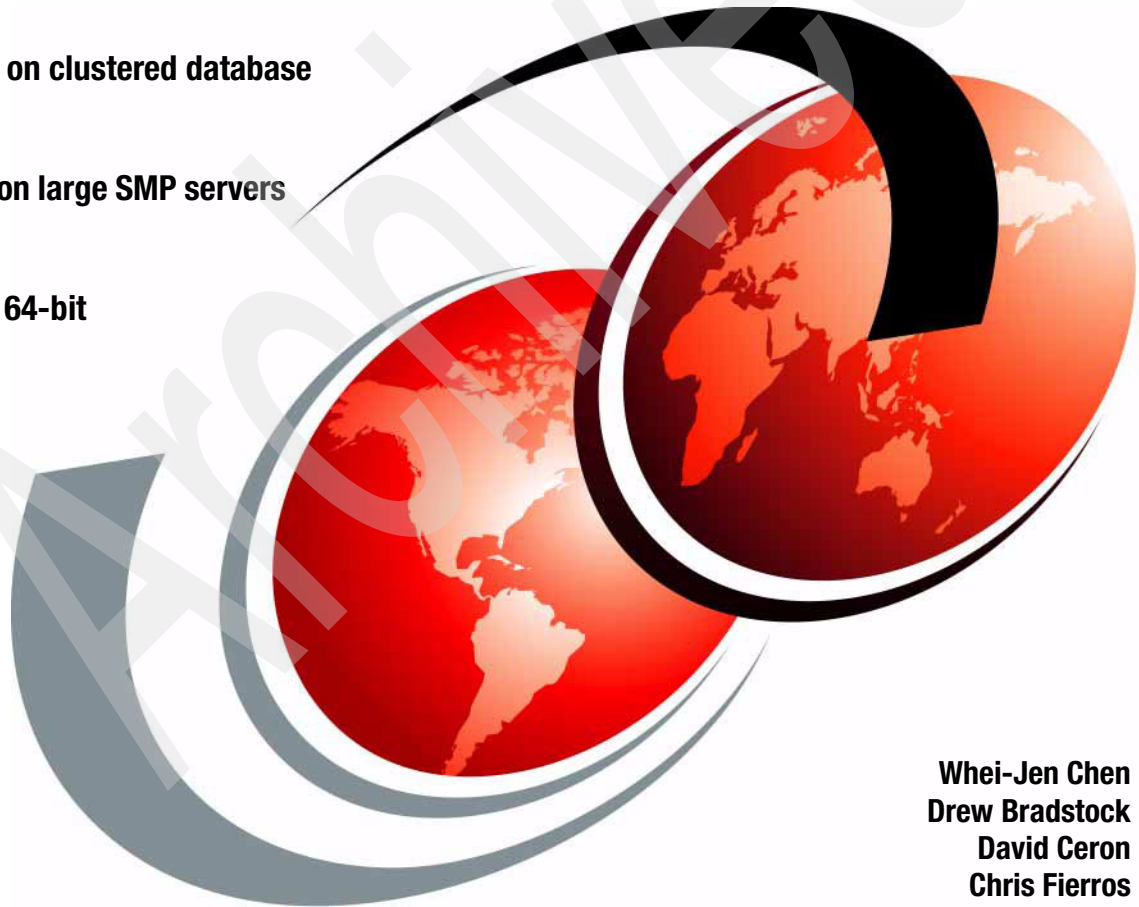


Scaling DB2 UDB on Windows Server 2003

Scaling out on clustered database
servers

Scaling up on large SMP servers

Leveraging 64-bit
technology



Whei-Jen Chen
Drew Bradstock
David Ceron
Chris Fierros



International Technical Support Organization

Scaling DB2 UDB on Windows Server 2003

August 2003

Archived

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

First Edition (August 2003)

This edition applies to DB2 Universal Database Version 8 and Microsoft Windows Server 2003.

© Copyright International Business Machines Corporation 2003. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	xi
Notices	xiii
Trademarks	xiv
Preface	xv
The team that wrote this redbook	xvi
Acknowledgement	xvii
Become a published author	xvii
Comments welcome	xviii
Chapter 1. Introduction	1
1.1 Scaling overview	2
1.1.1 Measuring scalability	2
1.1.2 Achieving scalability	3
1.2 Scaling with DB2 Universal Database	8
1.2.1 DB2 UDB for Windows Editions	8
1.2.2 Try and buy product availability	11
1.3 Scaling with Windows Server 2003	11
1.3.1 Windows Server 2003 Editions	13
1.4 Scaling with IBM xSeries Servers	17
1.4.1 IBM xSeries servers family	17
1.5 Design considerations for scaling	20
1.5.1 Planning considerations	21
1.5.2 Hardware considerations	22
1.5.3 Design considerations	24
1.6 Documentation	30
1.6.1 PDF documentation	30
1.6.2 HTML documentation	30
Chapter 2. Utilizing the capabilities of 64-bit	33
2.1 Migration	34
2.1.1 Migrating DB2 between versions of Windows	34
2.1.2 Migrating between versions of DB2 UDB	41
2.1.3 Migrating from 32-bit to 64-bit	43
2.1.4 Migrating everything at once	43
2.1.5 Migrating back to 32-bit	45

2.1.6 Migrating back to DB2 UDB V7 or V6	46
2.2 Installation considerations	46
2.2.1 Installation requirements	46
2.2.2 64-bit and 32-bit installation images	48
2.2.3 Fixpaks	49
2.3 32-bit installation	50
2.3.1 Existing references	50
2.4 64-bit installation	51
2.4.1 Installation steps	51
2.4.2 Installing DB2 UDB using a response file	61
2.4.3 Creating an Instance	67
2.4.4 Removing an instance	74
2.5 DB2 configuration	75
2.5.1 Configuration levels	75
2.5.2 Using the auto-configure tool	91
2.5.3 Configuring memory for 64-bit systems	97
2.5.4 Configuring the instance	98
2.5.5 Configuring the database	105
2.6 Additional considerations	112
2.6.1 Compatibility issues between V7 and V8	112
2.6.2 Paging	114
Chapter 3. Scaling up DB2	117
3.1 Scale up overview	118
3.2 Adding additional processors	118
3.2.1 Can your system scale	118
3.2.2 Moving from a 4-way to an n-way	119
3.2.3 Windows considerations	120
3.2.4 Taking advantage of greater parallelism	120
3.2.5 Observing the performance change	127
3.3 Adding additional memory	129
3.3.1 Physically adding more memory	130
3.3.2 Configuring Windows	130
3.3.3 xSeries information	133
3.3.4 Adjusting your DB2 configuration	133
3.3.5 How you see how much is allocated	133
3.3.6 Case studies	141
3.4 Adding additional storage	146
3.4.1 Storage and Windows Server 2003	147
3.4.2 DB2 storage considerations	149
3.4.3 DB2 tablespace design	150
3.4.4 Database managed space	155
3.4.5 DB2 DMS device considerations	157

3.4.6 Using raw I/O in DB2.	158
3.4.7 Tablespace disk I/O	163
3.4.8 Optimizing tablespace performance on RAID devices	164
3.4.9 DB2 Storage Management view	167
3.4.10 DB2 Health Monitor for storage management.	168
Chapter 4. Scaling out DB2	177
4.1 Overview	178
4.1.1 DB2 UDB and Windows clustering technologies	178
4.1.2 Implementing very large databases on Windows platforms	179
4.1.3 Cluster benefits and limitations	181
4.2 Preparing the environment for scaling out DB2 UDB	182
4.2.1 Installation overview	183
4.2.2 Windows Server 2003 Domain Controller setup	184
4.3 Installing a DB2 partitioned environment.	194
4.3.1 Before configuring DB2 on a physical cluster environment.	194
4.3.2 Installing the DB2 Instance Owning Node.	196
4.3.3 Installing a physical database partition	203
4.3.4 Verifying DB2 partitioning configuration	206
4.3.5 Creating additional partitions and redistributing data	210
4.3.6 Creating a partitioned environment with DB2 commands	220
4.3.7 Executing commands on all partitions.	228
4.4 Creating partitioned databases	229
4.4.1 Basic database structure.	229
4.4.2 Database partition groups	230
4.4.3 Partitioning maps and partitioning keys	232
4.4.4 Partition groups and tablespaces	234
4.4.5 Create database example.	234
Related publications	239
IBM Redbooks	239
Other publications	239
Online resources	240
How to get IBM Redbooks	241
Help from IBM	241
Index	243

Figures

1-1	System scalability	2
1-2	Adding resources to scale up.	4
1-3	Clustering for high availability	5
1-4	Clustering for scaling out	5
1-5	Scale out architecture	6
1-6	Partitions in a share-nothing architecture	7
1-7	4 GB Tuning.	14
1-8	Microsoft Address Windowing Extensions	16
1-9	Multiple logical partitions	28
1-10	Multiple physical partitions	28
1-11	Scaling out with multiple logical and physical partitions.	29
1-12	Splitting logical partitions	29
1-13	DB2 Information Center program location	31
1-14	Fixpak documentation change indicators.	32
2-1	Backup and restore combinations with DB2 UDB V8.1 for Windows . .	35
2-2	Executing the db2relocatedb command.	40
2-3	DB2 migrate command syntax.	43
2-4	32-bit DB2 UDB image installation on 64-bit Windows error message .	49
2-5	My Computer's properties information for Windows Server 2003	52
2-6	Windows Server 2003 64-bit indicator	53
2-7	DB2 installation options	55
2-8	Response file selection during DB2 installation.	56
2-9	DB2 installation component selection	57
2-10	DB2 Administration Server (DAS) installation options	58
2-11	DB2 contact list information	59
2-12	DB2 UDB V8.1 tools database options	60
2-13	Response file installation option	63
2-14	db2rspgn utility command syntax.	63
2-15	Setup command response file options.	65
2-16	Active Directory program location	67
2-17	Adding a new user	68
2-18	User specifications.	69
2-19	Adding a user to the Administrator group	69
2-20	Domain Controller program location	70
2-21	Selecting user rights in the Domain Controller.	70
2-22	Adding a user's rights	71
2-23	Services option with the Domain Controller.	72
2-24	db2icrt command syntax	72

2-25	Windows Server 2003 services file	73
2-26	DB2 contents in a services file	73
2-27	DB2 registry settings output	76
2-28	db2set error message	77
2-29	Database manager parameters from the command line	81
2-30	dbm cfg show detail steps	82
2-31	Windows command line DB2 error	82
2-32	Location in Windows of the DB2 Control Center utility	83
2-33	Control Center instance configuration option	84
2-34	Setting instance configuration parameters at the command line	85
2-35	Changing an instance parameter using the Control Center	86
2-36	Parameter change time frame	87
2-37	Selecting a database to configure	88
2-38	Configuring a database option	89
2-39	Updating the database configuration using the command line	90
2-40	Updating a database configuration parameter	90
2-41	Configuration Advisor: Memory specifications	92
2-42	Configuration Advisor: Transaction length and volume	93
2-43	Configuration Advisor: Administration policy	94
2-44	Configuration Advisor: Isolation level	95
2-45	Configuration Advisor: Scheduling the configuration task	96
2-46	Insufficient sort heap threshold limit example	100
2-47	Sort heap and sort heap threshold problem	101
2-48	Catalog cache hit ratio	109
2-49	Package cache hit ratio	110
2-50	V7 and V8 64-bit compatibility	114
2-51	Windows perfmon: Paging information fields	115
3-1	Approaches for scaling the number of CPUs for a system	120
3-2	Intra-query parallelism	123
3-3	Intra-partition parallelism	124
3-4	Inter-partition parallelism	125
3-5	Viewing system use with the Windows Task Manager	128
3-6	Selecting multiple CPUs for the Windows perfmon tool	129
3-7	Windows Task Manager output	134
3-8	Perfmon add monitor option	135
3-9	Perfmon DB2 memory options	136
3-10	Perfmon multiple DB2 database selection	137
3-11	Perfmon output: Text	138
3-12	DB2 Memory Visualizer	139
3-13	Memory Visualizer graphic output	140
3-14	Bufferpool hit ratio	144
3-15	Containers, tablespace, and tables	151
3-16	Extents and containers	152

3-17	Alter tablespace with manage stripe sets option	157
3-18	Windows disk management	160
3-19	Storage Management Setup Launchpad	170
3-20	Storage Management view	171
4-1	Shared-nothing architecture	180
4-2	DB2 UDB partitioned environment with two physical nodes	184
4-3	Manage Your Server Wizard	186
4-4	Defining a server role in Windows Server 2003	187
4-5	Active Directory Installation Wizard	188
4-6	Domain Controller type	189
4-7	Active Directory summary window	190
4-8	Completing the Active Directory Wizard	191
4-9	Domain Controller successfully set up window	192
4-10	System properties: Joining an existing domain	193
4-11	Computer Name Changes window	194
4-12	DB2 Setup Launchpad window	197
4-13	DB2 setup for partitioned database environment (instance-owning) . .	198
4-14	DB2 administration server information	200
4-15	Select the instance-owning computer	201
4-16	Add new database partition server	202
4-17	Review settings of the DB2 instance-owning partition	203
4-18	DB2 setup, partitioned database environment (additional partition) . .	204
4-19	Adding a new database partition	205
4-20	db2nodes.cfg file with the instance owning entry	206
4-21	Default ports reserved by DB2 instance-owning installation	207
4-22	Sample hosts file	208
4-23	db2start output with two partitions	209
4-24	Database partitions used by the employee table	210
4-25	The Add Partitions Launchpad	211
4-26	Add partitions to your instance	212
4-27	Copy Partition Settings	213
4-28	Set containers	214
4-29	System analysis results	216
4-30	Staged Redistribution Plan window	217
4-31	Redistribute Data Wizard Summary window	218
4-32	db2icrt command	222
4-33	db2ncrt command	224
4-34	db2stop and db2start after adding a physical partition	224
4-35	Database partitions groups	231
4-36	Sample output of volumes.xml	237

Tables

1-1	DB2 UDB sever editions	9
1-2	Windows Server 2003 Editions	13
1-3	Memory tuning features	26
2-1	Disk space required	47
2-2	CPU limits for DB2 UDB editions	54
2-3	Primary return codes from the response file	65
2-4	Additional return codes from the response file	66
2-5	Memory limits for Windows database editions	98
2-6	DB2 instance memory configuration variables	98
2-7	Database memory configuration variables	105
3-1	OLTP instance parameter recommendations	141
3-2	Recommended database changes	142
3-3	DSS instance confirmation settings	144
3-4	Database parameter configuration settings	145
3-5	RAID configurations	148
3-6	DB2 page size specific limits	153
4-1	Server information	220

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

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

AIX®	Distributed Relational Database	Redbooks (logo)  ™
BladeCenter™	Architecture™	Redbooks™
@server™	DRDA®	@server™
DataPropagator™	IBM®	WebSphere®
DB2 Connect™	Informix®	X-Architecture™
DB2 Universal Database™	iSeries™	xSeries®
DB2®	pSeries™	zSeries®

The following terms are trademarks of International Business Machines Corporation and Rational Software Corporation, in the United States, other countries or both.

Rational®

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This IBM® Redbook offers you a broad understanding of how DB2® UDB V8.1 is integrated with Windows Server 2003. The book is organized as follows.

- ▶ Chapter 1, “Introduction” on page 1

In this chapter we define scalability from a database management system perspective. We first introduce the concept of scalability, including the basic goals, measurements, and the two common approaches to scaling (which include vertical (scaling-up) and horizontal (scaling-out) scaling). We follow with an introduction to the DB2 Universal Database™ server editions, the Windows 2003 Server editions, and the IBM xSeries® Servers, pointing out the features of each that allow you to design a highly scalable database system. Finally, we cover design considerations for scaling up and scaling out with DB2 Universal Database for Windows Server 2003 on the IBM xSeries Servers.

- ▶ Chapter 2, “Utilizing the capabilities of 64-bit” on page 33

In this chapter we provide a step-by-step guide on how to take advantage of the new 64-bit capabilities of Windows Server 2003 with DB2 V8.1. The discussion covers in detail which DB2 configuration parameters need to be set for Windows and how to scale your memory effectively for the move to 64-bit. The steps required to migrate from DB2 V7 on Windows 2000 to DB2 V8.1 on Windows Server 2003 are also covered.

- ▶ Chapter 3, “Scaling up DB2” on page 117

This chapter focuses on how to utilize the increased resources of your database system. The scaling up of a system has been divided into separate sections on the use of additional CPUs, memory, disks, and other hardware resources. Each section covers the DB2 parameters, workloads, and utilities that will benefit from the scaling. A case study on how to utilize the new DB2 V8.1 Configuration Advisor is included to help new DB2 DBAs.

- ▶ Chapter 4, “Scaling out DB2” on page 177

In a multi-node environment, DB2 extends the database manager’s capability to parallel respond to today’s business demands: Flexibility, security, and power to store and retrieve data. Scaling out DB2 is the ability to add more servers to DB2 system. This is the way to improve scalability when a single system will not meet the needs of memory, processor, and storage for your business. This chapter describes how to set up DB2 on Windows Server 2003 Datacenter Edition multi-node configuration.

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, Kirkland Center.

Whei-Jen Chen is a Project Leader at the International Technical Support Organization, San Jose Center. She has extensive experience in application development, database design and modeling, and DB2 system administration. Whei-Jen is an IBM Certified Solutions Expert in Database Administration and Application Development, as well as an IBM Certified IT specialist.

Drew Bradstock is a DB2 Consultant who spends his time between the USA and Canada. He has worked extensively with IBM's Data Management partners in helping them to migrate their applications to DB2. He has been working with DB2 for four years and has presented at conferences, written papers, and helped to edit articles on DB2 performance and best practices. His specialties include DB2 performance and EEE. He is a certified DB2 V8 Advanced Administrator and was one of the authors of the book *DB2 SQL Procedural Language for Linux, Unix and Windows*. Drew can be reached at drewkb@ca.ibm.com.

David Ceron is an Information Management Pre-Sales Technical Specialist who works for the Software Group at IBM Mexico. He has spent eight years in IT and has extensive experience in DB2 and architecting solutions on AIX®, Solaris, Linux, and Windows Operating Systems. He has worked at IBM for five years. David is an IBM Certified IT Specialist and an IBM Certified Solutions Expert in DB2 UDB Administration, DB2 UDB Application Development, and Business Intelligence Solutions. David can be reached at dceron@mx1.ibm.com.

Chris Fierros is a DB2 Solutions Expert and principle consultant at Ten Digit Consulting, Inc. specializing in consulting, training, and support for DB2 Universal Database. Chris began his career with DB2 on distributed platforms in 1991 at IBM Corporation, where he worked in technical support, global services, and consulting roles. He has published articles, presented at conferences, and written books about DB2. Chris is an IBM Certified Solutions Expert (ICSE) in DB2 UDB Administration and Development, as well as an IBM Certified Advanced Technical Expert (CATE) in DB2 Clustering (EEE), DB2 Replication (DpropR), and DB2 Connect™ (DRDA®). Chris has spent the last couple of years helping customers move to DB2 UDB on large-scale 16-way and 32-way Intel servers running Windows 2000 Server, Windows Server 2003, and Red Hat Linux Advanced Server. Chris can be reached at chris@tendigit.com.

Acknowledgement

The authors express their deep gratitude for the help they received from **Michael Gao** from the IBM Toronto Laboratory. They would also like to thank the following people for their contributions to this project:

Mike Logan

IBM Data Management and Marketing

Susumu Arai, Richard Brown, Michael Lawson, Michael Lee, Shannon Thompson, James Yang

IBM Center for Microsoft Technologies

Dale Hagen, Andrew Hilden, Gene Kligerman, Mick Legare, Dale McInnis, Hiep Phuong, Michael Roecken

IBM Toronto Laboratory

Donn Bullock

IBM xSeries

Sandy Arthur, Al Bergstein, Eileen Brown, Tim Cerling, Jim Drennan, Ji-Young Kim, Thomas Kuklovszky, William Lyon, Bernd Pfann, Jim Steinmark

Microsoft

Emma Jakobs

International Technical Support Organization, San Jose Center

Julie Czubik

International Technical Support Organization, Poughkeepsie Center

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Introduction

In this chapter we define scalability from a database management system perspective. We first introduce the concept of scalability including the basic goals, measurements, and the two common approaches to scaling which include vertical (scaling up) and horizontal (scaling out) scaling.

We follow with an introduction to the DB2 Universal Database server editions, the Windows 2003 Server editions, and the IBM xSeries Servers pointing out the features of each that allow you to design a highly scalable database system.

Finally, we cover design considerations for scaling up and scaling out with DB2 UDB for Windows Server 2003 on the IBM xSeries Servers.

This chapter contains the following sections:

- ▶ Section 1.1, “Scaling overview” on page 2
- ▶ Section 1.2, “Scaling with DB2 Universal Database” on page 8
- ▶ Section 1.3, “Scaling with Windows Server 2003” on page 11
- ▶ Section 1.4, “Scaling with IBM xSeries Servers” on page 17
- ▶ Section 1.5, “Design considerations for scaling” on page 20

1.1 Scaling overview

We define system scalability as the ability of a system to perform at a level that is proportional to demands on the system. That is to say, a system that scales provides a given performance level for a given amount of system resources and that as we increase system resources we expect the system's performance level to increase proportionally.

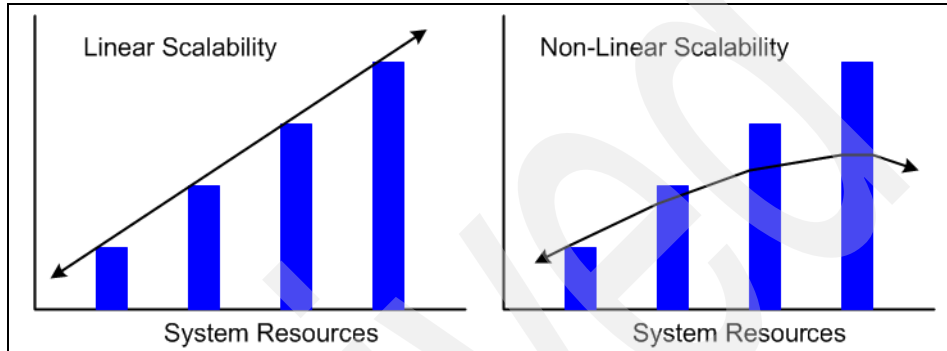


Figure 1-1 System scalability

1.1.1 Measuring scalability

There are several methods that can be used to measure system scalability. Some of the more common ways include:

- ▶ Workload
- ▶ Throughput
- ▶ Response time

Workload

We measure system scalability in terms of workload when we expect a given workload to increase proportionally to the amount of additional system resources. For example, if a system currently supports 100 users, we should expect to support 200 users (workload) with twice the amount of system resources and maintain the same level of service (response time) for all users, thus achieving scalability.

Throughput

We measure system scalability in terms of throughput when we expect a given throughput to increase proportionally to the amount of additional system resources. For example, if a system currently supports 100 transactions per second, we should expect to achieve 200 transactions per second with twice the

amount of system resources and still maintain the same level of service (response time) for the same number of users (workload), thus achieving scalability.

Response time

We measure system scalability in terms of response time when we expect a given response time to decrease proportionally to the amount of additional systems resources. For example, if a system currently provides two-second response time, we should expect a decrease in the response time to one second with twice the amount of systems resources, thus achieving scalability.

1.1.2 Achieving scalability

The goal of system scalability, as we defined it in the previous section, is to increase performance at a rate that is proportional to increases in system resources for a given workload.

There are two methods to achieving system scalability. They are:

- ▶ Scale up
- ▶ Scale out

Scaling up

Scaling up is achieved by adding resources, such as memory, processors, and storage, to an existing system that runs on a single server. It is sometimes referred to as vertical scaling.

The benefit to scaling up is that it is relatively easy, as in general it only requires hardware or software that is designed to take advantage of additional memory, processor, and storage resources. For example, your database server may start out on a 2-way SMP system with four GB of memory and six hard drives. As the database grows in size or the number of users increase, you can easily scale up by adding additional processors, memory, and disk resources to maintain the same level of performance. You may eventually need to replace the server with one that is capable of supporting more resources, but today on Intel-based servers you can scale up to systems that support 32 processors and 64 GB of memory on 32-bit versions of Windows Server 2003 with DB2 UDB; and 64 processors and 128 GB of memory on 64-bit versions of Windows Server 2003 with DB2 UDB. Adding storage capacity is virtually unlimited using enterprise Storage Area Networks. See Figure 1-2 on page 4.

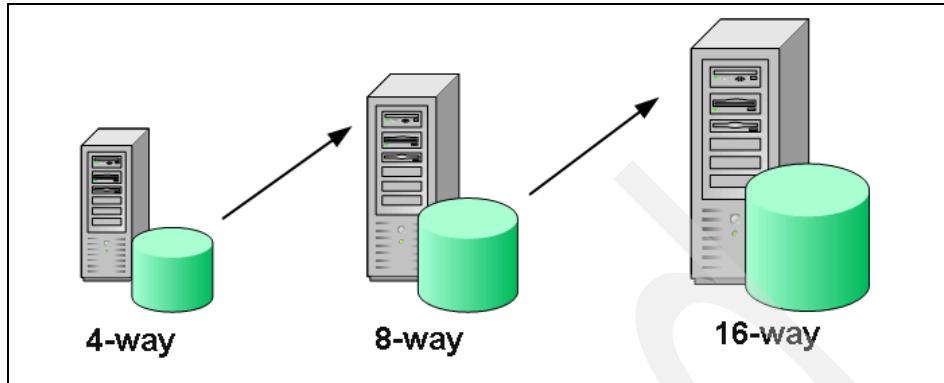


Figure 1-2 Adding resources to scale up

The drawback to scaling up is that it does not provide linear scalability. That is to say that as you add processors, memory, and storage, there is some overhead in resource management that limits the scalability of a single large systems. Thus, scaling up introduces a diminished return, as the overall system cost increases proportionally to the cost of resources, but does not provide an equally proportional return. For example, if the cost of upgrading your system to a 4-way with 8 GB of RAM and 12 hard drives doubles, you will not achieve twice the performance.

Scaling out

Scaling out is achieved by adding resources such as memory, processors, and storage to an existing system that runs on one or more physical servers. It is sometimes called horizontal scaling, as well as in general being referred to as clustering. However, clustering can sometimes be ambiguous in that there are distinct types of clusters, which include high availability, load balancing, and partitioning.

Clustering for high availability, also called failover clustering, is achieved by implementing a system that includes redundant servers that can take over processing in the event of a failure. On Windows Server 2003, failover clustering is implemented with the Microsoft Clustering Service (MSCS), which is available in both the Windows Server 2003 Enterprise and Datacenter Editions. All server versions of DB2 UDB for Windows support failover clustering with MSCS. Failover clustering for high availability is often implemented in conjunction with partitioning clustering, but is only required to achieve high availability. See Figure 1-3 on page 5.

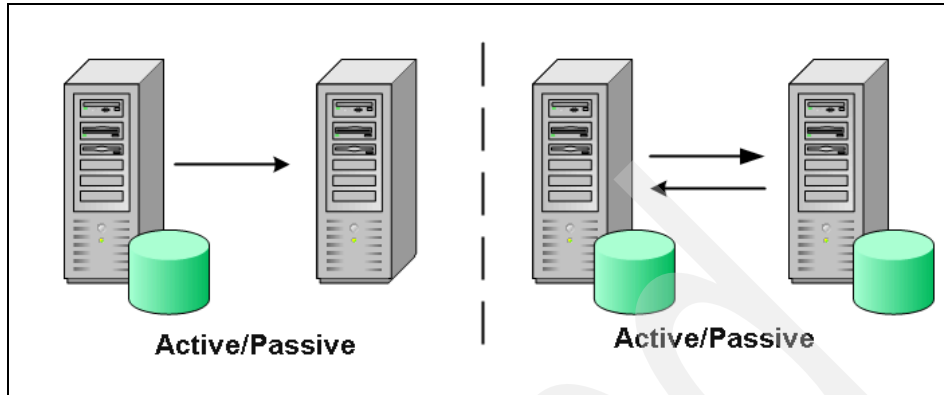


Figure 1-3 Clustering for high availability

Load balancing and/or partitioning is the goal of scaling out. That is to say, we scale out by adding one or more servers to an existing system to balance the system load as we add additional demands on the system. For example, your database server may start out on a 2-way system with 4 GB of memory and six hard drives. As the database grows in size or the number of users increase you scale out by adding another server with 2 processors, 4 GB of memory, and 6 disk drives to maintain the same level of performance. Although you do not necessarily have to add an additional server with the exact specifications, it does reduce the complexity of scaling out. See Figure 1-4.

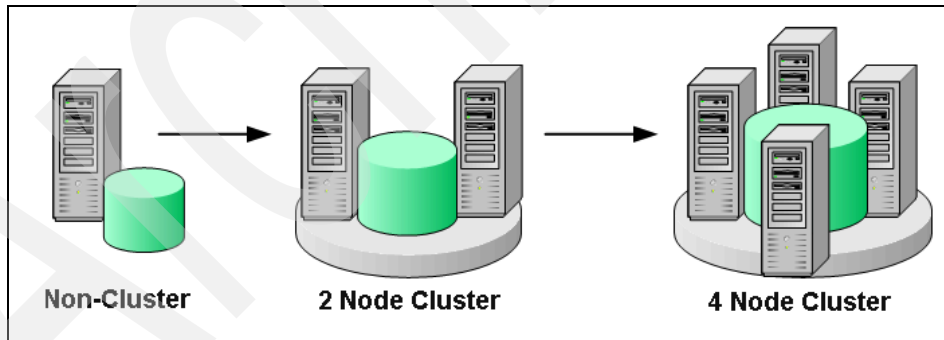


Figure 1-4 Clustering for scaling out

The benefit to scaling out is that you can achieve near linear scalability. That is to say, as you add each additional server to the system you effectively increase your system capacity proportionally. Thus, scaling out provides much better returns in terms of the additional costs associated with adding additional servers to the system. Another benefit inherent with scaling out is that a cluster of smaller servers generally costs less than a single large system. For example, a database

server running on a system that is built on sixteen 4-way SMP servers is generally much less expensive than a single 64-way SMP server, yet provides the equivalent processing power in terms of CPUs and memory.

The drawback to scaling out is that it requires system and database administrators that understand the technology well enough so that it can be implemented effectively. As more and more businesses embrace clustering technologies (and in doing so, clustering becomes more mainstream) this drawback diminishes considerably.

Another drawback is that clustering requires software that is specifically designed for the task. DB2 UDB V8.1 Enterprise Server Edition (ESE) includes a Database Partitioning Feature (DPF) that has a proven track record for superior scalability on clustered systems for over a decade.

Architectures for scaling out

There are two distinct approaches to scaling out database management systems. These are generally referred to as a shared architecture and a shared-nothing architecture. Both architectures attempt to achieve the same goal. That is, to implement a database management system that consists of a cluster of servers, provides linear scalability, and appears as single database to the end users. See Figure 1-5.

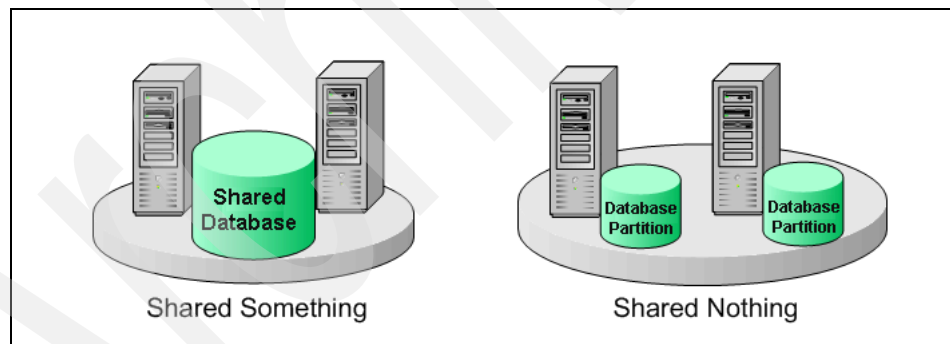


Figure 1-5 Scale out architecture

A shared architecture attempts to accomplish this goal while sharing the database. That is to say, as additional servers are added to the system, they all share or attempt to share the same database, which resides on shared storage, hence the name shared architecture. This is the approach some database vendors have taken with limited success in supporting very large databases (VLDB) in the multi-terabyte category.

On the other hand, a shared-nothing architecture accomplishes the same goal by dividing a very large database (VLDB) into smaller and more manageable “parts,”

called partitions. The term shared-nothing simply refers to the fact that as additional servers are added to the system, each server (called a partitioned database server) manages a clearly defined portion of the database.

The fact that the database is partitioned should not imply that the system cannot be implemented on shared storage. This is a common misconception of a shared-nothing architecture. In reality, multiple partitioned database servers can be implemented on a single server to more effectively leverage system resources on very large SMP servers. This is referred to as multiple logical partitioned (MLP) database servers. IBM has been very successful at implementing a shared-nothing architecture with DB2 UDB. See Figure 1-6.

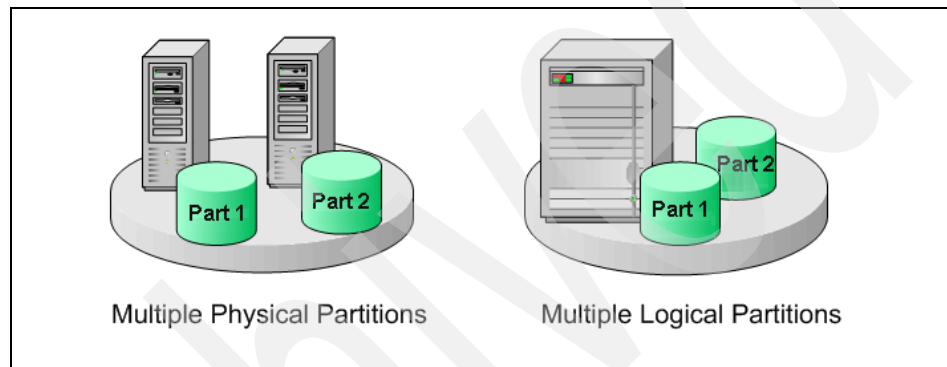


Figure 1-6 Partitions in a share-nothing architecture

Note: In versions of DB2 UDB prior to 8.1, implementing multiple partitioned database servers on a single system was commonly referred to as Multiple Logical Nodes (MLNs). In DB2 UDB V8.1 we have simply changed the terminology and now prefer the term *partition* to refer to what was previously called a node, as the term *node* introduced some confusion because it is commonly used to refer to a physical server in a cluster. On a single system with more than one partitioned database server we now use the term Multiple Logical Partitions (MLPs).

In summary, scaling up is relatively easy to implement as it is less complex and requires less system administration. Scaling up introduces diminishing returns inherent in system resource management and can be a more expensive solution, in terms of hardware, than scaling out. On the other hand, scaling out requires system administrators who can effectively implement clustering technology, and also requires software specifically designed to cluster multiple servers into a system that appears as a single system to the end user.

1.2 Scaling with DB2 Universal Database

DB2 Universal Database (UDB) is IBM's premier relational database management solution that has achieved worldwide market share leadership on a wide variety of platforms including the Windows operating system. DB2 UDB provides customers of all sizes with remarkable value by implementing innovative self-management and automation technology combined with superior price performance and scalability.

DB2 UDB has been running on Intel platforms for well over a decade. Back in the early years the product was called Database Manager, and the name can still be found in many aspects of the product today. For example, multiple DB2 instances are equivalent to what was once, many years ago, a single Database Manager. As Windows NT gained popularity in the mid 1990s, IBM announced DB2 for NT V2.1 in 1995.

The following is a quote from the original DB2 for NT announcement:

"If you need a reliable and powerful database on Windows NT in a stand-alone, departmental, or enterprise environment, IBM's premier family of relational database products, DB2, is now available."

Today, DB2 UDB V8.1 for Windows represents close to a decade of database development effort by IBM on the Windows platform.

1.2.1 DB2 UDB for Windows Editions

DB2 UDB for Windows is available in several editions that can scale from hand-held devices running Windows CE, to laptops and desktops running Windows XP, and all server editions of the Windows Server 2003 operating system.

DB2 UDB V8.1 is certified for Microsoft Windows Server 2003 for the Standard, Enterprise, and Datacenter editions of the operating system, including both the 32-bit and 64-bit versions.

Note: DB2 UDB V7.2 with Fixpak 4 or later is supported on 32-bit Windows Server 2003. This is due to the timing of the release of the 64-bit Windows Server 2003 not being generally available during the development of DB2 UDB V7.2. The code base was therefore not optimized for the 64-bit Windows Server 2003.

In this section we take a look at the server editions of DB2 UDB V8.1 that can be deployed on the Windows Server 2003 operating system. These include:

- ▶ Express Edition
- ▶ Workgroup Server Edition
- ▶ Workgroup Server Unlimited Edition
- ▶ Enterprise Server Edition

Note: DB2 UDB Personal Edition provides a single user database engine ideal for deployment on laptops and desktop personal computers for users that require a local database on their system. Since this edition does not support client/server connectivity over a network, it will not be discussed in this book.

Table 1-1 summarizes the server editions of DB2 UDB V8.1

Table 1-1 DB2 UDB sever editions

Edition	Processor	Description
Express	32-bit servers with up to 2 processors	Full-function database server designed to be embedded as a transparent part of a business solution with a named user licensing model.
Workgroup Server	32-bit servers with up to 4 32-bit processors	Full-function database server designed for small organizations or departmental use with a per-user licensing model.
Workgroup Server Unlimited	32-bit servers based with up to 4 32-bit processors	Full-function database server designed for small organizations or departmental use with a per-processor licensing model.
Enterprise Server	32-bit and 64-bit servers with unlimited processor support	Full-function database server designed for the enterprise with flexible scale-up and scale-out capability. Also offers connectivity and integration of other enterprise DB2 and Informix® databases.

DB2 UDB Express Edition

DB2 UDB V8.1 Express Edition is the newest member of the DB2 UDB product family. DB2 UDB Express Edition is a specially tailored full feature relational database for small and medium businesses, and is designed to be embedded as a transparent part of a business solution. This edition can be deployed on servers with up to two processors and is only available for 32-bit platforms.

DB2 UDB Workgroup Server Edition

DB2 UDB Workgroup Server Edition (WSE) is designed for small organizations and departmental use that supports a small number of connected users. Workgroup Server Edition uses a licensing model designed to provide an attractive price point for smaller installations while still providing a full-function database server. This edition can be deployed on servers with up to four processors and is only available for 32-bit platforms.

The Workgroup Server Edition of DB2 UDB is ideal for deployment on servers with up to four processors and when the number of concurrent or registered users is small (25–30 users), as a license is required for each concurrent or registered user in addition to the base server license.

DB2 UDB Workgroup Server Unlimited Edition

The Workgroup Server Unlimited Edition (WSUE) offers the same features provided with Workgroup Server Edition, but with a simpler per-processor licensing model. This edition is ideal for deployment on 32-bit servers with up to four processors when the number of concurrent or registered users is large (more than 25–30 users) or unknown (Web servers) since no additional license is required for each concurrent or registered user.

DB2 UDB Enterprise Server Edition

DB2 UDB Enterprise Server Edition (ESE) is designed for large enterprises that require a database management system that can scale to support multi-terabyte databases. It is available in both 32-bit and 64-bit versions. There is a migration path that allows you to migrate databases from 32-bit to 64-bit versions without having to rebuild the database from scratch.

The Enterprise Server Edition of DB2 UDB is ideal for deployment on servers of all sizes, as it provides a number of features that are not available in the Workgroup Server Edition of the product.

In general, this edition is required when you need support for more than four processors, have a large number of concurrent or registered users, or you require the scalability provided by 64-bit platforms and/or the Database Partitioning Feature (DPF).

The Database Partitioning Feature (DPF) allows you to partition a database within a single system or across a cluster of systems. This feature provides support for very large databases (VLDBs) by partitioning the database into smaller parts that can more easily be managed by individual partitioned database servers (engines). This approach not only supports very large databases with complex workloads, it also provides great scalability for database administration tasks such as loads, backups, and recovery.

Note: The Database Partitioning Feature is an optional licensed feature of DB2 UDB Enterprise Server Edition. The feature is already included in this edition of DB2 UDB and therefore does not require the installation of any additional components.

You also can scale by upgrading products, that is, you could start off with Express and then choose to upgrade to WSE and then ESE. A detailed description of DB2 UDB product offerings for Windows features of each product and licensing can be found in *DB2 UDB Evaluation Guide for Linux and Windows*, SG24-6934.

1.2.2 Try and buy product availability

IBM provides sixty day evaluation copies of most of the above products for download over the internet. The evaluation code can be obtained by following the instructions at this Web page:

<http://www-3.ibm.com/software/data/db2/udb/>

1.3 Scaling with Windows Server 2003

Windows Server 2003 is Microsoft's latest offering in the Windows server operating system family that builds on the technology from the Windows 2000 Server family while making it easier to deploy, manage, and use. It includes all the functionality customers need today from a Windows server operating system with enhanced security, reliability, availability, and scalability.

Before we go into the details of each individual editions of Windows Server 2003, let us first take a look at some of the new and improved features available with Windows Server 2003.

64-bit support

Both Windows Server 2003 Enterprise Edition and Datacenter Edition will be available in 64-bit versions to the Intel Itanium-based servers. These high-scalability versions of Windows Server 2003 will provide huge virtual memory address space, up to 64 GB on Enterprise Edition, and a mind numbing 512 GB on Datacenter Edition. As we mentioned in the previous section, DB2 UDB Enterprise Server Edition is already available for 64-bit platforms.

Hot-add memory

Both Windows Server 2003 Enterprise and Datacenter Edition will provide support for hot-pluggable memory. This high-availability feature, called hot-add

memory, will allow memory to be automatically added to the operating system addressable memory space without re-booting. Note that the feature is not called hot-swap memory, because it only allows you to add memory. It does not support removing or swapping memory. Support will require, at a minimum, hardware that supports hot-add memory. This feature, once implemented at the hardware level, will allow you to re-size your DB2 UDB buffer pools completely online, since DB2 UDB V8.1 already supports resizing buffer pools online (without recycling the engine).

NUMA

Both Windows Server 2003 Enterprise Edition and Datacenter Edition will provide support for NUMA, or Non-Uniform Memory Access. These editions of Windows Server 2003 provide NUMA awareness to applications and can manage threads and memory via NUMA application program interfaces, sometimes referred to in the industry as NUMA nuggets. Since support for NUMA is provided at the operating system level it does not require applications to directly implement the NUMA API. However, the NUMA API is available to applications to allow developers to obtain system information about the NUMA architecture and better leverage this technology.

Clustering Services

Both Windows Server 2003 Enterprise Edition and Datacenter Editions will include Microsoft Clustering Services (MSCS). This high-availability feature has been available in Windows since Windows NT Enterprise Edition and has been improved to allow up to eight servers to be members of a single high-availability cluster. In Windows 2000 Advanced Server you were limited to two nodes, and in W2K Datacenter Server you were limited to four nodes. One big enhancement of Windows Server 2003 in clustering is volumes without a drive letter, which increases the number of drives that can be used in a cluster environment. Another enhancement to MSCS in Windows Server 2003 is support for volume mount points on shared disks. Volume mounting has been supported since Windows 2000, but volume mount points could not be used in conjunction with MSCS. The additional node support greatly simplifies high availability clustering when scaling out with DB2 UDB Enterprise Server Edition.

Windows System Resource Manager (WSRM)

The Windows System Resource Manager (WSRM) provides resource management features that allow you to allocate resources, including processors and memory, among multiple processes based on business priorities. WSRM allows administrators to set hardware resource targets for any number of applications or users. It also creates utilization accounting records for management, service level agreement (SLA) tracking, and enabling charge-backs. This resource management tool can be used to manage resource

consumption when multiple instances of DB2 UDB are deployed on a single server.

1.3.1 Windows Server 2003 Editions

The Windows Server 2003 server operating system is available in four editions that can scale from single processor servers up to large 64-way (64-bit) servers supporting up to 512 GB of memory.

In this section we take a look at the different editions of Windows Server 2003, which include:

- ▶ Windows Server 2003 Web Edition
- ▶ Windows Server 2003 Standard Edition
- ▶ Windows Server 2003 Enterprise Edition
- ▶ Windows Server 2003 Datacenter Edition

The table below summarizes the features of the server editions of Windows Server 2003. These features will be discussed later in this section.

Table 1-2 Windows Server 2003 Editions

Edition	Web	Standard	Enterprise	Datacenter
Processor	2	4	8	32 (32-bit) 64 ^a (64-bit)
Memory (GB)	2	4	32 (32-bit) 64 (64-bit)	64 (32-bit) 512 (64-bit)
32-bit	Yes	Yes	Yes	Yes
64-bit	No	No	Yes	Yes
4 GB Tuning	Yes	Yes	Yes	Yes
AWE Support	No	No	Yes (32-bit)	Yes (32-bit)
Hot-add memory	No	No	Yes (32-bit)	Yes (32-bit)
NUMA Support	No	No	Yes	Yes
Cluster Service (MSCS)	No	No	8 nodes	8 nodes

a. Microsoft provides a 128-way SKU for Datacenter Edition for licensing Windows Server 2003 Datacenter Edition on a single server with 128 processors; however, the largest operating system partition can only support 64 processors.

Windows Server 2003 Web Edition

Windows Server 2003 Web Edition is a new member of the Windows Server 2003 family. Designed for building and hosting Web applications, it comes with support for up to two 32-bit processors and two GB of memory.

The Web Edition of Windows Server 2003 delivers a single-purpose solution for Internet service providers, application developers, and others who use or deploy specific Web functionality and is ideal for deployment with DB2 UDB Express Edition. The Web Edition for Windows Server 2003 has limited room in scaling up, therefore this edition is not listed further in this book.

Windows Server 2003 Standard Edition

Windows Server 2003 Standard Edition is the follow on to Windows 2000 Server. Designed for small organizations and departmental use, it provides support for up to four 32-bit processors and 4 GB memory.

Microsoft Windows NT and Windows 2000 technologies have always provided applications with a flat 32-bit virtual address space that provides four (4) gigabytes (GB) of virtual memory. The address space is typically divided in half so that two GB of address space is reserved for the Windows kernel and the other two GB is available for applications.

Windows NT Server Enterprise Edition (SP3) was the first version to provide applications with a 3-GB flat virtual address space. This memory tuning feature, which can be enabled with the /3GB boot.ini switch, allows 32-bit applications that are aware of the /3GB switch to increase their virtual address space by an additional one GB of memory for a total of three GB virtual address space. Windows 2000 (W2K) Advanced Server and W2K Datacenter Server also supported the 4 GB Tuning. See Figure 1-7.



Figure 1-7 4 GB Tuning

In the previous Windows server offerings 4 GB Tuning was not available with the entry-level editions of Windows NT Server or Windows 2000 Server. However, in response to customer requests, Microsoft has expanded the availability of this memory tuning feature to the 32-bit version of Windows XP Professional and all 32-bit versions of Windows Server 2003.

The Standard Edition of Windows Server 2003 is ideal for implementing DB2 UDB Workgroup Server Edition or Workgroup Server Unlimited Edition. Now that the Standard Edition supports a 3-GB flat virtual address space you should plan on tuning your database buffer pools beyond the 2-GB limitations imposed by other 32-bit operating systems.

The Standard Edition can also be used to implement the Database Partitioning Feature (DPF) included in DB2 UDB Enterprise Server Edition to scale out on 2-way and 4-way SMP servers, making it ideal for scaling out on 32-bit Intel blade servers. The only downside to this edition is that it lacks the high-availability features of Microsoft Clustering Service, which is only available in the Enterprise and Datacenter editions of the operating system.

Windows Server 2003 Enterprise Edition

Windows Server 2003 Enterprise Edition is the follow on to Windows 2000 Advanced Server. Designed for medium to large businesses, it provides support for both 32-bit and 64-bit servers. The 32-bit version supports up to 8-way SMP servers with up to 32 GB of memory. The 64-bit version supports up to 8-way Itanium SMP servers with up to 64 GB of memory. The Microsoft Cluster Service is included in this edition and provides support for up to 8 nodes in a single high availability cluster.

The 32-bit version of this edition provides support for the Intel IA-32 Physical Address Extensions (PAEs). Physical Address Extension is a memory-addressing extension that enables processors to expand the number of bits that can be used to address physical memory from 32 bits to 36 bits. PAE supports up to 64 GB of physical memory for applications running on most 32-bit (IA-32) Intel-based servers.

The Windows operating system supports the IA-32 PAE feature through the Microsoft Address Windowing Extensions (AWE) API. This API provides 32-bit Windows applications with the ability to address up to 64 GB of physical non-paged memory. The AWE API has been available on 32-bit versions of Windows XP Professional, Windows 2000 (W2K) Advanced Server, and W2K Datacenter Server for some time now. It is now supported on the 32-bit versions of Windows Server 2003. Although the AWE API is available on all versions of Windows Server 2003, support is limited to 4 GB of memory with Windows Server 2003 Standard Edition. Its primary purpose on the Standard Edition is for development and testing of applications that use the AWE API.

Note that Windows is currently the only 32-bit operating system that supports Intel's IA-32 PAE in both the kernel and application address space. This allows both the Windows operating system and applications to support large memory above 4 GB. See Figure 1-8 on page 16.

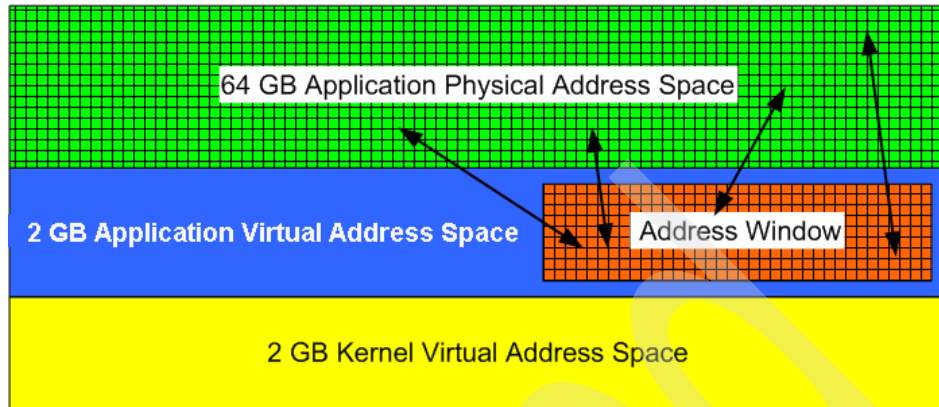


Figure 1-8 Microsoft Address Windowing Extensions

The Address Windowing Extension (AWE) API is enabled using the `/PAE` boot.ini switch. Note that without the `/PAE` boot.ini switch Windows will not load the PAE-enabled kernel, and the total system memory reported by Task Manager will be about 4 GB regardless of how much physical memory is actually installed. This is because the Windows kernel itself uses the AWE API for memory management.

The Enterprise Edition of Windows Server 2003 is ideal for implementing DB2 UDB Workgroup Server and Enterprise Server Editions. The availability of the Microsoft Address Windowing Extension API allows you to address up to 64 GB of memory with the 32-bit version of this edition. Support for AWE has been integrated into the server editions of DB2 UDB since Version 7.2.

For high availability the Microsoft Clustering Service (MSCS) provides support for up to eight nodes in either non-partitioned or partitioned (clustered) databases. Support for MSCS has been integrated into DB2 UDB since V5 and has been enhanced considerably in V8.

Windows Server 2003 Datacenter Edition

Windows Server 2003 Datacenter Edition is the follow on to Windows 2000 Datacenter Server. Designed for the enterprise customer who demands the highest levels of scalability, availability, and reliability required to deliver mission-critical solutions. The Datacenter Edition is available in both 32-bit and 64-bit versions and includes all of the functionality found in the Enterprise Edition, and adds additional features as well.

The 32-bit version supports up to 32-way SMP servers with up to 64 GB of memory. The 64-bit version supports up to 64-way Itanium SMP servers with up

to 512 GB of memory. The Microsoft Clustering Service is in this edition and provides support for up to eight nodes in a single cluster.

What sets the Datacenter Edition of Windows Server 2003 apart from other editions is that it is only available through original equipment manufacturer (OEM) partners. What this means is that it cannot be obtained directly from Microsoft. Instead it must be purchased from an OEM as a complete hardware, software, and services solution.

In addition to getting support through the usual Microsoft channels, customers of Windows 2003, Datacenter Edition, can join the Windows Datacenter High Availability Program. This program provides an integrated hardware, software, and service offering, delivered by Microsoft and qualified Datacenter Service Providers, such as OEMs.

Microsoft offers the Windows Datacenter High Availability Program that complements this edition. The Datacenter High Availability Program has a unique support and services model built for customers with mission-critical requirements. This program helps customers run mission-critical business applications with predictable and manageable systems.

1.4 Scaling with IBM xSeries Servers

The IBM eServer xSeries servers provide powerful Intel-processor based servers based on IBM's Enterprise X-Architecture™ with intelligent systems management capabilities and backed by world-class service and support teams.

The IBM X-Architecture technology is an evolving blueprint for xSeries servers that is drawn from the vast enterprise server heritage of the company. xSeries engineers take the technologies that have already revolutionized larger IBM systems and bring them to the Intel-based platform.

The strength behind X-Architecture technology pulls together many of the best features of the other systems: Availability characteristics achieved by zSeries®, scalability of the pSeries™, and self-maintaining capabilities of the iSeries™. The results: Industry-standard servers designed to provide enterprise scalability at very attractive prices.

1.4.1 IBM xSeries servers family

The IBM Intel-based xSeries server line is ideal hardware for the Microsoft Windows Server platform. The high-end xSeries servers feature Enterprise X-Architecture technology, offering revolutionary advances in performance, memory, and I/O capabilities, as well as a pay-as-you-grow approach to buying

Intel-based 32-bit and 64-bit high-end xSeries systems. The Enterprise X-Architecture strategy delivers enterprise-class technology to the high-end xSeries servers, with enhanced availability, scalability, and systems management, all on industry-standard hardware.

The IBM eServer xSeries Server family includes Intel-based universal, rack, blade, and scalable servers. In this section we provide a brief overview of IBM's eServer xSeries family of servers, which include:

- ▶ IBM xSeries Universal Servers
- ▶ IBM xSeries Rack-Optimized Servers
- ▶ IBM xSeries BladeCenter™ Servers
- ▶ IBM xSeries Highly-Scalable Servers

IBM xSeries Universal Servers

The IBM eServer xSeries Universal Servers are designed to meet a variety of needs including file, print, application, and mail serving. These servers are ideal for small businesses or departmental use and are available in various tower and rack configurations from single processor, dual processor, and 4-way processor models.

The top of the line xSeries Universal Servers offer the latest Intel Xeon MP processors with up to four processors and 12 GB of memory. They include high-availability features such as Memory Mirroring and Hot Spare Memory, which increases system availability by protecting the server and data from memory failures.

IBM xSeries Rack Servers

The IBM eServer xSeries Rack Servers offer a complete line of rack-ready, industry-standard solutions. These servers are designed for the IBM NetBay rack, which provides outstanding performance in cooling, airflow, weight, and power requirements while helping to save precious floor space. These highly dense rack-optimized servers are available in single processor, dual processor, and 4-way processor models.

The latest IBM offering in rack-optimized eServer xSeries servers is the x360 model. The powerful x360 is a space-efficient, rack-dense, 4-way server for demanding enterprise applications. The x360 model is the first rack-optimized server to incorporate IBM Enterprise X-Architecture technology. It features up to four Intel Xeon MP processors and up to 8 GB of memory, while packing power into a high dense 3U design ideal for the constrained data center.

IBM xSeries BladeCenter Servers

The IBM eServer xSeries BladeCenter offers the performance you expect from traditional rack-optimized servers at twice the density of a 1U server. Packed with

powerful 2-way Intel Xeon processor-based blades, the BladeCenter delivers density without sacrificing processor performance in a highly scalable and modular design. The reliability, advanced systems management, and integrated networking features of the BladeCenter make it ideal for centralized infrastructures.

The BladeCenter's high-availability ePlane provides a fault-tolerant connection from the blade servers to all modular components including up to four redundant hot-swap, load-balancing power supply modules and redundant switch modules supporting GB Ethernet and Fiber Channel connectivity.

BladeCenter is the ideal solution for scaling out with DB2 UDB on Windows Server 2003, as you can cluster up to 14 2-way blades with 8 GB of memory each in a single 7U dense form factor.

IBM xSeries Scalable Servers

The IBM eServer xSeries Scalable Servers include both the x440 and x450 rack-optimized Enterprise X-Architecture technology servers. The IBM x440 was the industry's first Xeon MP-based 8-way server and is now available in a 16-way server with Intel's 32-bit Xeon processors. It delivers 8-way (4U) and 16-way (8U) scalable performance in a modular rack-dense form factor providing space savings for crowded data centers without sacrificing scalability and performance. The xSeries x450 also leverages IBM's proven Enterprise X-Architecture chipset to deliver 64-bit systems with the latest Intel Itanium processors.

xSeries x440 Server

At the core of the xSeries 440 is a modular central electronics complex (CEC) with 32 MB of level-4 system cache that can be populated with 4 processors, 16 GB of memory, and six PCI slots. These 4-way CECs (nodes) provide the building blocks that can be expanded to 8-way SMP servers using an SMP Expansion Module and can scale all the way up to 16-way SMP by connecting two x440 chassis together using the SMP Expansion Ports.

Each SMP Expansion Module has 16 DIMM slots allowing you to expand the standard 2 GB ECC SDRAM memory up to 32 GB per SMP Expansion Module and up to 64 GB on a fully populated x440 server. The x440 server's 32 MB of level-4 system cache per CEC (node) maximizes CPU throughput by reducing main memory access under demanding workloads resulting in overall enhancement to system performance.

The xSeries x445 Server is the follow on to the x440. This model extends the scalability features of the x440 to support up to 32-way SMP processors. Each x445 can scale from 2-way to 8-way in a single rack-optimized 4U chassis. It can scale to 16-way simply by connecting two 8-way chassis or up to 32-way by connecting four chassis in a industry standard rack. The x445 also expands

memory capacity to 64 GB per SMP Expansion Module and up to 128 GB on a fully populated x445 server. The x445 level-4 system cache is also expanded to 64 MB per CEC (node).

xSeries x450 Servers

The xSeries 450 expands the Enterprise X-Architecture with the XA-64 chipset to include the Intel Itanium Processor Family (IPF), delivering highly scalable rack optimized 64-bit 4-way servers in a 4U form factor.

Unlike the x440, which uses an SMP Expansion Module to house processors, memory, and controllers, the x450 contains separate memory and processor board assemblies. The memory-board assembly provides support for up to 40 GB of standard PC2100 ECC DDR SDRAM DIMMs. It also contains both the memory and I/O controllers. The memory controller supports both 2-way and 4-way interleaving in order to ensure that the memory subsystem can supply data fast enough to match the throughput of the CPUs.

The processor-board assembly provides support for up to four 64-bit Intel Itanium2 processors and up to 64 MB of 200 MHz DDR memory for the level-4 system cache. The processors are connected together with a 200 MHz front side bus, but supply data at an effective rate of 400 MHz using the dual-pump design of the Intel Itanium2 architecture.

The xSeries x455 Server is the follow on to the x450. This model extends the scalability features of the x450 to support up to 32-way Itanium2 processors.

1.5 Design considerations for scaling

In the previous sections we covered the server editions of DB2 UDB, which include Workgroup Server Edition (WSE), Workgroup Server Unlimited Edition (WSUE), and Enterprise Server Edition (ESE). We also covered the Windows Server 2003 editions available from Microsoft, which include Standard Edition, Enterprise Edition, and Datacenter Edition. Finally we covered the IBM xSeries Server family.

