<SAPSID> and enter the following command:

```
db2 get db cfg for <SAPSID>
```

Check the output of the command for any unexpected conditions. Check values for Backup pending, Database is consistent, and Rollforward pending.

6.2.2 Database Will Not Start

6.2.2.1 Offline Backup Pending

If you have configured the database for archival logging, you must do a full offline backup before you can start R/3. If you attempt to start R/3 before completing a full offline backup, you will get an error message at start up.

Run a full, offline backup. When the backup is complete, try starting R/3. If you cannot do a full offline backup, you can change the database back to no archive mode until a backup can be completed.

6.2.2.2 Log Directory is Full

Log in to AIX and enter `df` on the AIX command line. Look at the filesystem `/db2/<SAPSID>/log_dir` to see if the log directory is full. If it is full, check the filesystem `/db2/<SAPSID>/log_archive`. If this directory is full, clear space in the directory by saving the archived log files. Move the log files from the `log_dir` directory to the `log_archive` directory. If it is not full, increase the space in `/db2/<SAPSID>/log_dir` and perform a crash recovery by restart.

6.2.2.3 Database Backup is Running

1. Log in to AIX as the database administrator, `db2<SAPSID>` and enter the following command:

```
db2 list tablespaces
```

If a backup is running, you get the following message:

```
SQL1035N The database is currently in use. SQLSTATE=57019
```

1. Enter the following command

```
ps -ef | grep brdb6bu
```

If you find processes running in the process stack, then a backup is running.

2. Look in the `db2diag.log` file located in `/db2/<SAPSID>/sqllib/db2dump`. If the last entries in the file are similar to the following line, a backup is running.

```
Backing up tablespace PSAPDDICI
```

6.2.2.4 Database Restore is Running

Log in to AIX as the database administrator, `db2<SAPSID>`, and enter the following command:

```
db2 list tablespaces
```

If there is a recovery pending, you will see a message similar to the following:

```
SQL1119N A connection to or activation of database "AUS" cannot be made  
because a previous restore is incomplete.  
SQLSTATE=57019
```

6.2.2.5 Database in Roll Forward Pending State

Log in to AIX as the database administrator and check the database configuration with the following command:

```
db2 get db cfg for <SAPSID>
```

If the database is in Roll Forward Pending state, "Rollforward pending" will be set to "DATABASE". If a tablespace is in the Roll Forward Pending state, "Rollforward pending" will be set to "TABLESPACE." If the database was not in Roll Forward Pending state, Roll Forward Pending would be NO.

Another way to check whether the database is in Roll Forward Pending state is to use the command line processor to list the tablespaces. You can list the tablespaces with the following command:

```
db2 list tablespaces
```

If the database is in Roll Forward Pending state you get a message similar to the following:

```
SQL1117N  A connection to or activation of database "AUS" cannot be made
because of ROLL FORWARD PENDING.
SQLSTATE=57019
```

6.2.3 Database Will Not Stop

If the database will not shut down, check whether there are applications connected to the database. You can check with the following command:

```
db2 list applications
```

If there are applications connected to the database, stop those applications and try shutting down the database again.

Check whether any db2bp processes are running. Sometimes these back-end processes are not cleaned up properly. To release the connection to the database, use the TERMINATE command.

6.2.4 Archive Log Directory Fills Up

As long as there is still space in the log directory, the database will continue to run. The archived log files will not be copied to the archive log directory. This filesystem should be monitored to ensure that it does not fill up. If it does fill up, you will need make additional space in the archive directory by either adding more space to the filesystem or by backing up the archive files and deleting them from this directory. Make sure you save the archive log files as they are critical for recovery in the event of a database failure. Once the space is available, you will need to move the inactive log files from the log directory to the archive log directory.

6.2.5 Tablespace is Full

Check the DB2 log file /db2/<SID./sqlib/db2dump/DB2DIAG.LOG. Look for a message similar to the following message:

```
Tablespace 25(PSAPP00LD) is full
```

See 4.3.2, "Redefining the Containers in a Tablespace (Redirected Restore)" on page 109 for information on adding additional space to a tablespace.