In this section we cover design considerations for implementing highly scalable database server solutions with DB2 UDB for Windows Server 2003 on the IBM xSeries Servers.

We take a look at design considerations for scaling on DB2 UDB ESE V8.1 for Windows Server 2003. Scaling up with DB2 UDB does not require the DB2 UDB Enterprise Server Edition. You can scale up with any of the other editions, but due to licensing you are limited to scaling up to two processors on DB2 UDB Express Edition and four processors on both Workgroup Server editions. Another

limitation with these editions is that they are only available for 32-bit platforms. For these reasons, you should consider DB2 UDB Enterprise Server Edition if you plan on scaling up beyond 4-way SMP servers.

Before getting into the details of design considerations for scaling up or scaling out, it should be noted that you do not have to commit to one solution or the other. If you spend a little time planning your solution, you should be able to scale up and subsequently scale out as needed.

1.5.1 Planning considerations

As you design for your DB2 UDB scale-up solution, there are a number of licensing considerations you should take into account. These include:

- Is the database scalable?

Will you be able to add processors to your system and continue to add licenses, or will you have to upgrade (re-install) your database management system. If you start out with DB2 UDB Workgroup Edition you will not be able to scale up beyond four processors due to the licensing model of this edition. You will not be able to scale out either, because the Database Partitioning Feature (DPF) is only included in Enterprise Server Edition.

The processor-based licensing of Enterprise Server Edition with the optionally licensed Database Partitioning Feature (DPF) solves this problem by allowing you to scale up the number of processors to the capacity of your hardware and scale out to 4,095 clustered servers.

- Is the operating system scalable?

Will you be able to add processors and memory to your system without having to upgrade the operating system? Windows Server 2003 Standard Edition is licensed for only four processors and 4 GB of memory. If you need more resources beyond this you will need to upgrade to Windows Server 2003 Enterprise Edition to solve this problem. If you start out with Enterprise Edition you cannot upgrade to Datacenter Edition, as this is not allowed, and under most circumstances you will likely have to fork lift the server.

- Is the hardware scalable?

Will you be able to add resources such as additional processors, memory, network, and storage? If you know that eventually you will need twice the resources as you have today to maintain the same level of service, you should consider planning for that eventuality with server hardware that can scale from where you are today to where you want to be tomorrow. Otherwise you will end up having to fork lift the server.

- Did you plan for high availability?

Will you be able to sleep at night without having to worry about whether your database is up and running due to a hardware or software failure? The Microsoft Clustering Services (MSCS) feature included with Windows Server 2003 Enterprise Edition and Datacenter Edition provides high-availability clustering, and all server editions of DB2 UDB support the MSCS feature. It should be noted that DB2 UDB must be licensed for a stand-by database server when implementing high availability. All processors must be licensed when implementing active/active configurations. Active/passive configurations require at a minimum a single processor license for ESE and a base server license for Express, WSE, and WSUE for the passive or idle stand-by server.

1.5.2 Hardware considerations

The consideration for system hardware selection and configuration is to obtain a well-balanced system that can sustain high rates of overall system scalability. To achieve a scalable system, we must adequately size individual hardware resources so that no single resource results in a system bottleneck. In this section we look at primary hardware considerations for scalable database systems.

You should take into consideration how you plan to scale your system, when planning for additional capacity requirements. Scaling out is generally achieved by adding more resources, such as memory, processors, and storage across multiple systems. DB2 UDB V8.1 ESE implements a shared-nothing architecture that partitions large databases into smaller, more manageable “parts” called partitions that are managed by partitioned database servers. Scaling up is achieved by adding more resources, such as memory, processors, and storage to a single system.

In general, scaling out is much easier in terms of hardware planning, as you scale out by simply adding additional hardware resources (processors, memory, etc.) by adding additional servers. A scale-up approach typically requires a little more planning, as you may reach hardware system limitations in terms of the amount of resources that can be added to the server. After the system is scaled out, due to the number of servers being increased, the tasks for administration and management may also be increased.

Memory

Since accessing data in memory is faster than accessing data on hard drives, the primary factor in terms of memory is quantity. Although memory speeds are also factors, it is seldom an option when configuring a system, unless you are willing to select and configure another system altogether. The amount of physical memory is a critical system hardware resource that can have a huge impact on overall

performance and scalability. In general, the cost of memory on commodity servers is usually an insignificant factor when compared to the cost of other hardware resources.

Today most systems based on the 32-bit Intel Architecture (IA-32) support the Physical Address Extensions (PAE) capabilities of the IA-32. Physical Address Extensions provide operating system software with an instruction set to address physical memory above four GB. Operating systems that take advantage of the PAE can address up to 64-GB physical memory.

Given the memory extensions of the IA-32, the primary factor in memory selection will most likely not be the cost of the memory itself, but rather the incremental cost moving from one edition of the Windows operating system to the next in order to address more physical memory.

Processor

Most systems are limited by the total number of central processing units (CPUs) they can support. Typically a 4-way cannot be upgraded to an 8-way unless it is indeed a true 8-way that was populated with only four processors. There are a few systems on the market today, such as the IBM x440, that can be expanded beyond the total number of original processors by adding additional processor expansion modules.

Besides quantity and speed, another important consideration in terms of processor selection is the size of the internal L2 cache. Slower processors with larger internal caches have shown significant throughput advantages for database applications over faster processors with smaller internal caches.

Another factor to consider when selecting the number of processors is the operating system software costs. There are incremental licensing costs associated with each additional processor for both DB2 UDB Enterprise Server Edition and Windows Server 2003 editions.

Storage

The disk subsystem has been a area of much debate over the last several years. Most disk subsystems will implement some form of redundancy that has always favored recoverability over performance. In recent years improvements in technology has been able to overcome many of the performance limitations imposed by implementing redundant disk arrays.

Performance characteristics of disk controllers include speed, throughput, channels, and cache. Care should be taken in the placement of disk controllers in the system. Although most disk adapters are backwards compatible, it should go without saying that you want to match the disk controller's speed with that of the systems' PCI bus. You should avoid placing faster 64-bit 66 Mhz disk controllers

in slower 32-bit 33 Mhz PCI slots. You should also consider the number of disk controllers in your system. Attaching a single disk controller with several I/O channels might be capable of driving your subsystem, but can quickly saturate a single PCI bus, not to mention introduce a single point of failure into your system. If possible you should also avoid placing disk controllers on PCI buses populated with other I/O-intensive resources.

Performance characteristics of disk subsystems include disk speed, size, cache, and the number of physical disks in the subsystem. You should favor a subsystem with a large number of small drives over a small number of large drives. If this is impractical, plan for growth by choosing a large number of large drives. Best performance will be achieved for database applications with a large number of physical disks (5–10) per processor.

Hardware implementations of disk arrays are now commonplace on Intel-based servers. Modern disk controllers support RAID levels 0, 1, 5, and 10, sometimes referred to as 0+1. As with most performance decisions there is always a give (cost) and take (performance) associated with choosing which RAID level to implement.

Network

Performance characteristics of network adapters include speed and throughput. As with disk controllers, care should be taken in the placement of network adapters in the system. You should also consider the number of network adapters in your system. If possible you should also avoid placing network adapters on PCI buses populated with disk controllers.

The speed of the network adapter can limit the total network throughput. You should consider using faster 64-bit 66 Mhz network adapters, especially when running on gigabit networks, as slower 32-bit 33 Mhz network adapters are not capable of driving gigabit networks.

Most network adapters today support teaming. Teaming network adapters provides several benefits. First, it provides network redundancy by preventing a single network adapter that can be a single point of failure in your system. Second, it provides better performance, as you transparently balance network traffic over two or more adapters and PCI buses.

1.5.3 Design considerations

Scaling up with DB2 UDB Enterprise Server Edition (ESE) is a relatively easy task to accomplish. The DB2 UDB engine is multi-threaded and can dispatch additional threads (called agents) to support a larger workload, throughput, or response time. Tuning DB2 UDB instance and database configuration parameters for scaling is covered in detail in other chapters of this publication.

Parallel processing

Leveraging additional processor resources with DB2 UDB is a relatively easy task to accomplish. The DB2 UDB engine is multi-threaded and can dispatch additional threads (called agents) to support a larger workload, throughput, or response time.

DB2 UDB's multi-threaded database engine provides support for both intra-parallel and inter-parallel processing. Intra-parallel processing takes place within a single instance of the database engine. This allows the database engine to leverage multi-processor systems to process queries and transactions in parallel. DB2 UDB also exploits SMP architectures in utility operations. For example:

- ▶ **Loads:** The LOAD utility takes advantage of multiple processors and performs parallel I/O when parsing, formatting, and writing data to tablespace containers in parallel.
- ▶ **Indexes:** Index creation takes advantage of multiple processors and performs parallel I/O. This helps to speed up index creation when a CREATE INDEX command is issued, during restart (if an index is marked invalid), and during REORG processing.
- ▶ **Recovery:** Backup and restore utilities take advantage of multiple processors and perform parallel I/O when performing backups and restores by assigning multiple buffer manipulators among the processors. You can also back up to or restore from multiple devices (for example, tapes) in parallel.

The Database Partitioning Feature of DB2 UDB Enterprise Server Edition also allows you to implement inter-parallel processing. Inter-parallel processing takes place across multiple database partitioned server engines. This can be accomplished on both SMP servers and MPP clustered servers. The Database Partitioning Feature of DB2 UDB Enterprise Server Edition allows you to run multiple database engines (one per database partition) to leverage a large number of processors with multiple database engines. For example:

- ▶ **Loads:** The load utility splits and loads data in parallel. It can run in parallel on multiple partitioned database servers, significantly speeding up the overall load process.
- ▶ **Indexes:** The CREATE INDEX command runs in parallel across multiple partitioned database servers.
- ▶ **Recovery:** Backup and restore operations can be run in parallel across multiple partitioned database servers.

Combining both intra-parallel and inter-parallel processing provides for a highly scalable database solution. For example, a single database engine can back up data in parallel by dispatching multiple threads within the database engine. Two

database engines, each managing half the data, can back up the database in almost half the time.

Tip: When Implementing DB2 UDB ESE with the Database Partitioning Feature (DPF), plan on at least two (2) processors for every logical or physical database partition, as fewer than this may result in limited parallel processing.

Memory

Probably the most challenging task when it comes to scaling on 32-bit servers is memory utilization. Fortunately, Windows Server 2003 provides a number of memory management features that allow 32-bit applications to address large memory. These include 4 GB Tuning and the Microsoft Address Windowing Extensions (AWE) API covered in the previous sections of this chapter. Here we will take a look at some considerations when implementing these features with DB2 UDB. Table 1-3 shows these features available on 32-bit versions of Windows Server 2003.

Table 1-3 Memory tuning features

Feature	Standard	Enterprise	Datacenter
Optimize Memory for Network Applications	Yes	Yes	Yes
4 GB Tuning (3 GB Flat Memory Address)	Yes	Yes	Yes
Address Windowing Extension (AWE) API	No	Yes	Yes

Storage

DB2 UDB provides a very flexible architecture for managing database storage. When creating a table, the name of the base, index, and long tablespace can be specified. Using index and long tablespaces to store indexes and LOB data, respectively, allows these structures to be kept separate from the rest of the table data. This flexibility can be used to increase database performance and availability. It should be noted that once a table is created with a defined tablespace definition you cannot change the configuration without dropping and re-loading the table. For example, if you start out with table, index, and long data all in one tablespace, you will have to drop, re-create, and re-load the table to place the table, index, and long data in separate tablespaces.

A tablespace can be spread over one or more physical storage devices, providing the media-spanning capability. As you add storage to your database server you can re-configure your tablespaces in several ways. If you increase storage by

adding logical volumes to your database server, you can add containers to your existing tablespaces. If you increase storage by expanding your existing logical volumes, you can extend or resize your existing tablespace containers.

DB2 UDB has the ability to store data directly on a device without incurring the overhead of using a file system. Direct media access, sometimes called raw I/O, can improve performance.

Tip: When planning your tablespaces, use 3–4 containers per tablespace supported by at least six physical drives (spindles) for optimal performance.

Partitioning

Scaling out with DB2 UDB Enterprise Server Edition (ESE) requires the Database Partitioning Feature (DPF). As discussed in the previous section, the DPF implements a shared-nothing architecture that partitions large databases into smaller, more manageable parts. This feature can be implemented on either a single server or a cluster of servers. In either case, the implementation of the DPF is very similar, with the primary difference being where each partitioned database server resides.

Regardless of whether you partition on a single server or multiple servers, there is only one DB2 UDB instance to manage. Although you can create more than one DB2 UDB instance if needed, the important thing to remember is that the DB2 UDB instance, also referred to as the database manager, is managed in the same way as a non-partitioned instance. For example, when you update a DB2 UDB instance (DBM) configuration parameter, the parameter takes effect on all partitioned database servers in the cluster.

A database created in a partitioned database instance is still only one database and is also managed as such, although there is flexibility in managing each partition of the database. For example, you can create an additional database buffer pool on just one of the database partitions to take advantage of additional memory that might be available.

Scaling out on a single server is accomplished by implementing multiple logical partitions (MLPs) within a single instance of DB2 UDB running on a single server. This is much simpler than scaling out on clustered servers, as you do not have to worry about installing and configuring the DB2 UDB software on separate physical machines. See Figure 1-9 on page 28.

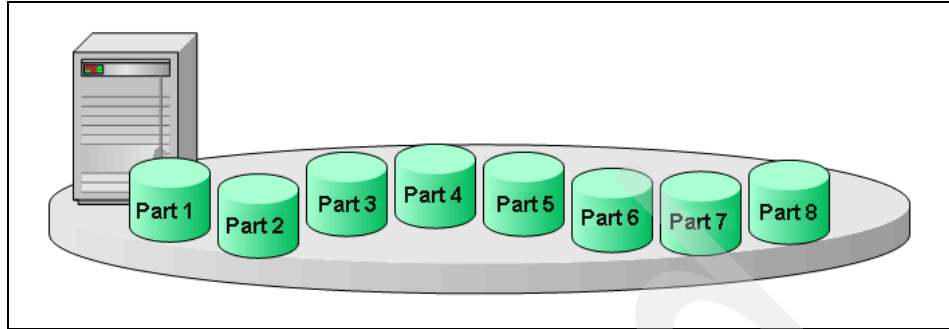


Figure 1-9 Multiple logical partitions

Scaling out on clustered servers is accomplished by implementing multiple physical partitions (MPPs) across multiple servers in a cluster. In this implementation you must install the DB2 UDB software on each physical server, but there is still only a single DB2 UDB instance and database to manage. See Figure 1-10.

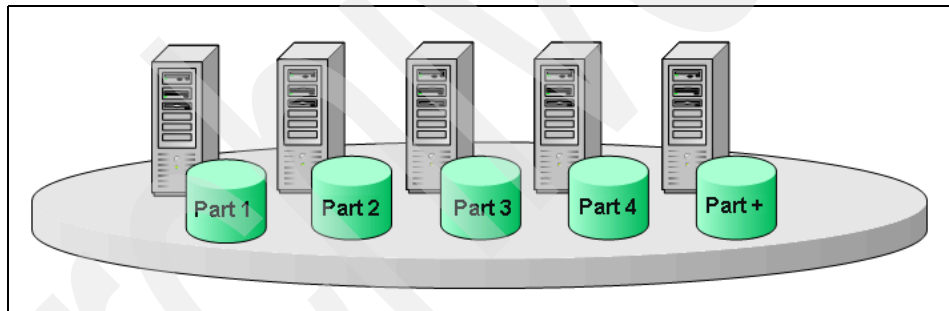


Figure 1-10 Multiple physical partitions

Finally, it is often feasible and, in many cases, desirable to implement a combination of both multiple logical and physical servers. This may be the case when we first scale up and reach a ceiling; then begin to scale out on a single server and reach a wall; and finally need to scale out across multiple servers, instead of fork lifting the box in favor of a larger single server. See Figure 1-11 on page 29.

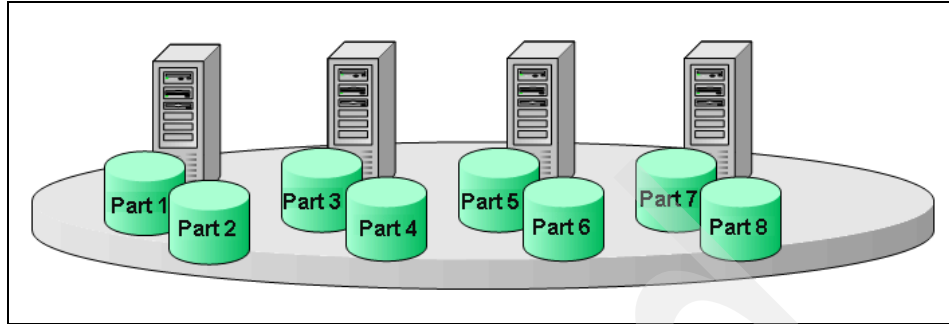


Figure 1-11 Scaling out with multiple logical and physical partitions

Another interesting scenario that can be implemented using both MLP and MPP is when you want to scale out without having to re-distribute the data across physical partitions. For example, your capacity planning efforts indicate that eventually you will need four database partitioned servers to support your database. So instead of starting out with two database partitions, you plan for the eventuality by implementing four partitions on two servers with two logical partitions per server, as illustrated in Figure 1-12.

Eventually your database grows beyond the capacity of each individual server. Now instead of adding partitions you simply move one database partition from each server onto a separate physical server. See Figure 1-12.

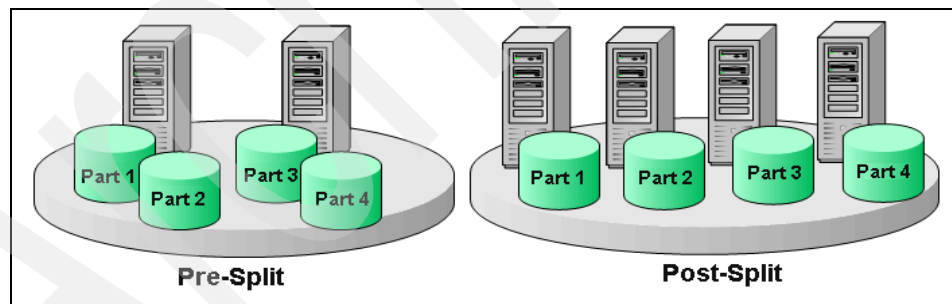


Figure 1-12 Splitting logical partitions

The movement of the actual database partition is accomplished using Storage Area Network utilities that very quickly move the logical unit (LUN) from one server to another and, in most cases, this is done by modifying the LUN security.

1.6 Documentation

The documentation throughout the IBM Redbook for DB2 UDB refers to two distinct sources of information. In previous editions of DB2 UDB the PDF and HTML documentation was very similar but this has changed with DB2 UDB V8.1. The documentation is included on a separate CD and must be installed separately. Each successive Fixpak also requires a documentation fixpak as well.

1.6.1 PDF documentation

The PDF documentation is used as the standard reference for all the documentation throughout this publication. This was chosen because it is the easiest method to point you directly to specific information in the DB2 UDB documentation. Unlike previous releases, however, the PDFs are not being updated with every Fixpak. The documentation is instead being updated with each major release, such as 8.1, 8.2, etc.

If you do not find a command or syntax that is referenced in this publication or in the PDFs it may be because a number of the references for DB2 UDB on 64-bit Windows were not included until V8.1 FP2. This Fixpak is when support began, so earlier 64-bit references were only for UNIX.

1.6.2 HTML documentation

The HTML documentation is now the most current source of information for DB2 UDB as the Fixpaks are released. The HTML documentation is initially installed using a separate zip file and a separate install is required for each Fixpak.

Accessing the documentation

The HTML documentation can be accessed using the DB2 Information Center. The Information is the new tool for searching through the HTML documentation. The tool has the documents summarized by category, and also an alphabetical index at the bottom of the page, which will allow you to quickly find the link to a document if you know the topic heading. The location of the information center is shown in Figure 1-13 on page 31.

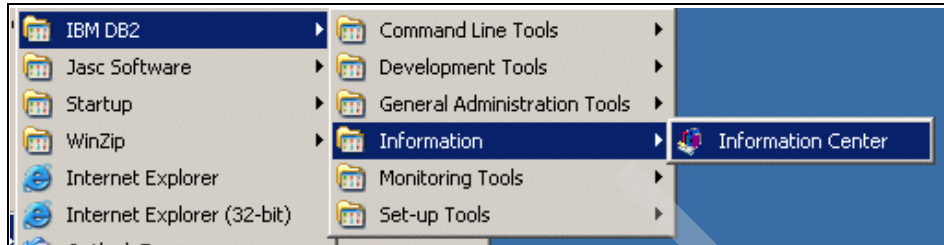


Figure 1-13 DB2 Information Center program location

Documentation updates

The updates for the HTML documents are included in a separate setup program from the DB2 Fixpak. This program contains the delta changes to the documents, and will update all of the HTML files that have been modified since the last fixpak. Using a delta update strategy will help to keep the update file size small.

The updates to the documentation are clearly labelled. If a section has been modified since the major release, V8.1, then the left-most column will have a number that indicates which fixpak the change was made in. This will allow you to be able to easily identify when changes were made to the product and what levels new functionality was added in. These indicators are shown in Figure 1-14 on page 32. The documentation indicators are listed below:

- ▶ 2: Changes made in DB2 UDB V8.1.2 (Fixpak 2)
- ▶ 1: Changes made in DB2 UDB V8.1.1 (Fixpak 1)
- ▶ I: Changes made after the V8.1 GA PDFs were published but before Fixpak 1

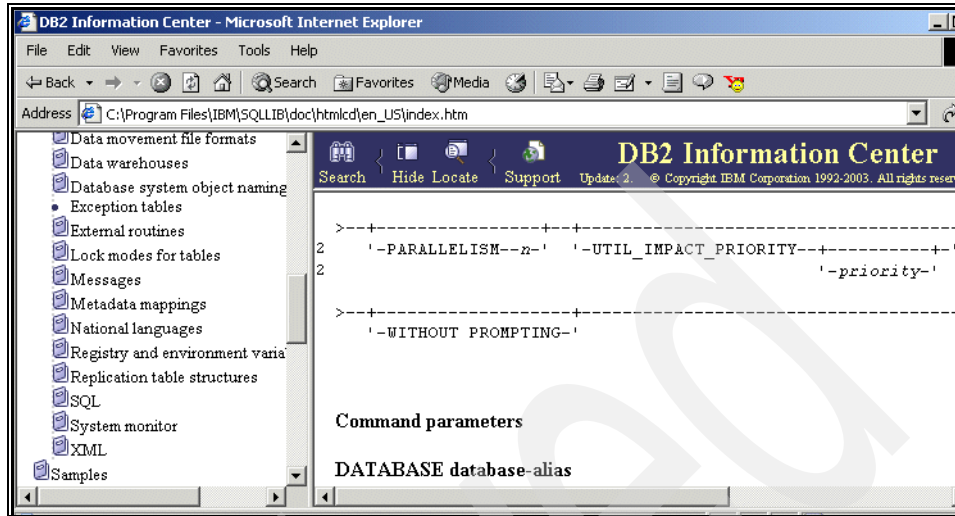


Figure 1-14 Fixpak documentation change indicators

Utilizing the capabilities of 64-bit

Businesses today are collecting and storing information at an alarming rate. As data repositories continue to grow, the limitations of a 32-bit architecture have grown more restrictive. With the release of Windows Server 2003, the Windows market can now overcome these issues and harness 64-bit.

This chapter describes step-by-step procedures on how to install DB2 UDB V8.1 on Windows Server 2003 Datacenter Edition. The instructions concentrate on a basic installation of DB2 UDB, and do not delve into the issues involved in setting up a partitioned database.

This chapter also leads you through the process to effectively utilize the 64-bit capabilities of DB2 UDB on Windows Server 2003. A series of best practices explains which parameters should be tuned and how to configure them. The case studies presented pull all the lessons together.

This chapter contains the following sections:

- ▶ Migrating to DB2 UDB V8.1
- ▶ Installation considerations
- ▶ 32-bit installation
- ▶ 64-bit installation
- ▶ DB2 configuration
- ▶ Additional considerations

2.1 Migration

The migration to Windows Server 2003 can be a complex process since you are most likely not only moving from Windows 2000 Server 2000 to Windows Servers 2003 and 32-bit to 64-bit, but also from DB2 UDB V7.2 to DB2 UDB V8.1. The migration process should be treated as two separate components: Upgrading versions of Windows and migrating versions of DB2 UDB.

2.1.1 Migrating DB2 between versions of Windows

Windows Server 2003 offers a great deal of new functionality. One of the greatest selling points for migrating to Windows Server 2003 is its ability to make use of the 64-bit architecture for a Windows system. This section covers what is required to migrate DB2 UDB between different releases of Windows.

Migrating DB2 from Windows 2000 to Windows 2003

The migration between different releases of the operating system can be a complex task because it often involves moving to a new system or cleaning out the contents of the C drive. There are a number of options available to you depending on which versions of DB2 UDB you are using. The following options apply only if the DB2 UDB version remains the same. The scenarios where the version of DB2 UDB has changed, such as from V7.2 to V8.1, are covered in the next section.

Using DB2 Backup and Restore

This migration is an easy process since the DB2 backup utility allows you to take a backup on one Windows server and restore to another Windows server. The backup image is OS version independent, so restoring to Windows Server 2003 from Windows 2000 Server, or vice versa, will work without any problems.

Steps

Following are the steps for using the DB2 Backup and Restore function to migrate a DB2 database from Windows 2000 Server to Windows Server 2003:

1. Install the same release of DB2 UDB on the new target Windows system.
2. Take the database offline on the original system.
3. Take a full offline backup of the database. Only offline backups can be used to restore to a newer version of DB2 UDB, as replaying of down level log records is not supported.
4. Copy the database backup image to the target system.
5. Start DB2 on the target system.

6. Restore the database. If the file systems that will contain the DB2 database are not identical on both systems, then a redirected restore will be required.
7. Check that the database was restored properly by checking on any restore error messages and items in the db2diag.log file.
8. Test a connection to the database on the target system.
9. Recatalogue the database on your client systems.

There are restrictions on which images can be restored from and to combinations of Windows, 32-bit and 64-bit, and releases of DB2 UDB. The diagram in Figure 2-1 illustrates the restrictions.

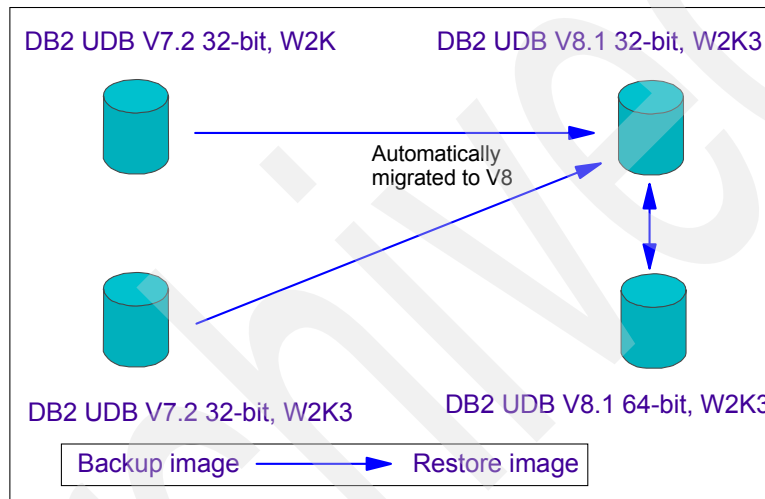


Figure 2-1 Backup and restore combinations with DB2 UDB V8.1 for Windows

Note: It is possible to take a backup image on DB2 UDB V7 and restore it on a different version of Windows on DB2 UDB V8.1. The new releases of Windows must be of the same bit level, so you cannot move from Windows 2000 32-bit to Windows Server 2003 64-bit.

If you try and restore from an unacceptable combination you will receive the following message:

SQL2570N A database cannot be restored on a platform that does not match the platform on which the backup image was created.

Moving disks to a new system

Sometimes the same external set of disks, be it in an array or a disk system, are going to be switched over to the new system. There are two distinct set of steps you will have to take, depending on whether the disk tables have changed.

Scenario 1: The disks assignments have not changed

If the disks are going to be laid out and labeled the same way on the new system then no work is needed. Once you install DB2 UDB you can catalogue the new database and DB2 will be able to use your old database. This option is sometimes referred to as “plug & play”. In order to use the “plug & play” approach you must ensure the following:

- ▶ The physical disk arrangement, file paths, and drive letters are preserved when existing disks are plugged into the target system.
- ▶ If raw disk is used for containers or a database log, they are referenced either as logical raw partitions (for example, “\\.\D:” on Windows) or as physical drives.

Note: On Windows this method is not supported for raw devices referenced by globally unique identifiers (GUID) that are unique to each server.

- ▶ Instance name(s) are the same on both systems.
- ▶ The number of database partitions must be the same on both source and target systems.

Steps of “plug & play”

Follow the steps below to use this method.

1. Install DB2 UDB on the target server with the same version and Fixpak as on the original server.
2. Stop DB2 UDB on the original server.
3. Unplug the disks (or disk subsystem) from the source machine in a manner approved by the operating system. On Windows you should disable the disk using the Device Manager utility before unplugging it in order to prevent system crash or data loss.
4. Enable the disks (or disk subsystem) on the target machine in an approved manner. On Windows you should use the Disk Management tool to rescan disks.
5. Verify that all disks are online and drive letters are identical on both systems.
6. Make the database available to the database manager on the target server (for example, by using **db2 catalog** command).

7. Proceed through the normal performance tuning tasks to ensure that DB2 UDB meets the performance expectations on the target server.

Note

In order to use this approach you must be aware of the following:

- ▶ The physical disk arrangement, file paths, and drive letters are preserved when existing disks are plugged into the target system.
- ▶ If raw disks are used for containers or the database log, they are referenced either as logical raw partitions (for example, "\\.\D:" on Windows) or as physical drives. Note that on Windows this method is not supported for raw devices referenced by globally unique identifiers (GUIDs), which are unique to each server.
- ▶ Instance names are the same on both systems.
- ▶ The number of database partitions must be the same on both source and target systems.

Scenario 2: The disks assignments have changed

Sometimes the DBA may not want to back up and restore the database since all of the database information is stored on a separate disk system, which can be unplugged from the old OS release system and connected to the new Windows 2003 server machine. The DB2 command **db2relocatedb** allows a DBA to relocate a database based on information in a configuration file. This command can also be used to inform DB2 how the location of the database files may have moved. Detailed information on the command can be found in the sources listed below.

PDF: *Command Reference*, Chapter 1, "System commands"

The command syntax is:

```
db2relocatedb -f configFileName
```

The *configFileName* file contains all the information describing the database. The command will alter the information for the database concerning where its files are stored. If the data still resides on a drive but the drive has been relabeled on the new system then all that needs to be done is to issue the command for DB2 to recognize the database and be able to use it once DB2 UDB has been installed. The format of the configuration file is described below.

The **db2relocatedb** command configuration file has the following keywords:

- ▶ **DB_NAME**

The name of the database to be relocated.

- ▶ **DB_PATH**
The path where the database is stored. This is the root directory for the database.
- ▶ **INSTANCE**
The name of the instance that the database belongs to.
- ▶ **NODENUM**
The node number of the database. This is used only for partitioned databases.
- ▶ **LOG_DIR**
The directory where the log files are stored.
- ▶ **CONT_PATH**
This is a list of container paths that need to be changed. There has to be a separate entry for each container that is being changed in the database. If you are moving a large number of database containers then you should use a script or editor like 'vi' to generate a script to make the db2relocatedb configuration file.
- ▶ The parameters are set using the following format:
`parameter=oldValue,newValue`

Steps of using db2relocatedb

Following is the procedure for using the **db2relocatedb** command to migrate a database.

1. Install DB2 UDB on the target server with the same version and Fixpak as on the original server.
2. Stop DB2 on the original server.
3. Unplug the disks (or disk subsystem) from the source machine in a manner approved by the operating system. On Windows you should disable the disk using the Device Manager utility before unplugging it, in order to prevent system crash or data loss.
4. Enable the disks (or disk subsystem) on the target machine in an approved manner. On Windows you should use the Disk Management tool to rescan disks.
5. Verify that all disks are online and drive letters are properly assigned.
6. Run the db2relocatedb utility to modify the desired configuration on the target system. (Note: **db2relocatedb** will automatically catalog the database indicated in the configuration file.) Proceed through the normal performance tuning tasks to ensure that DB2 meets the performance expectations on the target server.

Example of relocation

A database is currently laid out on a Windows 2000 server. The DBA would like to move the database over to a new Windows 2003 server. The new server, however, has an additional drive on it, which means that all the drive letters on the disk array that was used for the database will have to be changed. The old server is laid out like Example 2-1.

Example 2-1 Relocate database old system configuration

```
database name = redDB
instance name = redInst
database path = e:
dms container=e:\redbook\data
database log path = f:\db2\redDB\logs
```

The configuration file for the **db2relocatedb** command would be as shown in Example 2-2.

Example 2-2 Relocate database new system configuration

```
DB_NAME=redDB
DB_PATH=E:,F:
INSTANCE=redInst
LOG_DIR="F:\db2\redDB\logs","G:\db2\redDB\logs"
CONT_PATH="E:\redbook\data","F:\redbook\data"
```

The information above (Example 2-2) would be placed into a file and then the following command would be run.

```
db2relocatedb -f configFile
```

```

C:\redbook\scripts>db2relocatedb -f relocateFile.txt
DBT1006N The file/device "C:\DB2\NODE0000\SQL00002\SQLLOGCTL.LFH" could not be opened.

C:\redbook\scripts>db2 terminate
DB20000I The TERMINATE command completed successfully.

C:\redbook\scripts>db2 ? dbt1006n

DBT1006N The file/device "<file-device-name>" could not be opened.

Explanation:

  An error occurred while attempting to open the specified file/device.

User Response:

  Ensure that the file/device exists and that its permissions are correct. Correct the problem and resubmit the command.

C:\redbook\scripts>db2relocatedb -f relocateFile.txt
Files and control structures were changed successfully.
DBT1000I The tool completed successfully.

C:\redbook\scripts>_

```

Figure 2-2 Executing the `db2relocatedb` command

Before you run the `db2relocatedb` command you have to make sure that there are no connections to the database and that it has not been activated. The use of the `db2relocatedb` command is illustrated in Figure 2-2. If there are any connections remaining open you can force them off using the DB2 *force* command. The command syntax is:

force application {ALL | (application-handle [{,application-handle} ...])}

You can also use a command that is new in V8.1, the Quiesce command. If you place a database into Quiesce mode then no further connections are allowed to the database until the mode is changed. This will ensure that other users do not connect to the database between when you force off the connections and when you start your database relocation. The Quiesce command has a built in force option, which will force all applications off the database.

The command syntax is:

quiesce database {IMMEDIATE | DEFER} [FORCE CONNECTIONS]

Example 2-3 Preparing a database to be relocated

```

db2 list applications
>> Output:
DREWKB  db2bp.exe      7          *LOCAL.DB2.00B988234903      NETEDIT  1

db2 quiesce database FORCE
>> DB20000I The QUIESCE DATABASE command completed successfully.
db2 list applications

```

```
>> SQL1611W No data was returned by Database System Monitor.  SQLSTATE=00000
db2 relocatedb....
db2 unquiesce database
```

2.1.2 Migrating between versions of DB2 UDB

Migrating between versions of DB2 UDB has been simplified with DB2 system commands that are included with DB2 UDB V8.1. The migration of your databases is required if you would like to continue using upgraded instances from DB2 UDB V6 or V7 with the V8.1 installation. There is a detailed description of the migration process in the following DB2 UDB V8 manual

PDF: *Quick Beginnings for DB2 Servers*, Chapter 5, “Migrating DB2 Servers (Windows)”

Restrictions

Before you begin your migration make sure that you review the following requirements:

- ▶ Migration is only supported from DB2 UDB V6 and V7.
- ▶ The migration command can only be issued from a DB2 UDB V8 client or server, not from a V6 or V7 client or server machine.
- ▶ You cannot migrate a database between platforms. Only existing Windows databases can be migrated on Windows Server 2003.
- ▶ Migrating a partition database requires that DB2 UDB V8.1 be installed on all the computers with the DB2 partitions on them first.
- ▶ Windows only allows one version of DB2 UDB on a machine. If you already have DB2 UDB V6 or V7 on the machine, then the V8 install process will uninstall the previous version.
- ▶ User objects cannot have V8 reserved schema names as qualifiers. These qualifiers include SYSCAT, SYSSTAT, and SYSFUN. It is not recommended that you use DB2INFO, since this schema is used by the DB2 UDB V8.1 Cube Views product.
- ▶ The database must also be in a stable state. It cannot be in any of the following states:
 - Backup pending
 - Roll forward pending
 - Tablespace(s) not in a stable state
 - Transactions inconsistent

The following are the steps for a database migration:

1. Back up.

Make sure to take a full off-line database backup before you begin the migration. Once a database is migrated to DB2 UDB V8.1, it cannot be migrated back to the old release. The backup will allow you to uninstall V8.1 and reinstall your previous release with the old database.

2. Check that you have sufficient space for the migration.

3. Record your database and instance configurations.

4. Check the diagnostic level.

It is recommended that you set the DB2 instance parameter `DIAGLEVEL` to 4 during the migration to ensure that all error messages are captured. Use this command to set the DB2 diagnosis level:

```
db2 update dbm cfg using diaglevel 4
```

5. Verify that the databases are ready to be migrated.

The DB2 command **db2ckmig** will check that your database is ready to be migrated. The command will check that:

- A database is not in an inconsistent state
- A database is not in a backup pending state
- A database is not in a rollforward pending state
- Tablespace are all in a normal state

Command syntax is like:

```
db2ckmig databaseAlias -1 logFileLocation
```

Note: In Windows, the **db2ckmig** command must be run before DB2 V8 is installed. **db2ckmig** can be found on DB2 CD-ROM or install image under the directory `images\DB2\Windows\utilities`.

6. Take the database system offline.

The entire DB2 system will have to be taken offline. This includes not only stopping the instance but also the license server.

- a. Stop the DB2 license server: **db2licd -end**.
- b. Force all applications off the databases: **db2 force applications all**.
- c. Stop the instance: **db2stop**.
- d. If the stop fails then you may have to force it: **db2stop force**.

7. Install DB2 UDB V8.1.

8. Migrate the database using the DB2 **migrate** command.

Migrating the database

The databases from previous versions are migrated using the DB2 command **migrate**. The command will only migrate a database to the latest version, and will not migrate databases back to previous releases. If an error occurs during the usage of a command, the DB2 **terminate** command will need to be used before the command is used again. Additional documentation can be found in the *IBM DB2 UDB Command Reference Version 8, SC09-4828*.

The command syntax is shown in Figure 2-3.

```
>>-MIGRATE--+DATABASE--+database-alias----->
      '-DB-----'

>--+-----+----->
  '-USER--username--+-----+'
                        '-USING--password-'
```

Figure 2-3 DB2 migrate command syntax

Example 2-4 Migrating a database

```
db2 migrate database redDB user redTest using passwordTest
>> Files and control structures were changed successfully.
>>DBT1000I The tool completed successfully.
```

2.1.3 Migrating from 32-bit to 64-bit

With enhancements made in DB2 UDB V8, the process to move DB2 UDB V8 from Windows Server 2003 32-bit OS to 64-bit Windows Server 2003 is simple. As discussed in “Using DB2 Backup and Restore” on page 34, the backup and restore process is not supported for major releases and bit-ness. Figure 2-1 on page 35 illustrates which combinations are supported.

If you are moving from V7.2 on Windows 2000 to V8.1 on Windows Server 2003 64-bit you should refer to the next section.

Note: It is now possible to restore from a V7 backup to V8. This restoration is only supported for the same bit level of the restore.

2.1.4 Migrating everything at once

It is definitely not recommended to try and move both your Windows release and your DB2 UDB version at the same time. It is, however, quite likely that you may be moving from Windows Server 2000 with DB2 UDB V7.2 to Windows Server

2003 64-bit to DB2 UDB V8.1 64-bit. Two different methods are given to illustrate different migration strategies.

Method one

This method will isolate the changes to the new target server and will leave your original server intact and operational. This method works well if you want to be able to easily fall back to your original server if there are any problems with the new server.

Steps

Following are the migration steps:

1. Install Windows 2003 64-bit on the new target server.
2. Install DB2 UDB V8.1 on the target server. Install the appropriate fixpaks if required.
3. Take an offline backup of the DB2 database on the original server.
4. Check that the database is ready to be migrated by using the **dbckmig** command.
5. Record all the DB2 instance and database configuration parameters.
6. Copy the DB2 backup image to the target server from the original.
7. Take the database system offline on the original server.
8. On the target server change the DB2 instance configuration parameter **diaglevel** to 4. This will ensure that a full amount of information on the migration is captured.
9. Restore the database image on the target server. The restored database will automatically be migrated; therefore, the DB2 **migrate database** command will not be required.

We have restored a V7 FP4 image from a Windows 2000 machine to a systems with Windows 2003 Datacenter running DB2 UDB V8.1 FP2 for 32-bit. The output is shown in Example 2-5.

Example 2-5 Restoring a DB2 UDB V7.2 image onto V8.1

```
db2 restore database sample
>>SQL2517W The restored database was migrated to the current release.
Migration returned sqlcode "0" and tokens "".
```

The output from the db2diag.log log file (which is in C:\Program Files\IBM\SQLLIB\instanceName') should be checked to ensure that the restore completed successfully.

Method two

The second method has the migration performed on the original server. This reduces the complexity of the data movement and allows Windows Server 2003 to be installed in parallel on the target machine. The method does, however, make it difficult to fail back to the old configuration since the installation of V8.1 on the original server will cause DB2 UDB V7 to be uninstalled.

Steps

The method two migration procedure is:

1. Take an offline backup of DB2 on V7 or V6 on the original server.
2. While working on the DB2 database backup, you could install Windows Server 2003 on the new target server.
3. Install DB2 UDB V8.1 on the target server. Install the appropriate fixpaks if required.
4. Check that the database is ready to be migrated by using the `dbckmig` command.
5. Record all the DB2 instance and database configuration parameters.
6. Install DB2 UDB V8.1 on the original server.
7. Migrate the database to V8.1 using the DB2 `migrate` command.
8. Take an offline backup of the database on the original machine on V8.1.
9. Copy the backup image to the target server.
10. Restore the backup on the target server.
11. Apply the database and instance configuration parameter from the original server to the target server.

2.1.5 Migrating back to 32-bit

On DB2 UDB Version 8, a database can easily be moved back from a 64-bit system to a 32-bit system. The 64-bit properties of the operating system are only used for referencing memory and have no impact on the physical storage of the database.

The database can be moved back to a 32-bit system by first taking a backup of the database on the 64-bit system. After the backup is completed, zip up and copy the backup image to the target 32-bit machine. The backup can then be restored. The version of DB2 UDB on the target machine has to be the same release. Restoring a database to a different Fixpak is supported, as is migrating to V8 from a different release.

2.1.6 Migrating back to DB2 UDB V7 or V6

Once a database has been migrated to DB2 UDB V8, it is not possible to migrate the database back. It is actually a complex task to even move the database layout back to V7. The tools that are normally used to move a DB2 database between different operating systems, such as *db2move* and *db2look*, will capture the new layouts of the tablespaces. It is therefore quite important to first take a backup of the database before you migrate to a new release of DB2 UDB.

If you do uninstall DB2 UDB V8.1 and wish to reinstall DB2 UDB V7.2 and restore your old backup, make sure you clean up the V8.1 instance. You can use the **db2idrop** command to drop the instance before uninstalling DB2. When the DB2 UDB product is uninstalled it does not remove any of the instances or databases. If you have created the instance in the default location or have the same instance or database file location for both V7.2 and V8.1, an error will occur. This is due to DB2 UDB V7.2 trying to read the instance information for a V8 instance. The new versions have different information for the instances and databases, which leads to the error. You will have to completely delete all of the files for the V8.1 instances and databases after you uninstall or drop the instances and databases first before the uninstall.

2.2 Installation considerations

Before you begin installing DB2 UDB V8.1 on Windows Server 2003 it is important to consider a number of key points. It is also important that the installation Readme.txt file is read before you begin since the requirements do change from release to release.

2.2.1 Installation requirements

The installation is a simple process that requires almost no knowledge of DB2 UDB. You will, however, need the required permissions on the operating system for installing new software. A complete description of the server installation requirements can be found in the DB2 UDB documentation:

IBM DB2 UDB Quick Beginnings for DB2 Servers V8, GC09-4836, Chapter 7, "Installing DB2 Servers (Windows)"

The major requirements are summarized below.

Administrator rights

The user that you install DB2 UDB with must either be the administrator on the local system or domain, or have administrator security access. The user must also have the following security privileges: Act as part of the operating system.

For the details of setting up a user ID or verifying user rights, please refer to 2.4.3, “Creating an Instance” on page 67.

Disk space

During the installation process a message will indicate exactly how much disk space is required for all the DB2 components you have selected to install. You will, however, need to have sufficient space to unzip the installation files and then install DB2 UDB itself. You should therefore make sure that you have at least 450 MB for the zipped file plus the minimal space listed in Table 2-1.

Table 2-1 Disk space required

Installation type	Minimum disk space
Typical	350 MB
Custom	100 MB

Memory

DB2 UDB requires a minimum of 256 MB to run. Additional memory will be required for any database clients running locally on the system. You should also make sure to leave enough free memory for the operating system and any additional programs running on Windows or with DB2 UDB.

Readme information

Included with the unzipped DB2 UDB installation information is a ReadMe.txt file that is specific to each of the languages that DB2 UDB can be installed with. This file should definitely be read before you begin as it contains any updates to the installation process or new/removed installation prerequisites.

Windows domain

If you are planning on using DB2 UDB ESE's Partitioning Database Feature, it is required that the system be a member of a domain. All of the machines that you want to use for clustering must also be part of the same domain. The setup and configuration of the Domain Controller or domain is covered in detail in 4.2.2, “Windows Server 2003 Domain Controller setup” on page 184.

Online documentation

Web-based support for DB2 UDB products and Fixpaks can be found at the address below. Information on APARs can also be found there.

<http://www.ibm.com/software/data/db2/udb/winos2unix/support>

2.2.2 64-bit and 32-bit installation images

With the release of DB2 UDB V8.1 Fixpak 2, there are now two different images for the installation of DB2 UDB on Windows. For Windows Server 2003, DB2 UDB now has a 64-bit version that can run on 64-bit hardware, and also a 32-bit version.

Installation images

Unlike many 32-bit applications, the 32-bit version of DB2 UDB cannot be installed on the 64-bit release of Windows. This is due to the large amount of optimization of the DB2 UDB code for the specific functionality of the 32-bit and 64-bit versions of Windows Server 2003. The 64-bit DB2 UDB installation image can only be installed on the 64-bit release of Windows, and the 32-bit DB2 UDB image can only be installed on the 32-bit version of Windows.

Identifying the image

With the initial release of DB2 UDB V8.1 on Windows Server 2003 for 64-bit there is not a way to differentiate the zip files for 64-bit and 32-bit. The 64-bit DB2 UDB Fixpak 2 image is a full install for V8.1 since there was previously no 64-bit release for Windows. In the future the fixpaks for 32-bit and 64-bit DB2 UDB for Windows will most likely be identified by a '_32' and '_64' indicator on the zip file. If you are unsure which release of DB2 UDB for Windows you are installing then you will have to attempt the installation and see if a warning message is returned.

Incorrect image installation

If you try to install the incorrect image of DB2 UDB on the server then a warning message will appear during the beginning of the installation.

The 64-bit installation image can only be installed on the 64-bit version of the server. If you try to install the 32-bit release then you will receive the error message below (Figure 2-4 on page 49).

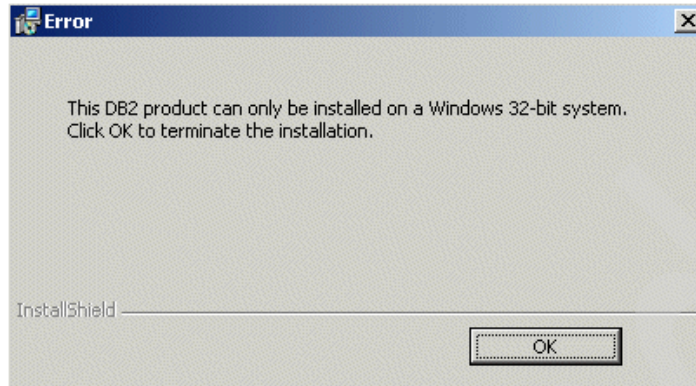


Figure 2-4 32-bit DB2 UDB image installation on 64-bit Windows error message

A similar message will be returned when the 64-bit version is installed on a 32-bit Windows Server 2003.

2.2.3 Fixpaks

A great deal of new functionality and improvements were built into DB2 UDB V8.1. IBM was not, however, able to get everything in that they wanted, and this extra functionality is then released in fixpaks before the next major release.

Tip: The odd-numbered fixpaks are designed to correct any problems that have been found, while the even numbered fixpaks offer new functionality in addition to improvements.

Required Fixpak level

There is a minimum level of DB2 UDB that is required for the product to be supported on Windows Server 2003. For DB2 UDB V8.1, the required Fixpak level is at least Fixpak 2. For DB2 UDB V7.2 the Fixpak level required is Fixpak 4 or later. It is recommended that the latest Fixpak be installed to ensure that all of DB2's new functionality is available and any known issues have been corrected.

Identifying the Fixpak

The level of the Fixpak can be identified by looking in the Fixpak release notes of the unzipped Fixpak image. Within the root directory of where the image is unzipped there will be a file, FixpakReadme.txt. At the top of the file will be a header that indicates which Fixpak it is. The DB2 command **db2level** can be used to determine your current DB2 UDB Fixpak level.

It is a good idea to read the Fixpak readme, as it identifies any new features or requirements that may have been added to the product since the last release. The current schedule for releases with DB2 UDB V8.1 is to have new functionality added only on even-numbered Fixpaks and any APARs or improvements added in the odd-numbered Fixpaks.

Identifying APARs

In addition to the adding of new database features, there are often partner and customer problems that have been resolved and added into the code. A list of the issues that have been corrected is included in the files APARList.txt and APARList.html. A complete list can also be found online at:

<http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/aparlib.d2w/report>

2.3 32-bit installation

The installation of DB2 UDB on 32-bit is executed using a simple GUI interface. One of the main changes from DB2 UDB V7 to DB2 V8 was that the documentation is now contained on a separate CD and must be installed separately.

2.3.1 Existing references

The IBM Redbook, *DB2 UDB Exploitation of the Windows Environment*, SG24-6893, contains a chapter that focusses on the installation of the product on 32-bit. Chapter 2, “Installation and deployment”, goes into detail on all the steps necessary to install DB2 UDB on Windows 2000. The installation procedure can be used for installing DB2 UDB on Windows Server 2003.

The online documentation included with DB2 UDB also has detailed instructions on how to install DB2 UDB on Windows. The documents can be found in the following source:

PDF: *Quick Beginnings for DB2 Servers*, Chapter 7, “Installing DB2 Servers (Windows)”

DB2 UDB V7.2 is also supported on Windows Server 2003, but its installation will not be covered in this publication. Further documentation on the installation process can be found in the DB2 UDB V8.1 manual *Server Quick Beginnings*.

2.4 64-bit installation

This section covers how to install DB2 UDB V8.1 on Windows Server 2003 for 64-bit. The edition of Windows Server 2003 used for the installation demo was Datacenter, but the installation steps will be the same for all supported editions. *DB2 UDB Exploitation on the Windows Environment*, SG24-6893, and the *Quick Beginnings for Server* documents cover the installation process in more detail.

2.4.1 Installation steps

Before you begin, there are a number of steps that you should check to make sure the installation goes smoothly.

Checking for 64-bit

In order to determine the maximum amount of memory that can be available for the system, and therefore your database, you need to know if the release is 64-bit. The normal method to determine the specification of the system is to use the Properties tab from the My Computer icon, or right click **My Computer** then select **Properties** (Figure 2-5 on page 52). The information provided in this window does not, however, indicate if the system is 64-bit.

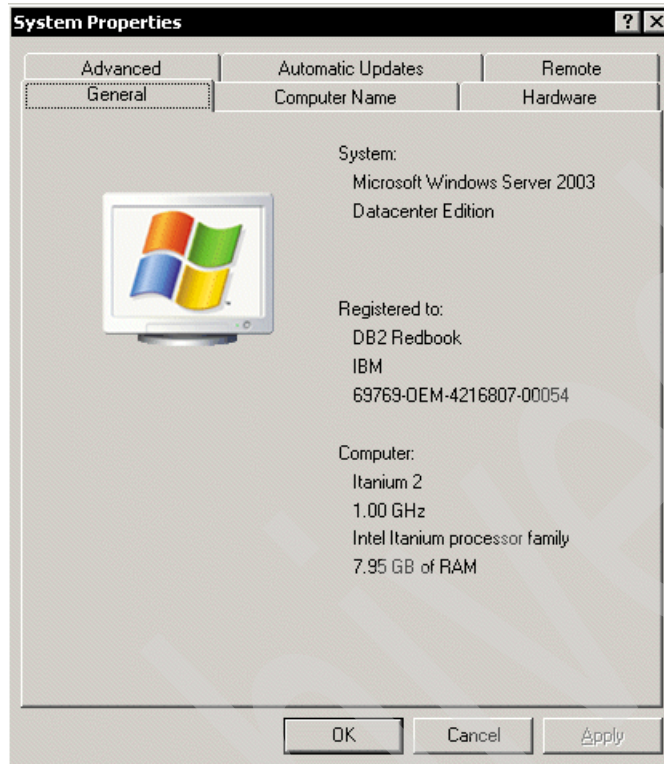


Figure 2-5 My Computer's properties information for Windows Server 2003

From the Windows command line you will have to issue the command **winver**. The program screen will indicate which edition of Windows Server 2003 is installed on the machine and also if the release is 64-bit (see Figure 2-6 on page 53).



Figure 2-6 Windows Server 2003 64-bit indicator

DB2 UDB product type

After you have determined if the system is 64-bit, you can install the appropriate image of DB2 UDB. Make sure that you are installing the 64-bit version on DB2 UDB; otherwise an error will occur (see Figure 2-4 on page 49).

There are also multiple versions of the DB2 UDB product that can be installed. The main installation CD contains the Enterprise Server Edition (ESE). There are also additional CDs that contain the following clients:

- ▶ Administration client
- ▶ Application development client
- ▶ Runtime client
- ▶ DB2 Connect client

Only the installation of the ESE version of DB2 UDB will be covered since this is what is typically installed on the server versions of the Windows operating system.

All of the different editions of DB2 UDB are built on the exact code base, but some do have built-in license limitations on the number of CPUs that they support (see Table 2-2 on page 54). Make sure to choose the one that best fits your business need.

Table 2-2 CPU limits for DB2 UDB editions

DB2 UDB Edition	Maximum CPUs
Express	2
Workgroup	4
Enterprise	Unlimited

Starting the installation

The installation is performed using a Windows Installer GUI interface that is called by the setup program. The program can be found on the CD in the \ESE\image\... directory. The setup process will automatically start when the CD is loaded or can be started if you double click the **setup.exe** icon.

Installation type

There are three main installation options available (Figure 2-7 on page 55):

- ▶ Typical

A standard set of DB2 options are selected.

- ▶ Compact

This option will only install the minimum components needed to run the DB2 server. If you are concerned about the amount of space available where you are installing DB2 UDB then this is a good option to select.

- ▶ Custom

The custom option allows you to pick which components and functionality of DB2 you would like to install. This option is only recommended for experienced DB2 DBAs who understand which components are essential for their planned use of DB2 UDB.

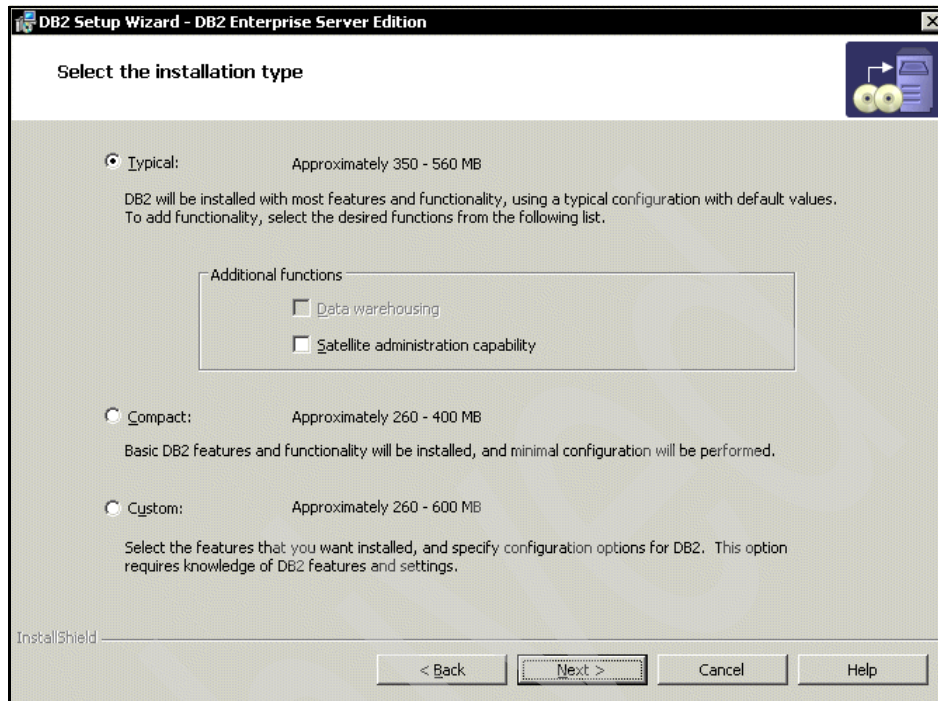


Figure 2-7 DB2 installation options

Response file

DB2 UDB has long had an option for automating the installation process, but not a lot of people have known about it or utilized it. The next section, “Why use a response file” on page 61, covers how to use a response file in detail. This option can be quite useful in reducing the amount of work to deploy DB2 UDB on multiple systems or in a clustered environment.

During installation, if you select **Save your settings in a response file** (Figure 2-8 on page 56), DB2 will create the response file for you with all the options you specified during the installation. You also can find the sample response file in the `db2/sqlib/sample` path. For more information on how to work with response files, please refer to the *DB2 Installation and Configuration Supplement* or *DB2 UDB Exploitation of the Windows Environment*, SG24-6893.

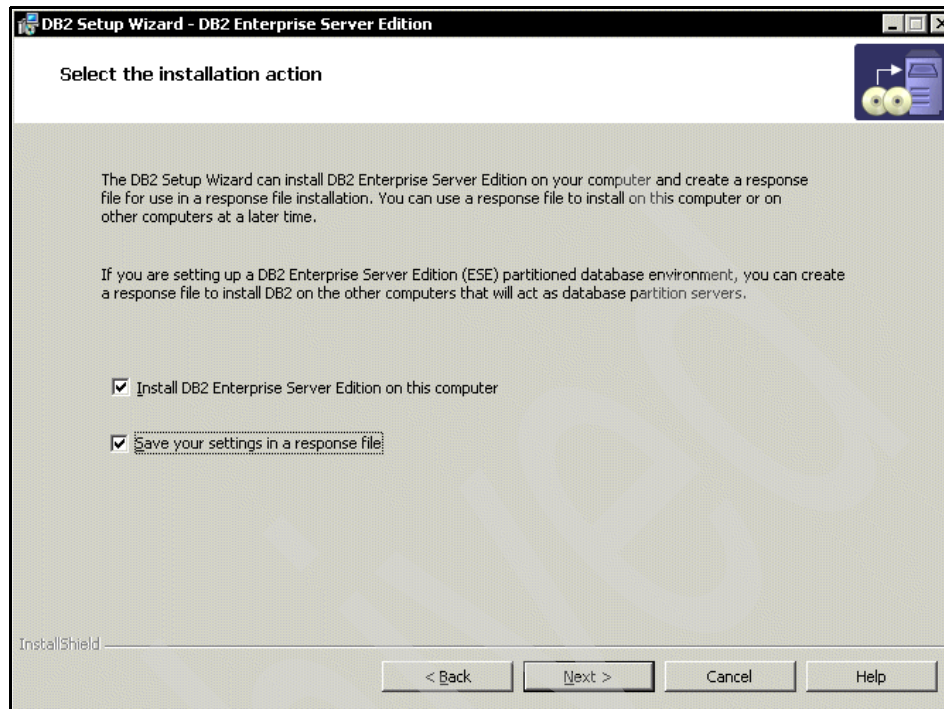


Figure 2-8 Response file selection during DB2 installation

DB2 UDB partitioning options

Within DB2 UDB ESE there is an option that allows DB2 to both logical partition data across a single SMP system and also to physically partition it across multiple SMP systems (MPP). This option is activated by an optional license required for partitioning. If the license has been purchased then during the installation there will be a screen asking you if you would like to set up the DB2 instance for partitioning and how it should be configured. The options for partitioning will be covered in detail in 4.3.2, “Installing the DB2 Instance Owning Node” on page 196.

Choosing components

The custom installation option allows you to choose the individual DB2 components that you wish to install (see Figure 2-9 on page 57). These components are covered in detail in *DB2 UDB Exploitation of the Windows Environment*, SG24-6893, in Chapter 2, “Installation and deployment”.

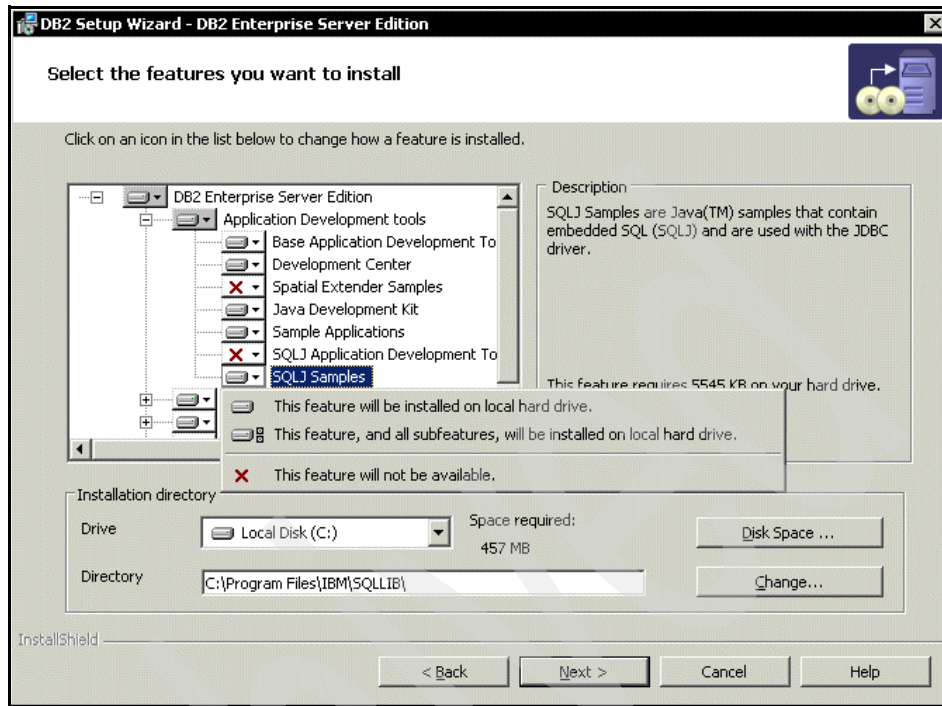


Figure 2-9 DB2 installation component selection

DB2 Administration Server

The DB2 Administration Server (DAS) was an instance but now is a Windows service. It is used to support local and remote administration. You must have a running DAS if you want to use available tools like the Configuration Assistant, the Control Center, or the Development Center. It is used to control information concerning all the instances on the system. If you want to be able to use the DB2 GUI tools then the DAS instance must be running on your system. Only one DAS is required to be running on each system even if multiple instances are running on the systems. During installation, DB2 will also install the DAS for you if requested (see Figure 2-10 on page 58).

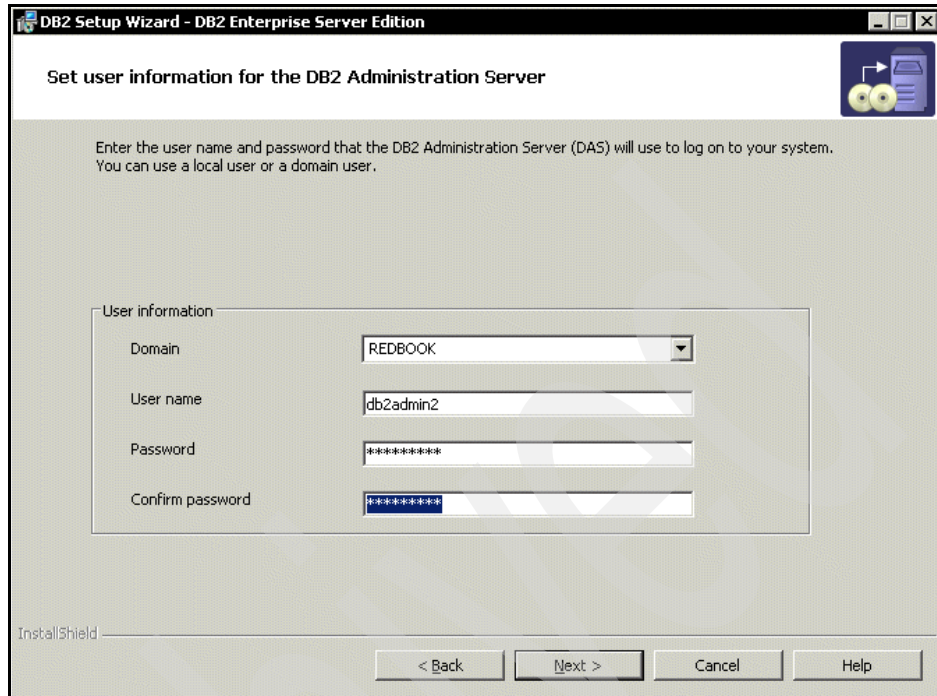


Figure 2-10 DB2 Administration Server (DAS) installation options

Setting up the contact list

The contact list is used by the DB2 Health Center tool, which is included in DB2 UDB V8. The Health Center notifies those on the contact list about any problems the database is experiencing or if any events specified by the user are occurring. The contact list can be specified later after the install is finished (see Figure 2-11 on page 59).

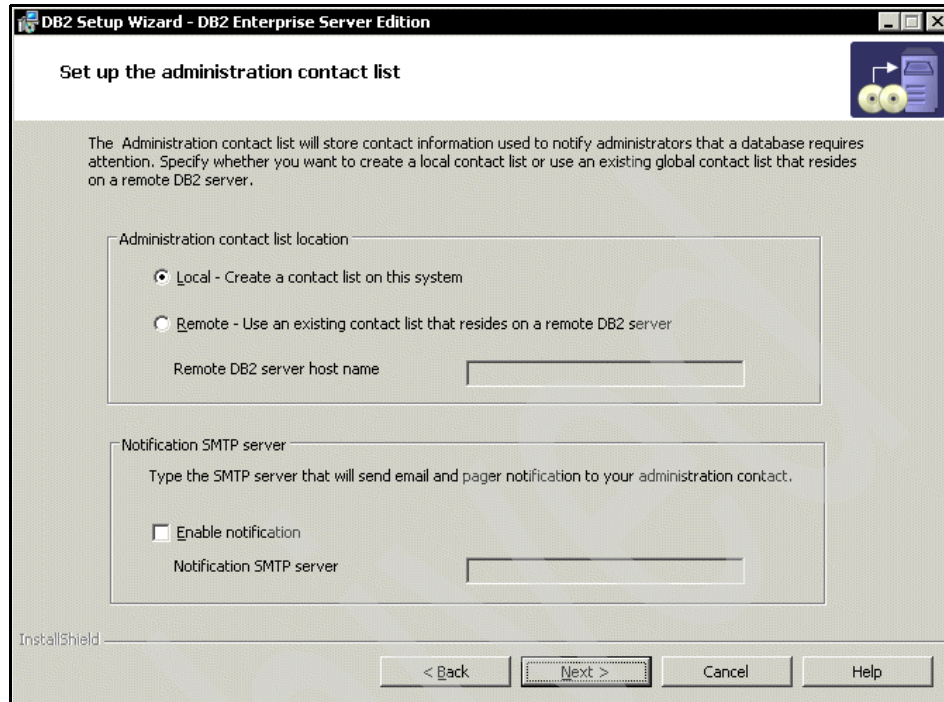


Figure 2-11 DB2 contact list information

Instance options

The default instance in Windows is automatically called DB2. You are able to alter the network communication settings for the instance and whether it automatically starts when Windows is booted up.

The default port used for TCPIP communication for DB2 is 50000. You can change the port to a different value but you have to make sure that it does not conflict with any ports that are already in use. The list of allocated ports is found in the services file. A description of setting the port numbers for DB2 in the services file is found in “Altering the Windows services file” on page 72.

Instance user account

Each DB2 instance has an operating system level user associated with it as the instance owner. The install program will automatically create the user and assign it all the required privileges if the requested instance owner does not exist. If, however, the user does exist then you may receive an error if it does not have all the required operating system privileges. You will most likely receive an error message indicating that the user ID does not have sufficient privileges to authenticate the user privileges. Do not worry, the installation will still complete.

The error message can be removed by assigning the required privileges to the user you would like to use as the instance owner. The steps for this are covered under in 2.4.3, “Creating an Instance”, which describes how to manually create a DB2 instance on Windows.

Port range

The port range options are used by a partitioned database. A port number is required for each partition that will be included in the cluster. This will be covered in detail in 4.3.4, “Verifying DB2 partitioning configuration” on page 206.

Tools database

The tools database is a new feature of DB2 UDB V8.1. The tools database is created to store all of the information used by the DB2 Task Scheduler. This utility is integrated into all the DB2 wizards and utilities to allow a DBA to save their actions and execute them later or at scheduled intervals.

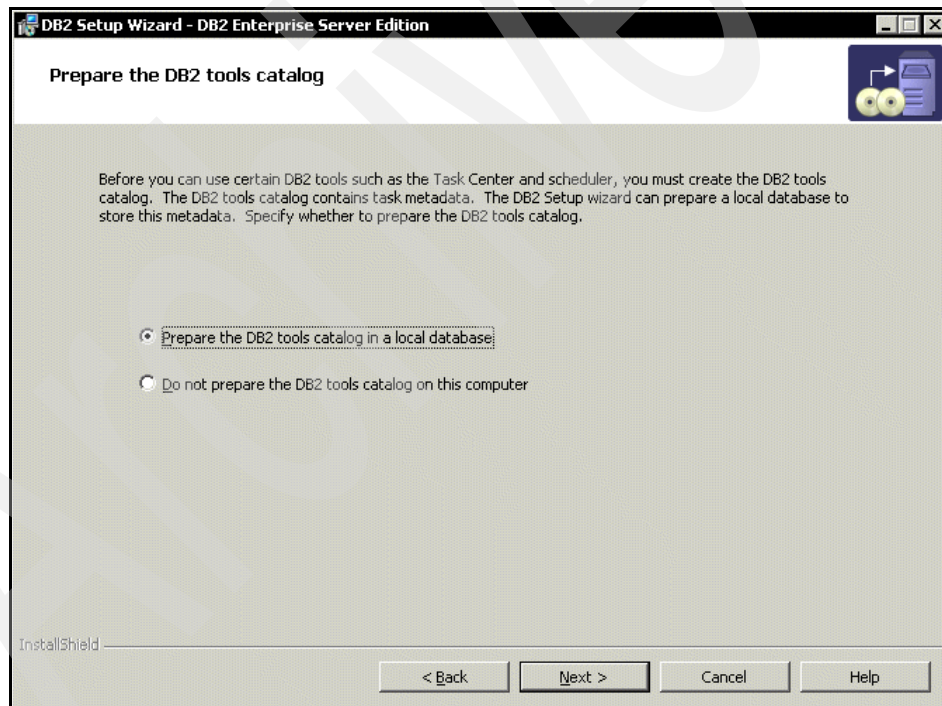


Figure 2-12 DB2 UDB V8.1 tools database options

It is a good idea to create the tools database (Figure 2-12) since it does not take up much space and the functionality it then offers greatly reduces the amount of administration that needs to be executed by a DBA.

2.4.2 Installing DB2 UDB using a response file

In addition to a GUI-based install, DB2 UDB can be installed using a user-defined script. The installation using this method is referred to either as a *silent install* or *response file install*. It offers great benefit to administrators who want to be able to automate their software installation or deploy identical installations out to many similar machines.

The DB2 response file installation is a background installation process that you can embed into an existing installation program, and it is transparent to end users. It is performed by DB2 setup utilities with user-generated response files. The response files are ASCII text files that contain setup and configuration information, which can be generated either with DB2 utilities or by manual editing.

The majority of the information on the silent install for DB2 UDB V8.1 was provided by Michael Gao from an IBM White paper he wrote, *Install IBM DB2 Universal Database Express Edition Transparently with Your Application*. This document can be found at the Web site:

<http://www7b.boulder.ibm.com/dmdd/library/techarticle/0306gao/0306gao.html>

Why use a response file

Unlike standard DB2 setup utilities, a DB2 response file installation lets you install DB2 UDB without interactive user input. This is useful not only for large-scale deployments of DB2 UDB, but also for embedding the DB2 installation process transparently inside your own installation program. The response file installation is one of the reasons why DB2 provides a complete solution for your embedded database needs.

Generating a response file

The first step of the response file installation process is to generate a response file. When installing DB2 UDB using the interactive setup utility, you are prompted to make installation choices and provide necessary configuration data before installation takes place. Response files simply contain these installation choices and configuration data, which are already prepared for embedded use.

There are three methods to generate the responses files. All of them apply to both DB2 UDB Server and DB2 Runtime Client installations.

- ▶ Using the DB2 Setup wizard
- ▶ Using the DB2 response file generator utility (Windows only)
- ▶ Generating the response file manually

Example 2-6 Sample of a partial response file

```
PROD=ENTERPRISE_SERVER_EDITION
```

```
INSTALL_OPTION=SINGLE_PARTITION  
LIC_AGREEMENT=ACCEPT  
FILE=C:\Program Files\IBM\SQLLIB\  
INSTALL_TYPE=CUSTOM
```

```
COMP=DEVELOPMENT_CENTER  
COMP=EVENT_ANALYZER  
COMP=JDBC_SUPPORT  
COMP=IBM_JDK  
COMP=IBM_JRE  
COMP=ODBC_SUPPORT  
COMP=OLE_DB_SUPPORT
```

A complete sample of a response file can be found on the installation CD at x:\db2\windows\samples, where x: is the drive letter for the CD-ROM.

Creating a response file using the GUI install

There is a very easy method to create a response file without having to learn all its syntaxes. The GUI install has an option that will record all of your selections from the installation process (see Figure 2-13 on page 63). You can then use this file for identical installations on other systems.

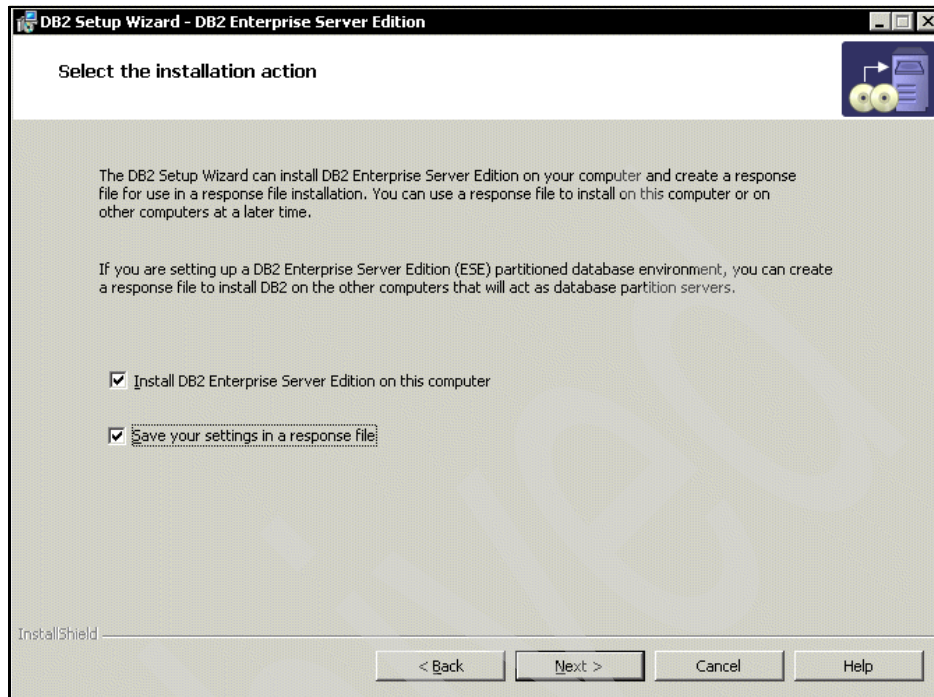


Figure 2-13 Response file installation option

You must manually modify the generated response files to accept the license agreement before they are usable.

Using the DB2 response file generator

In addition to the installation option for response file generation, there is a text utility, which will also create a response file. The purpose of this utility is for the installation and tuning of the instance to be completely first and then the response file is created. Figure 2-14 shows the syntax of the response file generator command **db2rspgn**.

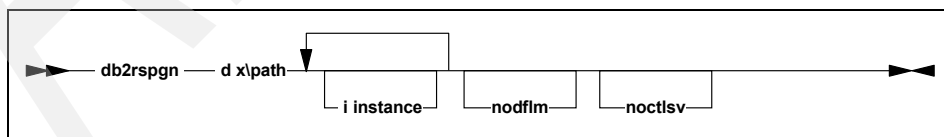


Figure 2-14 db2rspgn utility command syntax

By default, **db2rspgn** generates profiles for all instances on the master installation along with the response file. You can choose not to create profiles for the Data Links File Manager (DLFM) instance and control server instance by using the

-nodflm and -nocltsv options, respectively. You can also specify the individual instances for profile creation. The profile files are put in the same directory as the response file.

You must do some minor editing on the generated response files before they are usable:

- ▶ Manually modify the file to accept the license agreement.
- ▶ Enter the user ID and password for the new systems.

Manually generating a response file

Because the response files are ASCII text files, you can create them manually. You can choose to modify the sample response files provided with the DB2 UDB image or to type it from scratch. This method requires a solid understanding of the response file structure and keywords. Fortunately, the sample response files are well commented and self-explanatory.

Here are brief descriptions of some of the keywords that you specify in the response file:

▶ PROD

Specifies the product that you want to install:

- UDB_EXPRESS_EDITION for DB2 Express
- ADMINISTRATION_CLIENT for the DB2 Administration Client
- APPLICATION_DEVELOPMENT_CLIENT for the DB2 Application Development Client
- RUNTIME_CLIENT for the DB2 Run-Time Client

▶ FILE

Specifies the destination directory for a DB2 UDB product (for Windows only).

▶ INSTALL_TYPE

Specifies the type of installation: COMPACT, TYPICAL, or CUSTOM.

▶ COMP

Specifies the components that you want to install. This keyword is ignored unless your INSTALL_TYPE is CUSTOM.

▶ LANG

Specifies a language whose support you want to install.

▶ REBOOT

Specifies whether to restart the Windows system when the installation has completed.

► KILL_PROCESSES

Specifies whether the installation will terminate the currently running DB2 processes without prompt (for Windows only).

Other options include those for:

- DB2 Administration Server settings
- Instance settings
- Database settings

Installing using the response file

After you generate the response files, installation is straightforward. Use the same executable that you used in the GUI installation. However, instead of typing the command only at the prompt or double-clicking `setup.exe`, you must provide the proper response file name. You can also specify other options such as log file path and name (Figure 2-15).

```
setup [/f] [/I <language>] [/l <log file>] [/u <response file>] [/t  
<trace file>]  
/? - generates this usage message  
/f - forces any db2 processes to stop before installing  
/i - two letter code for the preferred language  
/l - full path and name of the log file  
/u - full path and name of the response file  
/t - creates a file with install trace information
```

Figure 2-15 Setup command response file options

The setup utility will return an error code upon completion of the installation. A program can be set up to trap these messages if you are embedding the installation. Table 2-3 describes the common error codes that are returned. Table 2-4 on page 66 details the codes that are less commonly seen.

Table 2-3 Primary return codes from the response file

Return Code	Description
0	Action completed successfully.
1	Action returned a warning.
1603	A fatal error occurring during the installation.
3010	Install was successful; however, a reboot is required to complete the installation.

Table 2-4 Additional return codes from the response file

Return Code	Description
3	The path was not found.
5	Access was denied.
10	An environment error occurred.
13	The data is invalid.
87	One of the parameters was invalid.
1602	The installation was cancelled by the user.
1610	The configuration data is corrupt.
1612	The installation source for this product is not available. Verify that the source exists and that you can access it.
1618	Another installation is already in progress.
1622	There was an error opening the installation log file.
1632	The Temp folder is either full or inaccessible. Verify that the Temp folder exists and that you can write to it.
1633	This installation package is not supported on this platform.
1638	Another version of the product is already installed. Installation of this version cannot continue.
1639	Invalid command line argument.

Creating the databases

The response file also allows you to create new databases during installation. The simplest way to add a database is to manually add a section to your response file. The options for adding a database are detailed in Example 2-7.

Example 2-7 Creating database in response file

```
DATABASE = MY_DB
MY_DB.INSTANCE = DB2INST1
MY_DB.DATABASE_NAME = MY_DB1
MY_DB.LOCATION = LOCAL
MY_DB.ALIAS = MY_DB1
MY_DB.USERNAME = username
MY_DB.PASSWORD = password
```

2.4.3 Creating an Instance

An instance can be created during and after the installation of DB2 UDB. It is recommended that you create the instance during installation since the process is much easier.

Using a GUI

In DB2 UDB V8.1 the only GUI available to create an instance is the install process. The instance will be created automatically when you install DB2 UDB on Windows Server 2003. There are currently no wizards available to create an instance after the installation has completed with DB2 UDB V8.1 for Windows.

Using the DB2 system commands

The command line tools allow you to create an instance without having to rely on any GUI tools, and give you the ability to build the instance creation into your scripts.

Authority needed

All of the instance creation commands require that the user be the Windows administrator or have administrator privileges for the OS.

Creating a new user

A user will have to be created before the instance is created. This user will then be assigned as the instance owner and will have instance administrator (SYSADM) privileges.

DB2 uses operating system level users so you can leverage the wizards and tools included with Windows. From the Windows Program menu select the program **Active Directory Users and Computers**, as shown in Figure 2-16.

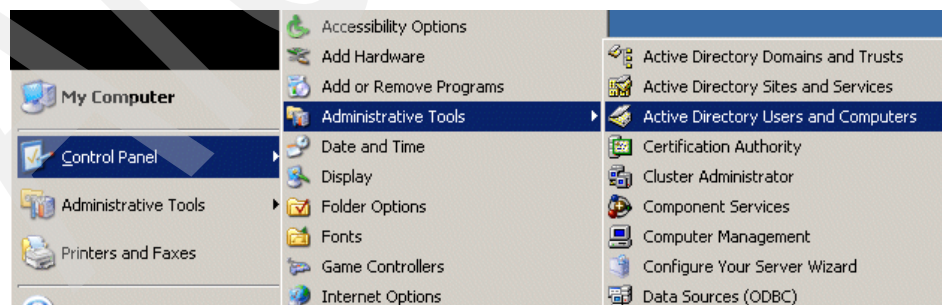


Figure 2-16 Active Directory program location

Within the Active Directory program there is a “User” folder that lists all of the users in the domain. When you create your instance you can use one of these users as the instance owner. A new user can also be created and become the new instance owner. Right clicking the **User** folder will give you the option to add a new user (Figure 2-17).

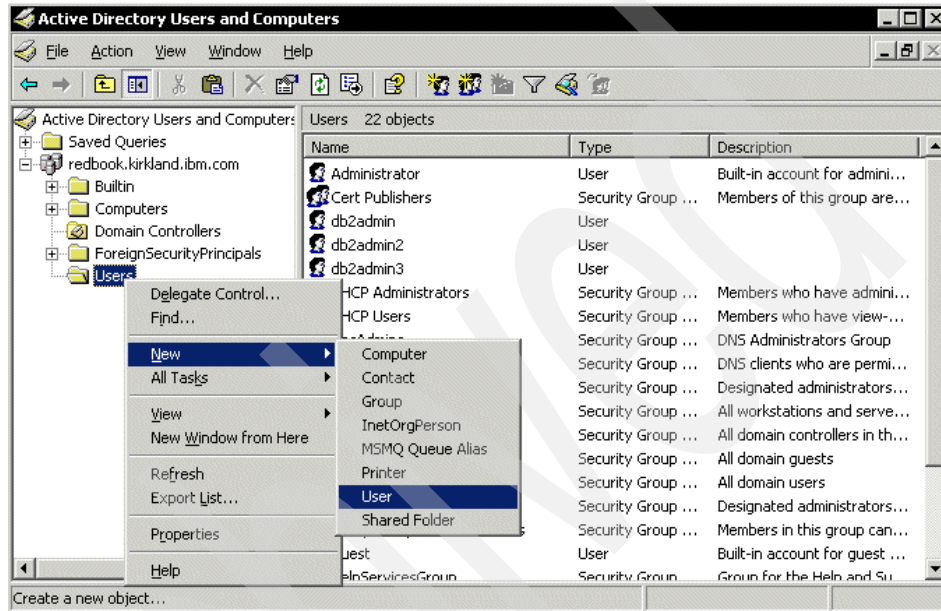
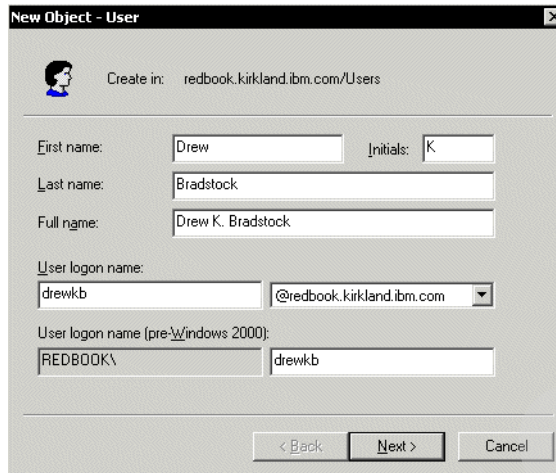


Figure 2-17 Adding a new user

When you create the new user it must be part of the domain if you are going to be using the instance as a partitioned database. The list of available domains is listed in the user specifications (Figure 2-18 on page 69).



New Object - User

Create in: redbook.kirkland.ibm.com/Users

First name: Initials:

Last name:

Full name:

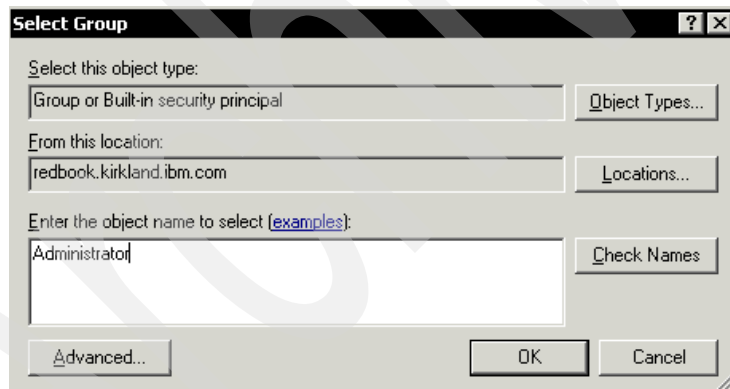
User login name:

User login name (pre-Windows 2000):

< Back Next > Cancel

Figure 2-18 User specifications

The user must also be added to the local Administrator group on the server (Figure 2-19). This will allow it to have the necessary privileges to create the instance and add the instance service to Windows required for it to run.



Select Group

Select this object type:

From this location:

Enter the object name to select (examples):

Figure 2-19 Adding a user to the Administrator group

The user will also require a number of security permissions. These can be added using the Windows Domain Controller. The Domain Controller can be found in the Program menu in the Administrative Tools folder (Figure 2-20 on page 70).

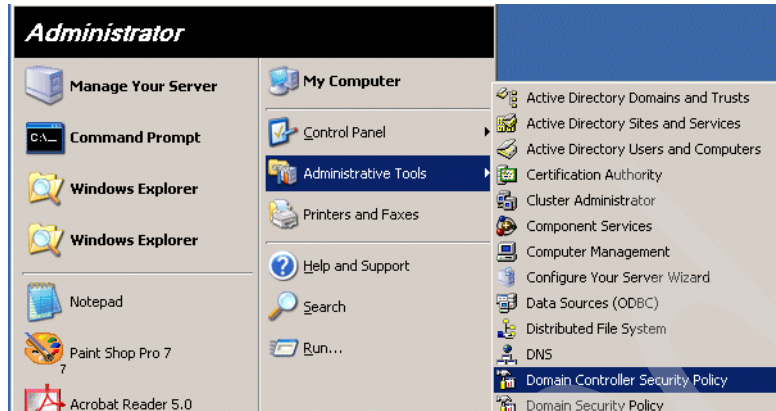


Figure 2-20 Domain Controller program location

Within the Domain Controller menu under Local Policies, the section for User Rights Assignments allows you to select which user right you would like to alter (Figure 2-21).

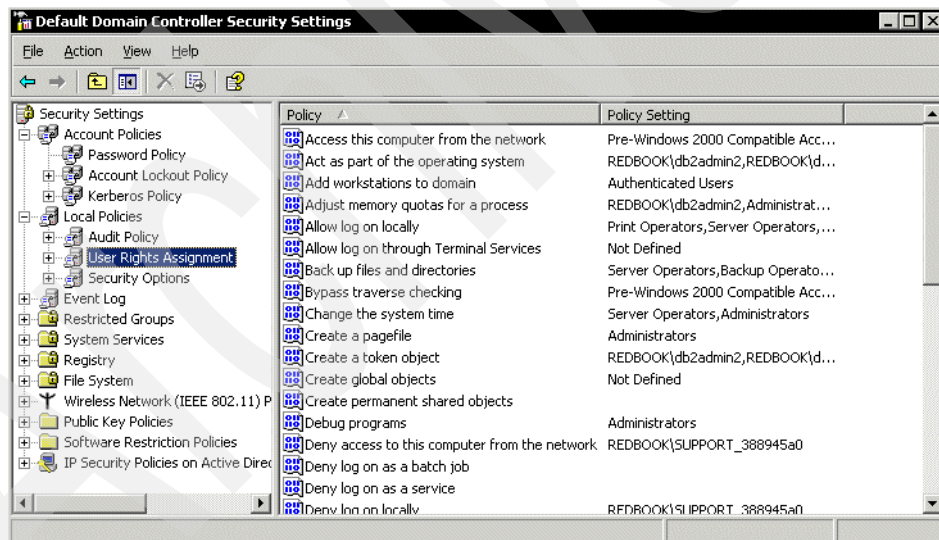


Figure 2-21 Selecting user rights in the Domain Controller

In the list of privileges, select the following list of authorities and add them for the instance owner.

- Act as part of the operating system
- Create token object
- Log on as a service

- ▶ Increase quotas
- ▶ Replace a process level token

Figure 2-22 shows multiple DB2 IDs with the privilege to “Act as part of the operating system”. During the testing for this publication we had added multiple users and given them all this privilege.

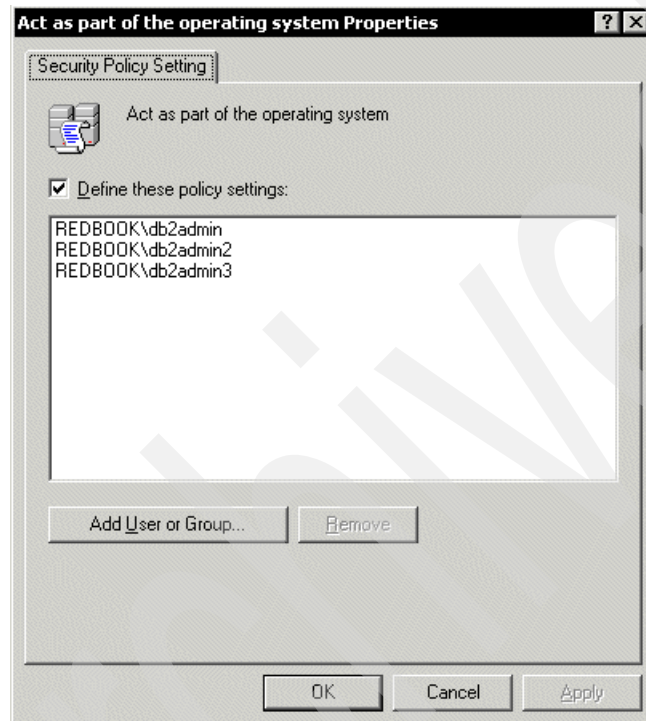


Figure 2-22 Adding a user's rights

Within the Domain Controller menu, the list of Windows Services for the system are listed (Figure 2-23 on page 72). You can control if you would like the instance to automatically start on Windows boot-up and also which ID the service should run under.

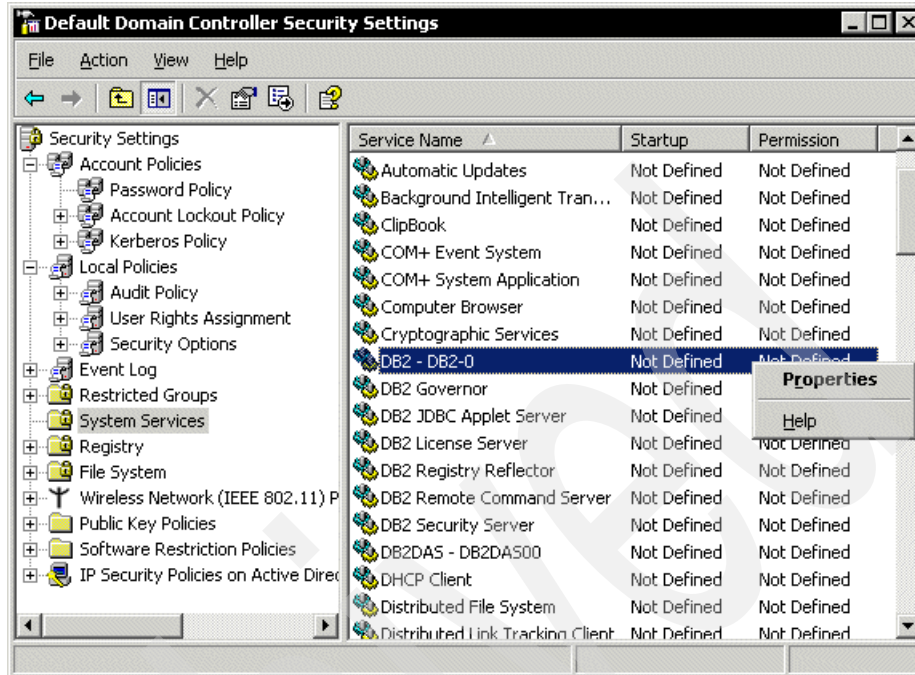


Figure 2-23 Services option with the Domain Controller

Creating the instance

The instance creation command **db2icrt** can be found off the drive where DB2 UDB was installed on the `sqllib\bin` directory. The command allows you to create a new DB2 instance and specify all of the initial settings. A complete explanation of the command can be found in the *DB2 Command Reference* manual.

Figure 2-24 shows the command syntax.

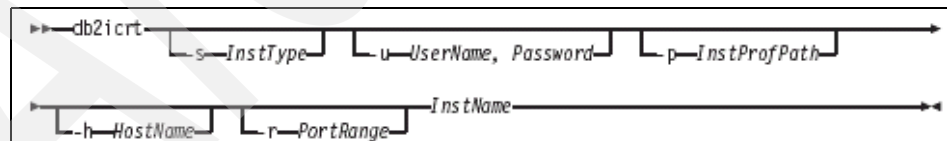


Figure 2-24 db2icrt command syntax

Altering the Windows services file

When a new instance is created using the **db2icrt** command, the Windows Server's services file is updated automatically. The services file can, however, be edited manually and is found in the `C:\WINDOWS\system32\drivers\etc\services` directory. In previous releases of Windows the file was found in the `C:\WINNT\system32\drivers\etc` directory (Figure 2-25 on page 73).

```

c:\ DB2 CLP
C:\WINDOWS\system32\drivers\etc>type services
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This file contains port numbers for well-known services defined by IANA
#
# Format:
#
# <service name> <port number>/<protocol> [aliases...] [#<comment>]
#
echo                7/tcp
echo                7/udp
discard            9/tcp      sink null
discard            9/udp      sink null
systat             11/tcp      users
systat             11/tcp      users          #Active users
daytime            13/tcp
daytime            13/udp
qotd               17/tcp      quote
qotd               17/udp      quote          #Quote of the day
chargen            19/tcp      ttytst source  #Character generator
chargen            19/udp      ttytst source  #Character generator
ftp-data           20/tcp
ftp                21/tcp      #FTP, data
telnet             23/tcp      #FTP, control

```

Figure 2-25 Windows Server 2003 services file

It is important that you follow the syntax of the services file exactly and use TABs instead of spaces to separate the different columns. The file is sensitive to how the text is laid out and an incorrect service file entry can lead to connectivity problems that will be difficult to debug later on (Figure 2-26).

```

c:\ DB2 CLP
ms-sql-m           1434/tcp      #Microsoft-SQL-Monitor
ms-sql-m           1434/udp      #Microsoft-SQL-Monitor
wins               1512/tcp      #Microsoft Windows Internet
wins               1512/udp      #Microsoft Windows Internet
Name Service
ingreslock         1524/tcp      ingres
l2tp               1701/udp      #Layer Two Tunneling Protoco
l
pptp               1723/tcp      #Point-to-point tunnelling p
rotocol
radius             1812/udp      #RADIUS authentication proto
col
radacct            1813/udp      #RADIUS accounting protocol
nfsd               2049/udp      nfs
knetd              2053/tcp      #Kerberos de-multiplexor
man                9535/tcp      #Remote Man Server
DB2_DB2            60000/tcp
DB2_DB2_1          60001/tcp
DB2_DB2_2          60002/tcp
DB2_DB2_END        60003/tcp
db2c_DB2           50000/tcp
C:\WINDOWS\system32\drivers\etc>

```

Figure 2-26 DB2 contents in a services file

For example:

System name: redSystem
Instance Name: redInst
Port Number (non-partitioned): 50051

The entry in the services file is:

```
DB2_REDINST redInst_DB2 50051/tcp #redInst instance port
DB2_REDINST_END 50051/tcp #redInst end port number
```

If you do not specify the specific port range then the following will be added to the services file:

```
DB2_DB2          60000/tcp
DB2_DB2_1        60001/tcp
DB2_DB2_2        60002/tcp
DB2_DB2_END      60003/tcp
```

2.4.4 Removing an instance

Before you drop an instance from your system you should be sure that you will not need any of the data that is contained in the databases contained in the instance. Once the instance is dropped there is no DB2 utility to recover it. The only recovery method would be to use disk storage recovery and restore an old copy of the file system.

Using the command tools

The **db2idrop** command will drop an instance completely from a system. This will remove all of the components on the system as well as remove it from the list of registered instances. The command syntax is as following:

```
db2idrop instanceName
```

The command will not, however, necessarily remove the instance from the Windows services file. You will have to remove the entries manually.

Restrictions

The base DB2 instance that is created during the install cannot be dropped using the **db2idrop** command. You will have to manually remove the base instance if you want to drop it from the system.

The DAS can be created or dropped using **db2admin** command. The command syntax is:

```
db2admin create DASInstanceName
db2admin drop DASInstanceName
```

2.5 DB2 configuration

After you have installed DB2 UDB and have your database created and populated with data you should tune your system to achieve optimal performance.

2.5.1 Configuration levels

There are three levels of configuration for DB2:

- ▶ DB2 registry
- ▶ Database manager (instance) configuration
- ▶ Database configuration

Each of these configuration levels have a different scope and impact on the performance of your system. It is important to understand what each section does in order to fully take advantage of Windows and the 64-bit capabilities now available for it.

Registry

The DB2 registry parameters are used to control OS-specific behavior and also the behavior of all instances on the system. The registry variables are normally used to activate functionality or control behavior that was added as part of a Fixpak or is operating system specific so it cannot be added to the database manager configuration settings.

Listing registry settings

A complete list of the available registry settings for your system and Fixpak level can be found in the following document:

PDF: *Administration Guide: Performance*, Chapter 13, “Configuring DB2”

The current list can also be given by the **db2set** command (Figure 2-27 on page 76). To list all set variables:

```
db2set -all
```

```

c:\Program Files\IBM\SQLLIB\BIN>db2set -all
[e] DB2PATH=C:\Program Files\IBM\SQLLIB
[i] DB2ACCOUNTNAME=REDBOOK\db2admin3
[i] DB2INSTOWNER=SU1
[i] DB2PORTRANGE=60000:60003
[i] DB2INSTPROP=C:\PROGRA~1\IBM\SQLLIB
[i] DB2COMM=TCPIP
[g] DB2SYSTEM=SU1
[g] DB2PATH=C:\Program Files\IBM\SQLLIB
[g] DB2INSTDEF=DB2
[g] DB2ADMINSERVER=DB2DAS00

```

Figure 2-27 DB2 registry settings output

To list all registry variables:

```
db2set -lr
```

Each registry setting is prefaced with a single letter that indicates what the scope of the variable is:

- ▶ e: Entire system
- ▶ g: Global, all instances
- ▶ i: Only the particular instance you are attached to

It is not recommended to set any of the registry variables unless the documentation or software specifically requests it.

Setting the variables

The variables can be set using the **db2set** command. The majority of the time this command is used is to set a variable for only a single instance. There is a great deal of functionality built into the **db2set** utility. The complete specifications for the command can be found by running the **db2set -?** command.

```
db2set [-i | -g] registryVariable=variableValue
```

It is very important that you do not leave any spaces between the registry variable name, the equals sign, and the variable value. The **db2set** command uses a blank space after the equals sign to indicate that the registry variable should be set to null and returned to its default value. Example 2-8 shows some **db2set** commands.

Example 2-8 Setting the DB2 registry variables

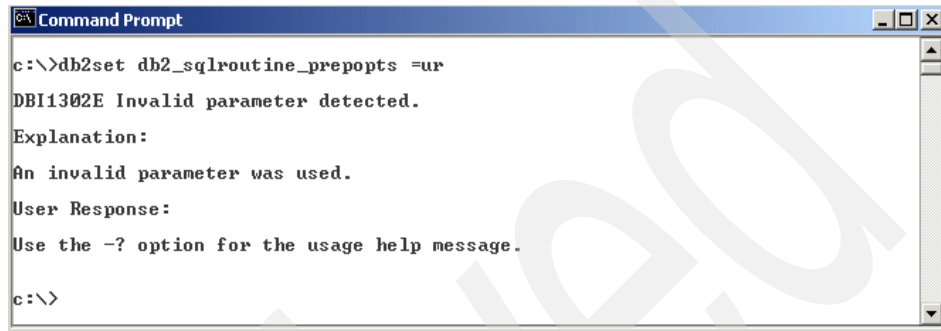
```

[1] db2set -i exampleVar=exampleValue
[2] db2set -i exampleVar =exampleValue
[3] db2set -i exampleVar= exampleValue
[4] db2set -i exampleVar=

```

The results are:

- ▶ [1] The exampleVar registry variable will be properly set to the 'exampleValue' value.
- ▶ [2] An error message will be returned indicating that a variable value was not given (see Figure 2-28).



```
Command Prompt
c:\>db2set db2_sqlroutine_preopts =ur
DBI1302E Invalid parameter detected.
Explanation:
An invalid parameter was used.
User Response:
Use the -? option for the usage help message.
c:\>
```

Figure 2-28 db2set error message

- ▶ [3] The registry variable will be reset to its default value.
- ▶ [4] The registry variable will be reset to its default value.

There is currently no GUI tool to set the parameters.

Important registry variables

The following registry variables are key for DB2 on Windows:

- ▶ DB2NTNOCACHE

Default: OFF *Setting:* ON or OFF

This is an important option for the creation of large databases that leverage large memory segments. It specifies whether DB2 opens database files with a NOCACHE option. If DB2NTNOCACHE=ON, file system caching is eliminated. If DB2NTNOCACHE=OFF, the operating system caches DB2 files. This applies to all data except for files that contain long fields or LOBs. Eliminating system caching allows more memory to be available to the database so that the buffer pool or sortheap can be increased.

In a Windows server, files are cached when they are opened, which is the default behavior. One MB is reserved from a system pool for every one GB in the file. Use this registry variable to override the undocumented 192 MB limit for the cache. When the cache limit is reached, an out-of-resource error is given.

► DB2_AWE

Default: NULL *Setting:* N/A

Used to activate the use of AWE for Windows 2000. This is not supported on 64-bit versions of DB2 UDB.

► DB2BPVARS

Default: path *Setting:* newPath

This parameter controls prefetching behavior, and also if scatter read is used for the bufferpools. Parameters are specified in an ASCII file, one parameter on each line, in the form parameter=value. For example, a file named bpvars.dat might contain the following lines:

- NO_NT_SCATTER = 1
- Numprefetchqueues = 2

If the file bpvars.dat was stored in the c:\db2Data directory then to set the variables you would issue the following command:

```
db2set DB2BPVARS=F:\vars\bpvars.vars
```

► Scatter read

The scatter-read parameters are recommended for systems with a large amount of sequential prefetching against the respective type of containers and for which you have already set DB2NTNOCACHE to ON. These parameters, available only on Windows platforms, are NT_SCATTER_DMSFILE, NT_SCATTER_DMSDEVICE, and NT_SCATTER_SMS.

Specify the NO_NT_SCATTER parameter to explicitly disallow scatter read for any container. Specific parameters are used to turn scatter read on for all containers of the indicated type. For each of these parameters, the default is zero (or OFF), and the possible values include zero (or OFF) and 1 (or ON).

Note: You can turn on scatter read only if DB2NTNOCACHE is set to ON to turn Windows file caching off. If DB2NTNOCACHE is set to OFF or not set, a warning message is written to the administration notification log if you attempt to turn on scatter read for any container, and scatter read remains disabled.

The parameter can also be set for the prefetching options. For further information on the parameter see the registry variable descriptions in the DB2 UDB documentation:

Administration Guide: Performance, Appendix A, “DB2 Registry and Environment Variables”

► DB2NTMEMSIZE

Default: varies by memory size *Setting:* varies by desired memory size

The Windows server requires that all shared memory segments be reserved at DLL initialization time in order to guarantee matching addresses across processes. DB2NTMEMSIZE permits the user to override the DB2 defaults on a Windows server if necessary. In most situations, the default values should be sufficient. The memory segments, default sizes, and override options are:

- Database Kernel: Default size is 16777216 (16 MB); override option is DBMS:<number of bytes>.
- Parallel FCM Buffers: Default size is 22020096 (21 MB); override option is FCM:<number of bytes>.
- Database Admin GUI: Default size is 33554432 (32 MB); override option is DBAT:<number of bytes>.
- Fenced Stored Procedures: Default size is 16777216 (16 MB); override option is APLD:<number of bytes>.

More than one segment may be overridden by separating the override options with a semi-colon (;). For example, to limit the database kernel to approximately 256 K, and the FCM buffers to approximately 64 MB, use:

```
db2set DB2NTMEMSIZE=DBMS:256000;FCM:64000000
```

► DB2NTPRICLASS

Default: NULL *Setting:* R,H, any other value

This sets the priority for the instance services (db2sysc.exe) on Windows. There are a couple of priority classes. For more information, please refer to the SetPriorityClass() API in the Win32 documentation.

- NORMAL_PRIORITY_CLASS (the default priority class)
- REALTIME_PRIORITY_CLASS (set by using "R")
- HIGH_PRIORITY_CLASS (set by using "H")

Note: Be careful when using this variable. The priority should only be changed when necessary since it can adversely affect system performance.

► DB2NTWORKSET

Default: 1,1 *Setting:* x,y

Used to modify the minimum and maximum working-set size available to DB2. By default, when the Windows server is not in a paging situation, the working set of a process can grow as large as needed. However, when paging occurs, the maximum working set that a process can have is approximately 1 MB. DB2NTWORKSET allows you to override this default behavior.

Specify DB2NTWORKSET for DB2 using the syntax
DB2NTWORKSET=min,max, where min and max are expressed in
megabytes.

Database manager

The database manager, also referred to as the instance, controls the connections to all the databases that are contained within it. The database manager parameters tend to control application and operating system level settings rather than database-specific issues.

The database manager can have a large impact on the performance of your applications since it controls a number of key memory heaps and also sets the limit on the memory available to all databases on the system.

A complete list of the database manager parameters can be found in the online help and in the PDFs:

PDF: *Administration Guide: Performance*, Chapter 13, “Configuring DB2”

Listing database manager variables

The database manager variable can be obtained using both GUI tools and from the command line.

Command line

The command line offers a quick method to view all of the parameters. It also allows you to work the output into any scripts you may be running. The command syntaxes are:

```
db2 get database manger configuration [show detail]  
db2 get dbm cfg [show detail]
```

The typical execution of the command is specified below (Figure 2-29 on page 81).

```

c:\>db2 get dbm configuration

Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

Database manager configuration release level           = 0x0a00

Maximum total of files open                <MAXTOTFILOP> = 16000
CPU speed <millisec/instruction>           <CPUSPEED>  = 1.086392e-006
Communications bandwidth <MB/sec>          <COMM_BANDWIDTH> = 1.000000e+000

Max number of concurrently active databases  <NUMDB> = 4
Data Links support                          <DATA LINKS> = NO
Federated Database System Support           <FEDERATED> = YES
Transaction processor monitor name           <TP_MON_NAME> =

Default charge-back account                 <DFT_ACCOUNT_STR> =

Java Development Kit installation path       <JDK_PATH> = C:\PROGRAM~1\IBM\SQLLI
B\java\jdk

Diagnostic error capture level               <DIAGLEVEL> = 3
Notify Level                               <NOTIFYLEVEL> = 3
Diagnostic data directory path               <DIAGPATH> =

```

Figure 2-29 Database manager parameters from the command line

When an instance parameter is reset the new value may not take effect immediately. The “show detail” option allows you to see what the current values for all the parameters will be and also what they will be once the instance is restarted.

In order to see what the current and future values of the parameters will be, you have to attach to the instance first. The instance will also have to be started. Example 2-9 illustrates the steps to use the “show detail” option. Figure 2-30 on page 82 shows the results of these commands.

Example 2-9 List dbm cfg

```

Start the instance: db2start
Attach to the instance: db2 attach to instName [user userName using password]
List the db2 database manager parameters: db2 get dbm cfg show detail

```

```

C:\>db2 get dbm cfg show detail
SQL1427N An instance attachment does not exist.

C:\>db2 attach to db2
SQL1032N No start database manager command was issued.  SQLSTATE=57019

C:\>db2start
06-04-2003 14:23:17      0      SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.

C:\>db2 attach to db2

      Instance Attachment Information

Instance server      = DB2/NT 8.1.2
Authorization ID     = DREWKB
Local instance alias = DB2

C:\>db2 get dbm cfg show detail

      Database Manager Configuration

      Node type = Enterprise Server Edition with local and remote clients

Description          Parameter    Current Value
-----
Delayed Value
-----
Database manager configuration release level      = 0x0a00
Maximum total of files open      <MAXTOTFILOP> = 16000
16000
CPU speed <millisec/instruction>      <CPUSPEED> = 1.086392e-006
1.086392e-006

```

Figure 2-30 dbm cfg show detail steps

If you receive an error when you try to execute DB2 commands at the command line (Figure 2-31), it is most likely due to the DB2 environment not being initialized. The DB2 tools Command Line Processor and Command Window both run a script to initialize the command line. These two tools are in the IBM DB2 program menu folder in the Command Line Tools folder.

```

C:\>db2 list db directory
DB21061E Command line environment not initialized.

C:\>db2cmd

C:\>_

```

Figure 2-31 Windows command line DB2 error

The environment can be initialized by running the utility **db2cmd**. A new command line will be opened in a different window. There are a number of options that control how the new command window is opened and closed. Further information on the command can be found in the DB2 manual *Command Reference*, Chapter 1, “System commands”.

Using the DB2 Control Center

The DB2 Control Center tool allows a user to graphically work with all the information and data in his database. Within this tool is a graphic interface to all of the database manager configuration parameters.

The DB2 Control Center (CC) can be found in the Windows program menu under **IBM DB2 -> General Administration Tools**.

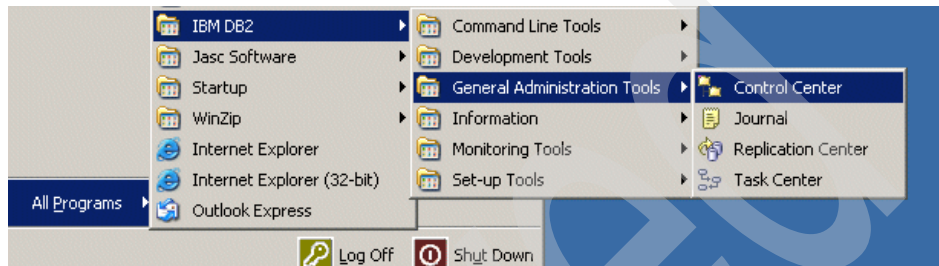


Figure 2-32 Location in Windows of the DB2 Control Center utility

The Control Center displays all of the systems, instances, and databases that you have catalogued on your local system. Each object can be double clicked to show the objects that are contained within it. If you right click the instance that you would like to work with a list of different options will appear. Select the **Configure Parameters** option (Figure 2-33 on page 84).

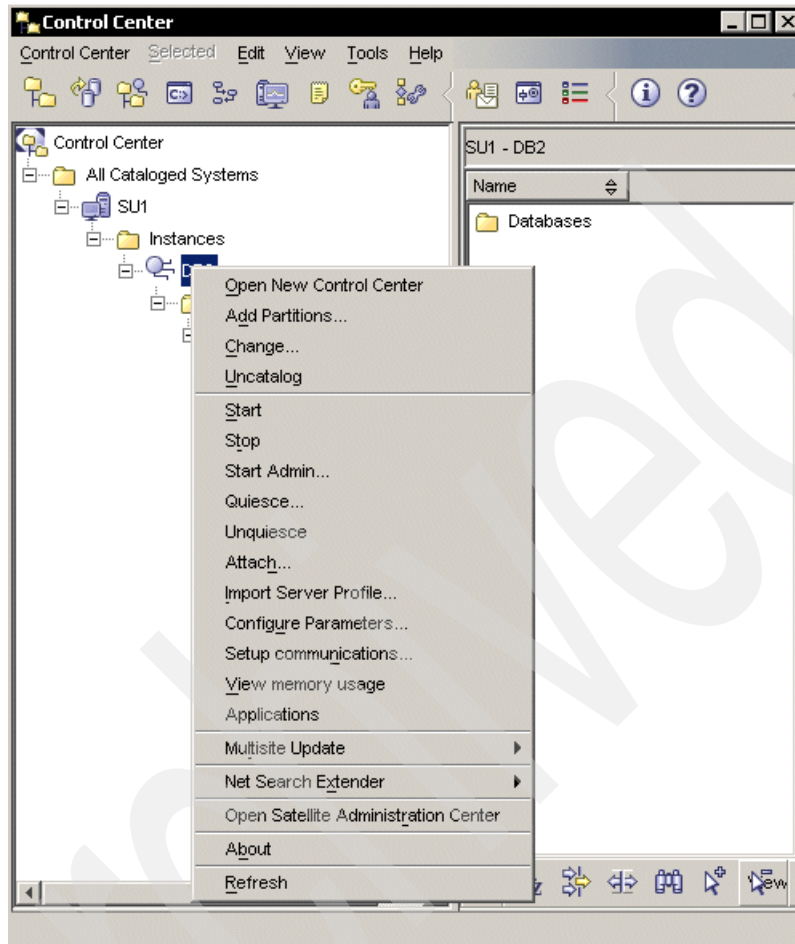


Figure 2-33 Control Center instance configuration option

The Control Center displays all of the parameters in related groups, which makes it easier to determine which parameters you need to work with. The information provided in the GUI tool is more extensive than that provided by the command line tool. The GUI provides the parameter, a description of the parameter, its current and future values, when the new value will take effect, and even a hint on how the parameter should be tuned. The additional information makes the GUI an ideal tool for new users to use to learn about configuring instances in DB2.

When the new values take effect

Previously in DB2 UDB V7.2, any changes in the instance configuration parameters, with the exception of MINCOMMIT and the DB2 snapshot switches, required the instance to be stopped and restarted. This was quite impractical for

production systems and required that DBAs would have to plan far in advance any configuration changes.

With DB2 UDB V8.1, the majority of the instance parameter changes take effect immediately. The GUI tool will indicate when the parameter change will take effect. A complete list of the instance parameters and when they take effect can be found in the *DB2 Administration Guide: Performance*.

Note: Make sure that you keep track of any parameters that you change. If you experience any problems or performance issues later on, you want to be able to quickly isolate the problem and check that the instance setting changes were not the cause.

Setting the parameters

There are two ways to set the DB2 database manager parameters.

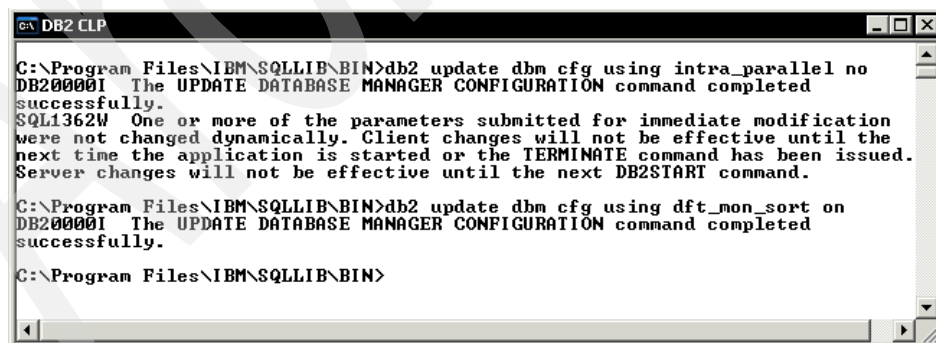
Command line

The Windows command line can be used to set the instance configuration parameters. The command syntax is:

```
db2 update dbm cfg using instanceVariable variableValue
```

Where the `instanceVariable` is the variable to be set and the `variableValue` is the value for the new variable.