6.2.6 Roll Forward Cannot Find Archive Log File

When performing a roll forward of the database, you may experience the following error that an archived log file is not available.

```
Output:
db2 rollforward database AUS to end of logs and stop
SQL1273N The archive log file 'S0000019.LOG' was not found.

Rollforward Status for Database: AUS

Next log file to be read      = S0000019.LOG
Log files processed           = -
Last committed transaction    = 1996-08-27-17.02.54.000000
```

Figure 152. Cannot Locate Archived Log File

If you get a message similar to the message displayed in Figure 152, you need to make the log file available. First check that the log file is available in the log_archive filesystem. If not, recover it from the backup. If it is in the filesystem, log in to AIX as the database administration user ID (db2<SAPSID>). Enter SMIT at the AIX command line. Select the following SMIT options:

- **Applications**
- **DB2admin for R/3: IBM DB2 for AIX Administration Utilities**
- **Manage Log Files**
- **Make Available Inactive Log Files**
- Select the **database name** by clicking on the correct entry.
- Enter the location of the log files. In most cases the location will be the log_archive filesystem. You can also press the **List** button to get a list.
- Press the **OK** button.
- Press the **List** button for Log Files to be made available. Select the log files from the list.
- Press the **OK** button.
- Try the roll forward again.

This option is useful when tablespace roll forward is necessary. DB2 for AIX does not retrieve the required log files through its User Exit feature when rolling forward tablespaces. For roll forward processing of databases, the log files are made available through the User Exit program.

6.2.7 ADSM Recovery Log Fills Up

While doing an ADSM backup, you get a message similar to the following:

```
ANR0526W Transaction failed for session 3 for node
F01N07E.ITSC.AUSTIN.IBM.COM (DB2/6000) - sufficient recovery log space is
not available.
```

```
ANR2010W Server is out of LOG space in adding entries to the Activity Log.
Console messages will not be logged until LOG space is available.
```

```
ANR0403I Session 3 ended for node F01N07E.ITSC.AUSTIN.IBM.COM (DB2/6000)
```

You need to add more space to the ADSM recovery log. Log in to AIX as root. From the AIX command line, enter the following:

```
cd /usr/lpp/adsmserve/bin
dsmfmt -log <filename> <space>
```

where <filename> is the filename you wish to add, and <space> is the amount of disk space in megabytes.

From the ADSM command line interface, enter the following:

```
define logvolume <filename>
```

where <filename> is the filename defined in the previous step

```
extend log <space>
```

where <space> is the amount of space you want to add in megabytes

6.2.8 Cannot Archive Inactive Redo Logs

While attempting to archive inactive redo logs, you get a message similar to the following:

```
BR278I Command output of '/db2/AUS/sapscripts/backup/backint -u AUS -f backup
-i/db2/AUS/sapbackup/.acsuudym.lst -t file':
```

```
sh: /db2/AUS/sapscripts/backup/backint: not found.
```

```
BR279E Return code from '/db2/AUS/sapscripts/backup/backint -u AUS -f backup
- i /db2/AUS/sapbackup/.acsuudym.lst -t file': 127
```

Copy or link the file /usr/sap/<SAPSID>/exe/run/brdb6backint to /db2/<SAPSID>/sapscripts/backup/backint.

6.2.9 Redirected Restore with ADSM Does Not Complete

Check the /db2/<SAPSID>/sqllib/db2dump/db2diag.log file for an error message similar to the following message:

```
Error in sqluend.C at line: 1072. ADSM rc: 2014
```

Login to AIX as the database administrator and enter the following command:

```
db2 list tablespaces
```

Check the State for any tablespaces you are restoring with a Redirected Restore. The state should be 0x0000, 0x0040, or 0x0080. See 4.2.7, "Tablespace States" on page 100 for an explanation of the states. If the value of State is 0x0100, the database is in Restore Pending state. The ADSM Server may have had a

communication timeout. Increase the value for COMMITimeout in /usr/lpp/adsmserve/bin/dsmserve.opt. The default value is 60 seconds. We changed the value to 6000 seconds. If you do not have access to the ADSM Server or cannot change this value, use another backup media to save your tablespace(s) for the Redirected Restore.

Appendix A. Special Notices

This publication is intended to help anyone who works with or supports R/3 on a DB2 for AIX platform. The R/3 database administrator will find it particularly interesting, but others will find that it provides a lot of useful information. The information in this publication is not intended as the specification of any programming interfaces that are provided by the DB2 Server product. See the PUBLICATIONS section of the IBM Programming Announcement for DB2 Server 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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

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.

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

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
BookManager	DATABASE 2
DB2	DB2/6000
DRDA	IBM
OS/2	OS/400
RISC System/6000	RS/6000

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, 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.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

C + +	American Telephone and Telegraph Company, Incorporated
Dialog	American Telephone and Telegraph Company, Incorporated
Gateway	Gateway Systems Corporation
IPX	Novell, Incorporated
Macintosh	Apple Computer, Incorporated
Microsoft	Microsoft Corporation
NT	Northern Telecom Limited or Microsoft Corporation
Oracle	Oracle Corporation
OSS	Optical Storage Solutions, Incorporated
SMS	Standard Microsystems Corporation
Sun	Sun Microsystems, Incorporated
Windows NT	Microsoft Corporation
Zip	Iomega Corporation

Other trademarks are trademarks of their respective companies.