Figure 2-34 shows the two update database manager commands. A message will indicate if the value has taken effect immediately or if you have to wait for the instance to be restarted.



```
GA DB2 CLP

C:\Program Files\IBM\SQLLIB\BIN>db2 update dbm cfg using intra_parallel no
DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
SQL1362W One or more of the parameters submitted for immediate modification
were not changed dynamically. Client changes will not be effective until the
next time the application is started or the TERMINATE command has been issued.
Server changes will not be effective until the next DB2START command.

C:\Program Files\IBM\SQLLIB\BIN>db2 update dbm cfg using dft_mon_sort on
DB20000I The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.

C:\Program Files\IBM\SQLLIB\BIN>
```

Figure 2-34 Setting instance configuration parameters at the command line

Using the Control Center

The Control Center also allows parameters to be changed, in addition to listing the variables. It offers an advantage over the command line in that when the variable is being set the complete allowable range is displayed in addition to a hint on how to set the parameter.

If you wish to change a parameter, open up the instance configuration form. Left click the value you would like to change for the desired variable. Clicking the small box that appears to the right of the variable will open up a new form that will let you change the variable (Figure 2-35).

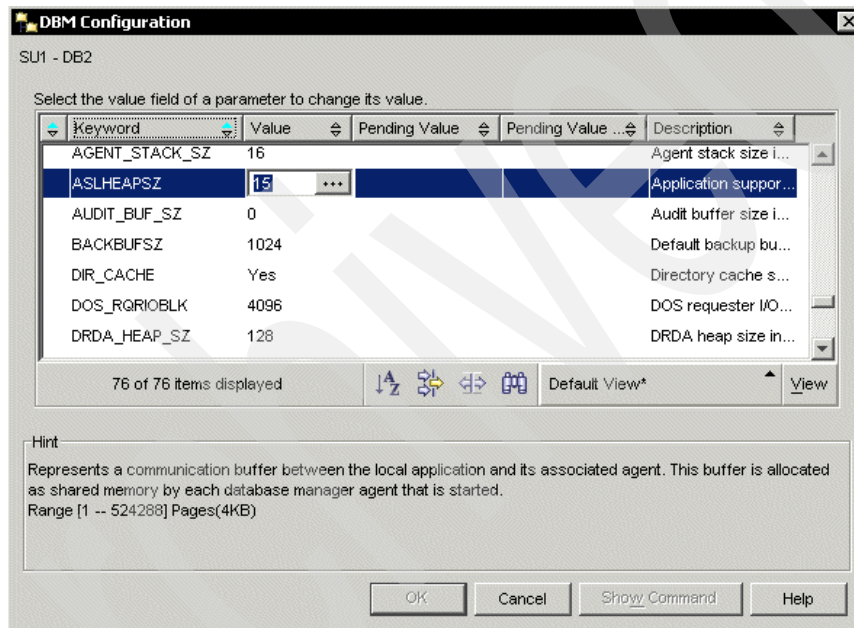


Figure 2-35 Changing an instance parameter using the Control Center

Once you have set the parameter you will be returned to the DBM configuration form. The column Pending Value Effective will now indicate when the parameter change will take effect (Figure 2-36 on page 87).

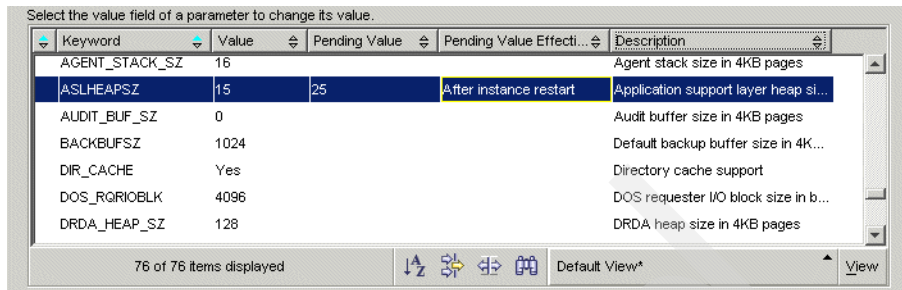


Figure 2-36 Parameter change time frame

Database

The database configuration parameters control how the database behaves and are used to ensure that the workload runs smoothly on the database. These parameters control all of the memory heaps that are for a specific database, and altering these values can have a significant effect on performance.

A complete list of the database parameters can be found in the online help and in the DB2 UDB manuals:

PDF: *Administration Guide: Performance*, Chapter 13, “Configuring DB2”

Listing database parameter values

As with the instance parameters, the database variables can be set both by using the command line and with the Control Center.

Command line

The command line offers a quick method to view all of the parameters. It also allows you to work the output into any scripts you may be running. The command syntax is as following:

```
db2 get database configuration for databaseName [show detail]
db2 get db cfg for databaseName [show detail]
```

The current and pending values of the command can be obtained by using the “show detail” option. You need to connect to database before issuing the command.

Control Center

Information on the databases in an instance can be found in the Control Center by double clicking the desired database. The database can then be configured by right clicking it.

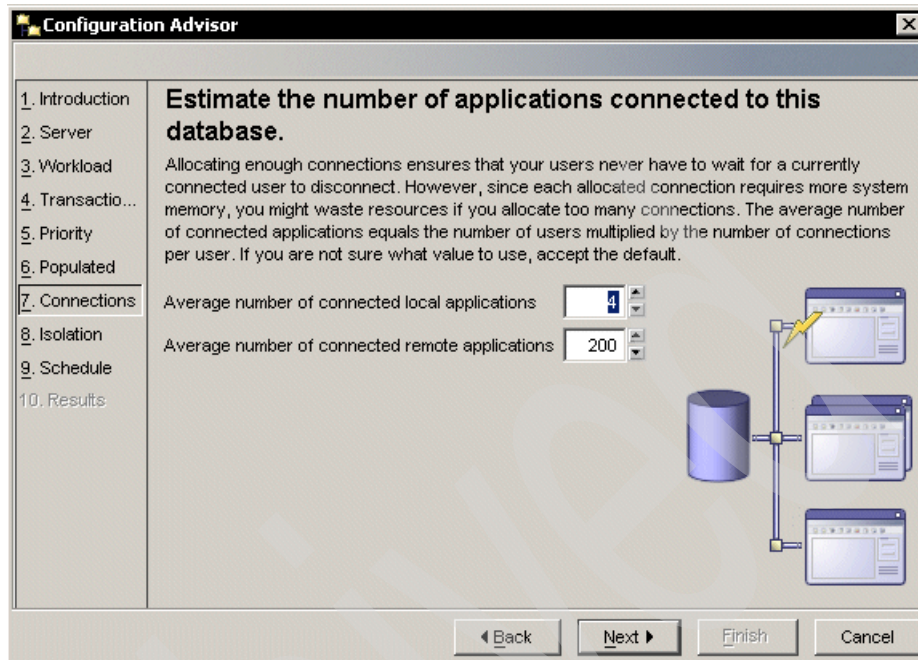


Figure 2-37 Selecting a database to configure

There are two options for configuring the database. The first option, Configuration Advisor, links to the DB2 auto-configuration wizard. This will be covered in the next section. The second option, Configure Parameter, is the option to directly configure the database parameters.

The database parameter interface is very similar to the instance parameter interface (see Figure 2-38 on page 89). The only difference is the parameters and how they are grouped. If you click a parameter, a full description of it will appear in the bottom window.

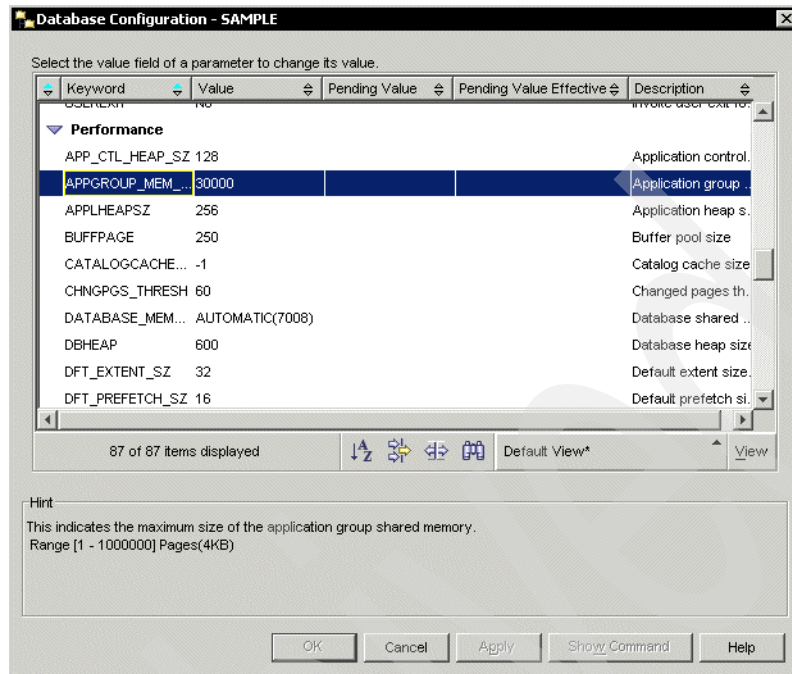


Figure 2-38 Configuring a database option

Setting the variables

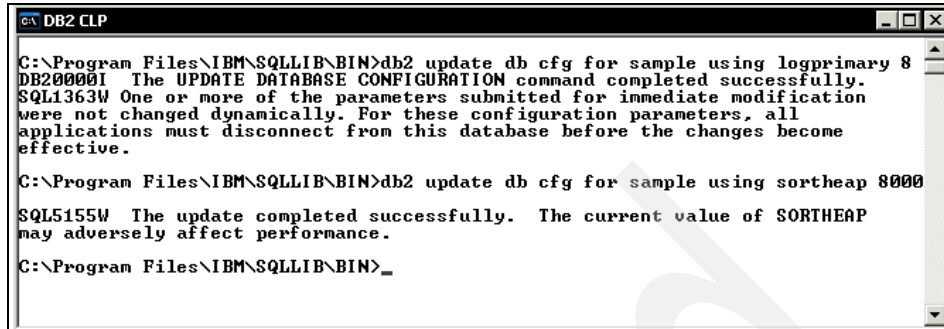
Previously in DB2 UDB V7.2, the updating of database parameters required all connections to be disconnected from the database before the new values took effect. Almost all of the values can now be updated dynamically with the reallocation of memory or changing their values. Those that are not able to be updated dynamically will return a warning message that indicates when the new values will take effect.

Command line

The command line interface for setting parameters is very similar to that used for the instance, and with the configuration of other variables in DB2. The command syntax is as follows:

```
db2 update database configuration for databaseName using variable varValue
db2 update db cfg for databaseName using variable varValue
```

Figure 2-39 on page 90 shows two update database configuration examples. A message will indicate if the setting took effect immediately or if a database reconnection is required for all users.

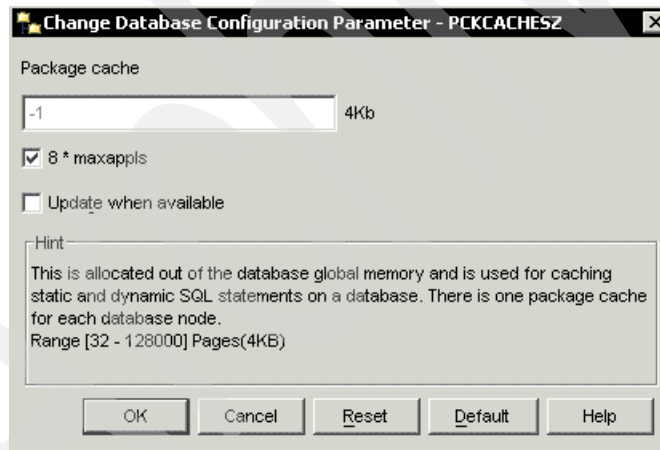


```
DB2 CLP
C:\Program Files\IBM\SQLLIB\BIN>db2 update db cfg for sample using logprimary 8
DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.
SQL1363W One or more of the parameters submitted for immediate modification
were not changed dynamically. For these configuration parameters, all
applications must disconnect from this database before the changes become
effective.
C:\Program Files\IBM\SQLLIB\BIN>db2 update db cfg for sample using sortheap 8000
SQL5155W The update completed successfully. The current value of SORTHEAP
may adversely affect performance.
C:\Program Files\IBM\SQLLIB\BIN>_
```

Figure 2-39 Updating the database configuration using the command line

Using the Control Center

The database parameters can also be updated using the Control Center interface. To alter a parameter, open up the DB configuration form that you used to view the parameters. Double clicking the desired parameter will open up the parameter configuration form. The form will vary in appearance depending on the type of parameter being updated. Figure 2-40 is the screen image for changing the package cache configuration.



The dialog box is titled "Change Database Configuration Parameter - PCKCACHESZ". It contains a text box for "Package cache" with the value "-1" and a label "4Kb". Below this is a checked checkbox labeled "8 * maxappls" and an unchecked checkbox labeled "Update when available". A "Hint" section contains the text: "This is allocated out of the database global memory and is used for caching static and dynamic SQL statements on a database. There is one package cache for each database node. Range [32 - 128000] Pages(4KB)". At the bottom are buttons for "OK", "Cancel", "Reset", "Default", and "Help".

Figure 2-40 Updating a database configuration parameter

Once you have set the parameter you will be returned to the DB configuration form. The column Pending Value Effective will now indicate when the parameter change will take effect.

2.5.2 Using the auto-configure tool

The auto-configure tool was available in DB2 UDB V7.2 as well, but its effectiveness has been greatly improved for V8. A significant amount of research was undertaken to improve the quality of the configuration recommendations that it gives. It is highly recommended for new DBAs to use this tool to initially tune their database since the default configuration settings are often not sufficient for the workload.

Configuration Advisor

The tool is included as part of the DB2 Control Center. The auto-configure tool is called by selecting the database in the Control Center and clicking the **Configuration Advisor** option. The DB2 DAS service must be running for the tool to work. Previously the DAS service was actually an instance and could be started and stopped from the command line using DB2 commands. It can no longer be controlled this way.

The DAS can be started and stopped in the Windows Services. The services section can be found by opening the Domain Controller utility. The menu for activating a service is shown in Figure 2-23 on page 72.

The DAS is under the DB2DAS - DB2DAS00 service. The DAS instance has a small memory footprint, so it is recommended that you set the DAS instance to automatically start at bootup.

How it has changed since V7.2

With each successive fixpak the utility is being refined and improved to give the best possible performance recommendation. The configuration utility in DB2 UDB V7.2 had less questions and also made more general recommendations for the workload. The new utility performs a large amount of auto-detection and adjusts to the system resources that are available.

Using the Configuration Advisor

The utility walks a DBA through a series of questions about the system resources and the workload that is expected on the database.

Memory allocation

The first screen (Figure 2-41 on page 92) allows a user to specify how much of the total memory will be made available to DB2. The application will auto-detect how many CPUs are available on the system. If you are using a 32-bit system there will be a limit of 64 GB (on Datacenter) of memory available due to the limits imposed by 32-bit addressing and how Windows has dealt with this limitation. If you have a large amount of memory available on the system, it is a

good idea to leave about 500 MB free for the operating system, other applications, and local connections.

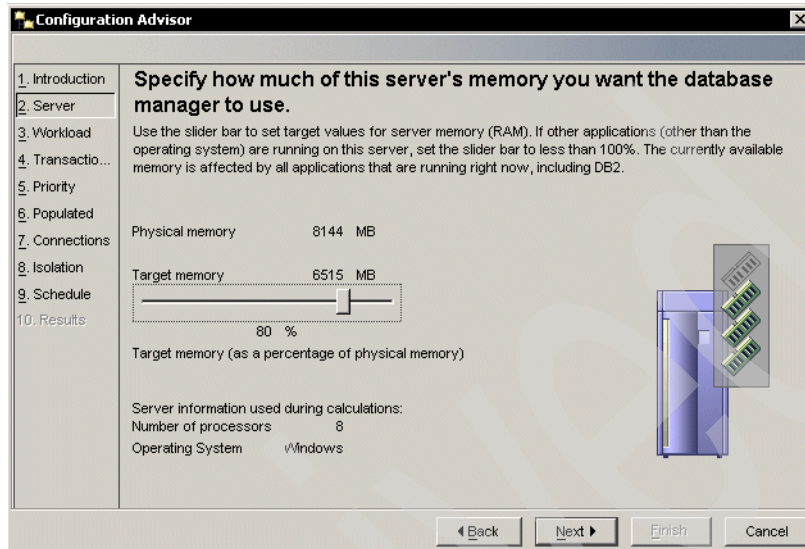


Figure 2-41 Configuration Advisor: Memory specifications

Query workload

The choice of the query workload has a large impact on how the system is configured.

- ▶ **Transaction:** Used for a pure order entry system such as an e-commerce application. A minimal amount of reporting should be running against the database since this option will not tune the system well for large complex queries.
- ▶ **Mixed:** Used for the majority of workloads running against today's databases. Most workloads now consist of a mixture of transactions through Web pages or internal tools, and occasional, but heavy, reporting against the data in the system. This option will tune the system to respond well to both types of workloads.
- ▶ **Queries:** Used for data warehousing, reporting, and mining systems. These workloads normally consist of very complex and long-running queries that can use a lot of system resources. The configuration used to obtain good results for these queries would have a negative impact on a high volume transaction workload.

Database transaction volume

The transaction volume information is used to determine the type of logging and commit frequency. The majority of order entry systems would choose “Fewer than 10”. ERP and CRM systems often have complex logic being performed for each transaction. “More than 10” should be selected. See Figure 2-42.

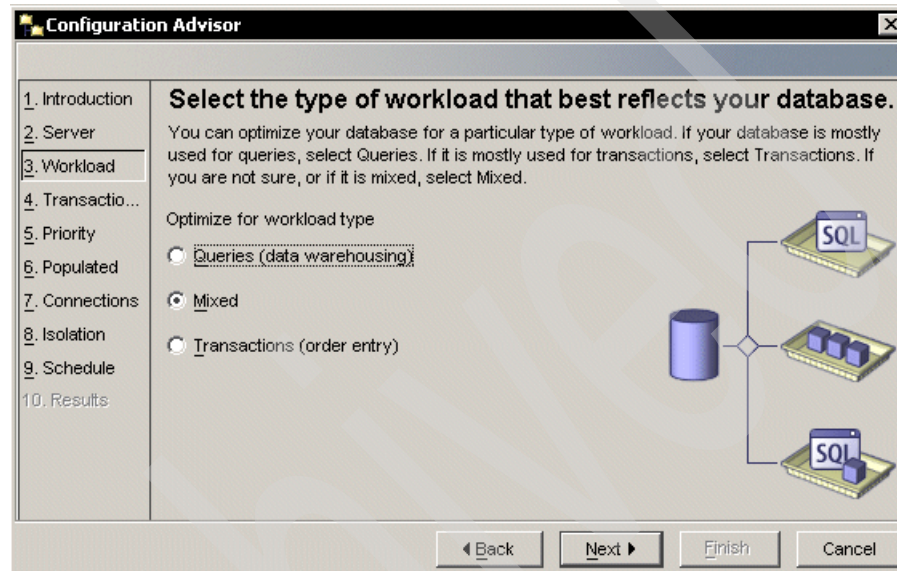


Figure 2-42 Configuration Advisor: Transaction length and volume

Administration policy

The administration policy screen (Figure 2-43 on page 94) is used to decide if it is more important that the database recovers very quickly or the speed of the workload is more important. Even by selecting **Faster transaction performance** the database will still recover quickly if the appropriate high availability settings and applications are chosen. This option impacts how logging is handled.



Figure 2-43 Configuration Advisor: Administration policy

Database population

The Configuration Advisor will use the existing statistics in the database to determine how large many memory parameters should be. It is important that you first run the Runstats utility on the database to ensure that the statistics are up to date. Basing the Configuration Advisor on out-of-date statistics could lead to poor recommendations.

Number of applications

The number of applications directly controls the maximum number of applications expected for the system. These choices also affect the average number of applications that DB2 expects to be running. This will have an impact on SQL optimization.

Isolation level

The default isolation level should be decided on by the application vendor that will be running on the database. The default level is Cursor Stability. Increasing the isolation level, upwards to repeatable read, will increase the database stability for transactions, but it can also lead to concurrency issues since more locks will be held for a greater duration. See Figure 2-44 on page 95.

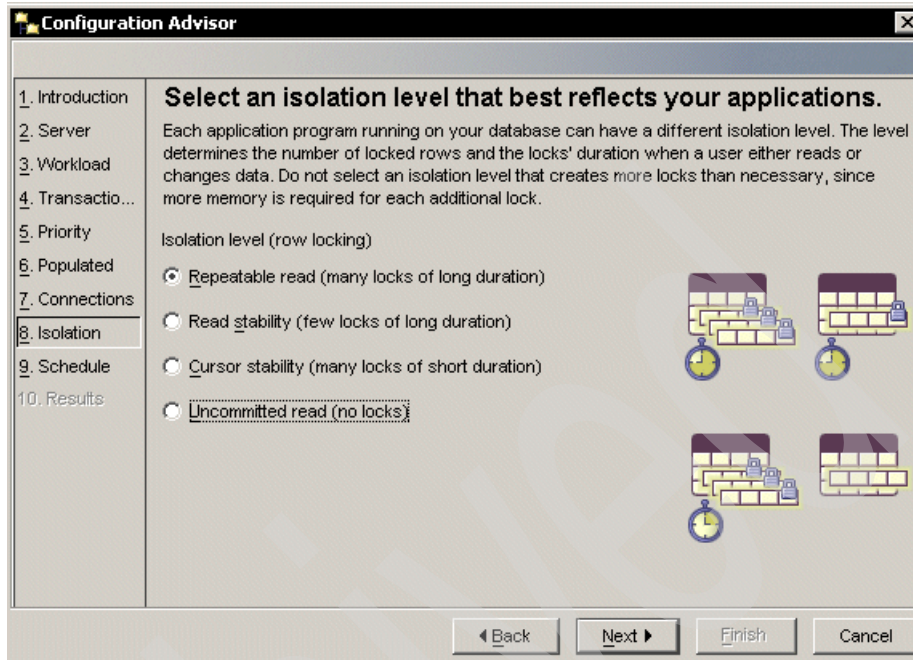


Figure 2-44 Configuration Advisor: Isolation level

Scheduling the task

Built into DB2 UDB V8.1 there is a task-scheduling facility for all the database wizards. This will allow you to save the database and instance configuration script that has been created by the advisor. The script can then be reviewed and checked to make sure that it is appropriate for you system and your workload. The task scheduler can then later execute the script and save the output. See Figure 2-45 on page 96.

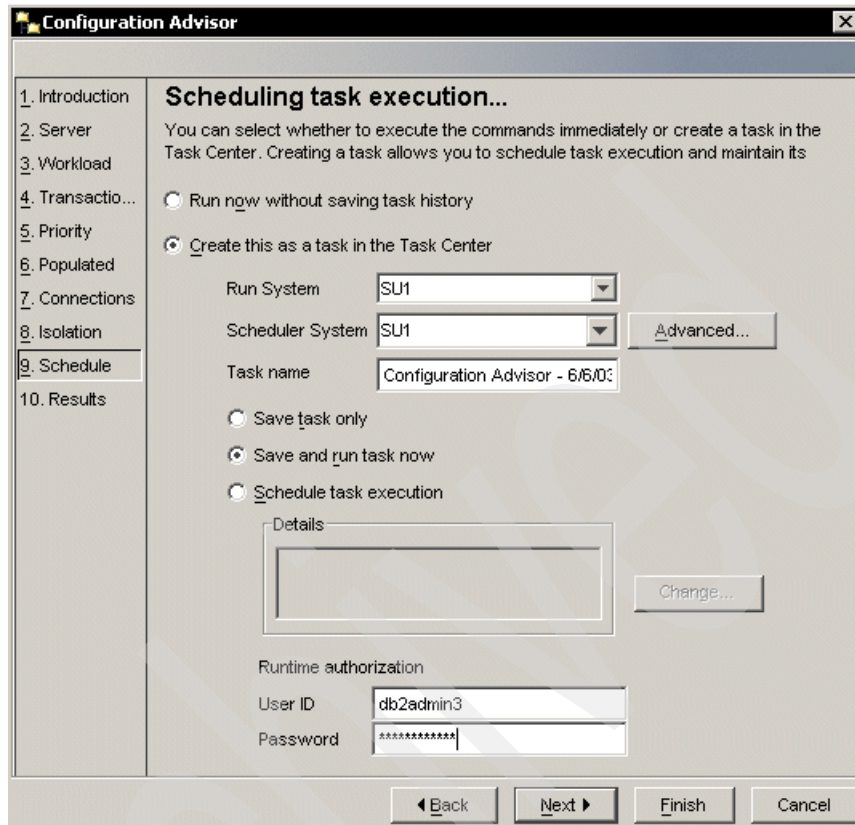


Figure 2-45 Configuration Advisor: Scheduling the configuration task

In order to use the task scheduler utility the tools database must have been created at installation time. If you did not create the tools database during installation it can still be created later. A DB2 command, CREATE TOOLS CATALOG, will create all the necessary tables, views, and schemas in the database that you are currently connected to. If you like a database dedicated to scheduling and tools then the tool has the option of creating a new database for you. You can also use an existing database, but you must specify which tablespace you would like all the new objects created on. See Example 2-10.

Example 2-10 Creating a tools database

```
db2 create database tools_db
>>DB20000I The CREATE DATABASE command completed successfully.
db2 create tools catalog toolsCat use existing database tools_db
>>DB20000I The CREATE TOOLS CATALOG command completed successfully.
```

2.5.3 Configuring memory for 64-bit systems

The availability of a 64-bit version of Windows Server 2003 will allow this product to fully take advantage of the scalable architecture of the IBM xSeries and other multiple CPU Intel systems. Previously with only a 32-bit version available, the large amount of memory available on many of today's systems could not be fully utilized. The trade off between different workload performance due to memory constraints will no longer be an issue.

The battle between sort vs. bufferpool is over

As the complexity of the applications running on modern databases has grown, the different demands on the system have made the life of a DBA difficult. There has traditionally been a trade off between OLTP workloads, which require large bufferpools to cache all the transactional data, and DSS or reporting workloads, which require very large sort areas for complex queries that return many rows. With 64-bit this decision will no longer have to be made since the required memory can be allocated to both types of workloads. The bufferpools can be made quite large and the sort space can be increased to handle the reporting as well.

Extended memory history on Windows: AWE

Look up history of AWE support for DB2 UDB on V7.2 the Microsoft Address Windowing Extensions (AWE) API provides 32-bit Windows applications with the ability to address up to 64 GB of physical non paged memory. Although it provides support for 64 GB of real physical memory space on 32-bit versions of Windows, it is sometimes difficult to implement on systems with over 16 GB of memory. This is because on systems with over 16 GB of memory, you must disable 4 GB Memory Tuning (also know as /3GB Switch), as the Windows kernel requires 2 GB of memory for managing AWE memory above 16 GB. This in turn reduces the applications virtual address space from 3 GB to 2 GB. Once the extra 1 GB of virtual address space has been removed from the application, it becomes difficult to allocate a large enough memory within a 2 GB address space to manage real physical memory above the 4 GB line.

Windows supports 64-GB memory; however, addressing real physical memory over 32 GB from a single 32-bit application is somewhat impractical.

You enable this feature by adding /PAE to the Windows Server 2003 boot.ini file, which can be used in combination with the /3GB switch on systems with 16 GB of memory or less.

All editions of DB2 UDB can take advantage of the large memory support provided by the Microsoft AWE API, but currently it is only implemented for database buffer pools. Tuning DB2 instance and database configuration

parameters to take advantage of large memory supported by AWE is covered in detail in “AWE Memory” on page 131.

How much you should use

The old memory limit of 3 GB (using /3GB) created a maximum for the amount of memory DB2 could use. Currently the limit for memory allocations for DB2 will be the physical memory limit or the particular edition of Windows that you have installed (see Table 2-5).

Table 2-5 Memory limits for Windows database editions

Windows edition	Memory limit (GB)
Standard	4
Enterprise (32-bit)	32
Enterprise (64-bit)	64
Datacenter (32-bit)	64
Datacenter (64-bit)	512

2.5.4 Configuring the instance

There are a large number of instance configuration parameters that can be tuned, but this section concentrates only on those that deal with memory allocation. A full description of all the database manager parameters can be found in the following help:

Administration Guide: Performance, Chapter 13, “Configuring DB2”, section “Database Manager Configuration”

Instance parameters

Table 2-6 lists all of the parameters that have memory allocated at the instance level. Each parameter lists whether it can be changed dynamically and also if the total memory allocated for it should be scaled up, as more is available with the 64-bit addressing.

Table 2-6 DB2 instance memory configuration variables

Parameter	64-bit scaling	Online configurable	Description
agent_stack_sz	No	No	Agent stack size
aslheapsz	No	No	Application support layer heap size

Parameter	64-bit scaling	Online configurable	Description
audit_buf_sz	No	No	Audit buffer size
fcm_num_buffers	Yes	Yes	Number of FCM buffers
instance_memory	Yes	No	Total instance memory
java_heap_sz	No	No	Maximum java interpreter size
min_priv_mem	Yes	No	Minimum committed private memory
mon_heap_sz	No	No	Database system monitor heap size
priv_mem_thresh	Yes	No	Private memory threshold
query_heap_sz	No	No	Query heap size
sheapthres	Yes	No	Sort heap threshold

Key parameters

Here we discuss the key database instance parameters in details:

► SHEAPTHRES: Sort heap threshold

Default: Automatic Range: 8 - 524,288

The sort heap threshold controls the total amount of sort memory available to all databases in an instance. Private and shared sorts take their memory from two different sources. The shared memory area is a fixed amount that has been set at the time of the first connect to a database after an instance restart. The size of the private memory sort area is unrestricted.

The parameter is used differently depending on the type of sort. For private sorts, the sort heap threshold is merely a soft limit on the total amount of memory available to sorts. If the total memory of the private sorts occurring concurrently approaches the parameter then less memory will be assigned to each successive sort.

If a large number of sorts are occurring concurrently then your database may run out of sort space. If the total amount of memory requested by the individual applications' sort heaps then not all of them will be given the full sort heap amount and you will see sort overflows occurring. There are two methods by which the sort heap overflow problem is normally dealt with. The first is to increase the sort heap threshold limit for the instance. This will allow more sort heaps to occur concurrently and some of the sort requests will no longer be given less memory then they had requested.

The problem of insufficient sort heap threshold space is illustrated in Figure 2-46. In the example there are six sort requests all for the maximum size of the sort heap at 5000 pages. They are not all given the amount of memory they requested since there is insufficient space for all of them to occur concurrently. If the DBA had increased the sort heap threshold limit to 30000 pages then there would be no sort overflows for this scenario.

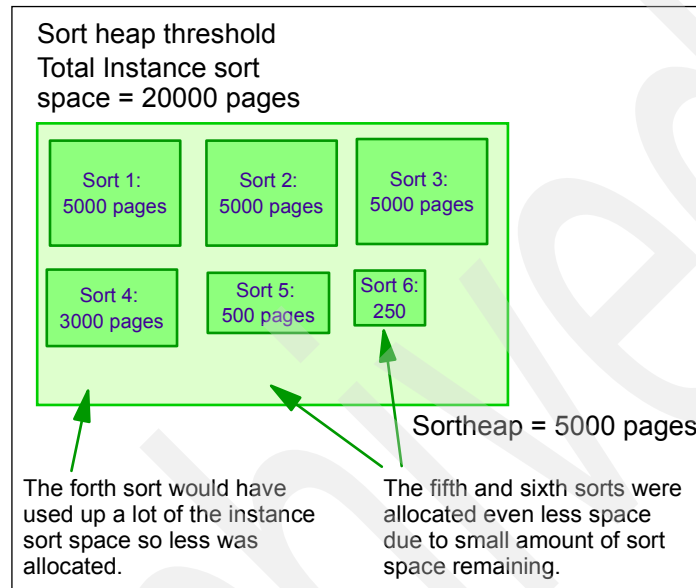


Figure 2-46 Insufficient sort heap threshold limit example

The second example, in Figure 2-47 on page 101, shows what could happen if the sort heap parameter and the sort heap threshold are not large enough. All of the sorts in the example are overflowing, which would normally lead a DBA to increase the sort heap parameter size. If this was the only parameter that was increased, the sort overflow problem would actually get worse. The first couple of sorts would get the new larger maximum sort heap but the rest would overflow even worse and take longer since they would be given less memory.

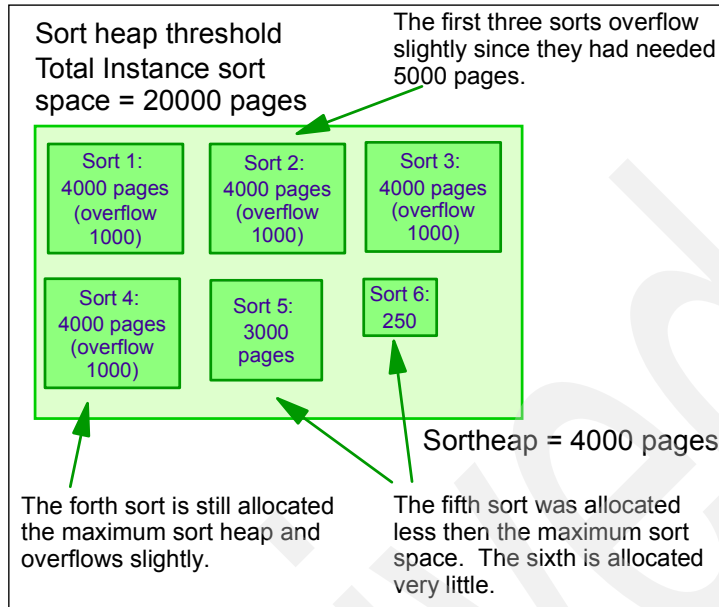


Figure 2-47 Sort heap and sort heap threshold problem

The ideal method to correct the problem would be to first increase the sort heap. If your sort overflow problem actually gets worse, then you need to increase the sort heap threshold as well give sufficient space for your sorts.

The greater amount of memory available for sorting due to 64-bit memory will eliminate the need to constrain the sort heap parameters since the sort heap threshold can be raised to a very large amount. It is recommended that the database level sort heap parameter be increased until the sort overflows disappear or are minimized. If you know that you have a few sort operations that are quite large (for example, with millions of rows in a data mining query), then you will most likely not be able to avoid a sort overflow. Raising the threshold and the sort heap could reduce the severity of the overflow and greatly speed up the performance of the query.

For shared sorts, the parameter is a hard limit on the amount of memory, and it will not be exceeded. Once the limit is reached, there will no longer be any shared sorts allowed until there is free memory. Increasing the amount of memory available for the sort heap threshold will allow more shared sorts to occur.

Recommendations:

- OLTP: 20 percent of total memory available for DB2
- DSS: 80 percent of total memory available for DB2

- ▶ **INSTANCE_MEMORY:** Instance memory
Default: Automatic Range: 8 - 524,288

This parameter controls the amount of memory that is required for instance management.

Recommendation: Leaving the parameter set to Automatic will allow DB2 to automatically size the parameter to the required amount. This will ensure that the value is always correct and sufficient memory has been allocated.

Memory parameters

Following are detailed descriptions of some memory-related database instance parameters:

- ▶ **AGENT_STACK_SZ:** Agent stack size
Default: 16 Range: 8 - 1000

The stack controls the maximum amount of memory that can be allocated for the query heap. The stack size must be sufficient to contain all the SQLDA, SQLCA, and all other query information when it is being compiled.

Recommendations: If you have very complex SQL that involves a large number of objects, or if the query references a number of LOBS, then the value may need to be increased. As a minimum the value should be five times the size of the application support layer heap size (aslheapsz, see below).

It will not need to be scaled as the amount of memory is increased so a low value is often sufficient.

- ▶ **ASLHEAPSZ:** Application support layer heap size
Default: 15 Range: 1- 524,288

The parameter is the communication buffer between a local application and its associated agents. Each database manager agent that is started will have this memory allocated for its use. The value is also used to determine the size of blocking cursors.

Recommendations: If the queries tend to be small then the default value could be reduced. If the queries are large and complex then it is a good idea to increase this parameter to avoid having to have more than one send and receive set.

The parameter does not need to be scaled up as the available memory is increased.

- ▶ **AUDIT_BUF_SZ:** Audit buffer size
Default: 0 Range: 0- 65000

The parameter controls the amount of memory used for auditing DB2. Having this value greater than zero will turn DB2 auditing on.

Recommendation: Leave the parameter at zero. This parameter does not need to be scaled as the memory increases.

- FCM_NUM_BUFFERS: Number of FCM buffers
Default: 512 Range: 128- 524,288

The parameter controls the number of 4-K buffers used for internal communication. These internal communication buffers are used heavily in a partitioned database environment.

Recommendation: If you have multiple logical nodes on a SMP machine then the parameter will need to be increased.

This value will not need to be scaled as the memory increases unless multiple logical nodes are added as well.

- JAVA_HEAP_SZ: Java interpreter heap size
Default: 512 Range: 0- 524,288

The Java interpreter heap controls the total amount of memory that is allocated for the Java compiler. This will only be used for Java stored procedures or functions.

Recommendation: The value should be set to zero if Java stored procedures and functions are not used. If there are a very large number of java procedures used in a high volume system then the parameter should be increased significantly.

The parameter does not, however, need to be scaled as additional memory is added, but instead as the java procedure/function workload increases.

- MIN_PRIV_MEM: Minimum private memory
Default: 32 Range: 32- 112,000

The parameter controls the number of pages the instance will reserve as virtual private memory when the instance is started.

Recommendation: The parameter should be left alone since setting the value too high could cause an issue when the instance is first started. If the instance requires more memory than is available in the private memory then it will request more from the operating system.

- MON_HEAP_SZ: Monitor heap size
Default: 32 Range: 0- 60,000

The monitor heap size is used by the DB2 monitoring tools. This memory will be used whenever a snapshot is taken, an event monitor is run, or one is reset.

Recommendation: The value should be increased if multiple event monitors will be running concurrently or multiple snapshots are taken during the same interval.

The value does not need to be scaled as the amount of memory grows.

- **PRIV_MEM_THRESH:** Private memory threshold
Default: 1296 Range: -1, 32 - 112,000

The parameter is used to determine how much unused agent private memory will be kept by DB2 ready to service new agents. When an agent is finished, instead of returning the memory to Windows, DB2 will keep a certain amount of the memory. The value of -1 will cause the value of min_priv_mem to be used.

Recommendation: This parameter is dependent on the volume of connect/disconnects by clients to the database.

This value does not need to be scaled with an increase in memory, but it should be examined as the volume and frequency of disconnecting clients is increased.

- **QUERY_HEAP_SZ:** Query heap size
Default: 1000 Range: 2 - 524,288

The query heap size stores all the memory objects required to execute a query. This value is the maximum amount of memory that can be allocated.

Recommendation: This value can be left at the default size unless an error message occurs stating that there is insufficient space in the query heap. The value does not have to be scaled as the amount of memory on the system is increased.

Another important parameter

This is another important instance parameter: **INTRA_PARALLEL:** Intra parallelism.

Default: NO Range: YES,NO

This parameter controls whether intra-parallelism is used for the query execution. The intra-partition parallelism is important for large complex queries since it allows the data to be broken down into sections and analyzed by different agents in parallel. This greatly speeds up the execution of a query against a very large data table. The parameter does, however, often result in a greater number of agents being created to process queries, which can lead to greater memory and CPU use. This extra analysis, and the processing required to coordinate the extra agents, can be detrimental for OLTP queries. It is recommended that the parameter only be turned on for large multi-CPU systems that process complex queries.

2.5.5 Configuring the database

Despite there being a large number of database parameters that can be scaled to use more memory, few of them actually require a large amount of memory. This section covers which parameters need to be increased and which ones may not require as much memory. A full description of all the database parameters can be found in the following DB2 manual:

PDF: *Administration Guide: Performance*, Chapter 13, “Configuring DB2”, section “Database configuration”

Database parameters

Table 2-7 lists all of the parameters that have memory allocated at the database level. Each parameter lists whether it can be changed dynamically, and also if the total memory allocated for it should be scaled up as more is available with the 64-bit addressing.

Table 2-7 Database memory configuration variables

Parameter	64-bit scaling	Online configurable	Description
app_ctl_heap_sz	No	No	Application control heap size
appgroup_mem_sz	No	No	Max size of application group memory set
applheapsz	No	No	Application heap size
catalogcache_sz	No	Yes	Catalog cache
database_memory	Yes	No	Database shared memory size
estore_seg_sz	No	No	Extended storage memory size
locklist	No	Yes	Maximum storage for lock list
logbufsz	No	No	Log buffer size
pckcachesz	No	Yes	Package cache
sheapthres_shr	Yes	No	Sort heap threshold for shared sorts
stat_heap_sz	Yes	No	Statistics heap
stmtheap	No	Yes	Statement heap
util_heap_sz	Yes	Yes	Utility heap size
bufferpools	Yes	Yes	Database bufferpools

Key parameters

Following are detailed descriptions of the key database parameters:

- ▶ **Bufferpools:** Database bufferpools
Default: 32 Range: 10- memory on the system

The size of the bufferpools in a database has traditionally been limited by the amount of memory available with 32-bit. The elimination of this limit will allow bufferpools to be greatly increased in size, which will ensure that less data has to be reread from disk.

Recommendation: Despite having a great deal of memory available, you should not just expand the bufferpools to use up all of it. Take a good look at the bufferpool usage using the bufferpool snapshot to determine how much the bufferpool is being used and what percentage of the time the data is being found in it versus having to be read from disk. If the data is having to be frequently read from disk then the bufferpools should be increased in size.

This is only true for OLTP type environments since DSS and warehousing workloads often do not gain a great deal by having memory in the bufferpools. This is due to the large queries that they run that may pull almost entire tables into the bufferpool.

This value should be scaled in direct proportion to the amount of memory that will be available for DB2.

- OLTP: 80–90 percent of available memory
- DSS: 20 percent of available memory

- ▶ **SORTHEAP:** Sort heap size
Default: 256 Range: 16 - 1,024,575

The sort heap controls the maximum number of private memory pages for a private sort and the maximum number of shared memory pages for a shared sort. If a sort requires more space than is available in the sort heap then the extra data and sorting space will have to be written back and forth from memory to disk. This sort overflow has an extremely negative impact on performance.

Recommendation: Sort overflows should be avoided as much as possible. If you observe that sort overflows are occurring in the database or application snapshot then the sort heap should be increased. This value is particularly important for large reporting queries and complex queries involving the joining of many large tables.

This parameter should be scaled to ensure that sort overflows do not occur. Its impact is not as significant for an OLTP workload.

- ▶ **DATABASE_MEMORY:** Database shared memory size
Default: Automatic Range: 0 - 4,294,967,295

The database shared memory parameter controls the minimum amount of shared memory that is reserved for a database. This memory does not include the memory assigned for the instance or for the database applications.

Recommendations: The value should be left to Automatic since this will allow DB2 to allocate more memory, as needed, in a 64-bit environment. This will allow DB2 to properly scale as more memory is made available to the system.

- SHEAPTHRES_SHR: Sort heap threshold for shared sorts
Default: sheapthres Range: 250- 2,147,483,647

This is a hard limit on the amount of memory that can be used for shared memory sorts in the database. If the value is set to zero then the value of sheapthres will be used instead. This shared sort space will only be used when the intra_parallel setting is set to yes or the connection concentrator is being used. These scenarios are when a shared sort space is needed.

Recommendation: This parameter should be scaled to the amount of memory available on the system. This ratio of the memory should be set similar to the sort heap threshold set at the instance level if lots of shared sorts are used.

- OLTP: scale to 20 percent of available memory
- DSS: scale to 80–90 percent of available memory

Memory parameters

Here we discuss some memory-related database parameters in detail:

- APP_CTL_HEAP_SZ: Application control heap size
Default: 64 Range: 1 - 64,000

This parameter controls two different functions depending on whether the database is partitioned or running with intra-parallel enabled. For partitioned databases or those running with intra-parallel, this parameter controls the average amount of shared memory allocated for an application. For non-partitioned and non-parallel databases, this parameter controls the size of the private memory heap. This heap is used primarily for communication between agents.

Recommendation: This parameter should be left as the default unless the queries you are using are highly parallelized. This parameter should be increased if they are to ensure that there is sufficient space for the queries to run efficiently with many parallel agents.

- APP_GROUP_MEM_SZ: Maximum size of the application group memory
Default: 10,000 Range: 1 - 1,000,000

This parameter determines the size of the application group memory size. This heap is used to share information between agents working on a query. In

a partitioned database or one with intra-parallel enabled, multiple applications will share this memory group.

Recommendation: This parameter should not be scaled and should be left at the default unless you are experiencing performance problems.

► **APPLHEAPSZ:** Application heap size

Default: 256 Range: 1 - 60,000

This parameter defines the number of private memory pages available to a database agent or subagent. The heap will be initialized when an agent is initialized for an application.

Recommendation: This parameter should not be scaled as it is for individual agents. This parameter should be left at its default value unless performance problems occur.

► **CATALOGCACHE_SZ:** Catalog cache

Default: -1 Range: 8- 524,288

The catalog cache is used to cache system catalog information. This parameter's memory is taken out of the database shared memory. There will be one catalog cache per partition if the database is partitioned.

Recommendation: It is important to ensure that the catalog cache has sufficient memory to buffer almost all the catalog information that it retrieves. The catalog information is required for SQL optimization and database administration so it is essential that an agent does not have to go to disk to retrieve the information that it needs.

This parameter does not need to be scaled, but you should ensure that the catalog cache hit ratio is close to 100 percent. The hit ratio can be determined from taking a database snapshot and determining what percentage of catalog searches were found in memory. Example 2-11 shows the database snapshot command and the output related to catalog cache. Figure 2-48 on page 109 shows the formula to calculate the catalog cache hit ratio.

Example 2-11 Determining the catalog cache hit ratio

```
db2 get snapshot for sample
>> Output from the snapshot
Catalog cache lookups           = 1025
Catalog cache inserts           = 105
Catalog cache overflows         = 0
Catalog cache high water mark   = 0
```

$$\text{Catalog Hit Ratio} = 100\% * \left(1 - \frac{\text{Catalog inserts}}{\text{Catalog lookups}} \right)$$

Figure 2-48 Catalog cache hit ratio

- **ESTORE_SEG_SZ:** Extended storage segment size
Default: 0 Range: 0- 2,147,483,647

The extended storage segment size controls how much extended storage space can be used. This was previously used to get around the 32-bit memory addressing limit by creating a separate memory pool from DB2 that could be referenced as an extension to the bufferpool. With the use of 64-bit this is no longer needed.

Recommendation: Always leave this value as zero if you have a 64-bit system since it is much more efficient to simply make the desired bufferpool larger.

- **LOCKLIST:** Lock list size
Default: 50 Range: 0- 60,000

This parameter controls the amount of memory that is allocated for the lock list. The information for all the locks being held by applications in DB2 is contained within this space. The locklist should be large enough to ensure that locks do not have to be escalated due to insufficient space in the lock list.

Recommendation: This parameter does not have to be scaled to the amount of memory available. It can, however, be given a great deal more space to ensure that lock escalation will never implicitly occur due to insufficient locklist space. As more applications access DB2 concurrently, this parameter can be increased.

The default value is normally insufficient for high volume workloads or those that have long transactions and therefore hold locks longer. Typical lock list sizes are in the low thousands for high volume systems. Use the DB2 memory visualizer tool to determine what the ideal level is.

- **LOGBUFSZ:** Log buffer size
Default: 8 Range: 4 - 65,535

The log buffer size controls how large the buffer is for the logs before they are written to disk.

Recommendation: The parameter should not be scaled to the available memory. A large log buffer will improve the logging of the system but it will also negatively effect the recoverability of a database since any records in the log buffer are lost when a system crashes.

- PCKCACHESZ: Package cache
Default: -1 Range: -1, 32- 524,288

The package cache stores all the prepared SQL statement execution plans for static and dynamic SQL. Having the access plans already compiled greatly improves query performance since compiling an SQL statement can often take as long as executing it for some workloads. By leaving the default value at -1, the parameter will be allocated a memory of eight times the maximum number of applications allowed on the database.

Recommendations: The package cache should not be scaled with the available memory on the system. For OLTP systems there should be sufficient memory in the parameter for the package cache hit ratio to be close to 100 percent. The package cache hit ratio is the percentage that SQL statements are found already precompiled in the cache instead of having to be recompiled. If the SQL queries in an application vary a great deal or do not use parameter markers, which will require them to be recompiled each time, then the ratio will likely not be improved.

Example 2-12 shows the DB2 snapshot command and the related package cache output. Figure 2-49 shows the formula for calculating the package cache hit ratio.

Example 2-12 Package cache hit ratio

```
db2 get snapshot for database on sample
>> output sample
Package cache lookups                = 23424
Package cache inserts                = 1011
Package cache overflows              = 0
Package cache high water mark (Bytes) = 6244031
```

$$\text{Package Cache Hit Ratio} = 100\% * \left(1 - \frac{\text{Package inserts}}{\text{Package lookups}} \right)$$

Figure 2-49 Package cache hit ratio

- STAT_HEAP_SZ: Statistics heap size
Default: 4384 Range: 1096- 524,288

The statistics heap is used to store information during the execution of the runstats utility and is only allocated while the utility is running. The default value is normally sufficient unless distribution statistics are being collected. The default is only sufficient to store information on one or two columns, so for distribution information the value should be greatly increased. Additional

information on how the memory is used can be found in the runstats utility documentation in the *DB2 Command Reference*.

Recommendation: The value should not be fully scaled with the amount of memory on the system. If, however, you are taking a lot of online updates of the database statistics with distribution, then the parameter should be greatly increased to speed up the process.

- ▶ **STMTHEAP:** Statement heap
Default: 2048 Range: 128- 65,535

The statement heap is used for the processing of the compilation of SQL statements. SQL statements of greater complexity, such as those that make heavy use of multi table joins, will require the value to be increased. The memory will be freed as soon as the statement is finished being prepared.

Recommendation: Leave the parameter as a default unless an error is returned during the binding of static SQL or the execution of dynamic SQL. The parameter does not need to be scaled with the memory available on the system.

- ▶ **UTIL_HEAP_SZ:** Utility heap size
Default: 5000 Range: 16- 524,288

The utility heap size is used by the many utilities in DB2 such as backup, restore, and load. Increasing this parameter will greatly speed up the execution time of the queries since more work can be done in memory instead of having to write sub-steps to disk. The memory will only be allocated when the utilities are running.

Recommendation: This parameter should be set much higher than the default if you have sufficient memory. Having more memory for the utilities will reduce the batch window required for them to complete. If you are running the utilities online, then the decreased runtime due to the extra memory will result in a shortened performance impact for your users. Having this variable quite large will improve your recovery time as well.

Configuration Advisor issues

The DB2 UDB V8.1 Configuration Advisor has been designed to take advantage of 64-bit, but the advice that it returns is constantly being improved. You should examine its results carefully to ensure that the memory variables have been scaled properly for your workload.

With the Configuration Advisor running on Windows Server 2003 Datacenter with DB2 UDB V8.1. Fixpak 2, almost all of the recommendations were excellent. However, the values for the utility heap were in the hundreds of thousands of 4-K pages, which was not correct. A case study illustrates how to use the tool in 3.3.6, “Case studies” on page 141.

2.6 Additional considerations

As you look at how you can adapt your system for 64-bit and Windows Server 2003 Datacenter, do not forget that memory allocation is most likely not your only issue.

2.6.1 Compatibility issues between V7 and V8

With the release of DB2 UDB V8.1, a number of fundamental architecture changes and improvements were made. One of these involved changing the database protocol that DB2 uses from DB2 Database Architecture (DB2DA) to Distributed Relational Database Architecture™, commonly referred to as DRDA. The impact of this was that not all old DB2 clients and servers are compatible for all functions with the V8.1 clients and servers.

V7 and V8 communication

The change of protocol to DRDA for DB2 will require that a number of additional steps be taken.

V8 clients to a V7 Server

The DRDA communication support must be enabled for any DB2 UDB V7 servers if you want V8 clients to be able to work with them. The steps for doing this are documented in the DB2 UDB V7 documentation. It is recommended that you migrate all of your V7 servers over to V8 in order to avoid any issues. If, however, this is not possible, then you must understand that the following restrictions apply:

- ▶ Data type support is lacking for:
 - Large object types (LOBs)
 - User-defined distinct types
 - DATALINK types
- ▶ Security issues:
 - Authentication type SERVER_ENCRYPT
SERVER_ENCRYPT is a method to encrypt a password. The encrypted password is used with the user ID to authenticate the user.
 - Changing passwords
You are unable to change your password from a V8 client for a V7 server.
- ▶ Communication issues:
 - No instance requests that require an ATTACH instead of a connection.
The ATTACH command is not supported from a V8 client to a V7 server.

- Network protocols other than TCP/IP are not supported.
- Application considerations:
 - The DESCRIBE INPUT parameter is only supported for ODBC and JDBC.
In order to support Version 8 clients running ODBC/JDBC applications that access Version 7 servers, a fix for DESCRIBE INPUT support must be applied to all Version 7 servers where this type of access is required. This fix is associated with APAR IY30655. Use the “Contacting IBM” information in any DB2 UDB documentation set (PDF or HTML) to find out how to get the fix associated with APAR IY30655
 - Two-phase commit cannot be used.
The Version 7 server cannot be used as a transaction manager database when using coordinated transactions that involve Version 8 clients, nor can a Version 7 server participate in a coordinated transaction where a Version 8 server may be the transaction manager database.
 - XA-compliant transaction managers.
An application using a Version 8 client cannot use a Version 7 server as an XA resource. This includes WebSphere®, Microsoft(R) COM+/MTS, BEA WebLogic, and others that are part of a transaction management arrangement.
 - Monitoring and utilities.
All monitoring that can be initiated from the client is not supported down level.

Note: In order to access a V7 server, you must have installed Fixpak 8 or later and have run the `db2upd v7` command.

V7 Clients to a V8 Server

A V7 client will be able to communicate with a V8 server. New functionality that was added in V8 will not be able to be called from a V7 client.

V7 64-bit

DB2 UDB first released a 64-bit version for AIX with V7.2. This version did have a number of restrictions on it that will impact V8.1 64-bit. With the previous release of DB2 UDB, only DB2 servers and clients of the same memory architecture could communicate with each other. This meant that only 64-bit clients could communicate with 64-bit database servers. This restriction was removed for V8.1, and now any combination of 32-bit and 64-bit clients and servers can communicate with each other.

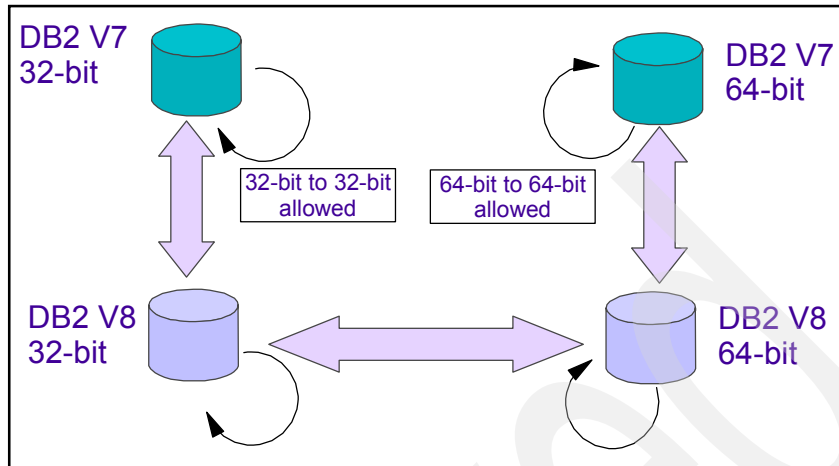


Figure 2-50 V7 and V8 64-bit compatibility

Development Center

As of DB2 UDB V8.1 FP2, the DB2 Development Center does not support a 64-bit environment.

2.6.2 Paging

Windows has the option to assign more memory than is physically available on the system. This then allows the system to page memory to disk if the physical memory is exhausted.

Despite having the ability to create a paging space, it is not recommended that you rely on this memory. Having to rely on paged memory will result in a very large performance hit, since instead of reading data from flash memory, the information will have to be read and transferred from disk. The maximum amount of memory that you allocate for the database should therefore be less than the physical memory available on your system.

You can find out if you are paging using the Windows *perfmon* tool. If you open up the tool and select an option to add, you will see an entry for Memory under the Performance Object pulldown. There are two options: Page reads and page writes (Figure 2-51 on page 115). These two variables will indicate how much the system is paging. You should try to tune the system to avoid paging since it slows down the system very much when paging occurs.

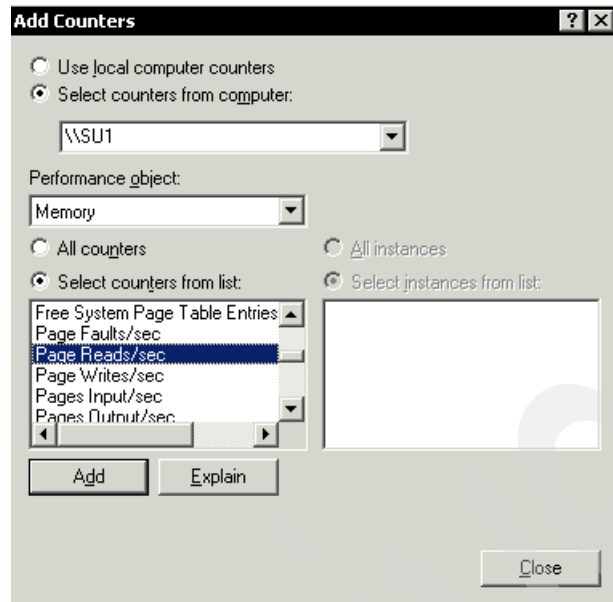


Figure 2-51 Windows perfmon: Paging information fields

Scaling up DB2

DB2 UDB V8.1 Enterprise Server Edition and Windows Server 2003 Datacenter Edition are raising the bar for scaling up on the Intel platform. With the 32-bit versions supporting large 32-way SMP servers with 64 GB of memory, and with the 64-bit versions supporting large 64-way servers with 512 GB of memory, the decision to scale up on this platform could not be easier.

DB2 UDB V8.1 ESE brings the performance and scalability features of enterprise-class relational database management systems to Windows Server 2003. It scales on Windows Server 2003 from one to 4-way SMP servers with Standard Edition, from four to 8-way SMP servers with Enterprise Edition, and all the way up to 64-way SMP servers with Datacenter Edition.

In this chapter we cover scaling up with DB2 UDB and Windows Server 2003. We cover relevant information about leveraging additional memory, processors, and storage resources.

This chapter contains the following sections:

- ▶ Section 3.1, “Scale up overview” on page 118
- ▶ Section 3.2, “Adding additional processors” on page 118
- ▶ Section 3.3, “Adding additional memory” on page 129
- ▶ Section 3.4, “Adding additional storage” on page 146

3.1 Scale up overview

Emergent technologies like e-business, e-commerce, e-analytical, etc. are under the scope of many companies around the world. This will be defined as a domain of opportunities and challenges to adapt, evolve, and innovate. These requirements need the technology infrastructure to be designed and built so that these types of operations can be supported and maintained.

Many companies are already in this “domain of opportunities”; they have technology that must be adapted to current demands. This is the time to start thinking about increasing hardware and software resources to their actual infrastructures.

Scale up is the first option to consider when it is necessary to upgrade existing environments to a more scalable platform in hardware and software technologies to respond to their business demands. DB2 UDB and Windows Server 2003 are software designed to meet these scaling requirements.

Companies using Windows 2000 Server can scale up with new versions of Windows Server 2003 or Windows Server 2003 Datacenter Edition.

If you plan to scale up your actual database server environment on Windows platforms, you should take the following considerations for adding additional memory, processors, and storage resources.

3.2 Adding additional processors

In the past, expanding your hardware capacity for your new business needs was an expensive and time consuming process. One of the most complicating components of this was increasing the number of CPUs. On some hardware and in previous releases of Windows it was not a simple task to add additional CPUs, and a number of steps had to be completed. This has been greatly simplified with Windows Server 2003 and will make expanding your system with hardware such as the xSeries x445 and x455 an attractive proposition.

This section covers how to take advantage of the additional CPUs with DB2 UDB. A section has also been included to measure the performance impact the additional CPUs have had on your system.

3.2.1 Can your system scale

Before you begin the process of deciding how many more CPUs you need to handle your workload, it is important to figure out if your system can scale with

your demands. Some systems can be joined together into a series of nodes that combine into a larger system, such as the x445s, but others can only be joined into clusters, such as the x450. If you want to be able to easily increase the number of CPUs for a system then you must select a system that be directly joined into a larger configuration, such as the x455s. Currently the x455 is the only 64-bit xSeries machine that can scale this way.

3.2.2 Moving from a 4-way to an n-way

There are two common methods for expanding the size of your system. The first method is to add another server to your system. This allows the data to be spread out over more partitions and the database queries to be parallelized more. This approach will only work if the database utilizes partitioning. The partitioning of a DB2 database is covered in detail in Chapter 4, “Scaling out DB2” on page 177. The two approaches are represented in Figure 3-1 on page 120.

The expansion of a system can either be accomplished by using a series of Blade servers or with server units. The Blade servers have a small amount of storage in addition to the CPUs and memory. The operating system can then be installed and run on each Blade, or the Blades can be linked off a storage area network (SAN) or other storage systems. This method allows the total number of nodes in a cluster to be quickly and easily expanded. The server units tend to have more memory, CPUs (4-ways), and power than the individual Blades (2-ways).

The second approach has been to keep all of your data on the same disk storage systems and to replace the central server with more powerful hardware. The “Rip and Replace” method is time consuming, expensive, and also does not allow you to easily scale the system as demand steadily increases. Improvements in the operating system, and the underlying hardware, now allow you to scale your hardware by adding additional units instead of purchasing a whole new server.

The other option is to connect together two or more distinct servers that each have their own dedicated CPUs, memory, and (possibly) disks. The combined system would have one copy of the operating system installed on one of the servers and would be able to use the combined resources of all the servers. This method is attractive since the systems can be used as individual units and they can also later be combined into a single larger system.

The method used in this publication testing was with two x455 4-way servers. The x455s can be connected together with a scalability cable and easily converted into an 8-way machine. Additional 8-way configurations can be added to allow the machine to scale up to a single 32-way. The total memory and CPUs on both systems then appeared to the operating system as a single machine with

one set of resources. DB2 UDB was able to leverage all 8 CPUs without having to set up a partitioned environment.

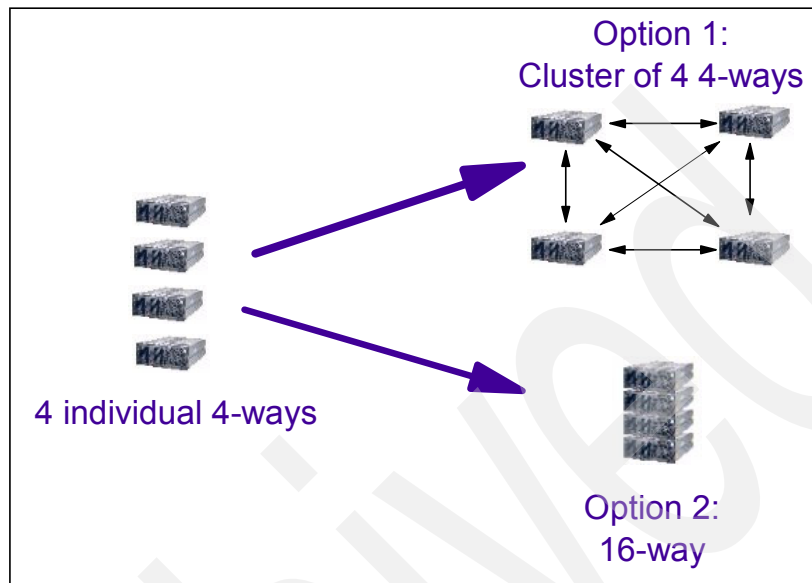


Figure 3-1 Approaches for scaling the number of CPUs for a system

3.2.3 Windows considerations

With Windows Server 2003 the process of adding additional CPUs to a system is seamless and requires no work by the administrator. The only restriction is that you do not add more memory than the edition of Windows Server 2003 you are using can recognize. If too much memory is added then the operating system will not recognize it and therefore it is wasted.

3.2.4 Taking advantage of greater parallelism

The DB2 UDB optimizer will transparently take advantage of the additional CPUs in a number of different ways. The DB2 UDB query optimizer will make use of any additional CPUs and capacity that you give it. The degree of parallelism available for a system can speed up the performance of large complex queries in two main ways.

Inter-query parallelism

Inter-query parallelism allows multiple queries to run in parallel on a database system. This is one of the fundamental components of any database system. Increasing the number of CPUs available on the system will not change how DB2

UDB schedules the execution of the queries, since the distribution of CPU cycles is handled by the operating system. Your workload will, however, run faster since each query will have more CPU cycles assigned to it in a smaller time period.

If your workload was running poorly due to a CPU bottleneck, then additional CPUs or power will elevate this problem. Throwing extra CPU power at a problem is not always the best solution. You should take a look at your database and determine if a few queries or transactions have been poorly designed and are using much more time and CPU power than they actually require. The performance of queries can be easily observed using a DB2 snapshot on Dynamic SQL. The snapshot will give you a summary of all executions of the dynamic SQL queries on the system with their user and system CPU usage displayed separately. The access plans of the longest running, and most active CPU using SQL queries should be examined to see if there is any indexing or issues.

Adjustments

With the addition of extra CPUs you should be able to squeeze more out of your system. If you were near the limit of your previous system then you may have put restrictions on the amount of work, and applications, that could use the database. The list below summarizes the parameters that can be increased to allow a greater amount of inter-query parallelism and throughput.

- ▶ **MAXAGENTS:** Maximum number of agents
Configuration parameter level: Instance

The parameter controls the maximum number of database agents that can be running for a DB2 instance. This parameter is often used to cap the workload on a system due to CPU limitations. For large data warehousing systems that have a large number of queries that use multiple agents, putting a lower limit on this parameter can cap the resources used by DB2.

If the number of CPUs is increased then the parameter should be increased too in order to allow for DB2 to use more of the system resources.

- ▶ **MAX_CONNECTIONS:** Maximum number of connections
Configuration parameter level: Instance

This parameter controls the maximum number of connections into all of the databases contained in the instance. This parameter is a very effective method to control the amount of work on the system if the CPU usage is constrained. The parameter should be increased as more CPU power is available.

- ▶ **MAX_COORDAGENTS:** Maximum number of coordinating agents
Configuration parameter level: Instance

Each connection to a database requires a coordinating agent. For queries that use intra-query parallelism, and the INTRA_PARALLEL parameter is set

to YES, then the coordinating agent is also used to control the combining of information from all the agents that are processing a query or data set in parallel. By controlling the number of coordinating agents, you limit the number of queries that can be active at one time.

- **MAXCAGENTS:** Maximum number of concurrently executing coordinator agents

Configuration parameter level: Instance

This parameter was new for DB2 UDB V8.1 and is used by the new Connection Concentrator functionality. If this parameter is less than the MAX_COORDAGENTS, then only MAXCAGENTS will be created. A query requires that a coordinating agent service it for it to be active. The available-coordinating agents will be allocated to the individual connections in a round robin fashion, as they require to be actively run in the database engine. This reduces the number of agents that are spawned and the amount of memory required for the agents. The connection concentrator can be viewed as a connection multiplexer.

This parameter will also have to be increased to allow DB2 to fully utilize the extra CPU power of a machine if the parameter had previously been used to control the amount of system resources used by DB2.

Intra-query parallelism

Intra-query parallelism is when DB2 takes a complex query and breaks it into sub-components that can be run separately, and then brings the results together to complete the query. This query optimization is automatically used by the optimizer, but having additional CPUs available could allow DB2 to consider more effective query execution plans.

Example 3-1 Intra-parallel query

```
select pop.lastName, bnk.bankAccount
from
  (select idNumber,lastName
   from population
   where city = 'Toronto'
  )as pop,
  (select idNumber, bankAccount
   from bankRecords
   where accountBal > 1000000
  ) as bnk
where pop.idNumber = bnk.idNumber
```

Example 3-1 is a sample query that can utilize the intra-query parallelism feature. The query could have one agent query the population table and another agent

query the bank records table. The coordinating agent would then bring the two results together and complete the query.

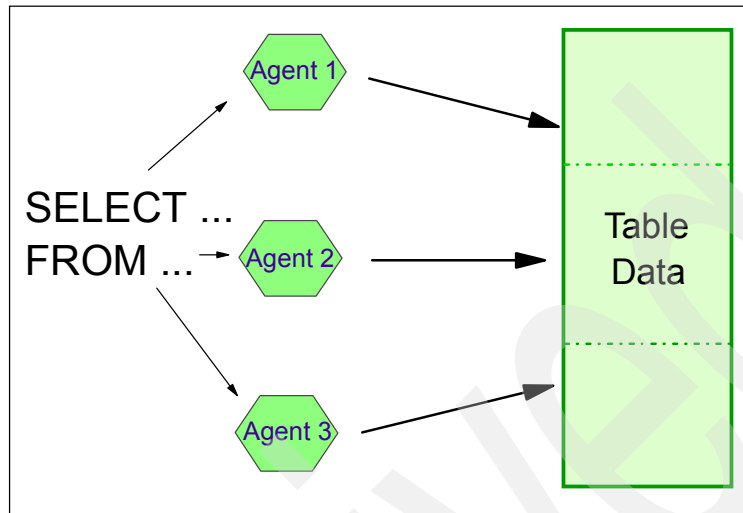


Figure 3-2 Intra-query parallelism

Adjustments

The parameters listed below will help better utilize the available CPUs for intra-query parallelism.

- ▶ **MAXAGENTS:** Maximum number of agents
Configuration parameter level: *Instance*

If the maximum number of agents is kept low, then DB2 may not have the resources to fully parallel the query. Increasing this value will allow more agents to be available for the instance.

- ▶ **MAXAPPLS:** Maximum number of applications on the system
Configuration parameter level: *Database*

The maximum number of applications is controlled at the database level. This control is similar to the maximum number of connections set at the instance level. If you used this to control the maximum amount of activity on your database then you should increase the parameter as more CPUs are added.

Intra-partition parallelism

Intra-partition parallelism is when DB2 has multiple agents execute the same query on different sections of the data on the same partition. This is useful when you are working with very large tables and it would take too long for one agent to scan the entire table.

Having intra-parallelism enabled will allow DB2 to spawn multiple agents to work on the same table or index. It is important to use prefetching as well since the agents can process the data quicker if they do not have to wait for an agent to retrieve it. This type of parallelism is rarely used in an OLTP system unless a report is being run, but it is the norm of DSS and data warehouse systems where the entire system is scanned and processed on a regular basis.

If your system was constrained by the CPU usage during use time periods, then the additional CPUs will help to reduce this problem.

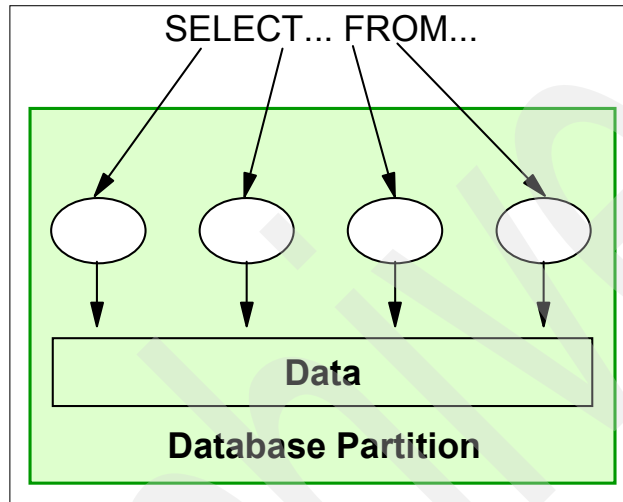


Figure 3-3 Intra-partition parallelism

Adjustments

The parameters listed below will help better utilize the available CPUs for intra-query parallelism.

- ▶ **INTRA_PARALLEL:** Intra-parallel parallelism
Configuration parameter level: *Instance*

The default for intra-partition parallelism is to have it set to NO. This is good for OLTP systems since the extra work required to coordinate running a query in parallel and the extra compilation time are often worse than the performance gained for short queries. For DSS and warehouse systems, however, this parameter can cause a dramatic improvement. The improvement comes at a CPU cost since extra database agents are used to process the query in parallel. If the CPU of your system is being used too heavily then turning this parameter off will reduce the workload, but will also slow down your query processing. The more complex queries could end up

greatly increasing the amount of CPU usage since they could be highly paralleled by DB2.

- ▶ **MAX_QUERYDEGREE:** Maximum degree of parallelism
Configuration parameter level: *Instance*

The maximum degree of parallelism controls the maximum degree of intra-partition parallelism. This is a good way to throttle how heavily the query is run in parallel and therefore how many agents are run. As you increase the number of CPUs on your system you should scale this value up as well if there will be extra processing capacity available.

- ▶ **NUM_QUANTILES:** Number of quantiles
Configuration parameter level: *Database*

The number of quantiles determines how many quantiles are captured when the Runstats statistics tool is run. Having a greater number of quantiles will give DB2 a better idea of how many agents are needed to effectively scan a table or index range and the agents will then be better allocated.

Inter-partition parallelism

By using physical and logical partitioning of a database, the data can be created in completely independent partitions, which then allow queries to be executed in parallel on all partitions. This parallelism is very effective for data warehouses and DSS. This parallelism is normally used in conjunction with the other two forms of parallelism to ensure that queries run as quickly as possible.

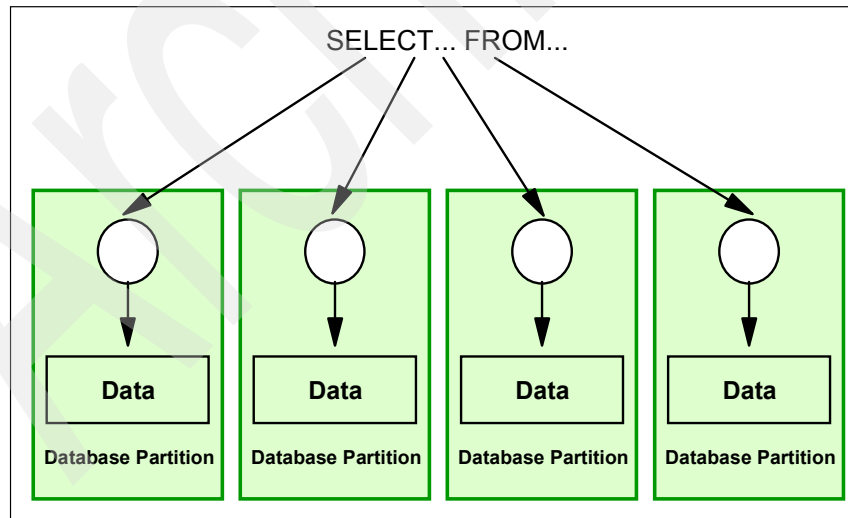


Figure 3-4 Inter-partition parallelism

Additional partitions

If you have CPUs added to the server then you may want to add additional partitions to a database instead of letting the extra CPU resources be used by the existing partitions. Adding a partition should only be used for databases that are already designed for partitioning since moving from a non-partitioned to a partitioned department is not a trivial task if you have not planned it out.

Previously it was considered normal to have two CPUs allocated for every logical partition that you would have on a server. Now with all the extra memory available on servers, many people are recommending that only one CPU is required per partition. When you are deciding on how many partitions to have on your server make sure that each partition will have sufficient memory to run effectively. You should also allocate sufficient memory for the application in case multiple copies of it are running in parallel against the database partitions. The creation of logical partitions is covered in detail in 4.3.5, "Creating additional partitions and redistributing data" on page 210.

Utility parallelism

A number of the DB2 utilities will also take advantage of the extra CPUs. The following sections cover only how to configure the utility for the extra CPU usage and will not go into detail on all the aspects of the tools.

Throttling the utilities

New with V8.1 Fixpak 2 is the ability to control the maximum CPU usage of the utilities. The instance parameter UTIL_IMPACT_LIM controls the maximum impact the utility will have on system performance. If the parameter was set to 25, then the utility will have no more than a 25 percent impact on performance of a workload while it is running. The value must be set to less than 100 for a utility to run in throttled mode. It is recommended that the value be set quite low, such as less than 10, to reduce the utilities impact. This value can be configured online, so that if you are running a restore the parameter should be set back to 100 to ensure the database recovery completes quickly.

Backup

The DB2 backup command has an option for parallelism, PARALLELISM n, that allows a DBA to control how many tablespaces will be read in parallel. By increasing the value from the default value of 1, more tablespaces will be read by additional agents and the database will be quicker.

Another parameter, UTIL_IMPACT_PRIORITY priority, allows you to control what impact the backup will have on workloads running on the system. By setting the value to less than 100, the impact of the backup is reduced. If you are running an offline backup, the parameter should be left at its default value of 100 since you

would want all the system resources to be used to ensure that the backup completes quickly.

Example 3-2 Backup command

```
db2 backup database redTest TO 'e:\' PARALLELISM 4 UTIL_IMPACT_PRIORITY 80
>> Backup successful. The timestamp for this backup image is : 20030610200648
```

Restore

When you are restoring a database it is essential that the restoration completes quickly, since normally a critical error has occurred and your production database is down. The DB2 restore command has an option for parallelism, PARALLELISM n, that specifies how many bufferpool manipulators will be created. Each bufferpool manipulator will help to speed up the restoration process. This allows a DBA to control how many tablespaces will be read in parallel.

Example 3-3 Restore command

```
db2 restore database redTest FROM 'e:\' PARALLELISM 4
>>SQL2539W Warning! Restoring to an existing database that is the same as the
>>backup image database. The database files will be deleted.
>>Do you want to continue ? (y/n) y
>>DB20000I The RESTORE DATABASE command completed successfully.
```

Load

The load utility makes very effective use of the system resources that are available to it, including the CPUs. There is a parameter, CPU_PARALLELISM n, that controls the number of processes that are spawned for the parsing, converting, and formatting of records during the load. The parameter is designed to exploit intra-partition parallelism. If the parameter is set to zero then the database determines an intelligent value, which is usually based on the number of CPUs available.

3.2.5 Observing the performance change

After you have configured your system and database to take advantage of the new CPUs it is important to be able to measure what improvement they resulted in.

Database query time

One of the most important indicators of the CPU impact is to see how much less time SQL queries and workloads take to complete. In order to be able to perform an effective comparison, you will need to benchmark how long the query workload took before the CPUs are added. The quickest methods to do this are

to use the **db2batch** command to observe how long a workload takes or to use DB2 snapshots. The Dynamic SQL snapshot will tell you exactly how long individual SQL statements took to complete.

The overall system throughput should be the most important indicator of the extra CPUs' impact. Linear scaling tests are helpful to determine if adding X more CPUs at a cost of Y is worth the performance improvement of Z. The values for these numbers is completely up to you.

Utility time

The DB2 utilities all return timestamps in the db2diag.log file, which indicate when they began and finished. These time differences between the timestamps will give you an exact measure of the performance improvement.

System use

The use of Windows system resources can be observed using both the Windows Task Manager and perfmon tools. The task manager output for a multiple CPU system is shown in Figure 3-5.

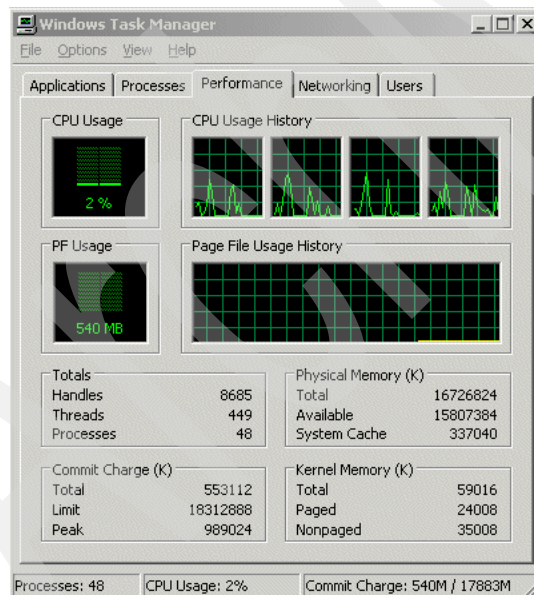


Figure 3-5 Viewing system use with the Windows Task Manager

The Windows perfmon utility can also be configured to view the workload of each individual CPU on the system separately. This is useful if you need to determine if your work is being spread evenly across all the CPUs or if it is being serialized and only one or a few CPUs are being used heavily. The use of the perfmon tool

is described in detail in 3.2.5, “Observing the performance change”. To view multiple CPU use, the processor option should be selected from the Add menu. In the right side of the menu a list of individual processor numbers and total CPU usage will be displayed for multiple CPU systems. The CPUs can be selected for the desired performance measurements, and they will then be shown on the main perform graphic display. This is shown in Figure 3-6.

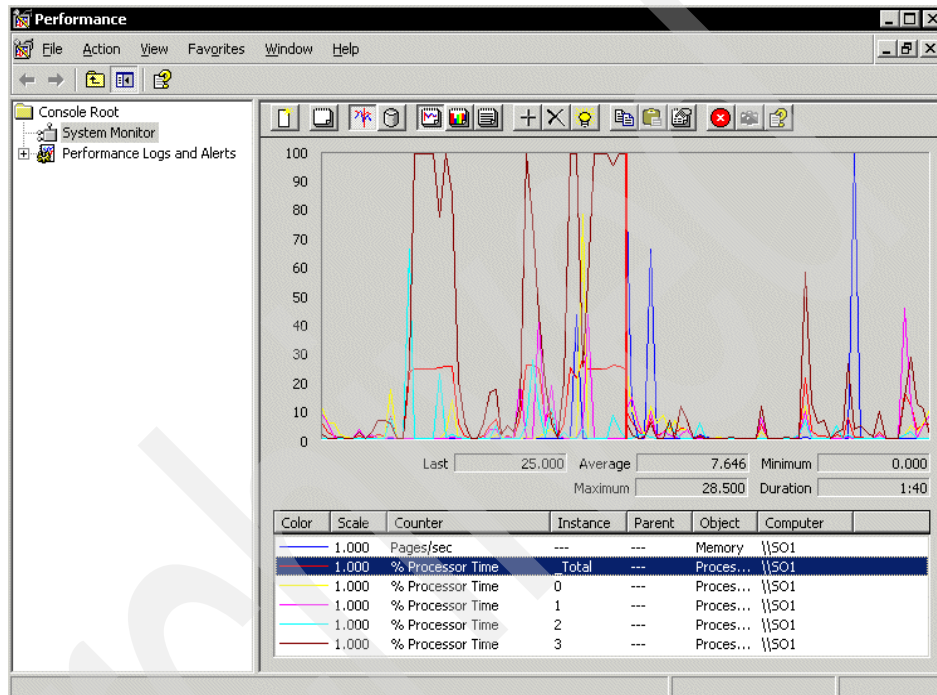


Figure 3-6 Selecting multiple CPUs for the Windows perfmon tool

Tip: If you are using a system with many CPUs, such as a 16-way or 32-way, it may be easier to observe CPU performance using the Task Manager tool. The perfmon tool displays all the separate CPUs performance on one graph, which can be difficult to read.

3.3 Adding additional memory

A lack of memory has traditionally been one of the largest bottlenecks for an application that uses databases heavily. With previous releases of Windows, the operating system limitations or 32-bit addressing may have been the only reasons your database could not use more memory. The movement to Windows

Server 2003 is an ideal time to re-evaluate how much memory is being allocated to DB2.

3.3.1 Physically adding more memory

Physically adding more memory to your system is a simple process. Windows Server 2003 allows two different processes for changing the amount of memory on the system.

Hot-add memory

The Datacenter and Enterprise Editions of Windows Server 2003 allow for additional memory to be added to a system without a reboot being required. This new functionality, called hot-add memory, does not support the removing or swapping of memory, but only the addition of new memory. The new memory will automatically be recognized by Windows and made available. Not all underlying hardware will support the functionality, so it is important to consider this option when you purchase your Windows servers.

Further documentation on this new feature can be found in the following Windows help:

<http://www.microsoft.com/whdc/hwdev/tech/hotadd/hotaddmem.mspx>

Cold-add memory

The amount of memory on a server can also be changed the traditional way by bringing the system down first. The memory can then be added or swapped and the system rebooted. DB2 can then be configured to take advantage of the greater amount of memory on the system.

3.3.2 Configuring Windows

A number of configurations will need to be set for Windows to take advantage of the additional memory, especially if the system is not 64-bit.

Optimizing memory

As Windows Server 2003, by default, comes optimized for file caching, for a system that will function primarily as a database server, the first consideration that pertains to memory is to optimize the Windows memory model.

This can be accomplished by selecting the “Maximize data throughput for network applications” option found on the Server Optimization tab of the “File and Printer Sharing for Microsoft Network properties” dialog. Selecting this option has the same effect as using the Windows Registry Editor to change the value of the LargeSystemCache REG_DWORD to 0 on the Memory Management subkey.

The first reason to change this setting from “Maximize data throughput for file sharing” to “Maximize data throughput for network operations” is to minimize the operating system’s use of memory for file caching. This allows you to maximize DB2’s working set memory for buffer pools, sort heaps, and lock lists.

The second reason is that if Windows is optimized for “Maximize data throughput for file sharing”, processes outside of the DB2 system controller process can consume almost all remaining available memory for file caching. This condition will almost always result in significant paging activity (for example, a user exit process that copies archive log files or database backup images).

Finally, when the Windows Server 2003 is optimized to “Maximize data throughput for network operations”, a larger portion of the operating system’s non-paged fixed memory pool is allocated for communication buffers, which, as the setting implies, maximizes throughput.

4 GB Tuning

The 4 GB Tuning feature provides applications with a 3-GB flat memory address space. On most 32-bit operating systems applications are limited to just 2 GB of address space. This limits the size of your database buffer pool, which we all know is the most significant performance variable within a database management system. For example, on 32-bit Linux the buffer pool is limited to approximately 1.6 GB of memory. On 32-bit Unix systems the buffer pool is limited to approximately 1.75 GB of memory.

All editions of DB2 UDB can take advantage of the 3 GB flat memory address space afforded by the 4 GB Tuning feature. And now that Microsoft has made this feature available on all editions of Windows Server 2003, you should plan on allocating larger buffer pools and/or larger sort heaps, up to 3 GB less DB2 UDB’s working set and memory for other heaps.

You enable this feature by adding /3GB to the Windows Server 2003 boot.ini file.

AWE Memory

The Microsoft Address Windowing Extensions (AWE) API provides 32-bit Windows applications with the ability to address up to 64 GB of physical non-paged memory. Although it provides support for 64 GB of real physical memory space on 32-bit versions of Windows, it is sometimes difficult to implement on systems with over 16 GB of memory. This is because on systems with over 16 GB of memory, you must disable 4 GB Memory Tuning, as the Windows kernel requires 2 GB of memory for managing AWE memory above 16 GB. This in turn reduces the application’s virtual address space from 3 GB to 2 GB. Once the extra 1 GB of virtual address space has been removed from the application, it becomes difficult to allocate a large enough memory within a 2 GB address space to manage real physical memory above the 4 GB line.

Windows 2003 Server Datacenter Edition supports 64-GB memory; however, addressing real physical memory over 32 GB from a single 32-bit application is somewhat impractical.

You enable this feature by adding /PAE to the Windows Server 2003 boot.ini file, which can be used in combination with the /3GB switch on systems with 16 GB of memory or less.

All editions of DB2 UDB can take advantage of the large memory support provided by the Microsoft AWE API, but currently it is only implemented for database buffer pools.

Note: Without the /PAE boot.ini switch Windows will not load the PAE-enabled kernel and the total system memory reported by Task Manager will be about 4 GB, regardless of how much physical memory is actually installed. This is because the Windows kernel itself uses the AWE API for memory management.

It is important to note that not all software and/or drivers will function properly with the Windows PAE kernel on Windows 2003 Server Standard or Enterprise Editions. This is because AWE support is not required for certification on these editions of Windows, only on Windows Server 2003 Datacenter Edition.

The DB2 registry variable, DB2_AWE, must be set in order to AWE-enable a buffer pool. Given the granularity of the DB2 registry variables at the instance level, we must define only one database in the instance or manage the buffer pool IDs across multiple databases within the same instance, as databases cannot share AWE memory.

The syntax is:

```
DB2_AWE=bp_id, bp_size, aw_size
```

Where:

- ▶ **bp_id:** This is the ID of the buffer pool to AWE enable. You can find the buffer pool ID with a simple query on SYSCAT.BUFFERPOOLS.
- ▶ **bp_size:** This is the size of the buffer pool in 4-K pages. The entire buffer pool resides above the 4-GB memory line and is backed by non-swappable physical memory.
- ▶ **aw_size:** This is the size of the addressing window in 4-K pages. The entire address window resides within the DB2 instance's virtual address space.

Example 3-4 on page 133 shows how to AWE enable buffer pool #2 to use 16 GB (4,000,000 4-K pages) of AWE memory with a addressing window of 1 GB

(500,000 4-K pages). We will also need to tune db2heap so that we have enough memory to manage the large buffer pool.

Example 3-4 DB2_AWE registry variable

DB2_AWE=2,4000000,500000

The bp_size must be at least equal to or greater than aw_size. If bp_size is smaller than aw_size your instance will start and your database will activate without warning or errors, but your buffer pool will not be AWE enabled. You will get a warning in the db2diag.log file, even if diaglevel = 1.

3.3.3 xSeries information

The xSeries machines that were used for the benchmark do support the Windows hot-add memory feature. This allows for the system's memory to be easily increased up to the limits of the machine. To use the hot-add feature the Configuration/Setup utility for the machine must be run. See the *User's Guide* on the *IBM xSeries Documentation CD* for additional information.

3.3.4 Adjusting your DB2 configuration

Now that you have the additional memory on the system you must decide on how to allocate the memory. The issue of adding additional memory and where it should be allocated was covered in detail in 2.5.3, "Configuring memory for 64-bit systems" on page 97. A case study in 3.3.6, "Case studies" on page 141, covers typical memory configurations for an OLTP or DSS system.

3.3.5 How you see how much is allocated

Once you have made all of your changes you have to be able to see what effect they have had on the system performance. There are three methods included with DB2 and Windows that will allow you to see how much memory is being used. The memory usage can be viewed at the Windows level with the Task Manager and perfmon tools, and with DB2 using the Memory Visualizer and DB2 snapshot functions.

Windows

The operating system can offer both a rough view and a detailed analysis of the memory being utilized by DB2. The Windows Task Manager can be used to provide the rough estimate of how much memory a DB2 instance is using. The Processes tab in the Task Manager indicates the total amount of CPU and memory being used by processes running on the system. You can search through the list of processes for db2syscs.exe, which is the process name for the

DB2 instance (see Figure 3-7). The memory associated with this process is the total allocated for the instances and all the databases contained within it. This is, however, quite rough and not of much practical use unless you need to determine the total resources DB2 is using on the system.

Image Name	User Name	CPU	Mem Usage
cmd.exe	Administrator	00	3,904 K
db2syscs.exe	db2admin3	00	68,456 K
db2reg64.exe	db2admin3	00	6,568 K
db2licd.exe	SYSTEM	00	7,320 K
db2bp.exe	Administrator	00	12,360 K
db2dasstm.exe	db2admin3	00	43,848 K
taskmgr.exe	Administrator	00	5,832 K
db2sec.exe	SYSTEM	00	6,984 K
WZQKPICK.EXE	Administrator	470x44%	7,400 K
VirtuaWin.exe	Administrator	00	7,912 K
WinList.exe	Administrator	00	4,576 K
notepad.exe	Administrator	00	720 K
db2jds.exe	SYSTEM	00	10,120 K
explorer.exe	Administrator	00	20,432 K
db2fmp64.exe	db2admin3	00	12,848 K
db2mh.exe	db2admin3	00	3,568 K
mmc.exe	Administrator	00	21,888 K
db2rcmd.exe	db2admin3	00	9,848 K
psp.exe	Administrator	13	3,888 K

Processes: 43 CPU Usage: 14% Commit Charge: 771M / 9806M

Figure 3-7 Windows Task Manager output

A few other DB2 processes are running on the system as well, which should be included in your memory totals:

- ▶ db2sec.exe: DB2 security server; used for authentication
- ▶ db2licd.exe: DB2 license server
- ▶ db2jds.exe: JDBC applet server
- ▶ db2dasrrm.exe: DB2 DAS instance; used for the GUI tools

The memory impact of these processes should be minimal and be under six Megabytes.

The Windows Performance tool, *perfmon*, offers a much more detailed analysis of the memory heaps used by DB2. Within the Performance tool there is a set of options available for the DB2 instance, DB2 databases, and DB2 applications. See Figure 3-8 on page 135.

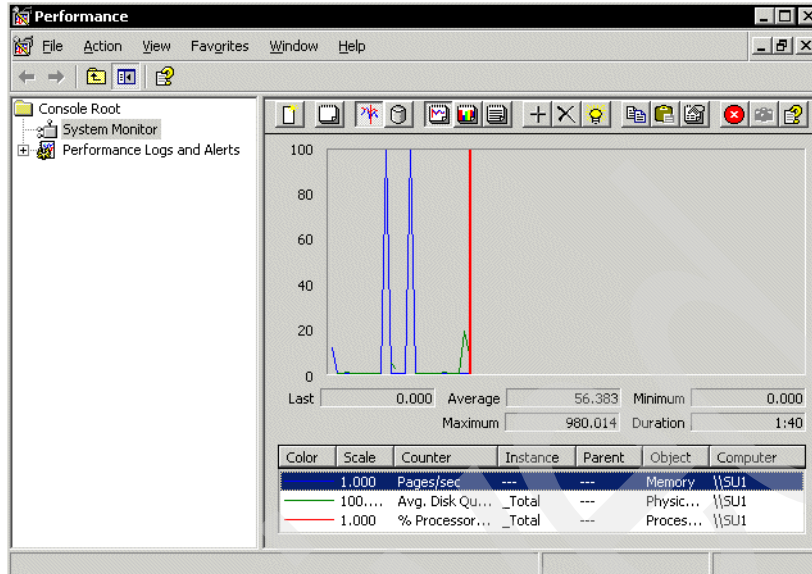


Figure 3-8 Perfmon add monitor option

If you do not have any connections to local databases on your systems then the database options will not appear. These options will be available when you click the **Add** option (Figure 3-9 on page 136).

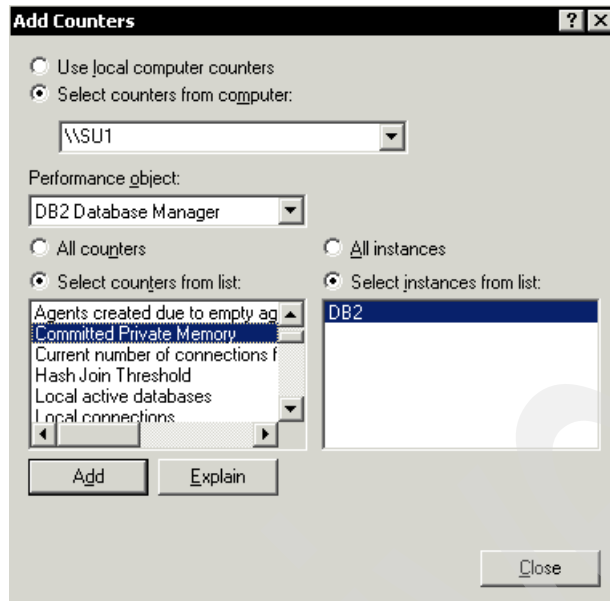


Figure 3-9 Perfmon DB2 memory options

If you have multiple local databases on the system then you can select which database's resources you would like to view (Figure 3-10 on page 137).

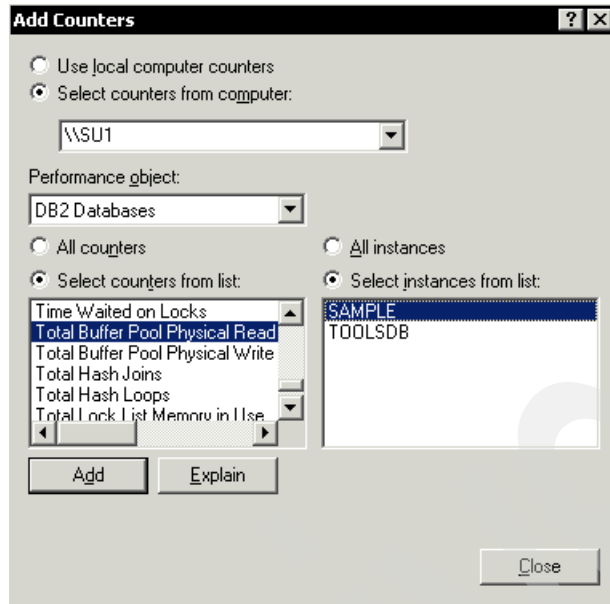


Figure 3-10 Perfmon multiple DB2 database selection

Once you have selected the memory heaps that you would like to monitor, they will be added to the Performance monitor screen. Do not try to monitor all the heaps at once since the output on the screen will be too confusing. It is a better idea to concentrate on the memory usage of a few specific items and have an idea of what you are looking for. The text output is shown in Figure 3-11 on page 138, while the graphic output was shown earlier in Figure 3-8 on page 135.

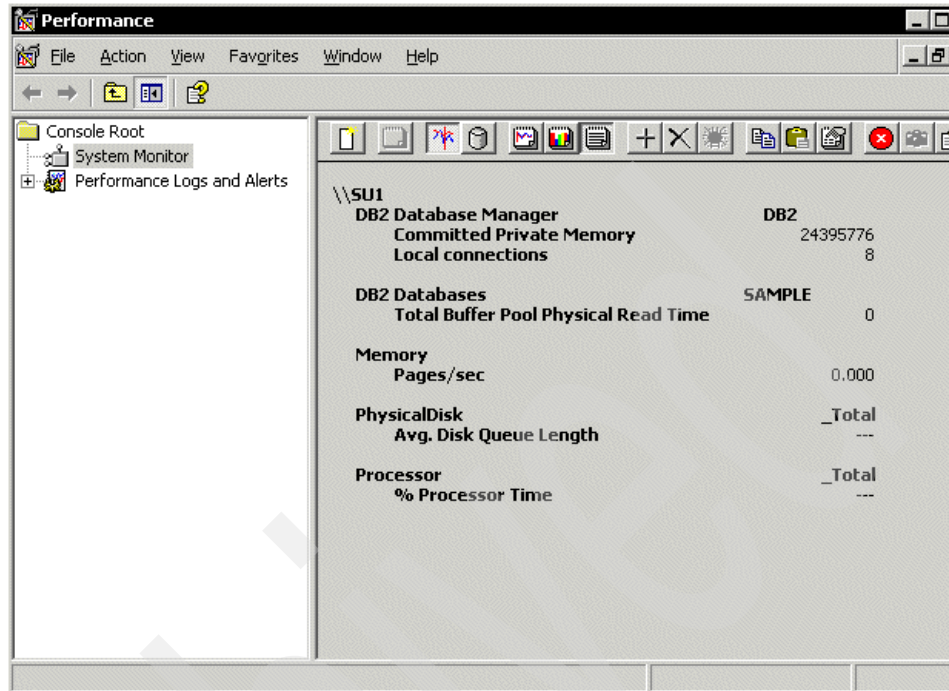


Figure 3-11 Perfmon output: Text

DB2 Memory Visualizer

A new tool has been added in DB2 UDB V8.1 to allow DBAs to see exactly how much memory has been allocated and used by DB2. The *Memory Visualizer* is available in the DB2 Control Center. To access the tool right click the instance you would like to monitor and select **View memory usage**.

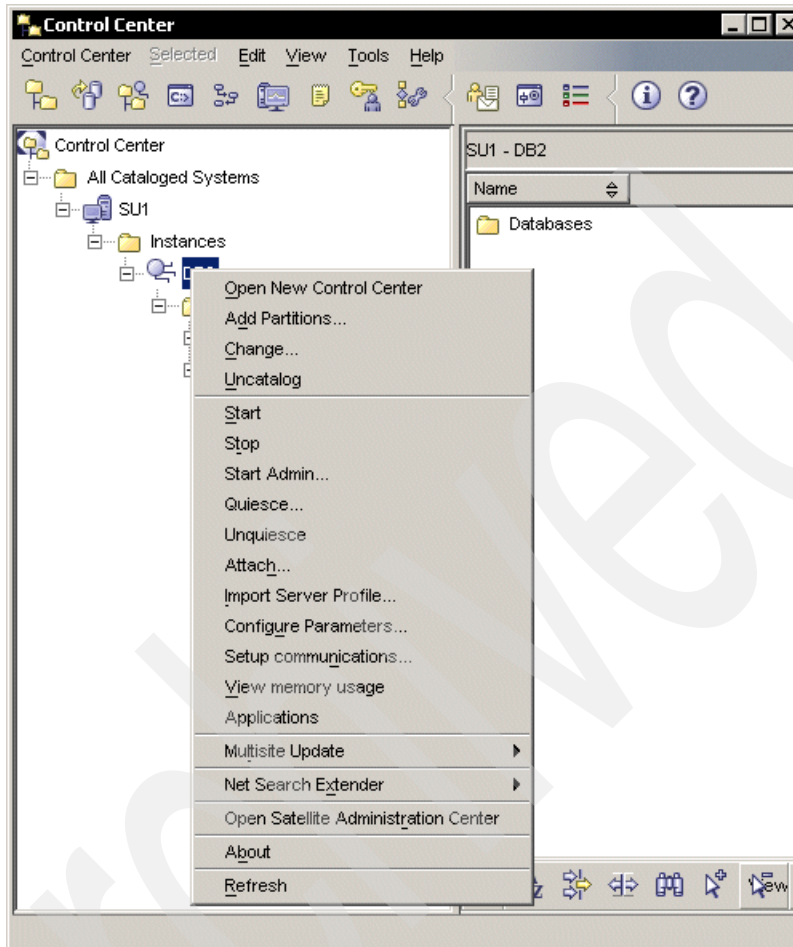


Figure 3-12 DB2 Memory Visualizer

The Memory Visualizer tool contains information on how all the heaps within the instance and the connected databases are being used. The information is provided both using text and a graphic chart format. The information is refreshed at regular intervals that can be selected by the user.

A memory heap can be displayed graphically by clicking the desired variable and selecting **Show Plot**. This is a very useful tool for monitoring how effectively you have allocated the memory across the many heaps. Previously with DB2 UDB V7.2 it was difficult to determine if you had allocated too much memory for a bufferpool or memory heap. This tool can now be used to see in real-time how much of the allocated memory is being used. Unless you are looking at the memory behavior over a short interval, it is better to increase the refresh interval

to a larger interval, possibly once every minute or every five minutes. This reduces the overhead of checking all the memory usage, and if you are using it as a background memory monitor, then you will probably be only interested in average behavior anyway.

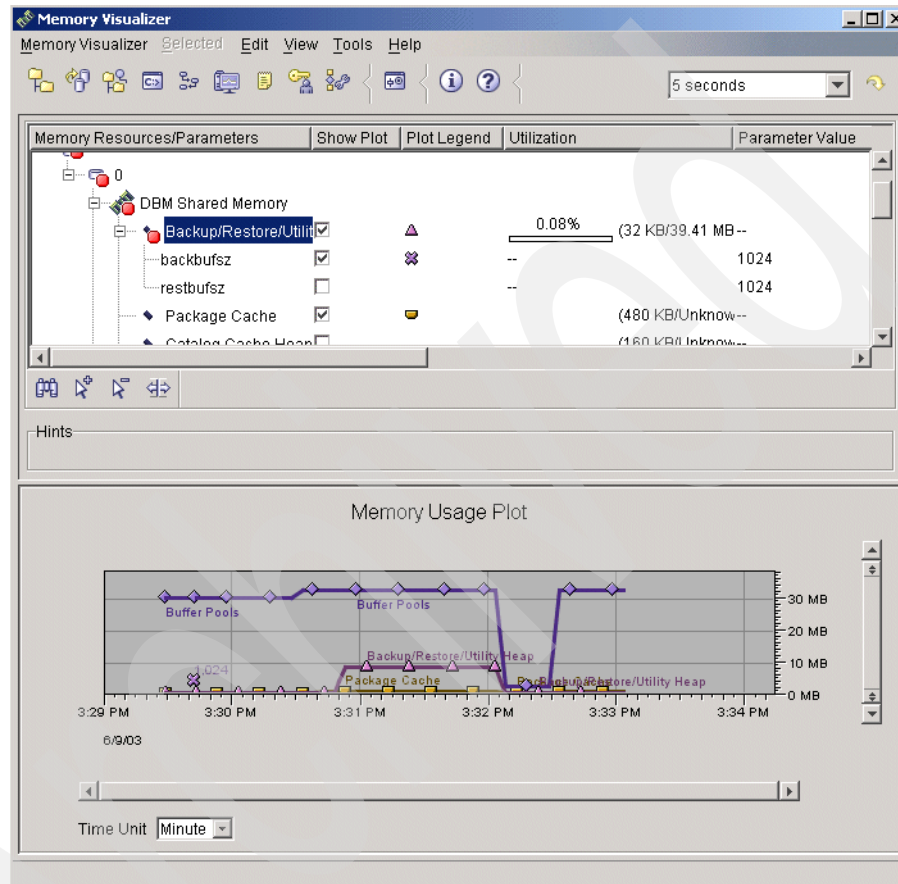


Figure 3-13 Memory Visualizer graphic output

DB2 snapshots

The DB2 snapshots provide a great deal of detailed information on how DB2 is behaving. The snapshots are normally used for performance tuning and have been improved in V8.1. The snapshots capture information on:

- ▶ Database memory and usage
- ▶ Instance memory and usage
- ▶ Tablespace
- ▶ Tables

- ▶ Bufferpools
- ▶ Locks
- ▶ Applications
- ▶ Dynamic SQL

The majority of the database memory heap usage is found in the database manager and database snapshots. A detailed description of how to use the information in the snapshots can be found in the following resource:

PDF: *System Monitor Guide and Reference*, Chapter 3, “Using the Snapshot Monitors”

3.3.6 Case studies

Two case studies are included below to illustrate what typical values should be used based on the workload on a system. Both workloads are based on the same system:

- ▶ 8-way x455 with 1 GHz CPUs
- ▶ 6 GB of RAM
- ▶ 1 FastT disk array with 130 GB of storage

OLTP case study

The OLTP case study was for a medium volume workload of simple transactional SQL statements being generated by a Web-based application. There were three bufferpools being used for the system and no LOB data was being transmitted. This is a simple example of the workload you may experience on your own system.

Example workload:

- ▶ 200 transactions a minute
- ▶ Two local connections
- ▶ 200 remote connections
- ▶ Cursor Stability isolation level

Instance recommendations

The Configuration Advisor recommended the settings listed in Table 3-1.

Table 3-1 OLTP instance parameter recommendations

Parameter	Value	Description
SHEAPTHRES	45077	Sort heap threshold
INTRA_PARALLEL	OFF	Intra-parallel partitioning

Parameter	Value	Description
MAX_QUERYDEGREE	1	Max query parallelism degree
MAXAGENTS	400	Maximum database agents
NUM_POOLAGENTS	400	Number of agents in the pool
NUM_INITAGENTS	0	Number of initialized agents
FCM_NUM_BUFFERS	4096	Number of FCM buffers
PRIV_MEM_THRESH	32767	Private memory threshold

The majority of the instance configuration parameter changes will restrict the amount of work on the system to the limits given during the configuration. These restrictions can be seen on the agents' settings. The parallelism of the queries has also been turned off, which is ideal for short OLTP type transactions. The sort heap threshold has been increased, but it is still a small percentage of the total memory.

Database recommendations

Table 3-2 contains all of the parameter changes that are recommended by the configuration tool.

Table 3-2 Recommended database changes

Parameter	Value	Description
APP_CTL_HEAP_SZ	128	Application control heap size
APPGROUP_MEM_SZ	14754	Application group memory size
CATALOGCACHE_SZ	270	Catalog cache
CHNGPGS_THRESH	40	Change page threshold
DBHEAP	600	Database heap
LOCKLIST	50	Lock list
LOGBUFSZ	76	Log buffer size
LOGFILSIZ	1024	Log file size
LOGPRIMARY	3	Number of primary log files
LOGSECOND	0	Number of secondary log files
MAXAPPLS	232	Maximum number of applications

Parameter	Value	Description
MAXLOCKS	76	Maximum percent of the lock list per application
MINCOMMIT	2	Minimum commit grouping
NUM_IOCLEANERS	2	Number of IO cleaners
NUM_IOSERVERS	3	Number of IO servers
PCKCACHESZ	707	Package cache size
SOFTMAX	30	Soft max
SORTHEAP	223	Sort heap
DFT_PREFETCH_SZ	32	Default prefetch size
UTIL_HEAP_SZ	342186	Utility heap size
BUFFERPOOL SIZE	1026560	Bufferpool size (not a db cfg parameter)

The parameter selections were designed to speed up the performance of short OLTP queries. The log file size was reduced and the MINCOMMIT was increased to improve the performance of the logging. The different memory heaps, such as the catalog and packages caches, were also increased in size in proportion to the expected workload complexity and volume. The number of I/O servers and cleaners should be adjusted to reflect the number of I/O controllers and CPUs available on the server.

You can see that the vast majority of the memory was allocated to the bufferpools on the system. A large percentage of the memory was also given to the utility heap. The amount of memory for the utility heap is too much and will not be effectively used. You should allocate about a fifth of the suggested amount. A good reference on tuning your system for OLTP performance is the article “DB2 Tuning Tips for OLTP Applications” written by Yongli An and Peter Shum. It can be found at

<http://www7b.boulder.ibm.com/dmdd/library/techarticle/anshum/0107anshum.html>

Key parameter changes

The key changes for the OLTP workload were to increase the size of the bufferpools and to minimize the impact of logging. If you have multiple bufferpools in your system then you should use the DB2 snapshots to determine which bufferpools are used more and which have the lowest hit ratio. It is also a good idea to examine how your bufferpools are being used. If they are all already at close to a 99 percent hit ratio, then there is no point adding additional memory

to them. This will not improve your performance, and the memory could be allocated somewhere else. Figure 3-14 is the formula for the buffer pool hit ratio.

Bufferpool
hit ratio

= 100% *

1 -

Bufferpool index and
data logical reads

Bufferpool index and
data physical reads

Figure 3-14 Bufferpool hit ratio

Data warehouse case study

The workload for a data warehousing system is quite different from OLTP, and even more so as the amount of memory for the system is expanded.

Our example workload:

- ▶ Transactions a minute not a configuration option
- ▶ Five local connections
- ▶ Five remote connections
- ▶ Uncommitted Read isolation level

Instance recommendations

The parameter recommendations for the instance form Configuration Advisor are listed in Table 3-3.

Table 3-3 DSS instance confirmation settings

Parameter	Value	Description
SHEAPTHRES	10000	Sort heap threshold
INTRA_PARALLEL	OFF	Intra-parallel partitioning
MAX_QUERYDEGREE	ANY	Maximum query parallelism
MAXAGENTS	400	Maximum number of agents
NUM_POOLAGENTS	-1	Number of pool agents
NUM_INITAGENTS	0	Number of initial agents
FCM_NUM_BUFFERS	4096	Number of FCM buffers
PRIV_MEM_THRESH	-999	Private memory threshold

The parameter recommendations given by the Configuration Advisor are designed to increase the performance of long-running complex queries. One parameter, which is key to this performance, was set incorrectly. The INTRA_PARALLEL parameter should definitely be set to ON for a DSS workload on a system with multiple CPUs. The sort heap threshold was also set fairly low, but that was due to the small database used for the example and the low number of connections on the server. For larger databases, which are the average for this type of workload, the SHEAPTHRES should be upwards of 80 percent of the available memory to allow the complex joins and aggregation to be completed quickly.

Database recommendations

The database recommendations focused more on the query performance and optimization. Table 3-4 lists the recommendations from the Configuration Advisor.

Table 3-4 Database parameter configuration settings

Parameter	Value	Description
APP_CTL_HEAP_SZ	128	Application control heap size
APPGROUP_MEM_SZ	30000	Application group memory size
CATALOGCACHE_SZ	-1	Catalog cache size
CHNGPGS_THRESH	60	Change page threshold
DBHEAP	600	Database heap
LOCKLIST	50	Lock list
LOGBUFSZ	8	Log buffer size
LOGFILSIZ	250	Log file size
LOGPRIMARY	3	Number of primary log files
LOGSECOND	2	Number of secondary log files
MAXAPPLS	40	Maximum number of applications
MAXLOCKS	22	Maximum percentage of the lock list
MINCOMMIT	1	Commit grouping
NUM_IOCLEANERS	1	Number of I/O cleaners
NUM_IOSERVERS	3	Number of I/O servers
PCKCACHESZ	-1	Package cache size

Parameter	Value	Description
SOFTMAX	100	Soft max
SORTHEAP	256	Sort heap size
STMTHEAP	4096	Statement heap
DFT_DEGREE	1	Default degree of parallelism
DFT_PREFETCH_SZ	16	Default prefetch size
UTIL_HEAP_SZ	5000	Utility heap size

A number of the parameter recommendations should be increased for a workload that will involve long-running complex queries. The sort heap recommendation of 250 4-K pages is far too small and should be closer to 5000. This parameter will often be much larger for DSS workloads and could be closer to 40,000 pages depending on the size of the queries.

As with the OLTP workload, the number of I/O servers and cleaners should be increased to match the system you are running on. Very complex queries may also require that the statement heap be increased as well, depending on how extensively views are used.

Thinking about partitioning

When you configure a data warehouse it is important to remember that the system will normally be partitioned. Each database partition will have the same database memory allocated for it. If you have multiple logical partitions on the server, then you must ensure that the total memory allocated to each of the database partitions does not exceed the maximum memory on the server.

3.4 Adding additional storage

This section is an introduction to important considerations that you should have in mind when you plan to add additional storage to a database server.

We discuss Windows Server 2003 storage recommendations, disk array concepts, and the difference between storage direct attach and storage area networks.

We also cover DB2 storage fundamentals, limits, and recommendations on tablespace design, performance, and a brief introduction to the Storage Management view.

3.4.1 Storage and Windows Server 2003

When adding additional disk capacity to a server or cluster, we recommend a basic rule. If using one server or a two-server clustering with Windows Server 2003, then SCSI or Fibre Channel can be configured to connect to the storage devices. In another way, if your architecture uses three or more servers then the technique for connecting storage devices should be Fibre Channel. Using fibre channel is highly recommended. SCSI or Fibre Channel configurations are *direct attach* technology. Fiber Channel is also commonly used as a SAN connection.

The technology called storage area networks (SANs) is also used by clustered servers and storage devices connected via high performance interconnections between secure servers and storage devices to deliver higher bandwidth and lower latency than comparable traditional networks.

A feature called Winsock Direct is implemented in Windows Server 2003 Datacenter and Enterprise Edition. This feature allows direct communication over a system area network using SAN providers. Applications can pass messaging requests directly to the SAN hardware interface. This capability eliminates unnecessary system calls and data copying since the communications are at the hardware level. The individual transport endpoints can be mapped directly into the address space of application processes running in user mode.

The SAN architecture typically uses two transfer modes:

- ▶ One consists of transferring control information (small transfers).
- ▶ The SAN hardware interface uses a bulk mode for larger transfers whereby data is transferred directly between the local system and the remote system (without CPU involvement on the local or remote system).

SAN configurations allow IT departments to consolidate storage needs; for example, share storage with non-Windows operating systems.

Depending on availability and performance needs, a cluster storage device should be optimized.

The Windows Datacenter Hardware Compatibility List provides a detailed list of acceptable Redundant Array of Independent Disks (RAID) configuration for clusters. Table 3-5 on page 148 provides an overview of common RAID configurations.

Table 3-5 RAID configurations

RAID level	RAID type	RAID description
5+1	Disk striping with parity + mirroring	Six or more volumes, each on a separate drive, are configured identically as a mirrored stripe set with parity error checking.
5	Disk striping with parity	Three or more volumes, each on separate drives, are configured as a stripe set with parity error checking. In the case of failure, data can be recovered.
1	Disk mirroring	Two volumes on two drives are configured identically. Data is written to both drives. If one drive fails, there is no data loss because the other drive contains the data. (Does not include disk striping.)
0+1	Disk striping with mirroring	Two or more volumes, each on a separate drive, are striped and mirrored. Data is written sequentially to drives that are identically configured.
0	Disk striping	Two or more volumes, each on a separate drive, are configured as a stripe set. Data is broken into blocks, called stripes, and then written sequentially to all drives in the stripe set.

The advantages for each RAID configuration are:

- ▶ 5+1: Provides a very high level of fault tolerance and very good performance.
- ▶ 5: Fault tolerance with less overhead than mirroring. Better read performance than disk mirroring.
- ▶ 1: Redundancy. Better write performance than disk striping with parity.
- ▶ 0+1: Redundancy with good read/write performance.
- ▶ 0: Speed/performance without data protection.

3.4.2 DB2 storage considerations

The storage architecture for your system implementation can influence the performance. There are some other implications associated with disk storage on performance.

The following are four aspects of disk-storage management that affect performance:

- ▶ Division of storage
How you divide a limited amount of storage between indexes and data and among tablespaces determines, to a large degree, how each will perform in different situations.
- ▶ Wasted storage
Wasted storage in itself may not affect the performance of the system. But wasted storage is a resource that could be used to improve performance elsewhere.
- ▶ Distribution of disk I/O
How well you balance the demand for disk I/O across several disk storage devices and controllers can affect how fast the database manager can retrieve information from disks.
- ▶ Lack of available storage
Reaching the limit of available storage can degrade overall performance.

If your database is demanding more storage space then you must consider which recovery method to use.

- ▶ The DB2 version recovery method requires space to hold the backup copy of the database and the restored database.
- ▶ The DB2 rollforward recovery method requires space to hold the backup copy of the database or tablespaces, the restored database, and the archived database logs.

If a table contains long field or large object binary (LOB) columns, you should consider placing this data in a separate tablespace. This will affect your storage space planning as well as your plan for recovery. With a separate tablespace for long field and LOB data, and knowing the time required to back up long field and LOB data, you may decide to use a recovery plan that only occasionally saves a backup of this tablespace. You may also choose, when creating or altering a table to include LOB columns, not to log changes of those columns. This will reduce the size of the required log space and the corresponding log archive space.

To prevent media failure from destroying a database and your ability to rebuild it, keep the database backup, the database logs, and the database itself on different devices.

Note: It is highly recommended that you use the *newlogpath* configuration parameter to put database logs on a separate device once the database is created. The logs should be spread out on multiple devices to ensure that the logging does not become an I/O bottleneck.

The database logs can use up a large amount of storage. If you plan to use the rollforward recovery method, you must decide how to manage the archived logs. Your choices are the following:

- ▶ Use the user exit program to copy these logs to another storage device in your environment.
- ▶ Manually copy the logs to a storage device or directory other than the database log path directory after they are no longer in the active set of logs.