Appendix B. Related Publications

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

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 253.

- *DB2 Version 2 Planning Guide for Database Administrators*, SG24-2523

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.

B.2 Other Publications

These publications are also relevant as further information sources:

- *DATABASE 2 Command Reference for Common Server V2*, S20H-4645
- *DATABASE 2 Administration Guide for Common Server V2*, S20H-4580
- *DATABASE 2 SQL Reference for Common Server V2*, S20H-4665
- *ADSM/6000 Installing the Server and Administrative Client*, SH35-0136

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL <http://www.redbooks.ibm.com>.

How IBM Employees Can Get ITSO Redbooks

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.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

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

```
TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
```

To get lists of redbooks:

```
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>

- **IBM Direct Publications Catalog on the World Wide Web**

<http://www.elink.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**

- **Online** — 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.ibm.link.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 Customers Can Get ITSO Redbooks

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

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	Internet
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark
--	--	--

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1) 415 855 43 29 (Outside USA)** — 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

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibm.com/pbl/pbl

- **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.ibm.com with the keyword `subscribe` in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

• Invoice to customer number

• Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

APA	All Points Addressable	LF	Long Field Data
BLOB	Binary Large Objects	LP	Logical Partition
CAE	Client Application Enabler	LPP	Licensed Program Product
CCMS	Computing Center Management System	LVM	Logical Volume Manager
CLI	Call Level Interface	PP	Physical Partition
CUT	Coordinated Universal Time	PROFS	Professional Office System
DARI	Database Application Remote Interface	PV	Physical Volume
DBA	Database Administrator	RM	Resource Manager
DBADM	Database Administration Authority	SAPSID	SAP System ID
DDCS	Distributed Database Connection Services	SDK	DB2 Software Developer's Toolkit
DMS	Database Managed Storage	SMIT	System Management Interface Tool
DUOW	Distributed Unit of Work	SMS	System Managed Storage
IBM	International Business Machines Corporation	SYSADM	System Administration Authority
IPC	Inter-Process Communication	SYSCTRL	System Control Authority
ITSO	International Technical Support Organization	SYSMAINT	System Maintenance Authority
JFS	Jounaled File System	VG	Volume Group
		VGDA	Volume Group Descriptor Area
		saposccl	SAP Operating System Collector

Index

A

ABAP/4 2
abbreviations 257
acronyms 257
Active Log Files 157
ADSM
 Administrative Client 166
 Application Client 167
 Assigning a Logical Volume to a Storage Pool 179, 188
 Backing up the Archive Log Files 230
 Backup/Archive Client 166
 Configuring for DB2 166
 Customizing the ADSM Backup/Archive Client 175
 Customizing the Server 168
 Defining Storage Pool Destination 180, 190
 Defining the Logical Volume 177
 Doing an Offline Backup 194
 Executing a non-Database Backup from the Client 184
 Installation 167
 Main Components 166
 Naming the Server 173
 Recovery Log Fills Up 247
 Recovery Using an Online Backup 206
 Registering a Backup/Archive Client 173
 Registering an Administrator 170
 Removing Files from log_archive File System 232
 Server 167
 Setting up ADSM to Backup to Disk 176
 Setting up AIX Logical Volumes 177
 Setting Up to Save Archived Logs to Disk 186
 Starting the Server 168
 Using the Administrative Client 169
 Verifying the Backup 185, 195
ADSM Command
 REGISTER NODE 173
AIX
 Processes 243
AIX Command