3.4.3 DB2 tablespace design

A tablespace is a storage structure containing tables, indexes, large objects, and long data. A tablespace has following characteristics:

- ▶ For tablespaces residing in database partition groups, the tablespace selected to hold a table defines how the data for that table is distributed across the database partitions in a database partition group.
- ▶ They allow you to assign the location of database and table data directly onto containers.
- ▶ A container can be a directory name, a device name, or a file name. This can provide improved performance and allow more flexible configuration.
- ▶ A single tablespace can span several containers. It is possible for multiple containers (from one or more tablespaces) to be created on the same physical disk (or drive).

Note: For performance improvement of large tables, each container should use a different disk.

Figure 3-15 on page 151 illustrates the relationship between tables and tablespaces within database 1 and database 2 under the same instance, and the containers associated with those databases. On database 1, the table 1, table 2, and table 3 are in the tablespace named A, which spans containers 0, 1, 2, and 3. The table 4 is in the tablespace named B in container 4. This example shows

each container existing on a separate disk. Database 2 has a tablespace named A with two tables (1, 2) and two containers (5, 6).

The database manager balances the data load across containers. As a result, all containers are used to store data. The number of pages that the database manager writes to a container before using a different container is called the *extent size*. The database manager does not always start storing table data in the first container.

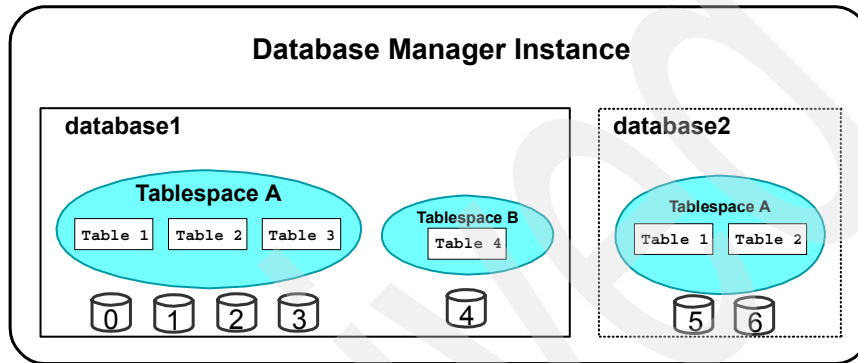


Figure 3-15 Containers, tablespace, and tables

Figure 3-16 on page 152 shows the HUMANRES tablespace with an extent size of two 4-KB pages; four containers, each with a small number of allocated extents. The DEPARTMENT and EMPLOYEE tables both have seven pages, and span all four containers.

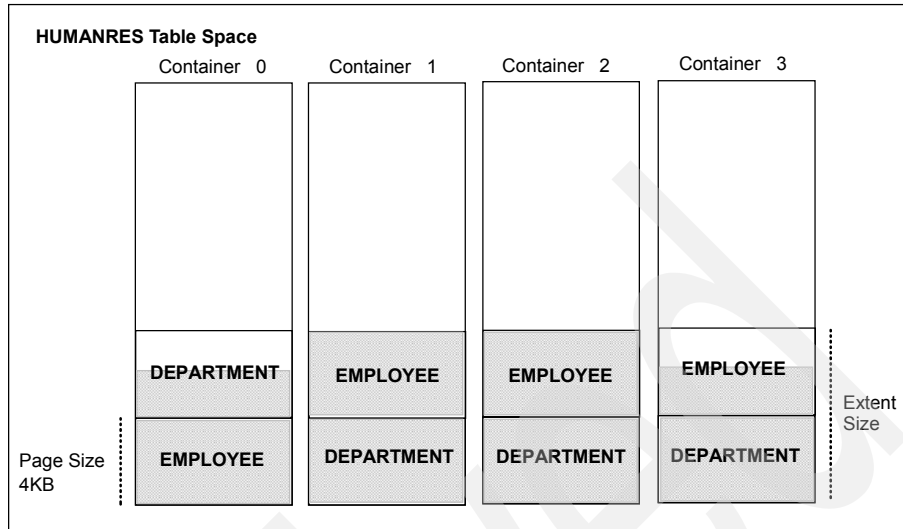


Figure 3-16 Extents and containers

A database must contain at least three tablespaces:

- Catalog tablespace

Contains system catalog tables for the database. This tablespace is called SYSCATSPACE and it cannot be dropped. IBMCATGROUP is the default database partition group for this tablespace.

- User tablespace

By default one tablespace called USERSPACE1 is created. IBMDEFAULTGROUP is the default database partition group for this tablespace.

Note: You should specify a tablespace name when you create a table, or the results in performance or storage capacity may not be what you intend. If the tablespace name is not specified, the default tablespace will be used. This tablespace normally is not configured across multiple devices and may have more I/O problems than others you have specifically configured for a heavy workload.

The page size of a table is determined either by row size or the number of columns. Possible values for page size are 4 KB (the default), 8 KB, 16 KB, and 32 KB. You can use a tablespace with one page size for the base table, and a different tablespace with a different page size for long or LOB data. If the number of columns or the row size exceeds the limits for a tablespace's

page size, an error is returned (SQLSTATE 42997). Table 3-6 lists the page size specification.

Table 3-6 DB2 page size specific limits

Description	4-K limit	8-K limit	16-K limit	32-K limit
Most columns in a table	500	1012	1012	1012
Maximum length of a row including all overhead	4005	8101	16293	32677
Maximum size of a table per partition (in GB)	64	128	256	512
Maximum size of an index per partition (in GB)	64	128	256	512
Most elements in a select list	500	1012	1012	1012
Maximum number of columns in a GROUP BY clause	500	1012	1012	1012
Maximum total length of columns in a GROUP BY clause (in bytes)	4005	8101	16293	32677
Most values in an INSERT statement	500	1012	1012	1012
Most SET clauses in a single UPDATE statement	500	1012	1012	1012
Maximum size of a regular DMS tablespace (in GB)	64	128	256	512

► **Temporary tablespace**

Contains temporary tables. Temporary tablespaces can be system temporary tablespaces or user temporary tablespaces. By default one system temporary tablespace is created and is called TEMPSPACE1. IBMTEMPGROUP is the default database partition group for this tablespace.

If a database has more than one temporary tablespace, when a new temporary object is needed, the optimizer will choose an appropriate page size for this object. That object will then be allocated to the temporary tablespace with the corresponding page size. If there is more than one temporary tablespace with that page size, then the tablespace will be chosen in a round-robin fashion.

Note: In most cases, it is not recommended to have more than one temporary tablespace of any one page size. The optimizer could spend extra time in deciding which temporary tablespace to use.

If queries are running against tables in tablespaces that are defined with a page size larger than the 4-KB default (for example, an ORDER BY on 1012 columns), some of them may fail. This will occur if there are no temporary tablespaces defined with a larger page size. You may need to create a temporary tablespace with a larger page size (8 KB, 16 KB, or 32 KB). Any Data Manipulation Language (DML) statement could fail unless there exists a temporary tablespace with the same page size as the largest page size in the user tablespace.

Note: You should define a single System Managed Space (SMS) temporary tablespace with a page size equal to the page size used in the majority of your user tablespaces. This should be adequate for typical environments and workloads.

Note: In a partitioned database environment, the catalog node will contain all three default tablespaces, and the other database partitions will each contain only TEMPSPACE1 and USERSPACE1.

There are two types of tablespace. You can use them in a single database:

- ▶ System Managed Space (SMS), in which the operating system's file manager controls the storage space
- ▶ Database Managed Space (DMS), in which the database manager controls the storage space

On SMS tablespaces you can improve performance by enabling the use of *multipage file allocation* for a database. Disk space is allocated one extent (last extent), rather than one page at a time, when data and index files are larger than one extent.

The DB2 system command **db2empfa** enables multipage file allocation. This command only affects the database partition on which it is executed, and you must have sysadm authorization to execute it. See Example 3-5.

Example 3-5 db2empfa command

```
x:\db2empfa database-name
```

The command **db2empfa** establishes a database connection and only needs one parameter that specifies the alias of the database for which multipage file allocation is to be enabled. Also it changes the value of the database configuration parameter `multipage_alloc` to YES.

Since **db2empfa** connects to the database partition in exclusive mode, it cannot be run concurrently on the catalog database partition, or on any other database partition.

3.4.4 Database managed space

In a Database Managed Space (DMS) tablespace, the database manager controls the storage space. The storage model consists of a limited number of devices or files whose space is managed by DB2. The database administrator decides which devices and files to use, and DB2 manages the space on those devices and files. The tablespace is essentially an implementation of a special purpose file system designed to best meet the needs of the database manager.

A DMS tablespace containing user-defined tables and data can be defined as:

- ▶ A regular tablespace to store any table data and optionally index data
- ▶ A large tablespace to store long field or LOB data or index data

When designing your DMS tablespaces and containers, you should consider the following:

- ▶ The database manager uses striping to ensure an even distribution of data across all containers.
- ▶ The maximum size of regular tablespaces is 64 GB for 4-KB pages; 128 GB for 8-KB pages; 256 GB for 16-KB pages; and 512 GB for 32-KB pages. The maximum size of large tablespaces is 2 TB.
- ▶ The containers that make up a DMS tablespace do not need to be the same size. However, this is not normally recommended because it will result in uneven striping across the containers, and sub-optimal performance.
- ▶ You need to allocate disk space before creating the tablespace. When using device containers, the device must also have enough space for the definition of the container. Each device can have only one container defined on it. To avoid wasted space, the size of the device and the size of the container should be equivalent. For example, if the device is allocated with 5,000 pages, and the device container is defined to allocate 3,000 pages, 2,000 pages on the device will not be usable.
- ▶ By default, one extent in every container is reserved for overhead. Only full extents are used. For optimal space management, you can use the following formula to determine an appropriate size when allocating a container:

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

Where `extent_size` is the size of each extent in the tablespace, and `n` is the number of extents that you want to store in the container.

- ▶ The minimum size of a DMS tablespace is five extents. Attempting to create a tablespace smaller than five extents will result in an error (SQL1422N).
- ▶ Three extents in the tablespace are reserved for overhead.
- ▶ At least two extents are required to store any user table data. These extents are required for the regular data for one table, and not for any index, long field, or large object data, which require their own extents.
- ▶ Device containers must use logical volumes with a “character special interface”, not physical volumes.

You can use files instead of devices with DMS tablespaces. No operational difference exists between a file and a device; however, a file can be less efficient because of the run-time overhead associated with the file system. Files are useful when:

- ▶ Devices are not directly supported.
- ▶ A device is not available.
- ▶ Maximum performance is not required.
- ▶ You do not want to set up devices.

If your workload involves LOBs or LONG VARCHAR data, you may derive performance benefits from file system caching. Please note that LOBs and LONG VARCHARs are not buffered by DB2's buffer pool.

Adding additional containers

You can add a new container without rebalancing by using the *Stripe Sets* option. This is recommended for adding containers for a large tablespace because adding new containers with same size when the tablespace is full forces the tablespaces to be rebalanced. With large tablespaces, rebalancing can take hours or even days to complete, which can have a significant impact on performance.

From the Control Center, select the database name and expand the object tree until you find the Tablespaces folder. Select this folder, then select the tablespace that you want. Right click the tablespace name, and click **Alter** from the pop-up menu. The Alter Tablespace notebook opens. Click the **Containers** page. Select the **Manage Stripe Set** option, then select **Add** to add a new stripe set to the new container. Complete the required fields on the Define Container dialog and click **OK** to add the new container. See Example 3-17 on page 157. The Define Container dialog closes, returning focus to the Containers page of the Alter Tablespace notebook.

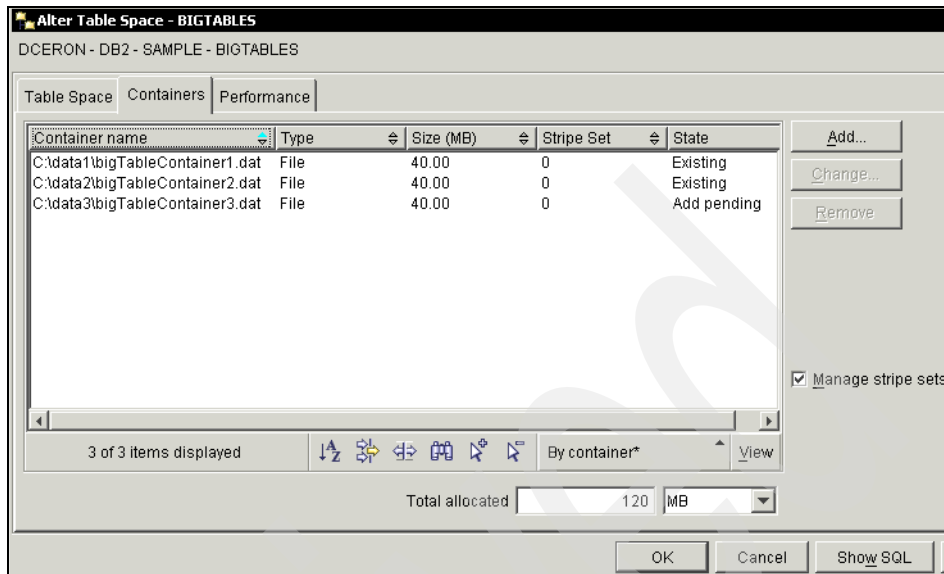


Figure 3-17 Alter tablespace with manage stripe sets option

From the Command Line Processor window, you can type as shown in Example 3-6.

Example 3-6 Altering a tablespace with stripe set option

```
alter tablespace BIGTABLES begin new stripe set ( FILE
'C:\data3\bigTableContainer3.dat' 10240 )
```

BIGTABLES is the name of the tablespace; *begin new stripe set* specifies that a new stripe set is to be created in the tablespace and the container is to be added to this new stripe set; the reset of the command is the type of the container, the container path, and size.

For adding a container to an existing tablespace with rebalancing, use a command like in Example 3-7.

Example 3-7 Alter tablespace using add option

```
alter tablespace BIGTABLES add ( FILE 'C:\data3\bigTableContainer3.dat' 10240 )
```

3.4.5 DB2 DMS device considerations

If you use Database Managed Storage (DMS) device containers for tablespaces in your design, consider the following factors for effective administration:

- File system caching

For DMS file containers and all SMS containers, the operating system might cache pages in the file system cache. For DMS device container tablespaces, the operating system does not cache pages in the file system cache. If you eliminate system caching then more memory will be available to the database.

On Windows, the registry variable DB2NTNOCACHE specifies whether or not DB2 will open database files with a NOCACHE option. If DB2NTNOCACHE=ON, file system caching is eliminated. If DB2NTNOCACHE=OFF, the operating system caches DB2 files.

- Buffering of data

For DMS file containers and all SMS containers, file system caching can eliminate I/O that would otherwise have been required. In some cases, a data page might be freed from the buffer pool before an application has actually used the page, particularly if the buffer pool space is required for other data pages.

DMS tablespaces using device containers do not use the file system or its cache. As a result, you might increase the size of the database buffer pool and reduce the size of the file system cache.

- Using LOB or LONG data

With LOB or LONG data the database manager does not cache the data in its buffers. Each time an application needs one of these pages, the database manager must retrieve it from disk. However, if LOB or LONG data is stored in SMS or DMS file containers, file system caching might provide buffering and, as a result, better performance.

Note: Because system catalogs contain some LOB columns, you should keep them in SMS tablespaces or in DMS-file tablespaces.

3.4.6 Using raw I/O in DB2

DB2 supports containers using direct disk access (raw I/O) to define tablespaces. The Windows family supports direct disk access (raw), with the exception of the Windows 9x operating system.

Before you start defining tablespaces with device containers, consider the following:

- You must know the device or file names of the containers you are going to reference when creating your tablespaces.

- ▶ You must know the amount of space associated with each device or file name that is to be allocated to the tablespace.
- ▶ You will need the correct permissions to read and write to the container.

On Windows, to specify a physical hard drive, use the following syntax:

`\\.\PhysicalDriveN`

Where N represents one of the physical drives in the system. In this case, N could be replaced by 0, 1, 2, or any other positive integer, for example:

`\\.\PhysicalDrive0`

On Windows, to specify a logical drive (that is, an unformatted partition) use the following syntax:

`\\.\N:`

Where N: represents a logical drive letter in the system. For example, N: could be replaced by E: or any other drive letter. To overcome the limitation imposed by using a letter to identify the drive, you can use a globally unique identifier (GUID) with the logical drive.

For Windows 2000 and above, there is a new method for specifying DMS raw tablespace containers. Volumes (that is, basic disk partitions or dynamic volumes) are assigned a globally unique identifier (GUID) when they are created. The GUID can be used as a device identifier when specifying the containers in a tablespace definition. The GUIDs are unique across systems, which means that in a multiple partitioned database configuration, GUIDs are different for each partition even if the disk partition definitions are the same. You can use the Windows disk management tool to find the GUID. Figure 3-18 on page 160 shows the disk management tool on Windows.

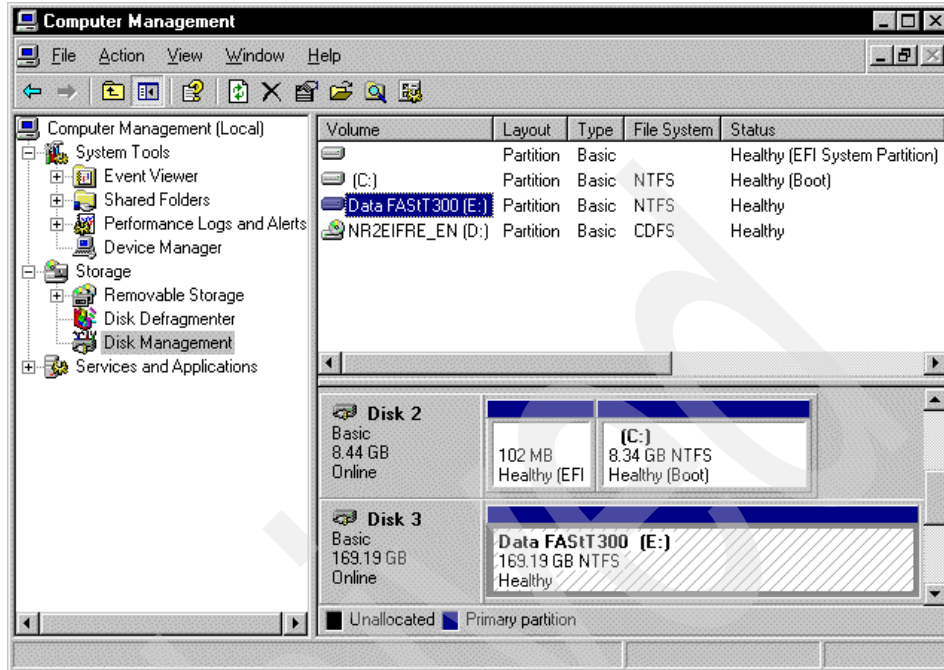


Figure 3-18 Windows disk management

A DB2 tool called `db2listvolumes.exe` is available (only on Windows operating systems) to easily display the GUIDs for all the disk volumes defined on a Windows system. This tool creates two files in the current directory where the tool is run:

- ▶ `Volumes.xml`: Contains information about each disk volume encoded in XML for easy viewing on any XML editor.
- ▶ `Tablespace.ddl`: Contains the required syntax for specifying tablespace containers. Example 3-8 shows a sample content of this file.

Example 3-8 tablespace.ddl file sample content

```
-- disk 2, offset 139829760
DEVICE '\\?\Volume{5d570940-20f4-01c3-a1f4-04622fd5ec6d}\\" 8542M,
-- disk 3, offset 134235136
DEVICE '\\?\Volume{39349c36-ba0d-4259-80b3-3b4dfcd1bfef}\\" 173246M,
```

Note: The file `tablespace.ddl` must be updated to fill in the remaining information needed for a tablespace definition.

The `db2listvolumes` tool does not require any command line arguments.

You can use a raw device for your database log. There are both advantages and disadvantages in doing so.

The advantages are:

- ▶ You can attach more than 26 physical drives to a system.
- ▶ The file I/O path length is shorter. This may improve performance on your system. You should conduct benchmarks to evaluate if there are measurable benefits for your workload.

The disadvantages are:

- ▶ The device cannot be shared by other applications; the entire device must be assigned to DB2.
- ▶ The device cannot be operated upon by any operating system utility or third-party tool that would backup or copy from the device.
- ▶ You can easily wipe out the file system on an existing drive if you specify the wrong physical drive number.

Note: You can configure a raw log with the *newlogpath* database configuration parameter. Consider the advantages and disadvantages listed above.

Additional considerations

In addition to what has been discussed, please also consider the following points if you plan to use raw I/O in DB2:

- ▶ Only one device is allowed. You can define the device over multiple disks at the operating system level. DB2 will make an operating system call to determine the size of the device in 4-KB pages.
- ▶ If you use multiple disks, this will provide a larger device, and the striping that results can improve performance by faster I/O throughput.
- ▶ DB2 will attempt to write to the last 4-KB page of the device.
- ▶ Information about the size of the device is used to indicate the size of the device (in 4-KB pages) available to DB2 under the support of the operating system. The amount of disk space that DB2 can write to is referred to as `device_size_available`.
- ▶ The first 4-KB page of the device is not used by DB2 (this space is generally used by the operating system for other purposes.) This means that the total space available to DB2 is `device_size = device_size_available - 1`.
- ▶ The *logsecond* parameter is not used. DB2 will not allocate secondary logs. The size of active log space is the number of 4-KB pages that result from `LOGPRIMARY x LOGFILSZ`.

- ▶ Log records are still grouped into log extents, each with a log file size (logfilsiz) of 4-KB pages. Log extents are placed in the raw device, one after another. Each extent also consists of an extra two pages for the extent header. This means that the number of available log extents the device can support is $\text{device_size} / (\text{logfilsiz} + 2)$.
- ▶ The device must be large enough to support the active log space. That is, the number of available log extents must be greater than (or equal to) the value specified for the logprimary configuration parameter. If the userexit configuration parameter is enabled, ensure that the raw device can contain more logs than the value specified for the logprimary configuration parameter. This will compensate for the delay incurred when the user exit program is archiving a log file.
- ▶ If you are using circular logging, the logprimary configuration parameter will determine the number of log extents that are written to the device. This may result in unused space on the device.
- ▶ If you are using log retention (LOGRETAIN) without a user exit program, after the number of available log extents are all used up, all operations that result in an update will receive a log full error. At this time, you must shut down the database and take an offline backup of it to ensure recoverability. After the database backup operation, the log records written to the device are lost. This means that you cannot use an earlier database backup image to restore the database, then roll it forward. If you take a database backup before the number of available log extents are all used up, you can restore and roll the database forward.
- ▶ If you are using log retention (LOGRETAIN) with a user exit program, the user exit program is called for each log extent as it is filled with log records. The user exit program must be able to read the device, and to store the archived log as a file. DB2 will not call a user exit program to retrieve log files to a raw device. Instead, during rollforward recovery, DB2 will read the extent headers to determine if the raw device contains the required log file. If the required log file is not found in the raw device, DB2 will search the overflow log path. If the log file is still not found, DB2 will call the user exit program to retrieve the log file into the overflow log path. If you do not specify an overflow log path for the rollforward operation, DB2 will not call the user exit program to retrieve the log file.
- ▶ If you have configured a raw device for logging, and are using DataPropagator™ (DPROP), or another application that calls the db2ReadLog API, the *overflowlogpath* database configuration parameter must be configured. DB2 may call a user exit program to retrieve the log file and return the log data requested by the db2ReadLog API. The retrieved log file will be placed in the path specified by the overflowlogpath database configuration parameter.

3.4.7 Tablespace disk I/O

The type and design of your tablespace determines the efficiency of the I/O performed against that tablespace.

Big block reads

Big block reads is a read where several pages (usually an extent) are retrieved in a single request. Reading several pages at once is more efficient than reading each page separately.

Whenever it is advantageous to do so, DB2 performs big block reads. This typically occurs when retrieving data that is sequential or partially sequential in nature. The amount of data read in one read operation depends on the extent size. The bigger the extent size, the more pages that can be read at one time.

Prefetching

Prefetching is the reading of pages in advance of those pages being referenced by a query. The overall objective is to reduce response time. This can be achieved if the prefetching of pages can occur asynchronously to the execution of the query. The best response time is achieved when either the CPU or the I/O subsystem is operating at maximum capacity.

Sequential prefetching performance can be further enhanced if pages can be read from disk into contiguous pages within a buffer pool. Since buffer pools are page based by default, there is no guarantee of finding a set of contiguous pages when reading in contiguous pages from disk. Block-based buffer pools can be used for this purpose because they not only contain a page area, they also contain a block area for sets of contiguous pages. Each set of contiguous pages is named a block, and each block contains a number of pages referred to as blocksize. The size of the page and block area, as well as the number of pages in each block, is configurable.

How the extent is stored on disk affects I/O efficiency. In a DMS tablespace using device containers, the data tends to be contiguous on disk, and can be read with a minimum of seek time and disk latency. If files are being used, however, the data may have been broken up by the file system and stored in more than one location on disk. This occurs most often when using SMS tablespaces, where files are extended one page at a time, making fragmentation more likely. A large file that has been pre-allocated for use by a DMS tablespace tends to be contiguous on disk, especially if the file was allocated in a clean file space.

You can control the degree of prefetching by changing the PREFETCHSIZE option on the CREATE TABLESPACE or ALTER TABLESPACE statements. (The default value for all tablespaces in the database is set by the dft_prefetch_sz database configuration parameter.) The PREFETCHSIZE parameter tells DB2

how many pages to read whenever a prefetch is triggered. By setting PREFETCHSIZE to be a multiple of the EXTENTSIZE parameter on the CREATE TABLESPACE statement, you can cause multiple extents to be read in parallel. (The default value for all tablespaces in the database is set by the dft_extent_sz database configuration parameter.) The EXTENTSIZE parameter specifies the number of 4-KB pages that will be written to a container before skipping to the next container.

For example, suppose you had a tablespace that used three devices. If you set the PREFETCHSIZE to be three times the EXTENTSIZE, DB2 can do a big block read from each device in parallel, thereby significantly increasing I/O throughput. This assumes that each device is a separate physical device, and that the controller has sufficient bandwidth to handle the data stream from each device. Note that DB2 may have to dynamically adjust the prefetch parameters at runtime based on query speed, buffer pool utilization, and other factors.

Page cleaning

As pages are read and modified, they accumulate in the database buffer pool. If the buffer pool is full of modified pages, one of these modified pages must be written out to the disk before the new page can be read in. To prevent the buffer pool from becoming full, page cleaner agents write out modified pages to guarantee the availability of buffer pool pages for future read requests.

For prefetching (or even reading) to be efficient, a sufficient number of clean buffer pool pages must exist. For example, there could be a parallel prefetch request that reads three extents from a tablespace, and for each page being read, one modified page is written out from the buffer pool. The prefetch request may be slowed down to the point where it cannot keep up with the query. Page cleaners should be configured in sufficient numbers to satisfy the prefetch request.

3.4.8 Optimizing tablespace performance on RAID devices

To optimize performance when data is placed on Redundant Array of Independent Disks (RAID) devices, you should consider doing the following items:

- ▶ Define a single container for the tablespace (using the RAID device).
- ▶ Make the EXTENTSIZE of the tablespace equal to, or a multiple of, the RAID stripe size.
- ▶ Ensure that the PREFETCHSIZE of the tablespace is the RAID stripe size multiplied by the number of RAID parallel devices (or a whole multiple of this product), and a multiple of the EXTENTSIZE.

- Use the DB2_PARALLEL_IO registry variable to enable parallel I/O for the tablespace.

DB2_PARALLEL_IO

When reading data from or writing data to tablespace containers, DB2 may use parallel I/O if the number of containers in the database is greater than 1. However, there are situations when it would be beneficial to have parallel I/O enabled for single container tablespaces. For example, if the container is created on a single RAID device that is composed of more than one physical disk, you may want to issue parallel read and write calls.

To force parallel I/O for a tablespace that has a single container, you can use the DB2_PARALLEL_IO registry variable. This variable can be set to “*” (asterisk), meaning every tablespace, or it can be set to a list of tablespace IDs separated by commas. See Example 3-9.

Example 3-9 Setting DB2_PARALLEL_IO

```
db2set DB2_PARALLEL_IO=*  
db2set DB2_PARALLEL_IO=1,2,4,6
```

The first command in Example 3-9 is to turn parallel I/O on for all tablespaces, and the second command is to turn parallel I/O on for tablespaces 1, 2, 4, and 6. DB2 must be stopped (**db2stop**) and the restarted (**db2start**) for the changes to take effect.

DB2_PARALLEL_IO also affects tablespaces with more than one container defined. If you do not set the registry variable, the I/O parallelism is equal to the number of containers in the tablespace. If you set the registry variable, the I/O parallelism is equal to the result of prefetch size divided by extent size. You might want to set the registry variable if the individual containers in the tablespace are striped across multiple physical disks.

For example, a tablespace has two containers and the prefetch size is four times the extent size. If the registry variable is not set, a prefetch request for this tablespace will be broken into two requests (each request will be for two extents). Provided that the prefetchers are available to do the work, two prefetchers can be working on these requests in parallel. In the case where the registry variable is set, a prefetch request for this tablespace will be broken into four requests (one extent per request), with a possibility of four prefetchers servicing the requests in parallel.

In this example, if each of the two containers had a single disk dedicated to it, setting the registry variable for this tablespace might result in contention on those disks since two prefetchers will be accessing each of the two disks at once.

However, if each of the two containers was striped across multiple disks, setting the registry variable would potentially allow access to four different disks at once.

Note: In earlier versions of DB2, the DB2_STRIPED_CONTAINERS registry variable was used to create tablespaces with an extent sized tag. However, because this is now the default behavior, this registry variable no longer has any affect.

DB2_USE_PAGE_CONTAINER_TAG

By default, DB2 uses the first extent of each DMS container (file or device) to store a container tag. The container tag is DB2's metadata for the container. In earlier versions of DB2, the first page was used for the container tag, instead of the first extent, and as a result less space in the container was used to store the tag.

When the DB2_USE_PAGE_CONTAINER_TAG registry variable is set to ON, any new DMS containers created will be created with a one-page tag, instead of a one-extent tag (the default). There will be no impact to existing containers that were created before the registry variable was set. See Example 3-10.

Setting this registry variable to ON is not recommended unless you have very tight space constraints, or you require behavior consistent with pre-Version 8 databases.

Setting this registry variable to ON can have a negative impact on I/O performance if RAID devices are used for tablespace containers. When using RAID devices for tablespace containers, it is suggested that the tablespace be created with an extent size that is equal to, or a multiple of, the RAID stripe size. However, if this registry variable is set to ON, a one-page container tag will be used and the extents will not line up with the RAID stripes. As a result it may be necessary during an I/O request to access more physical disks than would be optimal. Users are thus *strongly* advised against setting this registry variable in a RAID environment.

Example 3-10 Setting DB2_USE_PAGE_CONTAINER_TAG

```
db2set DB2_USE_PAGE_CONTAINER_TAG=ON
db2stop
db2start
db2set DB2_USE_PAGE_CONTAINER_TAG=
db2stop
db2start
```

The first db2set sets the variable to create containers with one-page container tags. The second db2set resets this registry variable.

The Control Center, the LIST TABLESPACE CONTAINERS command, and the GET SNAPSHOT FOR TABLESPACES command do not show whether a container has been created with a page or extent sized tag. They use the label “file” or “device”, depending on how the container was created. To verify whether a container was created with a page or extent size tag, you can use the /DTSF option of DB2DART to dump tablespace and container information, and then look at the type field for the container in question. The query container APIs (sqlbftcq and sqlbtcq) can be used to create a simple application that will display the type.

3.4.9 DB2 Storage Management view

On a single-partition or a partitioned database environment you can use the Storage Management GUI tool provided by DB2 to monitor the storage state, and take storage snapshots for a database, a database partition group, or a tablespace. The Storage Management view is the graphical interface to the Storage Management tables and stored procedures. See Figure 3-20 on page 171.

The following information is collected from the system catalogs and database monitor tables when a storage snapshot is taken:

- ▶ **Tablespace**
Statistical information is collected from the system catalogs and database monitor for tables, indexes, and containers defined under the scope of the given tablespace.
- ▶ **Database partition group**
Statistical information is collected for all the tablespaces defined in the given database partition group.
- ▶ **Database**
Statistical information is collected from all the database partition groups within the database.

Different types of storage snapshots can be used to help you monitor different aspects of storage:

- ▶ Space usage can be monitored through snapshots of tablespaces.
- ▶ On partitioned databases only: Data skew (database distribution) can be monitored best through snapshots of database partition groups.
- ▶ The cluster ratio of indexes can be captured through both database partition group snapshots and tablespace snapshots.
- ▶ The Storage Management view also enables you to set thresholds for data skew, space usage, and index cluster ratio.

Note: You can only monitor data skew thresholds for partitioned databases.

3.4.10 DB2 Health Monitor for storage management

The Health Monitor use health indicators to evaluate of the database manager or database performance. It measures either a finite set of distinct states or a continuous range of values to determine whether the state is healthy or unhealthy. The state defines whether or not the database object or resource is operating normally. If it is not, then it issues an alert through the specified reporting channels.

A health indicator is a specific measurement that gauges the healthiness of some aspect of a particular database object; for example, if some value from a tablespace object enters the threshold range of values that defines an alarm zone, then an alarm type alert is issued to indicate that the problem needs immediate attention. There are three types of alerts: Alarm, warning, and attention.

Health Monitor information can be accessed through the Health Center, Web Health Center, the CLP, or APIs. Health indicator configuration is available through these same tools.

The following are the categories of health indicators:

- ▶ Tablespace storage
- ▶ Sorting
- ▶ Database management system
- ▶ Database
- ▶ Logging
- ▶ Application concurrency
- ▶ Package and catalog caches, and workspaces
- ▶ Memory

In this section we discuss the tablespace storage categories.

SMS tablespace container utilization

This indicator tracks the consumption of storage for each SMS tablespace. If free space is not available on the file system to expand an SMS container, the associated tablespace becomes full. An alert may be issued for each container defined on the file system that is running out of free space. To calculate this see Example 3-11 on page 168.

Example 3-11 Indicator formula for SMS

$$(\text{fs.used} / \text{fs.total}) * 100$$

where fs is the filesystem in which the containers resides.

SMS tablespace utilization is measured as the percentage of space consumed, where a high percentage indicates less than optimal function for this indicator. The short-term and long-term growth rates, included in the additional information for this indicator, can be used to determine if the current rate of growth is a short-term aberration or consistent with longer term growth. The calculation of time remaining to fullness in the additional information is a prediction of how much time is remaining until all free space is consumed.

To analyze a comprehensive view of tablespace storage utilization, use the Storage Management tool. This tool can be configured to take storage snapshots as the basis of trend analysis.

From the Control Center, expand the object tree under the chosen database until you find the Tablespaces folder. Open this folder, right click the tablespace that you want on the detail view, and click **Manage Storage**.

If you are using the Manage Storage tool for the first time on the database, you will be brought to the Storage Management Setup Launchpad. See Figure 3-19 on page 170.

On the Storage Management Setup Launchpad there are three tasks that must be completed in order to manage storage:

- ▶ Specify snapshot storage.
This task creates the storage management tables. You must specify a tablespace or create a new one. You will not be able to change the table specified at a later time.
- ▶ Specify threshold settings.
This task allows you to specify warning and alarm thresholds for the following criteria: Space usage, data skew, and cluster ratio. Data skew measures the balance of data between the database partitions of a database, database partition group, or table. Cluster ratio measures the quality of coverage an index has on one table. Default warning and alarm threshold values are already provided. You can change these threshold values at any time.
- ▶ Specify snapshot schedule.
This task allows you to schedule an initial snapshot. You can also start a snapshot capture immediately with this task.

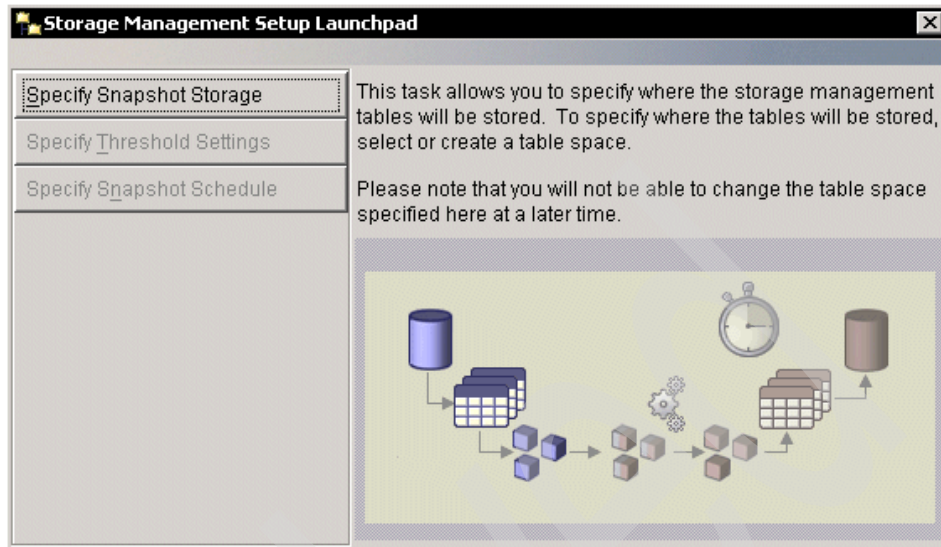


Figure 3-19 Storage Management Setup Launchpad

Once the setup is completed, you will be brought to the Storage Management dialog. The Storage Management Launch Setup only has to be done once per database. You will launch to the Storage Management view directly the next time you click Storage Management.

The storage management contains the following information:

- Tables: Table name, table schema, column count, number of rows, estimated size (MB), number of pages, F pages, tablespace name, index tablespace name, long data tablespace name
- Indexes: Table schema, table name, index schema, index name, cluster ratio, cluster warning threshold, cluster alarm threshold, cluster factor, column count, number of leaf pages, sequential pages, number of index levels, number of distinct first key values, number of two-column distinct key, number of three-column distinct keys, number of distinct full keys, page density

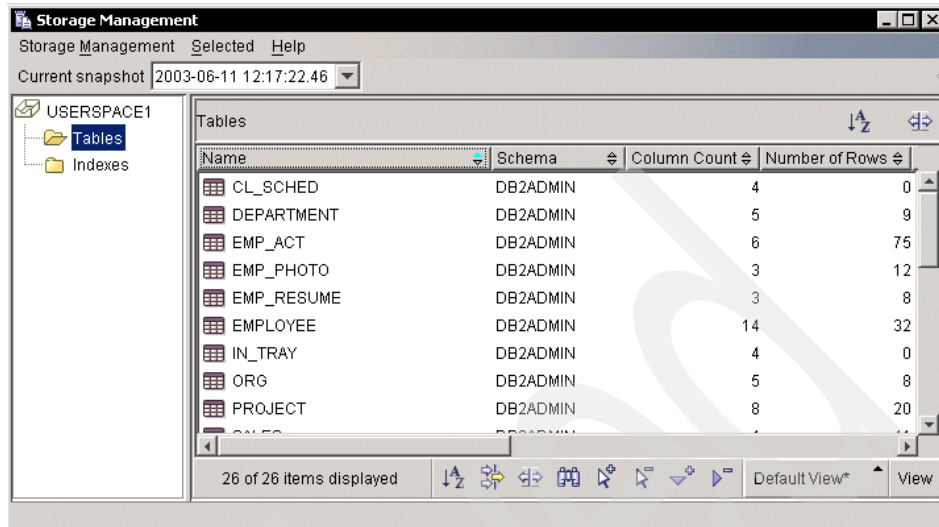


Figure 3-20 Storage Management view

DMS tablespace utilization indicator

This health indicator tracks the consumption of storage for each DMS tablespace. The DMS tablespace is considered full when all containers are full.

Example 3-12 Indicator formula for DMS

$$(\text{ts.used} / \text{ts.useable}) * 100$$

where `ts.used` and `ts.useable` are the system monitor elements “Used Pages” in tablespace and “Useable Pages” in tablespace respectively

To analyze a comprehensive view of tablespace storage utilization, launch the Storage Management tool. This tool can be configured to take storage snapshots as the basis of trend analysis.

From the Control Center, expand the object tree under the chosen database selected until you find the Tablespaces folder. Open this folder, right click the tablespace that you want on the detail view, and click **Manage Storage**. The Storage Management dialog opens. See Figure 3-20.

You can increase the number of tablespace containers to accommodate the anticipated rate of growth. Example 3-13 shows the formula the tool used to calculate the rate.

Example 3-13 Estimating capacity growth

Tablespace operational state health indicator

This health indicator tracks the state of a tablespace. A specific state can restrict activity or tasks that can be performed on that tablespace.

If the state is non-normal, an Attention alert may be generated. The non-normal state may be informational or critical.

The following is a list of different tablespaces states and the available commands to reset the states:

- **Tablespace in quiesced share state**

A user with the appropriate authority put the tablespace into QUIESCED SHARE. An explicit state reset by the user who issued the original quiesce can return the tablespace to normal state. The parent table in the tablespace with the QUIESCED SHARE state cannot be updated. However, other share mode requests to the tables in this tablespace are allowed.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the next command:

```
quiesce tablespaces for table table-name reset
```

- **Tablespace in quiesced update state**

The tablespace has been put into intent to update mode by an explicit request from a user with the appropriate authority. The tablespace is locked in intent exclusive (IX) mode and the parent table is locked in update (U) mode. An explicit state reset by the user who issued the original quiesce returns the tablespace to normal state.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the this command:

```
quiesce tablespaces for table table-name reset
```

- **Tablespace in quiesced exclusive state**

The tablespace has been put into QUIESCED EXCLUSIVE state by an explicit request from a user with the appropriate authority. The user who invoked the quiesce function (the quiescer) has exclusive access to the parent table and this tablespace. No other access to the tablespace is allowed. An explicit state reset by the user who issued the original quiesce returns the tablespace to normal state.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the command:

```
quiesce tablespaces for table table-name reset
```


► Load pending state

The tablespace has been placed in this state by a vendor's load utility. No tablespace access is allowed in Load pending state. To remove this state, refer to your load utility vendor's information.

► Delete pending state

This state is no longer used in version 8.

► Tablespace in backup pending state

The tablespace has been placed in backup pending state by the load utility because the load process has completed, and the database configuration parameter logretain is set to recovery or userexit is enabled, and the load option COPY YES is not specified, and the load option NONRECOVERABLE is not specified. The tablespace is still available for read access. To leave backup pending state, take a backup of the tablespace.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the command:

```
backup database database-name tablespace (table-space-name)
```

► Tablespace in rollforward pending state

The tablespace has been placed in rollforward pending state after a tablespace backup was restored or if the tablespace was taken offline by the database due to a media error.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the next command:

```
rollforward database database-name tablespace (table-space-name)
```

► Tablespace in restore pending state

The tablespace has been placed in restore pending state for one of two reasons. It could be done by the load utility because a rollforward operation was done through a LOAD command with the COPY NO option specified and the load operation completed successfully. Or, during a rollforward, a log record indicating that a point in time (PIT) rollforward tool place in the past was encountered, so the corresponding tablespace backup has to be restored. To leave restore pending state, perform a restore operation.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the next command:

```
restore database database-name tablespace (table-space-name)
```

- ▶ **Disable Pending State (DPS)**

DPS has placed this tablespace in the disable pending state because it hit an error during a database rollforward. You have to wait for the rollforward to complete for the tablespace to leave this state.

- ▶ **Not applicable for rollforward in progress state**

The rollforward utility is in progress on the tablespace due to an explicit request from the administrator. You have to wait for the utility to complete for the tablespace to return to normal state.

- ▶ **Reorg in progress state**

The reorg utility is in progress on the tablespace due to an explicit request from the administrator. You have to wait for the utility to complete for the tablespace to return to normal state.

- ▶ **Backup in progress state**

The backup utility is in progress on the tablespace due to an explicit request from the administrator. You have to wait for the utility to complete for the tablespace to return to normal state.

- ▶ **Restore in progress state**

The restore utility is in progress on the tablespace due to an explicit request from the administrator. You have to wait for the utility to complete for the tablespace to return to normal state.

- ▶ **Storage must be defined state**

The tablespace has been placed in storage must be defined state because, while defining containers for the tablespace during a redirected restore via the SET TABLESPACE CONTAINERS command, an error occurred while attempting to acquire the specified containers. To remove this state, reissue the SET TABLESPACE CONTAINERS command to make all the containers specified accessible.

- ▶ **Tablespace offline and not accessible state**

The tablespace is offline or not accessible due to a failure, possibly in tablespace initialization or rollforward processing. For example, with circular logging, if a tablespace container is not available on start up, and the database is in a consistent state, the tablespace will be placed in the offline state. If the database is in an inconsistent state, the database cannot be restarted until the tablespace is placed in the DROP PENDING state by using the DROP PENDING TABLESPACES clause in the RESTART DATABASE command. The same situation can occur if logretain is on. However, for the case where the database is in an inconsistent state, the tablespace is also placed in ROLLFORWARD PENDING STATE.

If the problematic container becomes accessible, then the tablespace can be placed online either by disconnecting and reconnecting to the database or, alter the tablespace using the SWITCH ONLINE option.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the next command:

```
alter tablespace table-space-name switch online
```

- Drop tablespace for drop pending state

The tablespace has been placed in drop pending state because a user requested the state through the DROP PENDING TABLESPACES clause in RESTART DATABASE. This state cannot be removed. The only allowable action is to drop the tablespace.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the next command:

```
drop tablespace table-space-name
```

- Storage may be defined state

The tablespace has been placed in this state because a user requested that a redirected restore take place, and now is the opportunity for the user to redefine containers for this tablespace. To remove this state, continue the restore.

- DMS rebalancer is active

The rebalancer is currently active on this tablespace. You have to wait for the rebalancer to complete for the tablespace to return to normal state.

- TBS creation in progress state

The tablespace was placed in TBS creation in progress state because the tablespace, although created, is not ready for use. You have to wait for creation to complete for the tablespace to return to normal state.

- TBS deletion in progress state

The tablespace was placed in TBS deletion in progress state because the tablespace deletion has not completed. There is no action to take to remove this state.

- Set write resume for database for write suspend state

The tablespace has been placed in this state due to an explicit request from a user by issuing a SET WRITE SUSPEND FOR DATABASE command. To remove this state, set write resume for the database.

From the Command Line Processor you can return the database to a normal mode with the next command:

```
set write resume for database database-name
```

Tablespace Container Operational State health indicator

This health indicator tracks the activity being performed on a tablespace container. The state of a tablespace container can restrict activity or tasks that can be performed on that tablespace container.

If the state is not accessible, an Attention alert may be generated. Tablespace in not accessible state: The container has been placed in not accessible state because DB2 was unable to open the container while initializing the tablespace for use. This could happen because the permissions are incorrect or because the physical container has been deleted. Ensure that the container exists and the permissions are correct, and either disconnect and reconnect to the database or alter the tablespace using the SWITCH ONLINE option.

From the Control Center or from the Command Line Processor you can return the tablespace to a normal mode with the next command:

```
alter tablespace table-space-name switch online
```

Scaling out DB2

On a multi-node environment, DB2 UDB extends the database manager's capability to parallel respond to today's business demands. These demands include flexibility, security, power to store and retrieve data, support for highly transactional systems, and very large databases for decision support systems.

This chapter describes how to set up DB2 UDB V8.1 on Windows Server 2003 Datacenter Edition multi-node configuration. We provide a description of the minimum operating system requirements and discuss the definition of the DB2 Instance Owning Node and the addition of participating physical database partitions.

We introduce the use of the DB2 UDB Enterprise Server Edition setup wizard for creating the instance-owning database node and additional physical database partitions in other participating nodes. We also introduce the use of the Add Partitions wizard. It is part of DB2 UDB ESE Database Partitioning feature.

This chapter contains the following:

- ▶ Scaling out overview
- ▶ Preparing the environment
- ▶ Setting up DB2 Instance Owning Node
- ▶ Setting up additional DB2 physical database partitions
- ▶ Adjusting DB2 configuration
- ▶ Adding logical and physical database partitions

4.1 Overview

Scaling out DB2 UDB is the ability to add more servers to the DB2 system. This is the way to improve scalability when a single system won't meet the needs in memory, processor, and storage for your business. Scaling out is sometime related to "extra effort," but it is worth the effort if it fits your business requirements.

4.1.1 DB2 UDB and Windows clustering technologies

For database systems that have reached the maximum quantity of memory and/or maximum number of processors and/or storage capabilities in a single Windows-based system, the option of moving this machine to a cluster environment should be considered.

A cluster is a group of computers that work together to run a common set of applications and provide the image of a single system to the clients and applications. Clustering services have become increasingly essential for organizations deploying business-critical e-commerce and line-of-business applications.

Note: DB2 UDB is a highly scalable system. It can run up to 999 nodes.

If you add high-availability support to your partitioned environment, the Microsoft Cluster Service (MSCS) provides high availability for mission-critical applications. DB2 UDB's high-availability feature supports the MSCS. Multiple servers (nodes) in a cluster remain in constant communication. If one of the nodes in a cluster becomes unavailable as a result of failure or maintenance, another node immediately begins providing service, a process known as failover. The session will be lost in the occurrence of a failure. However, the system can be configured so that the client is able to reconnect to the service seamlessly. In this case, users who are accessing the system continue to use the services, and are unaware that it is now being provided from a different server (node). By increasing the number of nodes in a server cluster, an administrator has many more options for deploying applications and providing failover policies that match business expectations and reduce risks. Larger server clusters provide more flexibility in building multi-site, geographically dispersed clusters that provide for disaster tolerance, as well as traditional node and/or application failure.

Note: If you design a high-availability system on a physical partitioned environment with DB2 UDB and MSCS on Windows Server 2003 Enterprise and Datacenter Edition, there is a limitation of 8-node cluster support from Windows. If you intend to implement high availability with MSCS in a database partitioned environment with more than 8 nodes you will need to organize your nodes into multiple MSCS clusters.

MSCS and DB2 UDB High Availability are not discussed in this publication. Refer to *DB2 Data Recover and High Availability Guide and Reference* documentation or the IBM Redbook *DB2 UDB Exploitation of the Windows Environment*, SG24-6893, if you need more information.

4.1.2 Implementing very large databases on Windows platforms

With proper architecture, the servers in a particular DB2 configuration can be scaled up or out as necessary to meet growing performance and throughput needs.

Windows Server 2003 can be considered a powerful and safe option when implementing very large databases (VLDBs).

The number of servers required depends on the anticipated workload, and the size and types of requests that the servers will handle. Processors and memory should be sized appropriately for the applications and services the servers will be running, as well as the number of simultaneous user connections.

Very large databases demand fast processors and large memory needs, DB2 UDB can take advantage of the increased memory space of computers running the 64-bit versions of Windows Server 2003. This fact provides an incredibly powerful platform for the most computer-intensive applications.

The DB2 UDB ESE Partitioning feature is a product that takes advantage of the shared-nothing architecture (Figure 4-1 on page 180).

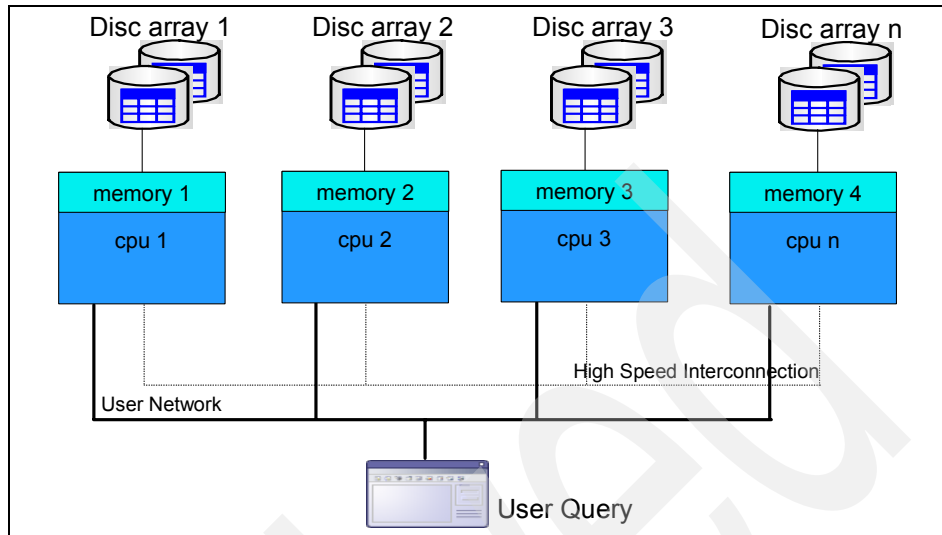


Figure 4-1 Shared-nothing architecture

In the shared-nothing architecture we can see that a processor or group of processors are linked by a high-speed interconnection. Each box or node has its own memory and accesses its own disks. Two important advantages exist in this architecture:

- ▶ Scalability in terms of storage, memory, and number of processors
- ▶ Performance gains from not having to share resources across the network

The memory growth in a single server or node is limited. When the maximum size is reached, the only way you can exceed this limit is by increasing the number of nodes because the memory is shared among machines. The same is true for disk storage and the number of processors. Another advantage that could be gained is in the number of operations that are performed. Each node only needs to do part of the work. Processing is therefore more distributed, and the database can manage a larger amount of data.

When a table is partitioned across two or more partitions, some of its rows are stored in one partition, and other rows are stored in other partitions. Because data is divided across database partitions, you can use the power of multiple processors on multiple physical nodes to satisfy requests for information. Data retrieval and update requests are decomposed automatically into sub-requests, and executed in parallel on the applicable database partitions. The fact that databases are split across database partitions is transparent to users and applications issuing SQL statements.

User interaction occurs through one database partition, known as the coordinator node for that user. Any database partition can be used as a coordinator node. This coordinator node is referred to as the *Instance Owning Database Partition*. Each database partition executes the operation on its own set of data. An exchange of information among the hosts may occur as a SQL statement is processed. The result from the operation is sent back to the coordinator node. The coordinator node assembles the data and returns it to the requestor.

A diagram of this model is shown in Figure 4-1 on page 180. The high-speed interconnection used between nodes is represented by dotted lines in Figure 4-1 on page 180. In a cluster, this would be a high-speed switch. The solid lines could represent a typical Ethernet network. The user query issues an SQL statement, for example, an SQL SELECT statement. Every database partition receives the operation from one node that works as the Instance Owning Database Partition. Applications and users accessing data in a partitioned database do not need to be aware of the physical location of the data.

4.1.3 Cluster benefits and limitations

When a database environment is migrated from a single node to a multi-node partitioned environment, the database design should be reviewed carefully in order to define which groups of tables should be partitioned and how many partitions should exist per table or group of tables. Adding nodes might result in communication and task-coordination issues. However, this choice provides the advantage of balancing data and user access across more than one system. DB2 UDB has supported this environment for many releases.

You can add nodes either while the database manager system is running or while it is stopped. If you add nodes while the system is running, however, you must stop and restart the system before databases migrate to the new node.

When you scale your system by changing the environment, you should be aware of the impact that such a change can have on your database procedures, such as loading data, backing up the database, and restoring the database.

When you add a new database partition, you cannot drop or create a database that takes advantage of the new partition until the procedure is complete, and the new server is successfully integrated into the system.

While clusters can be designed to handle failure, they are not fault tolerant with regard to user data. The cluster by itself does not guard against loss of a user's work.

The benefits of clusters are used to solve three typical problems in a data center environment:

► Need for high scalability

High scalability refers to the ability to add resources and computers while attempting to improve performance. A solution is highly scalable if it can be scaled up and out. Individual systems in a service offering can be scaled up by adding more resources (for example, CPUs, memory, disks, etc.). The service can be scaled out by adding additional computers.

DB2 UDB applies the concept of *function shipping* to improve performance on partitioned environments. It:

- Assists in the reduction of network traffic because a process, such as SQL queries, is shipped instead of data.
- Relational operators are executed on the node partition containing the data whenever possible, so the SQL, or part of the SQL, is moved to where the data resides.
- Function shipping is well suited to the shared-nothing architecture model.

► Need for high availability

A server cluster provides high availability by making application software and data available on several servers linked together in a cluster configuration. If one server stops functioning, a process called failover automatically shifts the workload of the failed server to another server in the cluster. The failover process is designed to ensure continuous availability of critical applications and data. A solution is highly available if it meets the organization's scheduled uptime goals. Availability goals are achieved by reducing unplanned downtime and then working to improve total hours of service operation.

► Need for high reliability

High reliability refers to the ability to reduce the frequency of system failure, while attempting to provide fault tolerance in case of failure. A solution is highly reliable if it minimizes the number of single points of failure and reduces the risk that failure of a single component/system will result in the outage of the entire service offering. Reliability goals are achieved using redundant, fault-tolerant hardware components; application software; and systems.

4.2 Preparing the environment for scaling out DB2 UDB

This section covers the basic tasks involved in the Windows Server 2003 environment to support a DB2 UDB physical partitioned environment.

4.2.1 Installation overview

Figure 4-2 on page 184 shows a DB2 UDB Enterprise Server Edition partitioned environment configuration with two partitions, one per computer. Setup instructions are based on this configuration but can easily be adjusted for partitioned configurations with a greater number of computers and database partition servers.

The steps involved in this section are focused on how to prepare the Windows Server 2003 environment for a partitioned installation of DB2 UDB Enterprise Server Edition. For a complete review of software and hardware requirements see 2.2.1, “Installation requirements” on page 46.

Before you install, you must prepare your environment for installation. In some work environments, the System Administrator will perform these tasks. To prepare your environment, you will:

- ▶ Verify that each computer meets the necessary operating system, memory, and disk requirements.
- ▶ Ensure that all computers belong to the same Windows domain.
- ▶ Ensure that all computers have consistent time and date settings.
- ▶ Verify that all computers can communicate with each other via TCP/IP.
- ▶ Add a domain user account to the local Administrator group on each computer.
- ▶ Optionally create DB2 user accounts for setup.

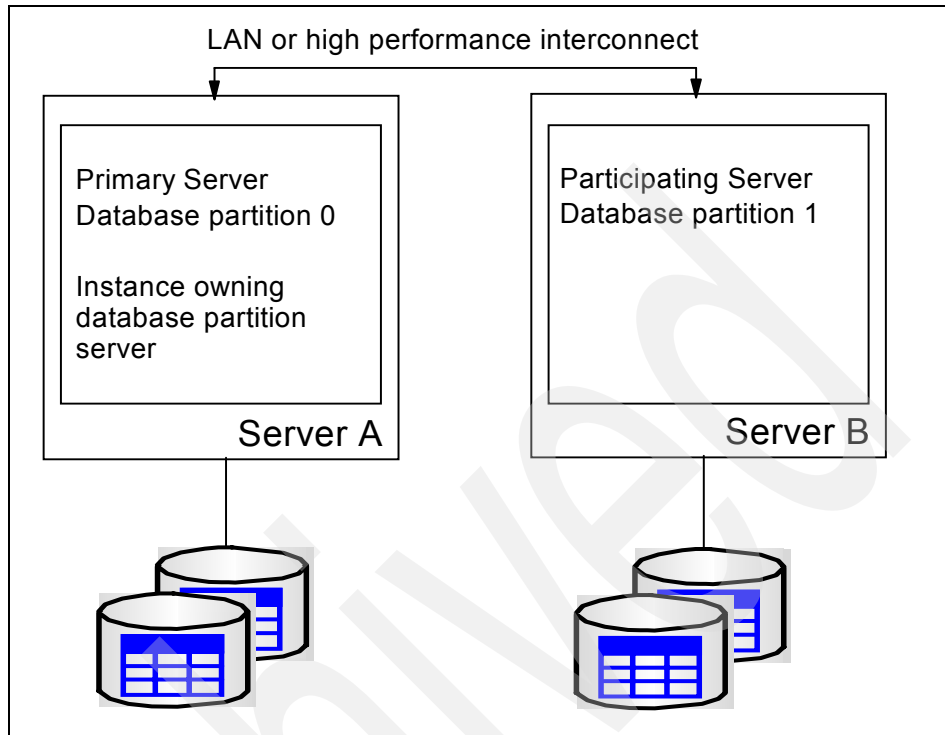


Figure 4-2 DB2 UDB partitioned environment with two physical nodes

4.2.2 Windows Server 2003 Domain Controller setup

If your primary server and participating servers are not under a domain or the domain does not exist, then you must create a Domain Controller server in order to install DB2 UDB as a partitioned database server. If you already have a configured cluster, be sure to fulfill the points mentioned in this section.

Before creating a Windows Server 2003 Domain Controller

Consider the following when starting a configuration of a DB2 UDB partitioned environment on a Microsoft Windows Cluster:

- Ensure that the primary computer and participating computers belong to the same Windows domain.

On Windows Server 2003, check the domain that the computer belongs to using the System Properties dialog, accessible through the Control Panel. If there is not a current domain, you must create one.

- Ensure that the time and date settings on the primary computer and participating computers are consistent. In a Windows 2000/2003 domain, the

computers in the domain get their times from the Domain Controller. You have to check that the communication and name resolution to the Domain Controllers is running properly.

To be considered as consistent, the difference in GMT time between all computers must be no greater than one hour. System date and time can be modified using the Date/Time Properties dialog, **Start -> Settings -> Control Panel -> Date/Time**.

After installing DB2 UDB, you can use the *max_time_diff* database manager configuration parameter to change this restriction. The default value of *max_time_diff* is 60, which allows a difference of less than 60 minutes. The following command shows how to update this value in the DB2 database manager configuration:

```
update database manager configuration using max_time_diff 1
```

This command sets the max time difference between nodes to one minute.

- Ensure that all participating computers can communicate with each other using TCP/IP.

On the primary participating computer, enter the **hostname** command, which will return the host name of the computer.

Example 4-1 hostname command

```
c:\>hostname
ServerA
c:\>
```

On another participating computer, enter the following command:

```
ping hostname
```

Where *host name* represents the hostname of the primary computer. If the test is successful, you will receive output similar to the following:

```
Pinging ServerA.ibm.com [9.21.27.230] with 32 bytes of data:
Reply from 9.21.27.230: bytes=32 time<10ms TTL=128
...
```

Repeat these steps until you are sure that all participating computers can communicate with each other using TCP/IP.

Note: Each computer must have a static IP address.

- Have a Windows Server 2003 CD or network install path available.

Installing the Domain Controller

To set up the Domain Controller, follow the steps below:

1. Start a session as the Administrator user in the Windows Server 2003 system selected as a Domain Controller server. Under this account you can start the Active Directory Installation Wizard to set up this server as a Domain Controller. To start the wizard, right click **Start** and then select **Manage Your Server** (see Figure 4-3). In this window select the option **Adding Roles to Your Server**.

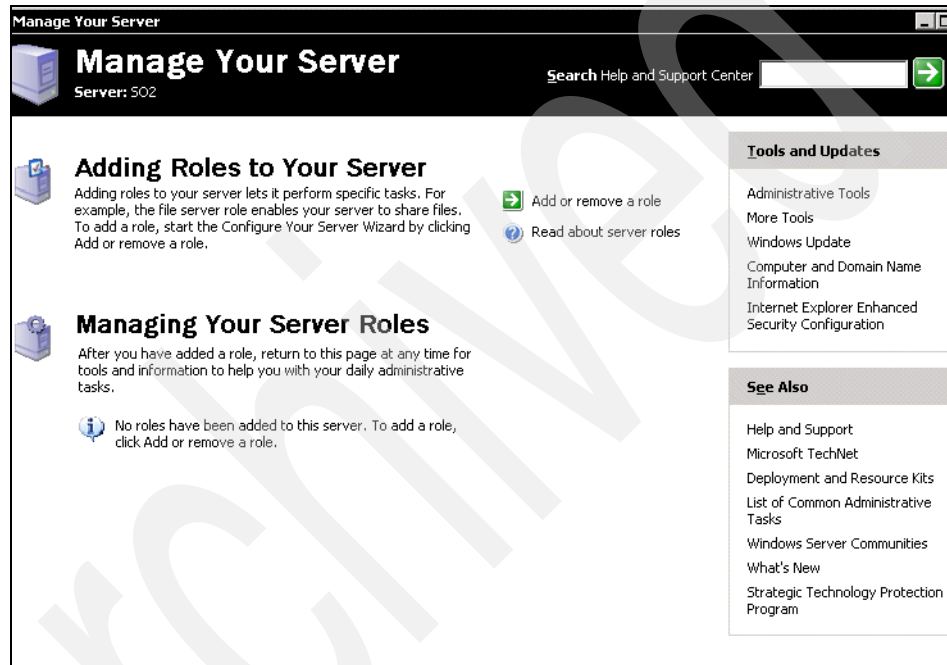


Figure 4-3 *Manage Your Server Wizard*

2. A Preliminary Steps window appears. Verify that the steps listed on it have been completed. Some of them are not needed for our example. The most important thing in this basic configuration is the correct installation and configuration of TCP/IP and the availability of a Windows Server 2003 CD or network installation path. Keep this in mind and click the **Next** button.
3. A list of server roles is displayed (see Figure 4-4 on page 187). From this list, select the option **Domain Controller (Active Directory)**. DB2 uses the services provided by the Domain Controller and focuses on logon processes and server services controlled through the domain. Click the **Next** button.

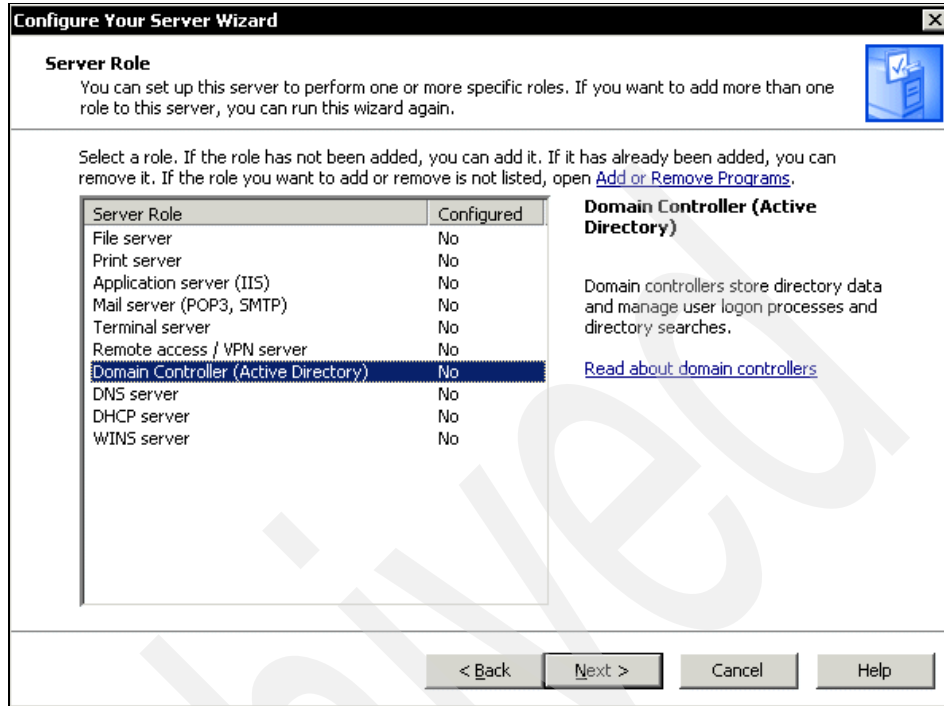


Figure 4-4 Defining a server role in Windows Server 2003

4. The Summary of Selections window appears. Review that the Active Directory Wizard will run. If it is OK, click the **Next** button.
5. The Active Directory Installation Wizard begins the process to create this server as a Domain Controller Server. This wizard is shown in Figure 4-5 on page 188. Start the wizard process by clicking the **Next** button.
6. The Operating System Compatibility window shows a message about compatibility issues regarding security between Windows Server 2003 and operating systems like Windows 95 and Windows NT 4.0 SP3 or later. Consider this important issue because users on these operating systems will not be able to log onto a Domain Controller running Windows Server 2003. For the configuration in this example, there are no security issues since all the participating servers run under Windows Server 2003. Click the **Next** button.



Figure 4-5 Active Directory Installation Wizard

7. In the next window you must specify the type of Domain Controller. See Figure 4-6 on page 189. You can create a new Domain Controller or add one for an existing domain. In our example, we select the option **Domain Controller for a new domain**. Click the **Next** button.

For further information about domain types refer to the online help or Microsoft Windows Server 2003 documentation.

8. The following window allows you to select the domain type. There are three types:
 - Domain in a new forest
Select this if it is the first domain in your configuration or if you want the new domain to be completely independent of your current forest.
 - Child domain in an existing domain tree
If you want the new domain to be a child of an existing domain.

- Domain tree in an existing forest

This will create a new domain tree that is separated from any existing trees.

For our example, select the first option, **Domain in a new forest**, and click the **Next** button.

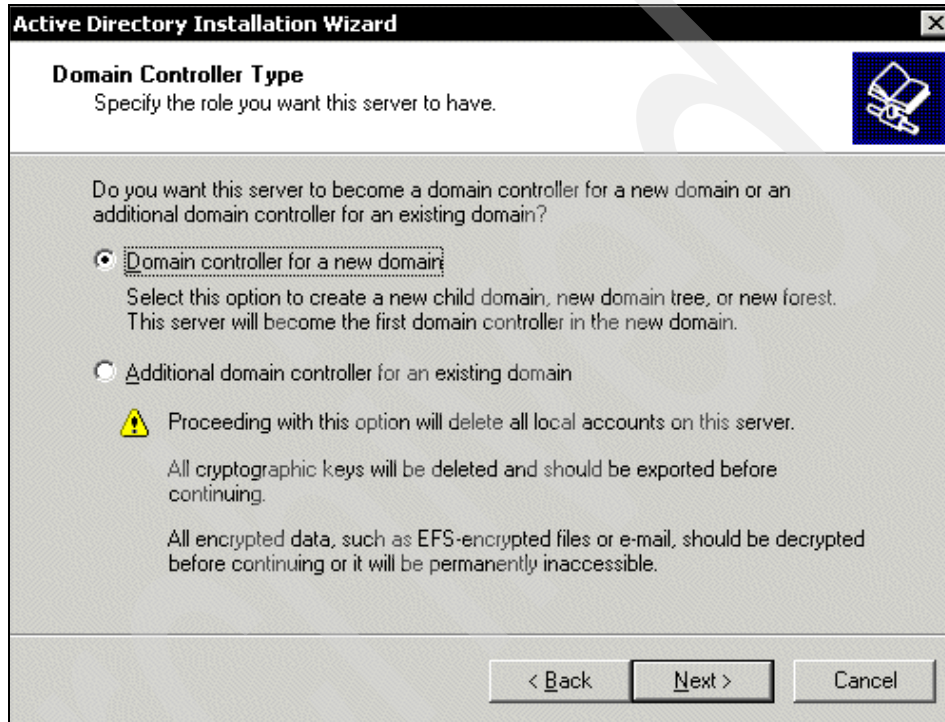


Figure 4-6 Domain Controller type

9. The new window will ask you to specify a name for the new domain. You can type any name. In our example, we choose a DNS name called redbooks.itso.ibm.com. Click the **Next** button to continue.
10. Now a request for a Domain NetBios Name is shown. This is the name that users of earlier versions of Windows will use to identify the new domain. You can accept the name shown or type a new name. Click **Next**.
11. For the Database and Logs Folders, accept the defaults and click **Next**.
12. For the Shared System Volume, accept the default and click **Next**.
13. On the Permissions window, select the option **Permissions compatible only with Windows 2000 or Windows Server 2003 operating systems** and click **Next**.

14. Type a password in the Directory Services Restore Mode Administrator Password. This is used when we start the computer in a restore mode (for Directory Services). The restore Administrator account is different from the domain Administrator account. The passwords for the account might be different, so be sure to remember both. For more information about Directory Services Restore Mode, see the Active Directory help or Microsoft documentation. Click **Next** to continue.
15. A review window is displayed showing a list of the options selected in previous steps (see Figure 4-7). If you agree, click the **Next** button to start the installation of Active Directory.

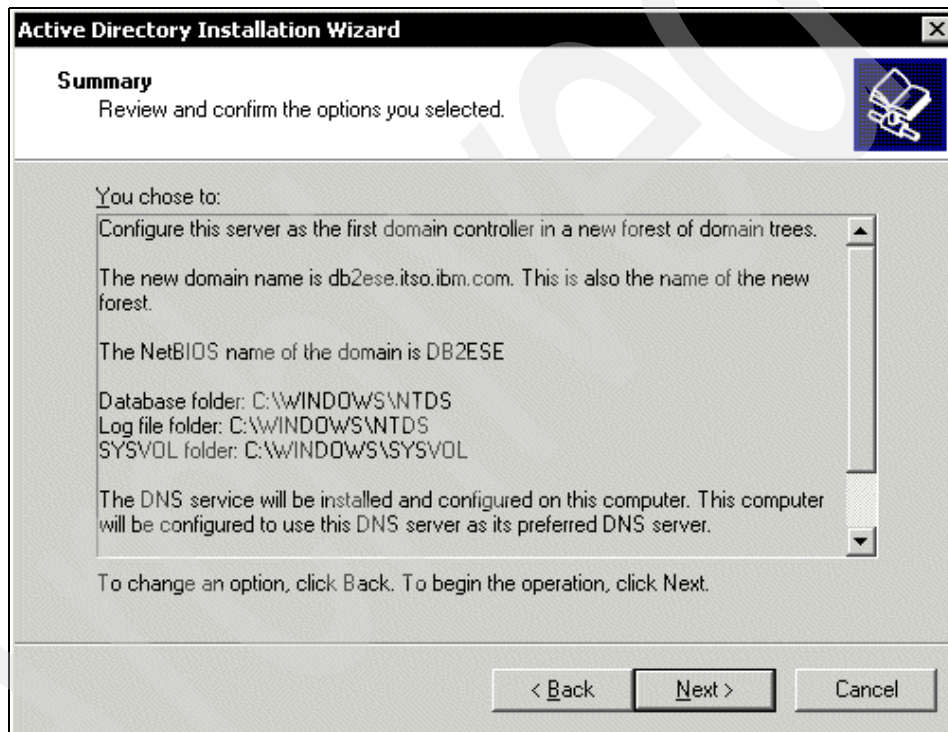


Figure 4-7 Active Directory summary window

You must wait until the wizard configures the Active Directory. This process configures the local Domain Controller to host Active Directory services in the selected server.

After a while during the installation process, you will be asked to provide the Windows Server 2003 CD-ROM or an alternate location, such as a floppy disk or a network server, in order to start copying files needed for the configuration process of the Active Directory services. Depending on the configuration of your server, this process could take several minutes.

16. The Active Directory is now installed and configured (see Figure 4-8). Click the **Finish** button.

Windows must be restarted before the changes made by the Active Directory Installation Wizard take effect.

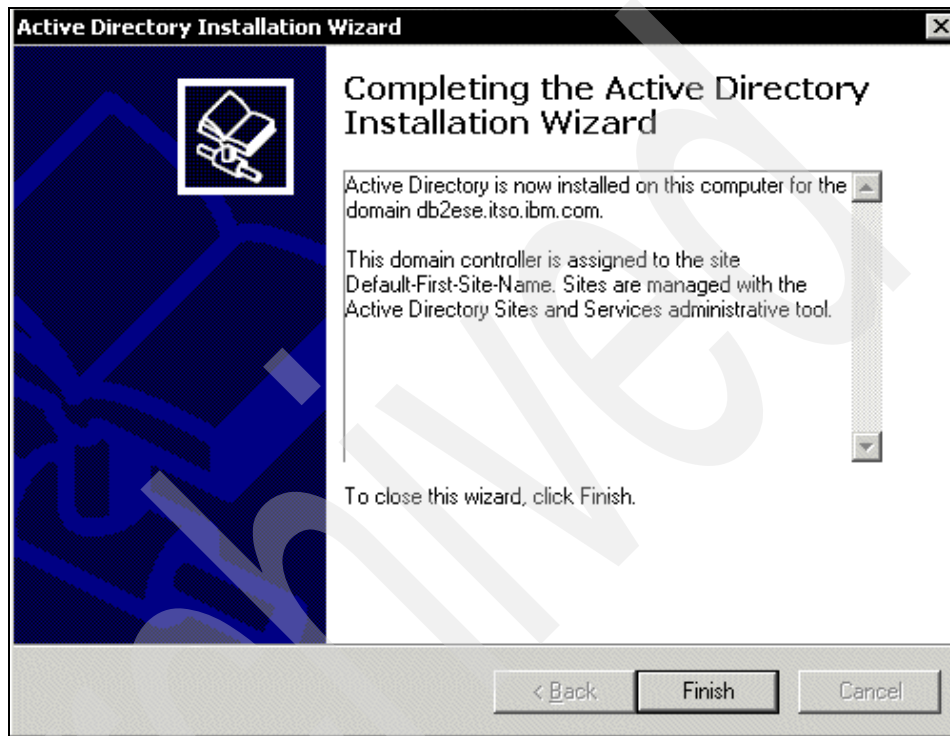


Figure 4-8 Completing the Active Directory Wizard

After rebooting the server, a window shows a message confirming that installation and setup were successful. See Figure 4-9 on page 192. Click the **Finish** button.

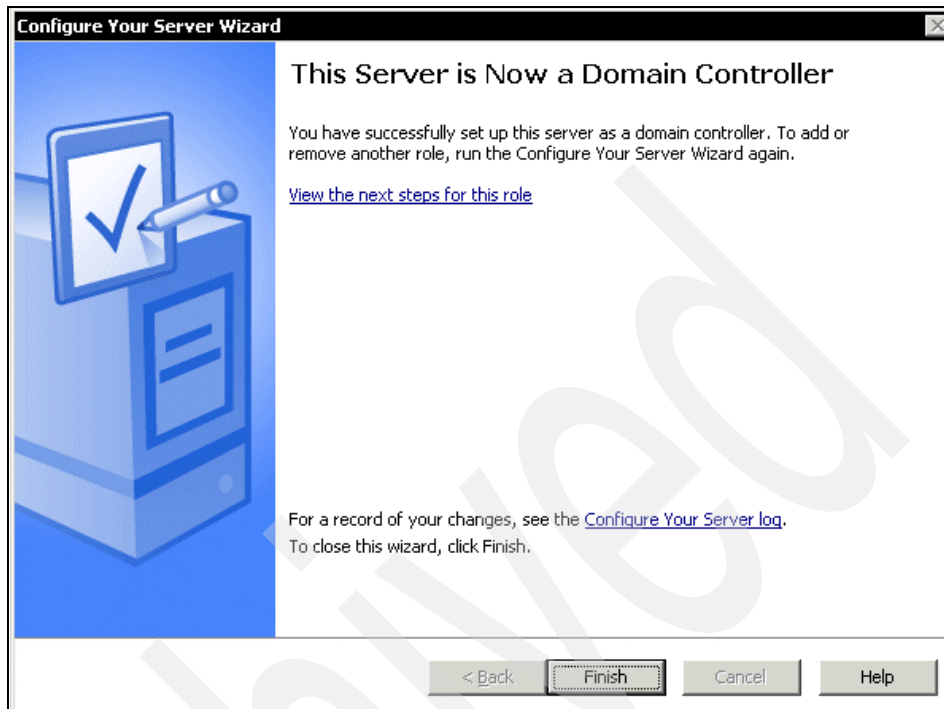


Figure 4-9 Domain Controller successfully set up window

Adding a server to the domain

Before we can start configuring DB2 UDB for a partitioned environment, we must add the participating servers for the partitioned environment to the same domain. The following steps walk you through this process.

Note: Each participating computer must have the same operating system. For example, you cannot have a partitioned database system that includes both Windows 2000 Server and Windows Server 2003 operating systems. This is a DB2 UDB requirement.

We highly recommend that you create a domain user under the Domain Administrator Group on the Domain Controller Server. This domain user will be used in the next steps to configure the physical partitions in the DB2 UDB partitioned environment. In the last step of this section, you will use this user to authenticate the server into the Domain Controller Server.

Before adding a server to a domain, check the IP settings of this machine and make sure that it communicates with the same DNS servers like the Domain

Controller (DC). In Windows 2000/2003 usually on a DC runs a DNS (AD integrated). You should add this IP address as the DNS server in your member server IP settings and test a ping to the fully qualified domain name (FQDN) of the DC.

1. Start by opening the System Properties window on the server. On the desktop select **My Computer**, right click and select **Properties**. Select the tab **Computer Name** and click the **Change** button in order to join this server to a domain. See Figure 4-10. You can also add a description of this server.

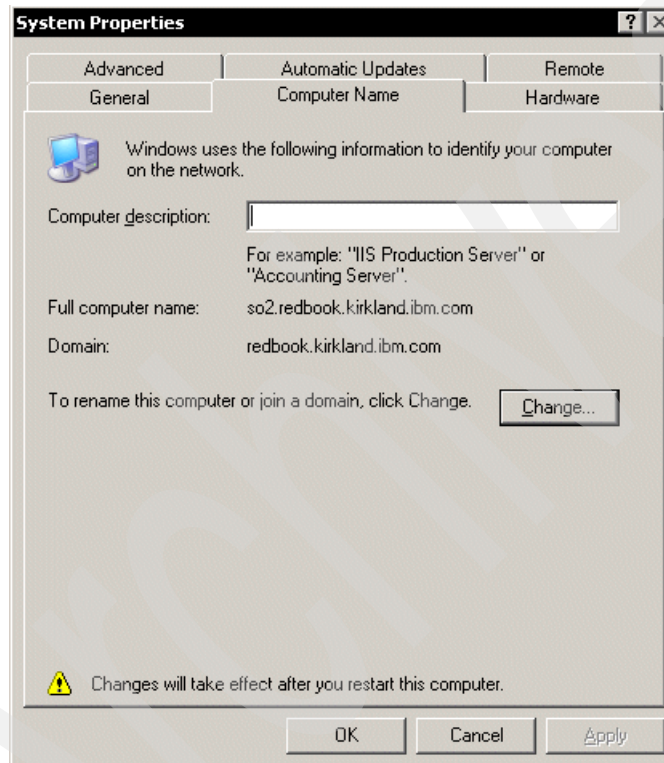


Figure 4-10 System properties: Joining an existing domain

2. Figure 4-11 on page 194 shows the Computer Name Changes window. In the Computer name entry enter a unique name for your server. On the Member of box select **Domain** option and enter the correct domain name in the Domain entry box. You will be asked to enter the name and password of an account with permission to join this computer to the domain. After the authentication is verified, click the **OK** button and this computer will be part of the domain. You must reboot your computer after completing this task.

Once we have two servers under the same domain, we are ready to start installing and configuring DB2 UDB for a physical cluster environment.

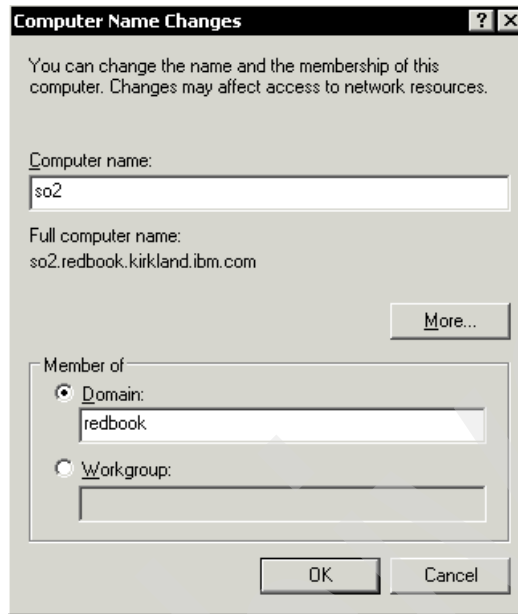


Figure 4-11 Computer Name Changes window

4.3 Installing a DB2 partitioned environment

This section will walk you through the steps of setting up the DB2 Instance Owning Node and additional physical database partitions using the DB2 Setup utility.

4.3.1 Before configuring DB2 on a physical cluster environment

You will have the same installation considerations as for installing DB2 UDB in a single server. There are additional comments regarding users, security, and the coordinator node in a server cluster for a partitioned database environment.

Users and security

During the DB2 UDB installation you will be asked to provide a local or domain user account that will be used by the DB2 Administration Server (DAS) to log onto the system and to start itself as a service. You can define a user now or have the DB2 Setup wizard create one for you. If you want to create a new

domain user using the DB2 Setup wizard, the account used to perform the installation must have the authority to create domain users.

On the primary computer, where you will install the instance-owning partition, you must have a domain user account that belongs to the local Administrators group. You must add the same user account to the local Administrators group on each participating computer. This user must have the “Act as part of the operating system” user right. You will log on as this user when you install DB2 UDB.

During the installation you will be asked to provide a domain user account to be associated with the DB2 instance. You can define a user now, or you can have the DB2 Setup wizard create a new domain user for you. If you want to create a new domain user using the DB2 Setup wizard, the account used to perform the installation must have the authority to create domain users. The instance user domain account must belong to the local Administrators group on all the participating computers, and it will be granted the following user rights:

- ▶ Act as part of the operating system.
- ▶ Create token object.
- ▶ Increase quotas.
- ▶ Log on as a service.
- ▶ Replace a process level token.
- ▶ Lock pages on memory.

Note: DB2, by default, gives full database and instance rights to the Public group. Make sure that you change the privileges of the Public group and remove the access it has to the database if you want to only allow a restricted list of users or groups access to the instance and database.

DB2 Instance Owning Node

A partitioned database environment allows a database to remain whole logically, despite being physically divided across more than one partition. The fact that data is partitioned remains transparent to most users. Work can be divided among the database managers on the different physical machines. Each database manager in each partition works against its own part of the database. This configuration combines the advantages of SMP and MPP parallelism. A query can then be executed on a single partition across multiple processors. It also means that a query can be executed in parallel across multiple partitions.

In DB2 UDB, user interaction occurs through one database partition, known as the *coordinator node* for that user. The coordinator runs on the same database partition as the application, or in the case of a remote application, the database partition to which that application is connected. Any database partition can be used as a coordinator node. This node is also known as the DB2 Instance Owning Node.

You should identify one physical computer to be the DB2 Instance Owning Node. This will be the first computer you install DB2 UDB on in the partitioned environment.

Note: Ensure that you install DB2 UDB to the same drive on each participating computer. For example, do not install DB2 UDB on the c: drive of the instance owning database server, on the d: drive of a database partition server, or on the j: drive of another database partition server. If you install DB2 UDB on the c: drive of the instance owning database server, then install DB2 UDB on the c: drive of any participating database partition servers.

4.3.2 Installing the DB2 Instance Owning Node

To install the instance owning database partition server, log on to the system with a user who belongs to the local Administrators group. This user is also a domain user and belongs to the Domain Administrators group.

1. Insert the DB2 UDB Enterprise Server Edition for Windows CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup Launchpad (Figure 4-12 on page 197). This installation runs like any other DB2 for Windows setup. For a complete reference on how to install and set up 64-bit DB2 UDB on Windows Server 2003, please refer to 2.4, “64-bit installation” on page 51.

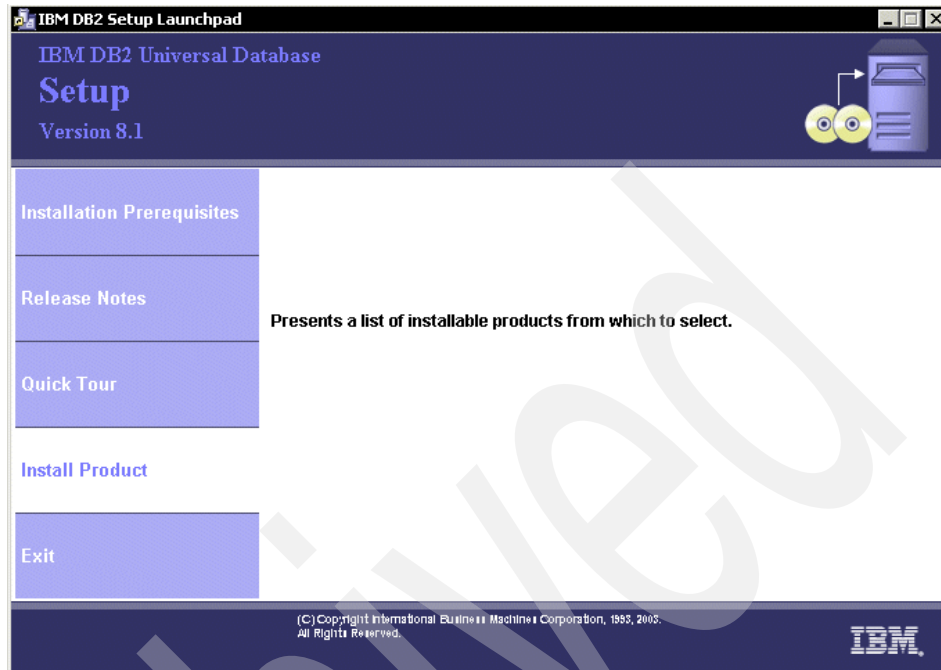


Figure 4-12 DB2 Setup Launchpad window

2. Following the steps guided by the Setup wizard, you will see a window asking how this computer will be used (see Figure 4-13 on page 198).

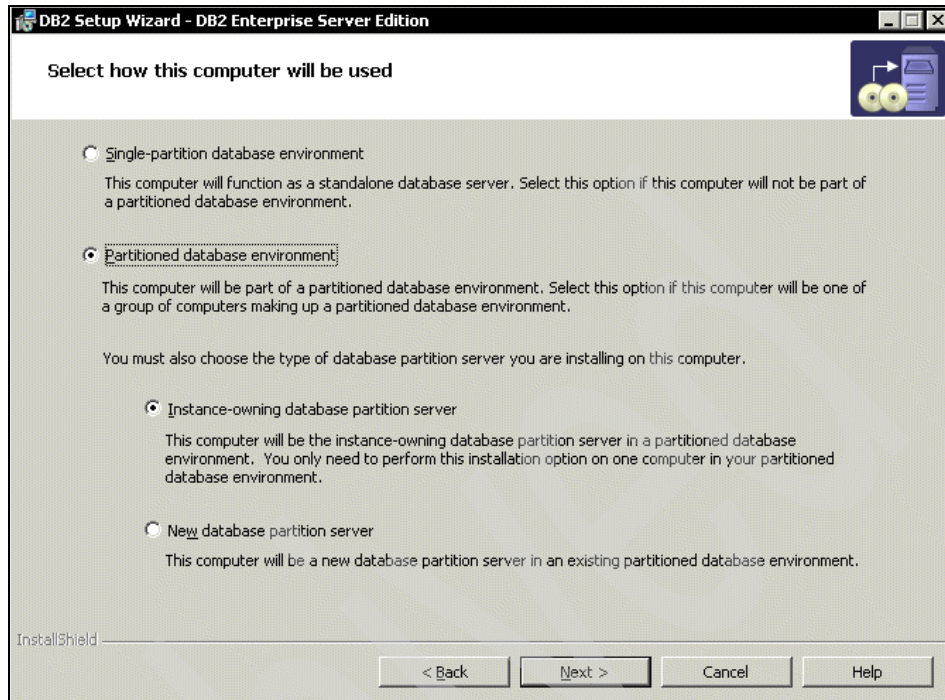


Figure 4-13 DB2 setup for partitioned database environment (instance-owning)

On the “Select how this computer will be used” screen, you can select the type of partitioned database environment:

- Single-partition database environment

This is a typical standalone database server configuration. If you select this option then you cannot use this server for a partitioned database environment.

- Partitioned database environment

This option allow you to set up and configure this server to be part of a partitioned database environment. When you select this option you must choose one of the partitioned database server types:

- Instance-owning database partition server

In a DB2 partitioned environment you at least need a partition server of this type. This is the partition through which your users and applications communicate to the database server in a partitioned database environment.

- New database partition server

In a physical DB2 partitioned environment, when you add additional participating servers to the partitioned database environment, use this option to install and set up a new partition.

As we are configuring a new instance-owning database partition, we must select the **Partitioned database environment** radio button and **Instance-owning database server** radio button. Click **Next** to continue.

3. The next screen will set up the *administration contact list*. Select **Local**. This selection will create a file on the primary computer (instance-owning database server) that will store contact information for your system.

The contact information is used by DB2 to send notifications and alerts to a system administrator. A notification may state that a job has completed. An alert may state that a system threshold has been breached. You can specify notification and alert parameters after setup is complete. Participating computers will remotely access the contact list on this computer. Click the **Next** button.

4. The following window sets the user information for the DB2 Administration Server (DAS). This user will perform administration and configuration tasks locally or remotely on this server and/or other partitioned database servers, so it is important to have this user under a domain users group to grant access to all participating database partitions. From the Domain drop-down box, select the domain in which your partitioned database will exist (see Figure 4-14 on page 200). You can also specify a domain name by entering the domain name in the Domain field. In the other fields, enter the user ID and password for the administration server user. Click the **Next** button.

Note: You must have the DAS user and instance on every participating server to perform remote database administration through the DB2 Control Center.

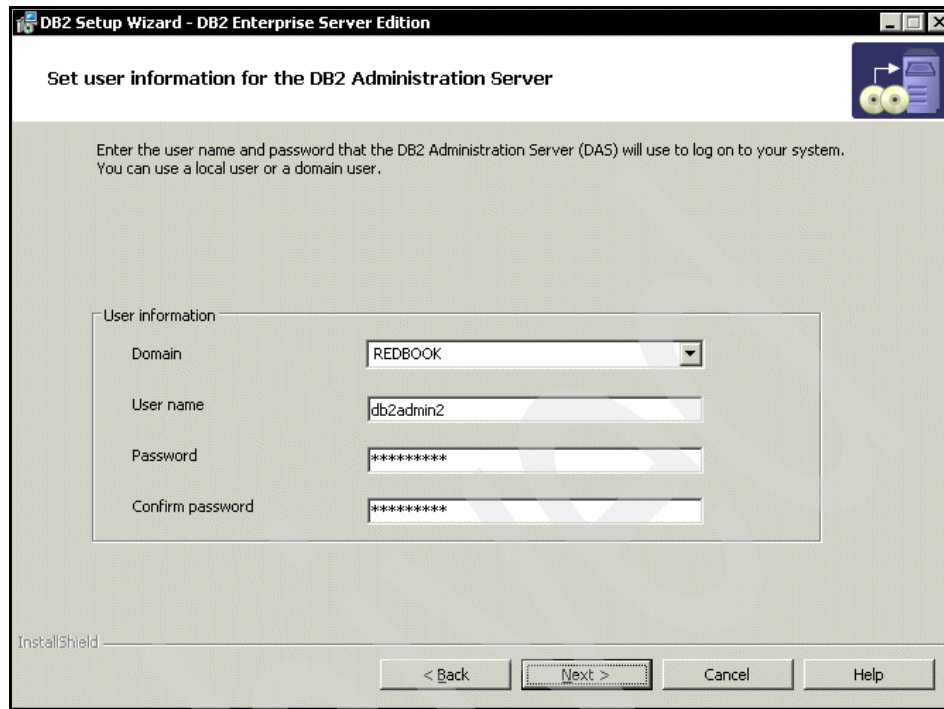


Figure 4-14 DB2 administration server information

5. The next screen is “Add new database partition server”. In this screen, DB2 will gather information pertinent to configure the new instance-owning computer. The default logical partitions for a computer is four. If you have one database partition per server, only one port is required. If you keep the default value of four, four ports will be reserved for database partition server communication. DB2 setup will attempt to reserve identical port numbers when you install database partition servers on participating computers.

In the “Add new database partition server” window, select the instance-owning computer by entering the name in the field. Or, if you click the drop-down button on the right side of the field, a new box is displayed, showing the participating servers under the domain. Select the computer you have identified as the instance-owning system and click the **OK** button. Figure 4-15 on page 201 shows a domain with two computers, SO2 and SU1. SU1 is selected as the instance-owning computer and SO2 will be used later to hold a new database physical partition.

For the Instance name field you leave the default (DB2) or type in a new one. In the Partition number field, use the value 0 because this is the dispatcher

partition in the database partitioned environment. Remember to use a domain user account to set up this partition.

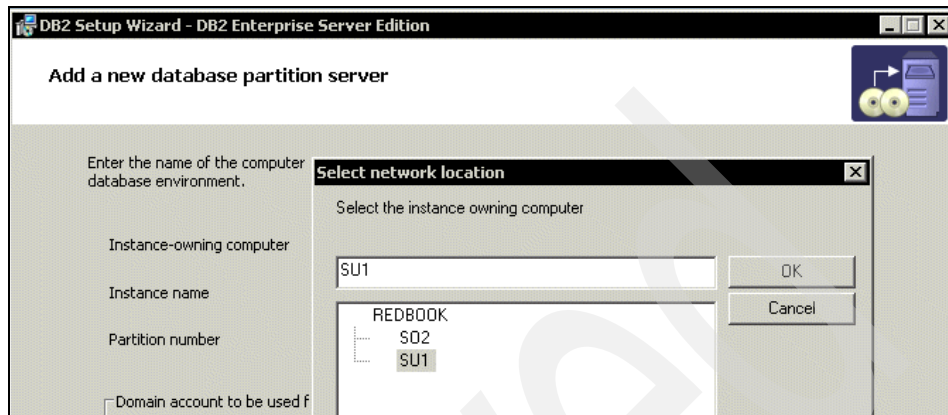


Figure 4-15 Select the instance-owning computer

Figure 4-16 on page 202 shows the “Add a new database partition server” panel with complete information to set up the instance-owning database server. In our case we are using a DAS account called db2admin2 to set up this server since this user ID is a local user in the Administrator group and also is a domain user belonging to Domain Administrators group. Click the **Next** button.

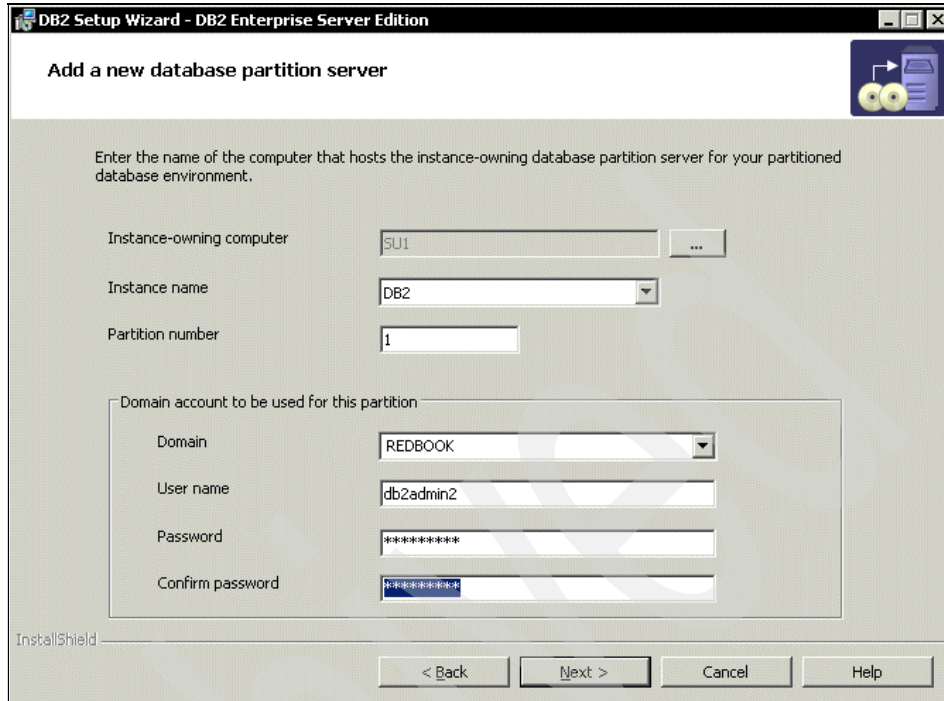


Figure 4-16 Add new database partition server

6. The setup program is ready to start copying files, verify the settings, and give a name to the response file (if you selected to create one at the beginning of the setup process). At the end, click the **Finish** button to complete the installation (see Figure 4-17 on page 203).

At this point you have DB2 UDB installed and the instance-owning computer configured.

What if something goes wrong

For information on errors encountered during installation, see the db2.log file. The db2.log file stores general information and error messages resulting from the install and uninstall activities. By default, the db2.log file is located in the My Documents\DB2LOG\ directory. The location of the My Documents directory will depend on the settings on your computer.

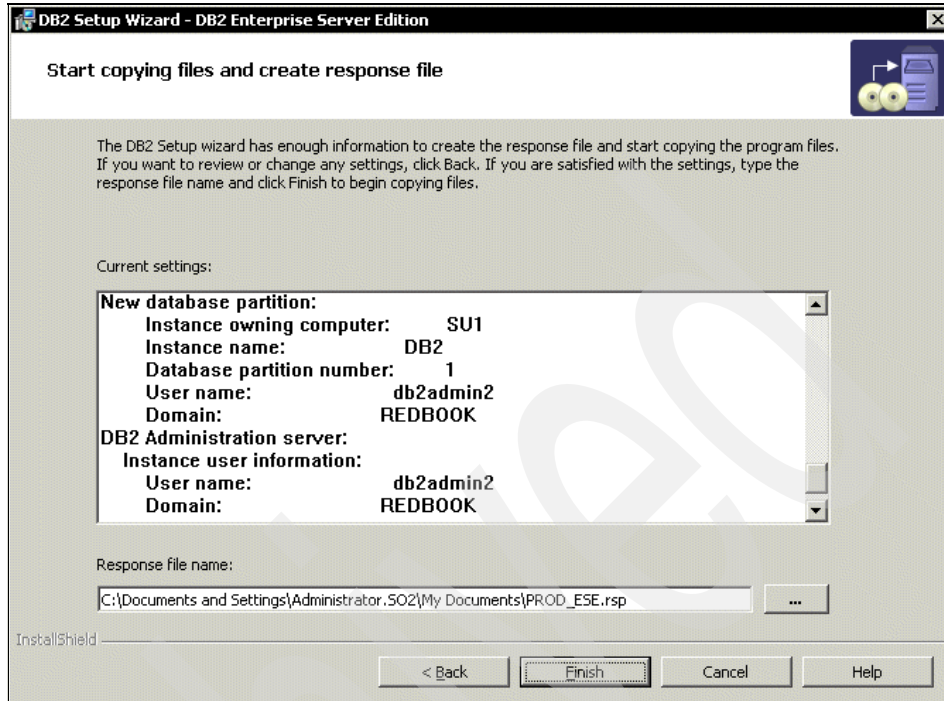


Figure 4-17 Review settings of the DB2 instance-owning partition

4.3.3 Installing a physical database partition

This section walks you through the steps of setting up a physical database partition using the DB2 Setup utility. Based on Figure 4-2 on page 184, we are going to install and configure Server B.

To install a physical database partition on a participating server:

1. Log on to the system with a user who belongs to the local Administrators group. This user is also a domain user and belongs to the Domain Administrators group.
2. Insert the DB2 UDB Enterprise Server Edition for Windows CD-ROM into the drive. If enabled, the auto-run feature automatically starts the DB2 Setup Launchpad (see Figure 4-12 on page 197). This installation runs like any other DB2 installation on Windows. For a complete reference on how to install and set up 64-bit DB2 UDB on Windows Server 2003, refer to 2.4, “64-bit installation” on page 51.
3. Following the steps guided by the DB2 Setup Wizard, you will see a window asking how this computer will be used (see Figure 4-18 on page 204).

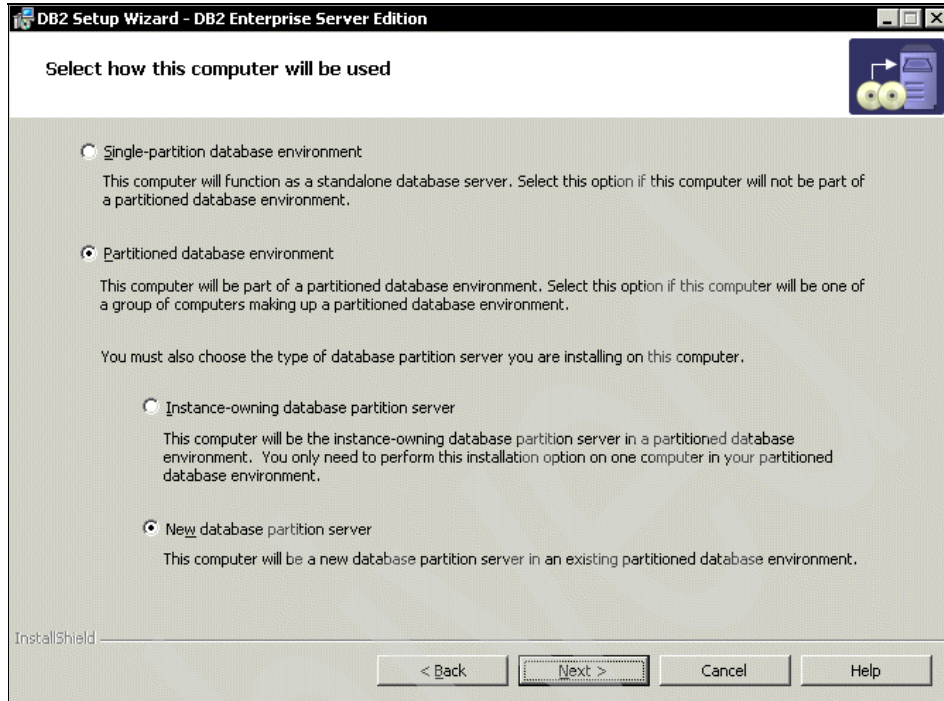


Figure 4-18 DB2 setup, partitioned database environment (additional partition)

On this window you must select the option **Partitioned database environment** and **New database partition server**, then click **Next**.

4. The next window will set up the administration contact list. Select **Remote**. Specify the host name of the primary computer (instance-owning server) and setup the contact list. Click **Next**.
5. The "Add new database partition server" window shown will gather information pertinent to configure the new database partition server. The default logical partitions for a computer is four. If you have one database partition server per computer, only one port is required. If you keep the default value of four, four ports will be reserved for database partition server communication. DB2 setup will attempt to reserve identical port numbers when you install additional database partition servers on participating computers.

In this dialog, first select the instance-owning computer by entering the name in the field. If you click the drop-down button on the right side of the field, a new box is displayed showing the participating servers under the domain. Select the correct instance-owning computer and click the **OK**.

For the Instance name field you leave the default (DB2) or enter the desired instance name. In the Partition number field use the value 1 because this is the first partition in the database partitioned environment. Remember to use a domain user account to set up this partition.

Figure 4-19 shows the “Add a new database partition server” panel with complete information to set up the additional physical database partition. In our case, we are using a DAS account called db2admin2 to set up this server since this user ID is a local user in the Administrator group and also is a domain user belonging to Domain Administrators group. Click **Next**.

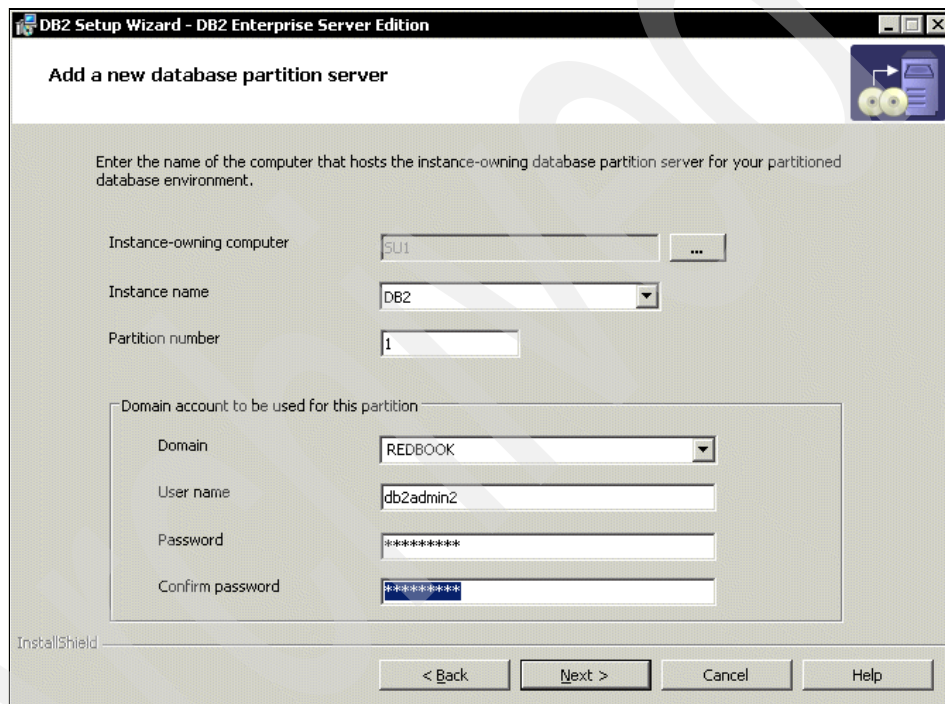


Figure 4-19 Adding a new database partition

6. Continue with the normal installation procedure of DB2 UDB. At this time the setup program checks that the primary node is correctly configured and running. It checks the TCP/IP ports and reserves the same ones for this new partition. The file db2nodes.cfg is updated on the primary server and this additional partition is configured. A sample db2nodes.cfg file with the instance-owning entry and an additional partition is shown in Figure 4-20 on page 206.

A new Windows system service is also created and registered in this new physical database partition. This service supports the instance in this new

database partition and is controlled by the instance-owning server. This is one of the reasons why you must have a domain user.

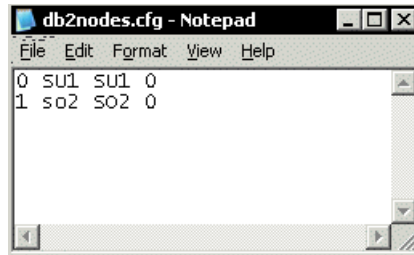


Figure 4-20 db2nodes.cfg file with the instance owning entry

4.3.4 Verifying DB2 partitioning configuration

Once the DB2 partitioned system is installed, you will need to verify that the installation is completed.

TCP/IP considerations

For TCP/IP connectivity, you do not need any additional software. This also applies for Named Pipes and NetBIOS.

Services file

During the instance creation a number of ports equal to the number of logical nodes that the instance is capable of supporting will be reserved in the following file:

x:\windows\system32\drivers\etc\services

Where x is the operating system installation drive. This path corresponds to a Windows 2003 Server.

The ports reserved in the services file will be used by the DB2 Fast Communication Manager. The reserved ports will be in the format:

```
DB2_InstanceName
DB2_InstanceName_1
DB2_InstanceName_2
DB2_InstanceName_END
```

The only mandatory entries are the beginning (DB2_InstanceName) and ending (DB2_InstanceName_END) ports. The other entries are reserved in the services file so that other applications do not use these ports.

When you install the instance-owning database partition server on the primary computer, DB2 reserves a port range according to the specified number of database partition servers per node. The default range is four ports. The DB2 Setup wizard will attempt to reserve an identical port range when database partition servers are installed on participating computers. If the port has been used, the Setup wizard will use the first available port range after 60000. However, if the port number in the participating servers do not match the ports in the Instance-owning server, you will receive communication errors when you start the DB2.

To verify the port range availability on participating computers:

- ▶ Open the services file located in the path specified above for Windows Server 2003 (or it should be located in %SystemRoot%\system32\drivers\etc).
- ▶ By default, DB2 reserves the first four available ports after 60000. Figure 4-21 is an example of how the entries appear in this file after DB2 is installed.
- ▶ On each participating computer, open the services file and verify that the ports reserved for DB2 FCM in the services file of the primary computer are not used.
- ▶ In the unlikely event that the required ports are in use on a participating computer, identify an available port range for all computers and update each service file, including the services file on the primary computer.

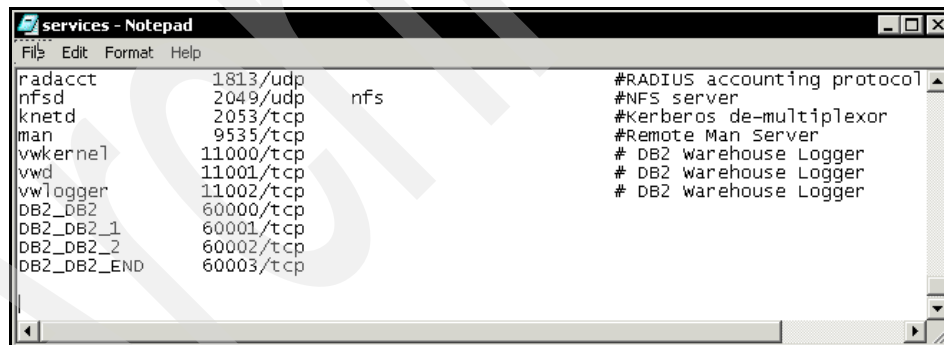
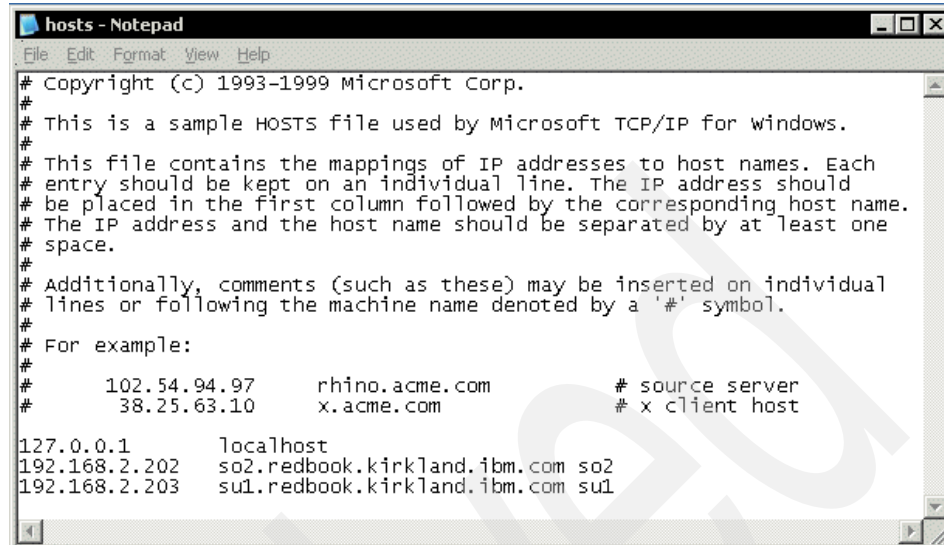


Figure 4-21 Default ports reserved by DB2 instance-owning installation

Hosts file

Another important thing to consider is the correct definition of the hosts file. The hosts file is located in the same directory as the services file. You must have an entry defining the IP address, server name, and domain name for each participating server in this file. This will ensure the correct configuration of the additional physical partitions at setup time. Figure 4-22 on page 208 shows a sample hosts file with two entries; one corresponds to the partition-owning server and the other to an additional physical partition.



```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com          # source server
#       38.25.63.10       x.acme.com              # x client host

127.0.0.1       localhost
192.168.2.202   so2.redbook.kirkland.ibm.com so2
192.168.2.203   su1.redbook.kirkland.ibm.com su1
```

Figure 4-22 Sample hosts file

From the entries shown in Figure 4-22, we could translate from Figure 4-2 on page 184, as that so2 could correspond to Server A and su1 could correspond to Server B. Each participating server must have a hosts file with entries of the members participating in the physical database partition configuration.

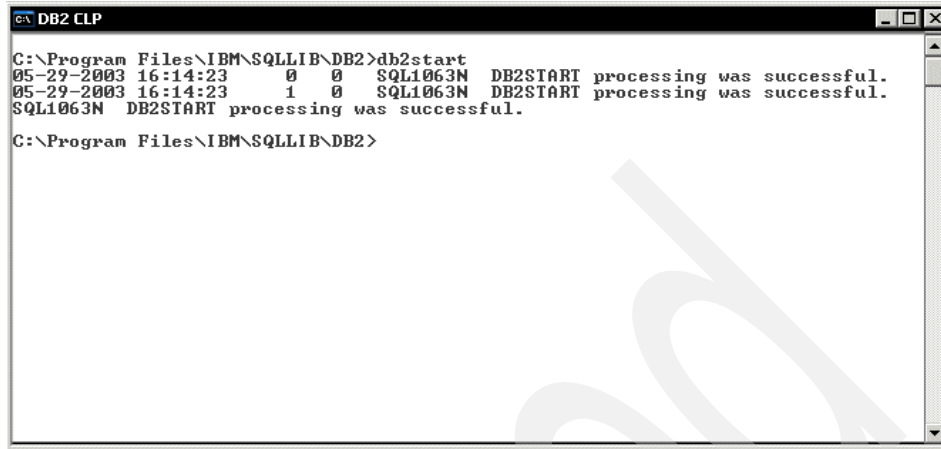
If we add more participating servers to increase the number of physical database partitions, we must manually insert the corresponding entry in the hosts file on each participating server.

Creating a sample database

A simple way to verify that your DB2 server installation was successful is by creating a database. You can create a sample database, run SQL commands to retrieve data from the database, and verify that the data has been distributed to all participating database partition servers.

To create the sample database:

1. Log onto the primary computer (Server A, Figure 4-2 on page 184) with the DB2 instance owner ID.
2. Open a DB2 Command Line Processor window and start the database manager by entering the **db2start** command. Figure 4-23 on page 209 shows a DB2 command window with the successful start of two partitions.



```
C:\Program Files\IBM\SQLLIB\DB2>db2start
05-29-2003 16:14:23      0      SQL1063N  DB2START processing was successful.
05-29-2003 16:14:23      1      SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
C:\Program Files\IBM\SQLLIB\DB2>
```

Figure 4-23 db2start output with two partitions

3. On the same DB2 Command Line Processor, type `db2samp1` to create the SAMPLE database. This command may take a few minutes to process. There is no completion message. When the command prompt returns, the process is complete.

4. Verify that the SAMPLE database was automatically cataloged with the database alias SAMPLE. Enter the following command to see the DB2 database directory:

```
db2 list database directory
```

5. Enter the following DB2 commands to connect to the SAMPLE database and retrieve a list of all the employees that work in department 20:

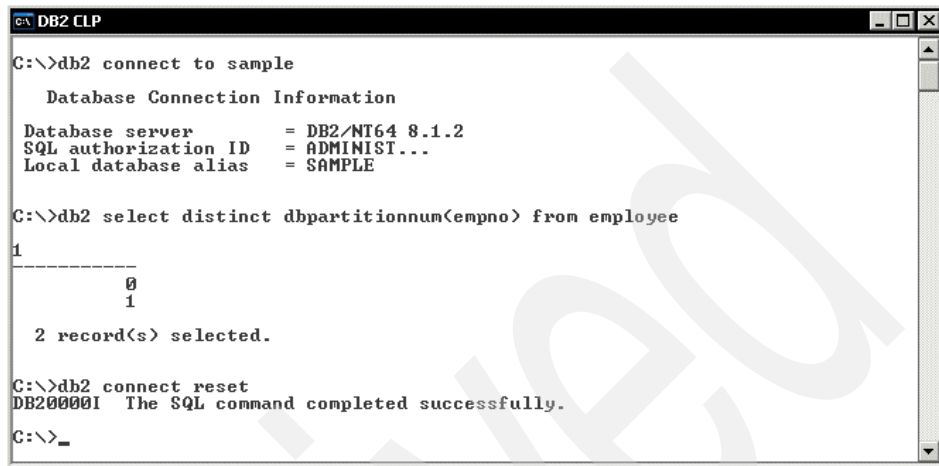
```
db2 connect to sample
db2 "select * from staff where dept = 20"
```

6. To verify that data has been distributed across database partition servers, enter the following commands from a DB2 command window:

```
db2 select distinct dbpartitionnum(empno) from employee
```

The output is shown in Figure 4-24 on page 210. You can see the database partitions used by the employee table. The DB2 function `db2partitionnum` returns the existing partition numbers for the given table in the SQL statement. Value 0 corresponds to data inserted in the partition database on the primary computer (Server A, Figure 4-2 on page 184), and value 1 corresponds to data inserted in the partition database on a participating computer (Server B, Figure 4-2 on page 184).