 kill -9 -1 63
 ps 63
Alert Monitor 97
APPC 7
Applications Server 2
Archive Log Directory Fills Up 245
AUTORESTART 198

B

bibliography 251
BUFFPAGE 52

C

Cannot Archive Inactive Redo Logs 247
Central Instance 4
Checking for Reorganization of Data 241
Circular Logging 156
Container
 Displaying Information with Database Director 96
 Displaying Information with R/3 Alert Monitor 97
 Displaying Information with R/3 Performance Monitor 95
Crash Recovery 165, 198

D

Database Alerts 241
Database Director 10, 77, 84, 96, 192
Database Monitoring 241
Database Server 3
Database Will Not Start 244
Database Will Not Stop 245
DB2
 Backup and Restore Utilities 191
 Common Server 6
 Components 7
 Event Analyzer 153
 Family of database servers 7
 Firewall 61
 Logging 156
 Performing a Backup 192
 Performing an Offline Backup 193
 Performing an Online Backup 195
 Prefetching 55
 Processes 64, 66
 SYSADM 69
 SYSCTRL 69
 Tools and Utilities 73
DB2 Administratorxd5 s Toolkit 10
DB2 Client Application Enabler (CAE) 8
DB2 Command
 BACKUP DATABASE 192
 db2 75
 db2start 56
 FORCE 99
 LIST APPLICATIONS 99
 LIST TABLESPACE CONTAINERS 87, 92, 93
 LIST TABLESPACES 87, 92
 quit 75
 RESTART DATABASE 198
 ROLLFORWARD DATABASE 165, 210, 212
 terminate 75
DB2 Command Line Processor (CLP) 9, 74
DB2 for MVS 7
DB2 for OS/2 7

- DB2 for OS/400 7
- DB2 for UNIX (various versions) 7
- DB2 for VM 7
- DB2 for VSE 7
- DB2 for Windows NT 7
- DB2 Server 7
- DB2 Servers 7
- DB2 Software Developer's Kit (SDK) 7
- DB2 Statement
 - ALTER TABLESPACE 87
 - CREATE TABLESPACE 87
 - DROP 87
 - ROLLBACK 165
- DB2admin 89
- db2agent 64, 68
- DB2ALERT.LOG 238
- db2cart 68
- db2dbabg 68
- db2dd 68
- DB2DIAG.LOG 238
- db2dlock 64
- db2event 68
- db2gds 63
- db2ipccm 68
- db2ippcm 63
- db2licd 63
- db2loggr 64
- db2pclnr 64, 68
- db2pfchr 64, 68
- db2resyn 63
- db2sysc 63
- db2tcpcm 68
- db2tcpim 67
- db2wdog 63
- DBHEAP 53
- Determining Backed Up Log Files 234
- DFT_PREFETCH_SZ 55
- Dispatcher 2
- Distributed Database Connection Services (DDCS) 7

E

- Enabling Archive Logging 162
- Enqueue Server 4
- Event Monitoring 149
- EXTENTSIZ 55

I

- I/O Cleaner 137
- I/O Server 137
- IPX/SPX 7

L

- Locks and Deadlocks 138
- Log File Information 163
- LOGBUFSZ 53, 159

- LOGFILSIZ 159
- LOGPRIMARY 156, 159
- LOGRETAIN 157, 159, 162, 163, 166
- LOGSECOND 156, 159

M

- Managing Archive Log Files 230
- MAX_APPLS 52
- MAXLOCKS 53
- Methods of Recovery 197
- MINCOMMIT 160
- Missing Indexes 241

N

- n-n process model 61
- NetBIOS 7
- NEWLOGPATH 159
- NUM_IOCLEANERS 160

O

- OFFLINE 192
- Offline Archived Log Files 157
- ONLINE 192
- Online Archived Log Files 157
- OVERFLOWLOGPATH 160

P

- Performance Monitor 10, 78
- Point-in-Time Recovery 213
- PREFETCHSIZE 55
- Presentation Server 2
- process is for log retain. 68

R

- R/3
 - Advantages 5
 - Alert Monitor 81, 83
 - architecture 1, 2
 - Buffer Pools 242
 - Database Administration 73
 - Database Performance Monitor 79
 - DB2admin 83, 192, 199, 211
 - Naming of Log Files 164
 - Performance Monitor 95
 - Profiles 161
 - Stopping 60
 - System Log 238, 242
 - Using Performance Monitor to Display Monitor Information 136
- R/3 Command
 - stopsap r3 60
- R/3 Performance Monitor 89
- R/3 Processes
 - co 68

R/3 Processes *(continued)*

- ms 68
- saposocol 68
- se 68
- Recovery Using an Offline Backup 199, 201
- Recovery Using an Online Backup 204
- Redirected Restore 247
- Removing Log Files 230
- Restore or Version Recovery 198
- Restoring Offline Archive Log Files 233
- Roll Forward 212
- Roll Forward Database 210
- Roll Forward Pending 209
- Roll Forward Recovery 165, 199

S

- SAPGUI 7
- sapr3 69
- SAPSID 149
- SEQDETECT 56
- Snapshot Monitor 78
- SOFTMAX 159
- SQLOGDIR 164
- SYSCAT.DBAUTH 70
- SYSCAT.INDEXAUTH 70
- SYSCAT.PACKAGEAUTH 70
- SYSCAT.TBAUTH 70

T

- Tablespaces
 - Checking for Free Space 241
 - Displaying Information 87
 - Displaying Information with DB2admin 89
 - Displaying Information with R/3 Performance Monitor 89
 - Full 245
 - Managing 86
- TCP/IP 7

U

- USEREXIT 158, 160, 162, 163, 166

V

- Visual Explain 10

W

- Work processes 2



Printed in U.S.A.

SG24-4871-00