The specific output is the number of partitions in the database and the number of partitions in the partition group that is used by the tablespace where the employee table was created by default.



```
C:\>db2 connect to sample

Database Connection Information

Database server      = DB2/NT64 8.1.2
SQL authorization ID = ADMINIST...
Local database alias = SAMPLE

C:\>db2 select distinct dbpartitionnum(empno) from employee
1
-----
      0
      1

2 record(s) selected.

C:\>db2 connect reset
DB20000I The SQL command completed successfully.
C:\>_
```

Figure 4-24 Database partitions used by the employee table

Congratulations! You have a DB2 partitioned environment with two physical partitions; each physical partition has one logical database partition. If you want to remove the SAMPLE database to free up disk space, use the **db2 drop database sample** command to drop the database. In the following sections, we will continue using the SAMPLE database.

4.3.5 Creating additional partitions and redistributing data

To add additional partitions you can use the Add Partition Wizard. In the DB2 Control Center, right click the database instance name and select **Add Database Partition**. Follow the step-by-step instructions to define new logical partitions for already created instances. If you have already created a database, this tool allows you to redistribute the data between the old and new partitions defined across the cluster. The Add Partition Wizard performs the following tasks:

- ▶ It modifies the db2nodes.cfg file adding the new partition.
- ▶ It alters the affected database partition groups.
- ▶ It alters the appropriate tablespaces.
- ▶ You can run or schedule this task.
- ▶ Optionally you can redistribute the data.

Adding partitions

In this example, we are going to add a new logical database partition server on a participating server. Follow the next steps to create the new partition:

1. Open the DB2 Control Center, select the correct instance name, right click and select **Add Partitions**. On the Add Partitions Launchpad, click **Add Partitions** (see Figure 4-25).

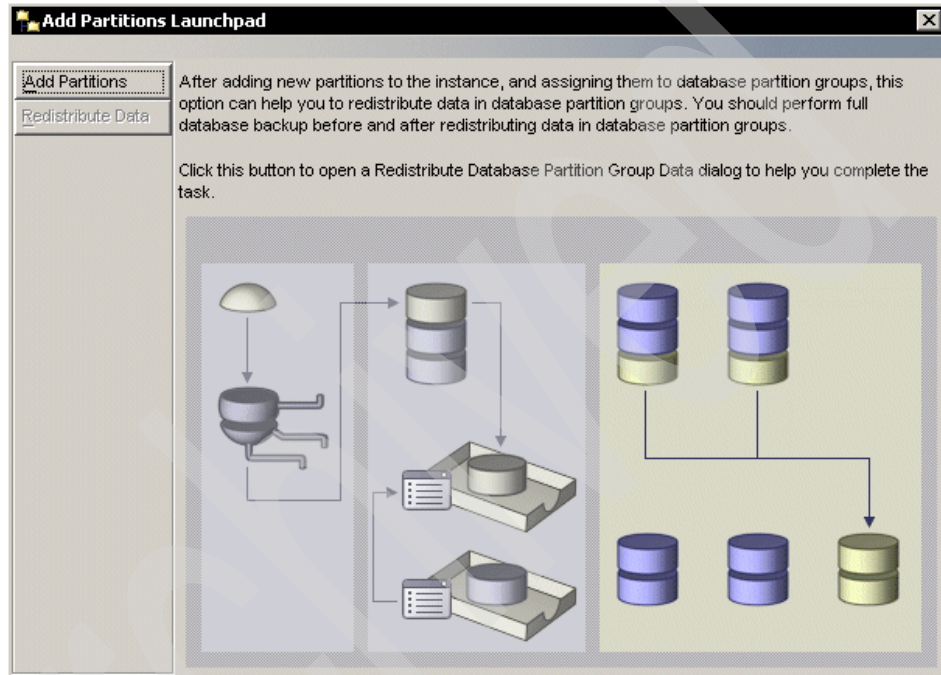


Figure 4-25 The Add Partitions Launchpad

2. Now you should see the Introduction screen on the Add Partitions Wizard. This screen shows the name of the affected instance. Click **Next** to continue.
3. On the “Add Partitions to Your Instance” screen (Figure 4-26 on page 212), make sure that **Show Existing Partitions** is selected and click **Add** to add the new partition. A new item will be shown with the partition number, host name, logical port, and switch name. Use the browse button (...) to list all of the host name in this partition environment. Select the server you want to add a partition to. On this screen you also must enter a user name, password, and domain name of the instance-owning user. Click **Next** to proceed.

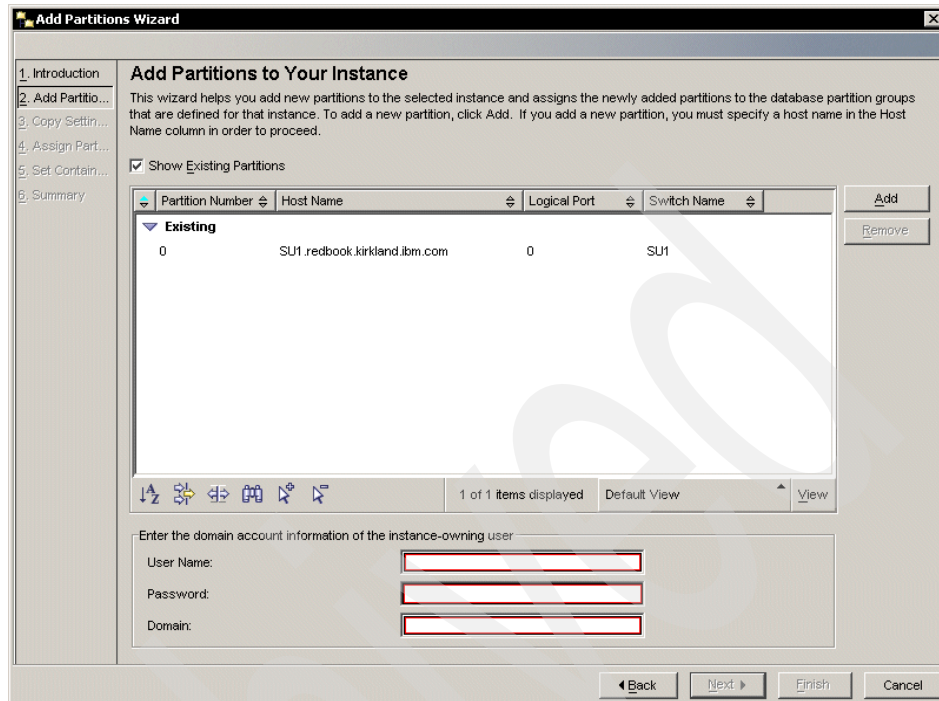



Figure 4-26 Add partitions to your instance

- On the “Copy setting from existing partitions to new partitions” screen, you can copy the tablespace container settings of existing partitions to the new partitions you are adding. Click the browse button () to list the container name from an existing partition. Figure 4-27 on page 213 shows the Copy Partition Settings window. If you want to modify the copied settings or create customized settings, click the **Next** button.

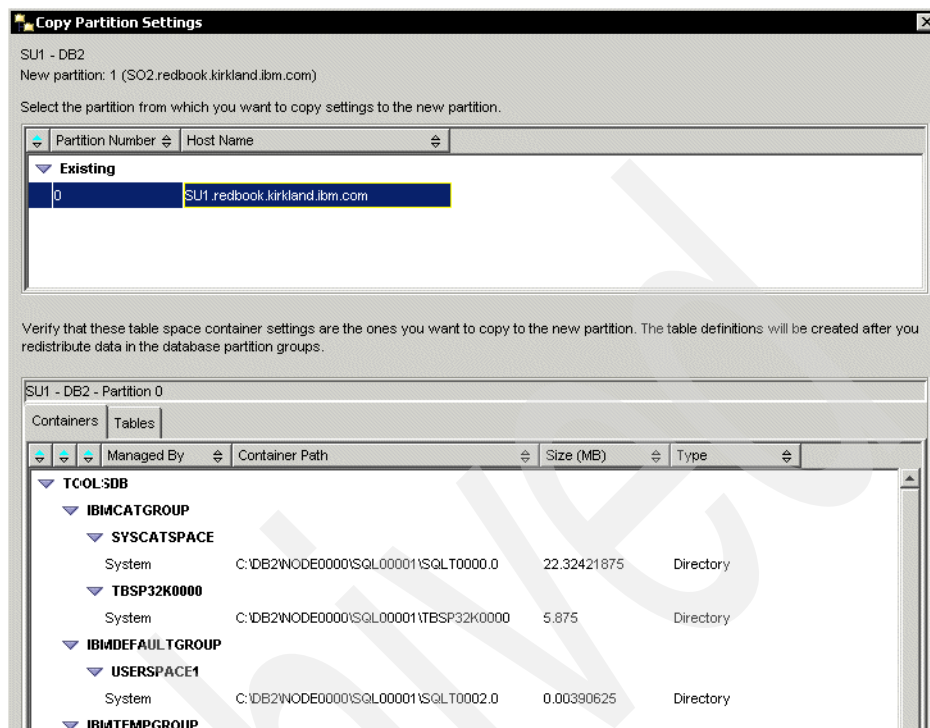


Figure 4-27 Copy Partition Settings

5. On the *Assign partitions to database partition groups* screen, you can see the defined partition groups. Select the checkboxes to specify the new partitions you want to add to each database partition group. Note that the check boxes for IBMCATGROUP and IBMTEMPGROUP are read-only. You can only set this for the IBMDEFAULTGROUP or other user-created database partition groups. Click **Next** to continue.
6. On the Set Containers screen (Figure 4-28 on page 214), review the container names. You can add or change the container definition for the new partition in the specified database. Remember to use unique container names for each partition. If you have a large number of container names use the Find field and the Replace All button.

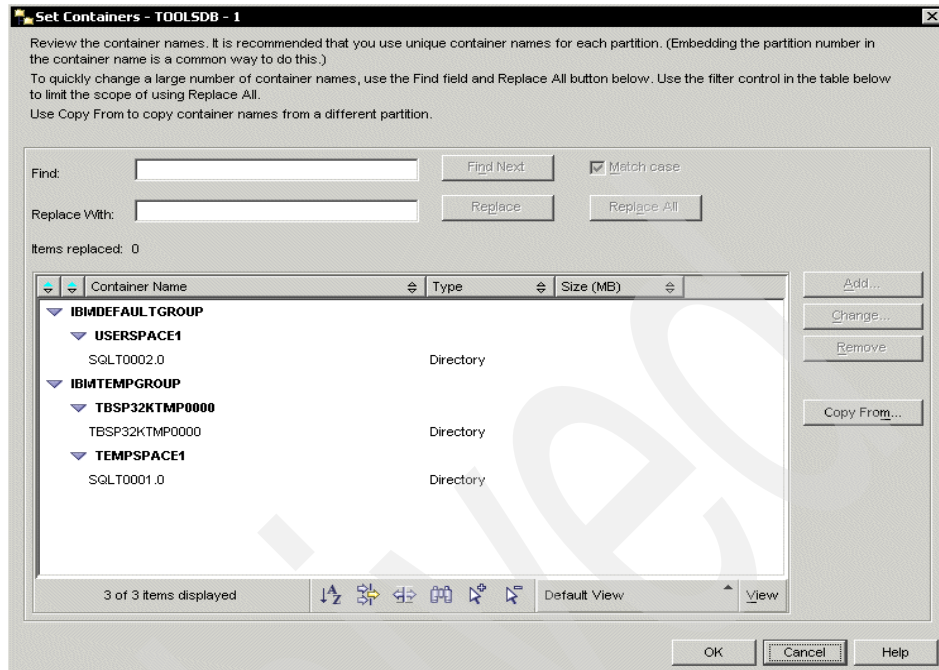


Figure 4-28 Set containers

7. On the Summary screen, you will see the number of new partitions and how many database partition groups are being altered. When you click the **Finish** button the following processes will run:

- Adds the new partitions to the instance
- Stops the instance
- Restarts the instance
- Assigns the newly added partitions to the selected database partition groups
- Sets container definitions on the new database partitions

The Show Command option lets you review the commands used for the tasks listed.

Redistribute data

After you add a new partition with the Add Partition Wizard on an instance with populated databases, it is necessary run a process called data redistribution. This process can help you to distribute data in database partition groups.

Note: A full database backup should be performed before and after redistributing data in database partitions groups.

To redistribute data using the wizard, click the **Redistribute Data** button on the Add Partitions Launchpad. See Figure 4-25 on page 211. Select a database partition group then click the **Redistribute** button. There are eight steps:

1. *Introduction:* Shows the name of the selected database partition group. Click **Next** to continue.
2. *Method:* For selecting the Redistribution Method. You may select to redistribute your data Uniformly or Fine tune the distribution of the database partition group. Select the first option if you have an evenly distributed database partition group and you want to redistribute the data on to newly added database partitions or off of database partitions to be dropped. Select the second option if you want to set up a weighted data distribution, or to distribute your data evenly across all of the database partitions in the database partition group. For this example, select **Uniformly** and **On to newly added database partitions**. Click the **Next** button.
3. *Strategy:* For selecting the Redistribution Strategy. You may select between a manual redistribution plan or let the wizard decide the strategy. For the first option you will reduce the time required for analysis. If you create the plan then keep in mind that you need to know the size and data skew of the largest table in your database partition group, the log size of each database partition, and the log overhead. For the second option you can use the storage management record if at least one storage snapshot for this database partition group has been taken recently or you can have a real time system analysis. For this exercise we use this option since we can afford the time required for the wizard to complete this task due to a small sample database. Click on the **Next** button.
4. *Analyze:* The System analysis screen shows the results of the analysis. In this example the redistribution process will run on three partitions, with 1 MB of data to be moved between the three partitions. See Figure 4-29 on page 216. Click the **Next** button.

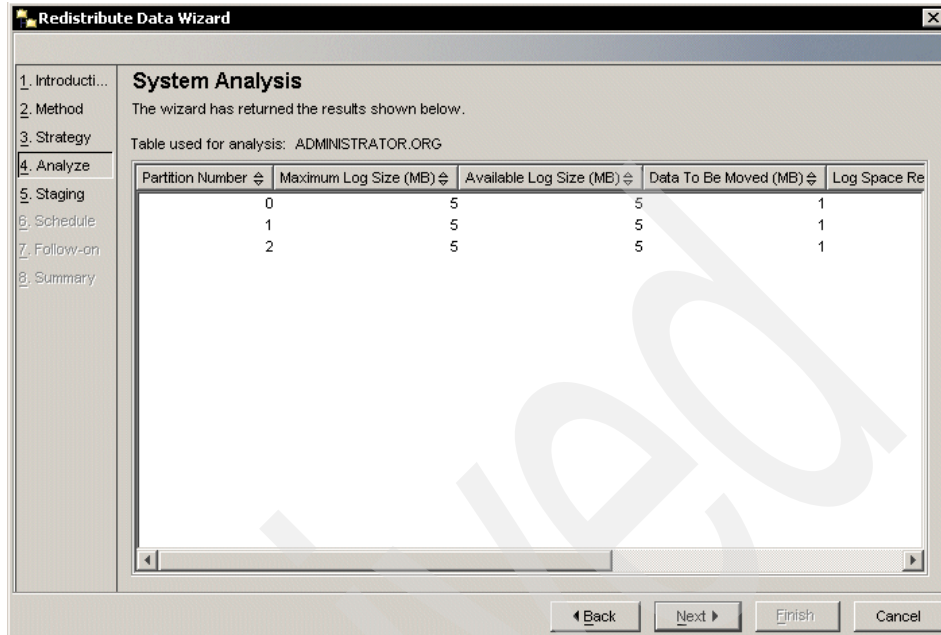


Figure 4-29 System analysis results

5. *Staging*: The Staged Redistribution Plan (Figure 4-30 on page 217) specifies how many steps you want the staged redistribution to take. Keep in mind that the data is committed after each step with the number of rows moved at a time. Depending on the number of rows you specified, DB2 creates the steps needed to redistribute the data among the database partition group. For this example, we use the DB2 recommended values. Click the **Next** button.

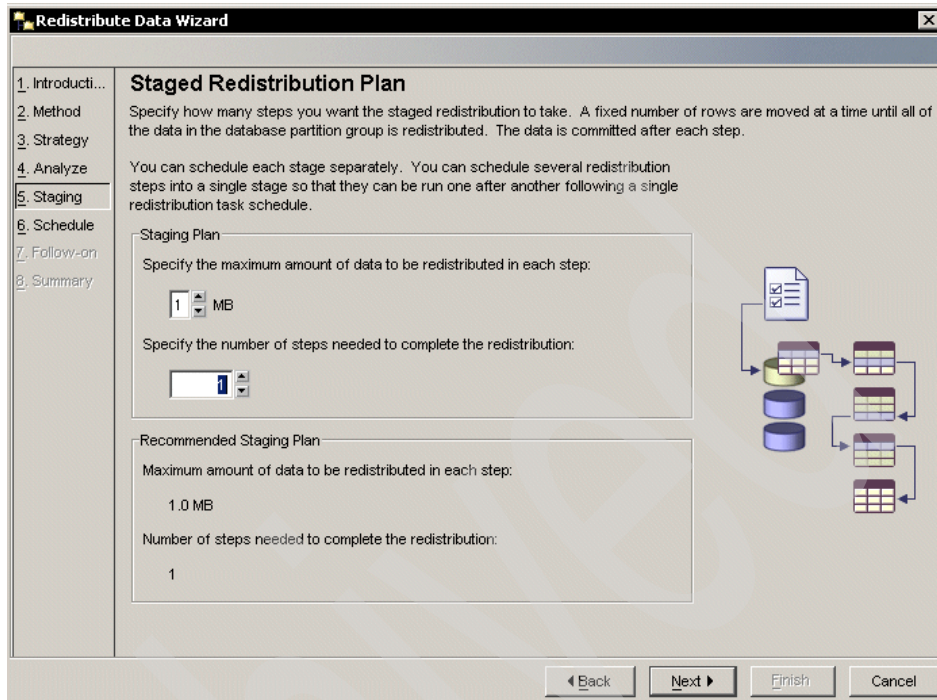


Figure 4-30 Staged Redistribution Plan window

6. *Schedule*: In the Scheduling task execution step, you choose **Run now without saving the task history or Create this as a task in the Task Center**. Select **Run now without saving the task history** and click the **Next** button.
7. *Follow-on*: The screen Subsequent Action lets you run other highly recommended steps after data redistribution. You can execute a run statistics process and/or rebind packages. For this example select both options and click the **Next** button.
8. *Summary*: The final step is the Summary window; review the process. This is your last chance to change the configuration for this data redistribution process. Click the **Finish** button. See Figure 4-31 on page 218.

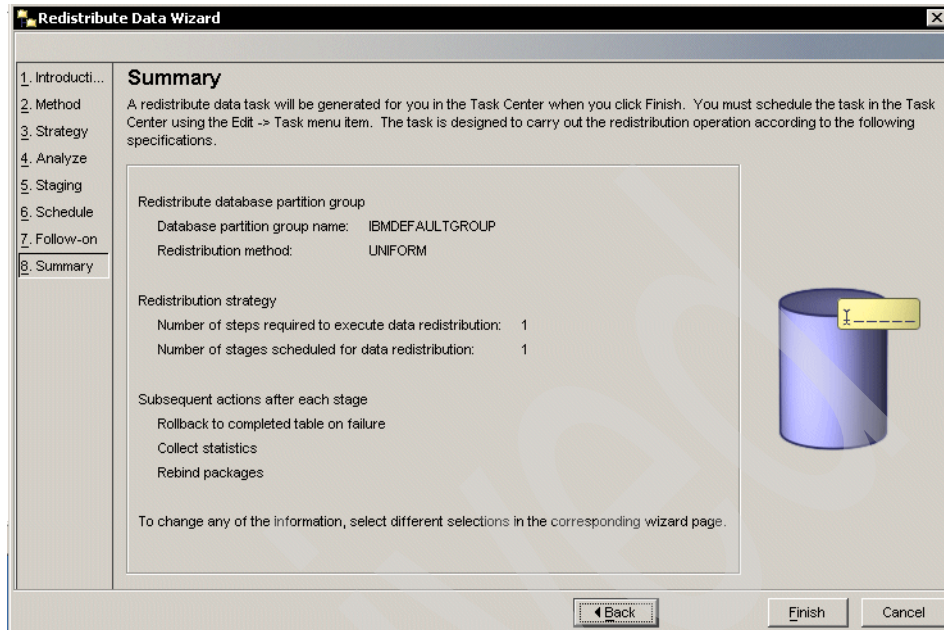


Figure 4-31 Redistribute Data Wizard Summary window

The Data Redistribution utility performs the following steps:

1. Obtains a new partitioning map ID for the target partitioning map, and inserts it into the syscat.partitionmaps catalog view.
2. Updates the rebalance_pmap_id column in the syscat.dbpartitiongroups catalog view for the database partitioning group with the new partitioning map ID.
3. Adds any new database partitions to the syscat.dbpartitiongroupdef catalog view.
4. Sets the in_use column in the syscat.dbpartitiongroupdef catalog view to "D" for any database partition that is to be dropped.
5. Does a commit for the catalog updates.
6. Creates database files for all new database partitions.
7. Redistributes the data on a table-by-table basis for every table in the database partitioning group, in the following steps:
 - a. Locks the row for the table in the systables catalog table.
 - b. Invalidates all packages that involve this table. The partitioning map ID associated with the table changes because the table rows are redistributed. Because the packages are invalidated, the compiler must

- obtain the new partitioning information for the table and generate packages accordingly.
- c. Locks the table in exclusive mode.
 - d. Uses DELETES and INSERTs to redistribute the data in the table.
 - e. If the redistribution operation succeeds, it issues a COMMIT for the table and continues with the next table in the database partitioning group. If the operation fails before the table is fully redistributed, the utility issues a ROLLBACK on updates to the table, ends the entire redistribution operation, and returns an error.
8. Deletes database files and deletes entries in the syscat.nodegroupdef catalog view for database partitions that were previously marked to be dropped.
 9. Updates the database partitioning group record in the syscat.nodegroups catalog view to set pmap_id to the value of rebalance_pmap_id and rebalance_pmap_id to null.
 10. Deletes the old partitioning map from the syscat.partitionmaps catalog view.

You can find out whether data is distributed unequally among partitions with a SQL command (see Example 4-2). With current distribution information, you can use this information to create a custom redistribution file or partitioning map later on.

Determine if any database partitions have unequal distributions of rows. For the largest table, use an appropriate partitioning column and enter a query such as in Example 4-2.

Example 4-2 Looking for data distribution information

```
select partition(column_name), count(*) from table_name
group by partition(column_name)
order by partition(column_name) desc
fetch first 100 rows only
```

The partition and dbpartitionnum SQL functions determine the current data distribution across hash partitions or database partitions. The partition function returns the partitioning map index for each row of the table. The dbpartitionnum function returns the partition number of the row.

Execute this query for other large tables that are partitioned across the database partition group.

Use the information to create both a distribution file and a target partitioning map.

Note: The log must be large enough to accommodate the INSERT and DELETE operations at each database partition where data is being redistributed. The heaviest logging requirements will be either on the database partition that will lose the most data, or on the database partition that will gain the most data.

4.3.6 Creating a partitioned environment with DB2 commands

This section will guide you on how to set up DB2 Physical Database Partition Servers using DB2 commands. We are assuming that DB2 has correctly installed and there are no instances.

The scenario for this configuration is using a two server cluster, each with Windows Server 2003 Datacenter Edition and DB2 UDB ESE 8.1 64-bit.

The following information is important to keep in mind if we are starting a partitioned database configuration. In this example only two physical database partitions are created. Server A is used as the instance-owning server and Server B is the additional participating server. A diagram of this configuration is shown in Figure 4-2 on page 184. Table 4-1 lists the values needed for our database partitioned environment. This table also provides the information that you need to prepare ahead of time if you want to create a partitioned environment using DB2 commands.

Table 4-1 Server information

Server	Parameters	Current Value
Server A	Host name IP address Domain Administrator domain user Ports reserved Range of ports Instance-owning server	SO2 192.168.2.202 REDBOOK1 db2admin 6 60000 to 60005 Yes
Server B	Host name IP address Domain Administrator domain user Ports reserved Range of ports Instance-owning server	SO1 192.168.2.201 REDBOOK1 db2admin 6 60000 to 60005 No

Follow these steps to set up the partitioned environment:

1. Log on to Server A as an administrator user, open a DB2 Command Line Processor window and execute the command *db2icrt*. See Figure 4-32 on page 222.

```
db2icrt -s ESE -u db2admin,password -h S02 -r 60000,60005 DB2
```

Where:

- -s is the type of DB2 server.
- -u is an administrator user/domain administrator user.
- -h is the host name of the server.
- -r is the port range; in this example we reserve only 6 ports.
- DB2 is the name of the instance.

The following environment variables are important (these variables must exist in each physical partition server). You can verify them or set them with the **db2set** command and the **set** windows operating system command:

```
db2set
DB2ACCOUNTNAME=REDBOOK1\db2admin
DB2INSTOWNER=S02
DB2PORTRANGE=60000:60005
DB2INSTPROF=\\SU1\DB2-DB2
DB2COMM=TCPIP
```

```
set
USERDNSDOMAIN=REDBOOK1.KIRKLAND.IBM.COM
USERDOMAIN=REDBOOK1
USERNAME=db2admin
USERPROFILE=c:\documents and settings\db2admin
```

Note: The db2set and Windows environment variables used by DB2 will have to be set on each physical partition that is part of the DB2 partitioned database. The commands will not automatically replicate the values out to all the physical machines.

2. On the same DB2 Command Line Processor window or another one, execute the command **db2ilist** to see if the new instance has been added to the system. See Figure 4-32 on page 222.

```
c:\db2ilist
DB2
```

3. Now we can see if the instance was properly created in our system by executing a **db2start** command. You should see a message like the following:

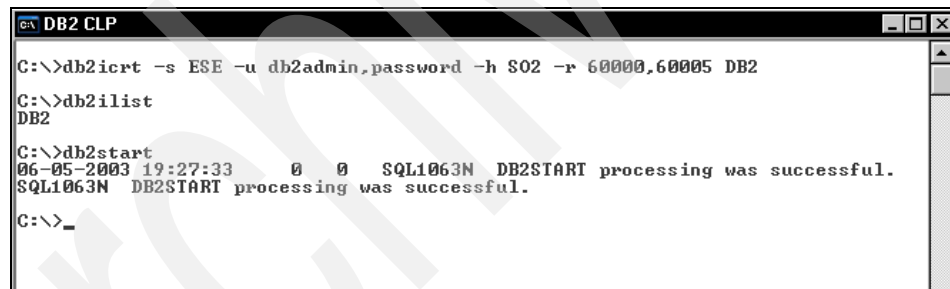
```
c:\db2start
05-29-2003 16:14:23 0 0 SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.
```

This means the instance-owning server is up and running. The two zeros between time and SQL code represent that this is the first partition (0) running on the reserved port 0. See Figure 4-32.

4. Verify the TCP/IP ports reserved for DB2 in the services file. The services file is located in x:\windows\system32\drivers\etc, where x is the operating system drive. The services should look like this:

```
DB2_DB2      60000/tcp
DB2_DB2_1    60001/tcp
DB2_DB2_2    60002/tcp
DB2_DB2_3    60003/tcp
DB2_DB2_4    60004/tcp
DB2_DB2_END  60005/tcp
```

With this range of partitions we are able to have six partitions. If you want to have more than six partitions then you must add more ports for DB2 in the services file. For this exercise we only use two ports.



```
DB2 CLP
C:\>db2icrt -s ESE -u db2admin,password -h S02 -r 60000,60005 DB2
C:\>db2ilist
DB2
C:\>db2start
06-05-2003 19:27:33      0 0  SQL1063N DB2START processing was successful.
SQL1063N DB2START processing was successful.
C:\>_
```

Figure 4-32 db2icrt command

5. At this time we should have a db2nodes.cfg file. This file contains information about partitions and servers. This file is created when you execute the db2icrt command. When you add additional partitions, they are reflected in this file. This file is located in x:\<db2_install_path>\sqllib\<inst_name>\. Do not edit this file manually. A blank space, for example, could cause undesirable behavior. You can execute a **type** command to see its content:

```
c:\type "\\Program Files\IBM\SQLLIB\DB2\db2nodes.cfg"
0 S02 S02 0
```

6. It is time to create the physical database partition on Server B. From a DB2 Command Line Processor window in Server A, execute the command **db2ncrt**. See Figure 4-33 on page 224.

```
db2ncrt /n:1 /u:REDB00K1\db2admin,password /i:DB2 /m:S01 /p:0 /h:S01 /o:S02
```

Where:

- /n is the partition number for this server (Server B).
- /u is a db2 administrator user, under a domain administrators group. The user name must be qualified with the domain name.
- /i is the instance name.
- /m is the machine name of the server where the partition is being added.
- /p is the logical port used for the database partition server. If it is the first partition on this server then it must start with port number 0. The port number cannot exceed the port range reserved for FCM communications in the x:\windows\system32\drivers\etc\ services. For example, in our case a range of 6 ports is reserved for the current instance, then the maximum port number is 5. Port 0 is used for the default logical database partition server.
- /h is the host name used by FCM communications. It must be specified if the partition is being added on a remote server.
- /o is the instance-owning server name. If you are executing this command on a server different than the Instance Owning Node, then you must specify this option. In this example it is optional.

Please note that if a database partition server is added to a computer where an instance already exists, a database partition is added as a logical database partition server to the computer. If a database partition is added to a computer where an instance does not exist, the instance is added and the computer becomes a new physical database partition server. The **db2ncrt** command should not be used if there are databases in an instance. Instead, the **db2start addnode nodenum** command should be used. This ensures that the database is correctly added to the new database partition server. It is also possible to add a database partition server to an instance in which a database has been created.

```

C:\>db2ncrt /n:1 /u:REDBOOK1\db2admin,password /i:DB2 /m:S01 /p:0 /h:S01 /o:S02

DBI1937W The db2ncrt command successfully added the node. The
         node is not active until all nodes are stopped and
         started again.

Explanation:

The db2nodes.cfg file is not updated to include the new node
until all nodes are simultaneously stopped by the STOP DATABASE
MANAGER (db2stop) command. Until the file is updated, the
existing nodes cannot communicate with the new node.

User Response:

Issue db2stop to stop all the nodes. When all nodes are
successfully stopped, issue db2start to start all the nodes,
including the new node.

C:\>

```

Figure 4-33 db2ncrt command

7. After the **db2ncrt** command is issued you must recycle the DB2. Use the **db2stop** command to stop DB2. When you have issued the **db2stop** command the system only stops the current configured instance. In this point the **db2nodes.cfg** file is updated with information for DB2 to communicate with the new node. When you start DB2 again you will see a message with two nodes starting. See Figure 4-34.
8. Now you have a database partitioned environment with two physical nodes. Verify installation and setup with the procedure in “Verifying DB2 partitioning configuration” on page 206.

```

C:\>db2stop
06-05-2003 20:35:57      0      0      SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.

C:\>db2start
06-05-2003 20:38:18      0      0      SQL1063N  DB2START processing was successful.
06-05-2003 20:38:20      1      0      SQL1064N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.

C:\>

```

Figure 4-34 db2stop and db2start after adding a physical partition

9. After this point you can start creating partitioned databases. If after creating a database you need to add additional logical database partitions, for example, on server A or server B on this sample configuration, you can use the DB2 command:

```
add dbpartitionnum
```

This command creates a database partition for all databases on the new database partition server. The user can specify the source database partition

server for the definitions of any system temporary tablespaces to be created with the new database partition, or specify that no system temporary tablespaces are to be created.

```
add dbpartitionnum like dbpartition db-partition-number
add dbpartitionnum without tablespaces
```

The first command copies the definitions of the system temporary tablespaces from another database partition. The second command does not create any temporary tablespace definition.

Note: The **add dbpartitionnum** command must be issued from the database partition server that is being added with **sysadm** or **sysctrl** authority.

10. Optionally you can create additional partitions with the **start database manager** command or the **db2start** command. The following options are needed to add and configure an additional logical partition with this command:

```
start database manager add dbpartitionnum hostname S01 port 1 computer S01
user REDB00K1/db2admin password password netname S01
```

Where:

- **hostname** is the host name used by FCM communications. This is added to the **db2nodes.cfg** file.
- **port** is the logical port used for the database partition server. If it is the first partition on this server then it must start with port number zero. The port number cannot exceed the port range reserved for FCM communications in the **x:\windows\system32\drivers\etc** services. Remember that port zero is used for the default logical database partition server.
- **computer** is the computer name of the server where the partition is being added.
- **user** is a DB2 administrator user. This user must belong to the domain administrators group. The user name must be qualified with the domain name.
- **password** is the user's password.
- **netname** is the name used for the **db2nodes.cfg** file. If it is not specified, the value from **hostname** is used.

The options from the **add dbpartitionnum** command are supported.

Note: You can add a partition to a running database system, and while applications are connected to databases. However, a newly added physical partition server does not become available to all databases until the database manager is shut down and restarted.

The procedure to add a new partition while the system is running is the following:

- ▶ On any existing database partition, run the **db2start** command or the **start database manager** command with similar options as the last example.
- ▶ When the **db2start** command is complete, the new server is stopped.
- ▶ Stop the database manager on all partitions by running the **db2stop** command. Remember, the node configuration file is not updated with the new server information until **db2stop** is executed.
- ▶ Start the database manager by running the **db2start** command.
- ▶ The newly added database partition is now started along with the rest of the system.

After adding a new partition, you should redistribute data across the database partitions in a database partition group. The DB2 command for data redistribution uses an algorithm that selects the partitions to be moved based on the current data distribution. This command affects all database partitions in the database partition group, and you need sysadm, sysctrl, or dbadm authorization.

```
redistribute database partition group database-partition-group
```

You may use the following options for the command:

```
uniform  
using distfile distfile  
using targetmap targetmap  
continue  
rollback
```

Where:

- ▶ database-partition-group is the name of the database partition group. You can find this name in the syscat.dbpartitiongroups catalog table.

Note: Tables in the IBMCATGROUP and the IBMTEMPGROUP database partition groups cannot be redistributed.

- ▶ uniform specifies that the data is uniformly distributed across hash partitions (that is, every hash partition is assumed to have the same number of rows), but the same number of hash partitions do not map to each database

partition. After redistribution, all database partitions in the database partition group have approximately the same number of hash partitions.

- ▶ using distfile. If the distribution of partitioning key values is skewed, use this option to achieve a uniform redistribution of data across the database partitions of a database partition group.

Use the distfile option to indicate the current distribution of data across the 4 096 hash partitions.

Use row counts, byte volumes, or any other measure to indicate the amount of data represented by each hash partition. The utility reads the integer value associated with a partition as the weight of that partition. When a distfile is specified, the utility generates a target partitioning map that it uses to redistribute the data across the database partitions in the database partition group as uniformly as database partitions in the database partition group as uniformly as possible. After the redistribution, the weight of each database partition in the database partition group is approximately the same. (The weight of a database partition is the sum of the weights of all partitions that map to that database partition.)

For example, the input distribution file may contain entries as follows:

```
10223
1345
112000
0
100
...
```

In the example, hash partition 2 has a weight of 112 000, and partition 3 (with a weight of 0) has no data mapping to it at all. The distfile should contain 4 096 positive integer values in character format. The sum of the values should be less than or equal to 4 294 967 295.

- ▶ continue. Continues a previously failed redistribute database partition group operation. If none occurred, an error is returned.
- ▶ rollback. Rolls back a previously failed redistribute database partition group operation. If none occurred, an error is returned.

Note: This command can only be issued from the catalog database partition. Use the LIST DATABASE DIRECTORY command to determine which database partition is the catalog database partition for each database.

When a redistribution operation is done, a message file is written to the \sqlib\redist\ directory on Windows operating systems, using the following format for subdirectories and file name:

database-name\first-eightcharacters-of-the-database-partition-group-name\date\time

The redistribute message file contains a list of all the tables that have undergone redistribution.

Note: It is also recommended to update statistics by issuing RUNSTATS after the redistribute database partition group operation has completed.

Database partition groups containing replicated summary tables or tables defined with *data capture changes* cannot be redistributed. Redistribution is not allowed if there are user temporary tablespaces with existing declared temporary tables in the database partition group.

4.3.7 Executing commands on all partitions

It is important to remember that all database configuration commands and utilities, such as backup, run locally only on the database partition you are attached to. You have to explicitly run the commands on each partition in your database if you want it to affect all the nodes. The commands are only issued against an individual node in order to give database administrators greater flexibility in how they administer and tune their database system.

If, however, you would like to execute a command once against all the partitions in your database, then DB2 has a command to allow you to do so. The **db2_a11** command or **rah** will execute a specified command against all nodes asynchronously. The syntax for both commands is quite simple. Multiple commands in the command string are separated by the "&" symbol.

db2_a11 command
rah command

If a semicolon (;) is placed at the beginning of the list of commands to be executed, then they will be run in parallel on all nodes. If the character is not given, then the commands will be executed serially from the first to the last partition. The character “ ” cannot be used in the command string. If you need to include this character in your command string then you can just execute the **db2_a11** command without any command string. You will then be prompted for the command, and the quote character can be inputted without any errors occurring. A complete description of the syntax can be found in the DB2 documentation *Administration Guide: Implementation*.

4.4 Creating partitioned databases

After you have completed your logical database design, there are a number of considerations about the physical environment in which your database and tables will reside. These include understanding the files that will be created to support and manage your database, understanding how much space will be required to store your data, and determining how you should use the tablespaces that are required to store your data.

4.4.1 Basic database structure

When you create a database, a subdirectory with the name of the instance is created. You can specify where you would like the database to be created. If you do not specify the location of the directory path or drive when you create the database, the default location is used. Below this subdirectory, a subdirectory named NODE0000 is created. This subdirectory differentiates partitions in a logically partitioned database environment. Below the directory NODE0000, a subdirectory named SQL00001 is created. This represents the database being created.

The database directory contains the following files:

- ▶ SQLBP.1 and SQLBP.2 contain buffer pool information. The file number 2 is a backup.
- ▶ SQLSPCS.1 and SQLSPCS.2 contain tablespace information. The file number 2 is a backup.
- ▶ SQLDBCON contains database configuration information. To change configuration parameters use either the Control Center or command line statements like *update database manager configuration* and *reset database manager configuration*.
- ▶ DB2RHIST.ASC and DB2RHIST.BAK contain history information about backups, restores, loading of tables, reorganization of tables, altering of a tablespace, and other changes to a database.
- ▶ DB2TSCHNG.HIS contains a history of tablespace changes at a log-file level. Tablespace recovery uses information from this file to determine which log files to process during tablespace recovery.
- ▶ SQLOGCTL.LFH and SQLOGMIR.LFH contain information about active logs. Recovery processing uses information from these files to determine from how far back in the logs for the database should the recovery be started. The SQLOGDIR subdirectory contains the actual log files.
- ▶ SQLINSLK helps to ensure that a database is used by only one instance of the database manager.

- ▶ SQLT* subdirectories contain the default System Managed Space (SMS) tablespaces required for an operational database. Three default tablespaces are created:
 - SQLT0000.0 is the catalog tablespace with the system catalog tables.
 - SQLT0001.0 is the default temporary tablespace.
 - SQLT0002.0 is the default user data tablespace.

A file, SQLTAG.NAM, in each container marks the subdirectory as being in use so that subsequent tablespace creation does not attempt to use these subdirectories.

A file, SQL*.DAT, stores information about each table that the container contains. The asterisk is a set of digits that identifies each table. For each .DAT file might be the following tables (one or more, depending on the table):

- ▶ SQL*.BMP, blocks allocation information if it is a MDC table
- ▶ SQL*.LF, LONG VARCHAR or LONG VARGRAPHIC data
- ▶ SQL*.LB, BLOB, CLOB, or DBCLOB data
- ▶ SQL*.LBA, allocation and free space information about SQL*.LB
- ▶ SQL*.INX, index table data
- ▶ SQL*.DTR, temporary data reorganization of an SQL*.DAT file
- ▶ SQL*.LFR, temporary data reorganization of an SQL*.LF file
- ▶ SQL*.RLB, temporary data reorganization of an SQL*.LB file
- ▶ SQL*.RBA, temporary data reorganization of an SQL*.LBA file

4.4.2 Database partition groups

Before you start creating tables in a partitioned database environment, you must review how many partition groups are defined. These partition groups will store the tablespace definitions of your database and under each table definition there is a defined number of containers. A database partition group is a subset of one or more partitions in a partitioned database environment. If you have a subset with two or more database partitions then you may call your database a multipartition database partition group.

Note: Multipartition database partition groups can only be defined with database partitions that belong to the same instance.

You can create a new database partition group using the following command:

```
CREATE DATABASE PARTITION GROUP
```

You can modify it using:

```
ALTER DATABASE PARTITION GROUP
```

In database partition groups data is divided across all the partitions. Depending on your needs, you may add or drop one or more database partitions from a database partition group.

A database partition group can contain at least one database partition, or as much as the entire set of database partitions. See Figure 4-35.

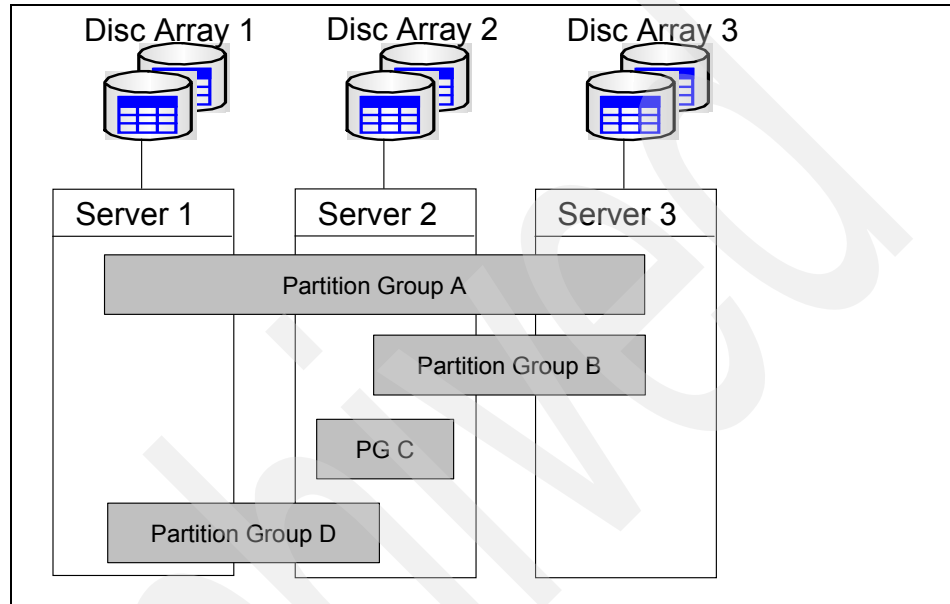


Figure 4-35 Database partitions groups

Consider these points when designing a database partitioned environment:

- ▶ In a multipartition database partition group, you can only create a unique index if it is a superset of the partitioning key.
- ▶ Depending on the number of database partitions in the database, you may have one or more single-partition database partitions groups, and one or more multipartitions database partition groups present. See Figure 4-35.
- ▶ Each database partition must be assigned a unique partition number. The same database partition may be found in one or more database partition groups.
- ▶ To ensure fast recovery of the database partition containing system catalog tables, avoid placing user tables on the same database partition. This is accomplished by placing user tables in database partition groups that do not include the database partition in the IBMCATGROUP database partition group.

Note: Do not extend medium-sized tables across too many database partitions. There is no formula to calculate how many partitions perform better for a specific table size. But keep in mind that a 100 MB table may perform better on a 16-partition database partition group than on a 32-partition database partition group.

Collocation is a strategy where you can place rows on different tables that contain related data in the same database partition. This is transformed in better join strategies. Consider the following for a collocated table:

- ▶ The tables reside in a multipartition database partition group.
- ▶ The tables have the same number of columns in the partitioning key.
- ▶ The columns in the partitioning key are partition compatible.

Tip: You can use database partition groups to separate online transaction processing (OLTP) tables from decision support (DSS) tables, to ensure that the performance of OLTP transactions is not adversely affected.

4.4.3 Partitioning maps and partitioning keys

A *partitioning map* is an internally generated array. This array is used by the database manager to know where to find the stored data. This array can contain up to 4096 entries. DB2 uses a round-robin algorithm to specify the partition numbers for multiple-partition database partition groups. There is only one entry in the array for a database with a single partition because all the rows of a database reside in one partition database partition group.

The database manager uses a *partitioning key* to accurately determine the location where the data is stored.

In the next sequence of numbers you can see the partitioning map for a database partition group with 6 partitions:

0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 ...

In another case, if the database partition group is defined only with 2 partitions (1 and 2) then the partitioning map would be:

1 2 1 2 1 2 1 2 1 2 1 ...

When you are ready to store data into a partitioned table and the partitioned key is an integer with possible values between 0 and 100 000, the partitioning key is *hashed* to a partition number between 0 and 4095. Any number between 0 and 4095 is used as an index into the partitioning map to select the correct database partition for that row.

Note: The redistribution utility allows you to rebalance or introduce skew into the data distribution. This is an easy way to change data distribution.

A partitioning key is a column (or group of columns) that is used to determine the partition in which a particular row of data is stored. The **create table** command allows you to define a partitioning key. If a partitioning key is not defined, one of the following may occur:

- ▶ The partition key is created from the first column of the primary key.
- ▶ If no primary key is specified, the partition key is the first non-long field column.

If you want a partition key of a table in a tablespace defined over a single-partition database partition group then you must define the partitioning key explicitly. Tables without a partitioning key are only allowed in single-partition database partition groups.

You can add or drop partitioning keys at a later time, using the **alter table** statement. Altering the partition key can only be done to a table whose tablespace is associated with a single-partition database partition group.

Choosing a good partitioning key is important. You should take into consideration:

- ▶ How tables are to be accessed
- ▶ The nature of the query workload
- ▶ The join strategies employed by the database system

Tip: A good partitioning key for a table is one that spreads the data evenly across all database partitions in the database partition group.

Consider the following when working with partitioned tables:

- ▶ An inappropriate partitioning key can cause uneven data distribution.
- ▶ The number of distinct values must be large enough to ensure an even distribution of rows across all database partitions in the database partition group.
- ▶ The partitioning key cannot be more than 16 columns. Fewer columns result in better performance.
- ▶ A partitioned table that contains only long data types (LONG, VARCHAR, LONG VARGRAPHIC, BLOB, CLOB, or DBCLOB) is not supported.
- ▶ The partitioning key definition cannot be altered.
- ▶ The partitioning key should include the most frequently joined columns.

For more information about concepts like table collocation and partition compatibility see the DB2 UDB V8 *Administration Guide*.

4.4.4 Partition groups and tablespaces

In a partitioned database environment, each tablespace is associated with a specific database partition group. This allows the characteristics of the tablespace to be applied to each partition in the database partition group. The database partition group must exist, and the association between the tablespace and the database partition group is defined when the tablespace is created using the **create tablespace** statement.

You cannot change the association between tablespace and database partition group using the **alter tablespace** statement. You can only change the tablespace specification for individual partitions within the database partition group. In a single-partition environment, each tablespace is associated with the default database partition group. The default database partition group, when defining a tablespace, is IBMDEFAULTGROUP, unless a system temporary tablespace has been defined, in which case IBMTEMPGROUP is used.

4.4.5 Create database example

In this example we use the SAMPLE database which is created across two physical database partitions. This database was created with default values and the first thing we are going to create is two new database partitions groups.

Example 4-3 Create database partitions

```
connect to sample
```

```
create database partition group "ONEPARTGROUP" on dbpartitionnums (0)
create database partition group "TWOPARTGROUP" on dbpartitionnums (0,1)
```

```
connect reset
```

In Example 4-3, the first **create database partition group** command creates a partition called "ONEPARTGROUP" on the physical database partition 0.

The second **create database partition group** command creates a partition called "TWOPARTGROUP" on the physical database partitions 0 and 1.

The next step is to create the buffer pools for the tablespaces that we will create later. In general, one bufferpool for one database partition group has better performance. We create two bufferpools in our example (Example 4-4 on page 235).

Example 4-4 Create buffer pools

```
connect to sample
```

```
create bufferpool BPONEPART immediate nodegroup "ONEPARTGROUP" size 125000  
except on nodes (0 size 125000) pagesize 4 K
```

```
create bufferpool BPTWOPART immediate nodegroup "MYTWOPARTGROUP" size 125000  
pagesize 4 K ;
```

```
connect reset
```

The first **create bufferpool** command creates a bufferpool named BPONEPART on the database partition group (or node group) ONEPARTGROUP.

Please note that the second **create bufferpool** command does not use the clause *on nodes* since it will be created over the two existent physical database partitions.

The next step creates the tablespaces (see Example 4-5).

Example 4-5 Create tablespaces

```
connect to sample
```

```
create regular tablespace TBLSPONEPART in database partition group  
"ONEPARTGROUP" pagesize 4 K managed by database using ( file  
'C:\data1\cont_1.dms' 204800, FILE 'C:\data1\cont_2.dms' 204800 ) on  
dbpartitionnum(0) extentsize 16 overhead 10.5 prefetchsize 16 transferrate  
0.14 bufferpool "BPOOLONEPART" dropped table recovery on
```

```
create regular tablespace TBLSPWOPARTS in database partition group  
"TWOPARTGROUP" pagesize 4 K managed by database using ( file  
'c:\data3\cont0_1.dms' 76800 ) on dbpartitionnum (0) using ( file  
'c:\data3\cont1_1.dms' 76800 ) on dbpartitionnum (1) extentsize 16 overhead  
10.5 prefetchsize 16 transferrate 0.14 bufferpool "BPTWOPART" dropped table  
recovery on
```

```
connect reset
```

The first **create tablespace** command creates a DMS tablespace with two containers on a partition group with a single physical database partition. This tablespace is linked to the bufferpool BPPONEPART.

The second **create tablespace** command creates a DMS tablespace with one container, but on a partition group with two physical database partitions (0 and 1). This tablespace is linked to the bufferpool BPTWOPART.

The DMS containers used in this example are based on files. On Windows 2000 Server and Windows Server 2003 platforms you can define DMS raw tablespaces containers as *physical drives* and as *logical drive*. Example 4-6 shows two commands using raw devices.

Example 4-6 Create DMS tablespaces using raw devices

```
create regular tablespace ... pagesize 4 K managed by database using ... (  
device '\\.\E:' 256000 ) on dbparttionnum(0) ...
```

```
create regular tablespace ... pagesize 4 K managed by database using ... (  
device '\\.\PhysicalDrive0' 256000 ) on dbparttionnum(0) ...
```

The first command shows the syntax of using the logical drive. The second command shows the syntax of using the physical drive.

Hint: You can use the DB2 system command **db2listvolumes** to gather volume information of your server. This command creates two files, `volumes.xml` and `tablespace.ddl`. For the XML file you need an XML editor to view this file. This is a recommended option because the volume information is well organized and presented in the XML editor. You can see a sample of this file in Figure 4-36. The `tablespace.ddl` file must be edited in order to use it as a create tablespace or alter tablespace script.

Tree Structure	Values
✕ xml	version="1.0" encoding="US-ASCII"
📄 [DOCTYPE]	volumes
[-] ➤ volumes	(host="S02")
[+] ➤ volume	(num="0")
[+] ➤ volume	(num="1")
[+] ➤ volume	(num="2")

Source View Tree View IE view

Figure 4-36 Sample output of volumes.xml

After creating database partition groups, bufferpools, and tablespaces, now you can create a table on the available tablespaces. The next command is an example of a **create table** statement with a partition key (Example 4-7).

Example 4-7 Create partitioned table

connect to sample

```
create table TWO_PARTS_TABLE
( billing_id character (10) , customer_id character (10) , customer_name
varchar (50) , project_id character (10) , project_name varchar (50) ,
employee_id character (10) , employee_name varchar (50) , hourly_rate decimal ,
hours_billed smallint , date_billed date , notes varchar (1000) )
in TBLSPWOPARTS
index in TBLSPWOPARTS
partitioning key ( BILLING_ID, CUSTOMER_ID)
```

connect reset

The **create table** command creates a table in a tablespace split over two partitions. This means that the table is physically created over two physical database servers with a partition key defined with columns.

Related publications

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

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 241. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *DB2 UDB Exploitation of the Windows Environment*, SG24-6893
- ▶ *DB2 UDB Evaluation Guide for Linux and Windows*, SG24-6934

Other publications

These publications are also relevant as further information sources:

- ▶ *IBM DB2 UDB Command Reference Version 8*, SC09-4828
- ▶ *IBM DB2 Universal Database What's New*, SC09-4848
- ▶ *IBM DB2 Universal Database Administration Guide: Planning*, SC09-4822
- ▶ *IBM DB2 Universal Database Administration Guide: Implementation*, SC09-4820
- ▶ *IBM DB2 Universal Database Administration Guide: Performance*, SC09-4821
- ▶ *IBM DB2 Universal Database Data Movement Utilities Guide and Reference*, SC09-4830
- ▶ *IBM DB2 Universal Database Data Recovery and High Availability Guide and Reference*, SC09-4831
- ▶ *IBM DB2 Universal Database Federated Systems Guide*, GC27-1224
- ▶ *IBM DB2 Universal Database Guide to GUI Tools for Administration and Development*, SC09-4851
- ▶ *IBM DB2 Universal Database SQL Reference, Volume 1*, SC09-4844
- ▶ *IBM DB2 Universal Database SQL Reference, Volume 2*, SC09-4845

- ▶ *IBM DB2 Universal Database System Monitor Guide and Reference*, SC09-4847
- ▶ *IBM DB2 Universal Database Application Development Guide: Building and Running Applications*, SC09-4825
- ▶ *IBM DB2 Universal Database Application Development Guide: Programming Client Applications*, SC09-4826
- ▶ *IBM DB2 Universal Database Application Development Guide: Programming Server Applications*, SC09-4827
- ▶ *IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1*, SC09-4849
- ▶ *IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2*, SC09-4850
- ▶ *IBM DB2 Universal Database Data Warehouse Center Application Integration Guide*, SC27-1124
- ▶ *IBM DB2 XML Extender Administration and Programming*, SC27-1234
- ▶ *IBM DB2 Universal Database Quick Beginnings for DB2 Clients*, GC09-4832
- ▶ *IBM DB2 UDB Quick Beginnings for DB2 Servers V8*, GC09-4836
- ▶ *IBM DB2 Universal Database Installation and Configuration Supplement*, GC09-4837

Online resources

- ▶ IBM White paper he wrote, *Install IBM DB2 Universal Database Express Edition Transparently with Your Application*
<http://www7b.boulder.ibm.com/dmdd/library/techarticle/0306gao/0306gao.html>
- ▶ Hot-add memory support
<http://www.microsoft.com/whdc/hwdev/tech/hotadd/hotaddmem.mspx>
- ▶ "DB2 Tuning Tips for OLTP Applications" written by Yongli An and Peter Shum. It can be found at
<http://www7b.boulder.ibm.com/dmdd/library/techarticle/anshum/0107anshum.html>

DB2

- ▶ Database and Data Management home page
<http://www.ibm.com/software/data/>

- ▶ DB2 Universal Database home page
<http://www.ibm.com/software/data/db2/udb/>
- ▶ DB2 Technical Support
<http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/index.d2w/report>
- ▶ DB2 online library
<http://www.ibm.com/db2/library>
- ▶ DB2 for Windows
<http://www.ibm.com/db2/windows>

eServer

- ▶ xSeries home page
<http://www.pc.ibm.com/us/eserver/xseries/index.html>

Windows

- ▶ Microsoft home page
<http://www.microsoft.com/>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

Symbols

/3GB 14, 131

/PAE 16, 132

Numerics

2-way SMP 3

32-bit installation 50

4 GB Tuning 131

64-bit installation 51

64-bit memory, configuration 97

64-bit, checking for 51

64-bit, database configuration 105

64-way Itanium 16

A

Active Directory 67

Active Directory Installation Wizard 186–187

Active Directory Summary Window 190

Active Directory Users and Computers 67

Active Directory Wizard 187

Active Directory, adding users 68

active log space 162

add dbpartitionnum 224

Add new database partition server 200, 204

Add Partition Wizard 210

Add Partitions Launchpad 211, 215

Adding a Server to the Domain 192

Adding additional disk 146

Additional partitions 126

administration contact list 199

Administration policy 93

Administration Server 57

Administrator 186

Administrator rights 47

Administrators group 195

Agent stack size 102

Agent_stack_sz 102

ALTER DATABASE PARTITION GROUP 230

alter table 233

alter tablespace 175, 234

APAR 50

APP_CTL_HEAP_SZ 107

APP_GROUP_MEM_SZ 107

APPLHEAPSZ 108

Application control heap 107

Application heap 108

Application support layer heap 102

artitioning key 233

ASLHEAPSZ 102

Audit buffer size 102

AUDIT_BUF_SZ 102

aw_size 132

AWE 15, 26, 97

AWE Memory 131

B

Backup 34, 126

Backup in progress state 174

backup pending state 173

Big block reads 163

Blade servers 119

BladeCenter Servers 18

BMDEFAULTGROUP 152

boot.ini 14, 132

bp_id 132

bp_size 132

Bufferpool hit ratio 144

Bufferpools 106

C

Capacity growth, estimating 171

Case study OLTP 141

Case study, data warehouse 144

Catalog cache 108

Catalog tablespace 152

CATALOGCACHE_SZ 108

certified 8

circular logging 162

cluster 178

Cluster benefits and limitations 181

Cluster, issues 182

Clustering Services 12

Cold add 130

Collocation 232

Command Line Processor 82

- Command Window 82
- Communication, v7 AND v8 112
- COMP 64
- Compatibility issues 112
- Computer Name Changes 193
- configFileName 37
- Configuration Advisor 88, 91
- Configuration Advisor, issues 111
- Configuration Advisor, memory allocation 91
- Configure Parameter 88
- Containers, adding additional 156
- Control Center 83, 138, 169, 171, 211
- Control Center, using 86
- coordinator node 181, 195
- Copy Partition Settings 212
- CPU limits for DB2 54
- CPU_PARALLELISM 127
- create database 233
- CREATE DATABASE PARTITION GROUP 230
- Create database, example 234
- CREATE INDEX 25
- create table 233
- CREATE TABLESPACE 164
- create tablespace 234
- CREATE TOOLS CATALOG 96
- Creating a new user 67
- Creating an Instance 67

D

- DAS 57, 194
- dasidrop 74
- Data Links File Manager 63
- Data Manipulation Language 154
- data redistribution 214
- Data type support 112
- Data warehouse case study 144
- Database configuration 87
- Database environment, partitioned 198
- Database environment, single partition 198
- Database manager 80
- Database manager variables 80
- Database manager, Pending Value Effective 86
- Database manager, setting 85
- Database manager, show detail 81
- Database parameters 105
- Database partition groups 230
- Database Partitioning Feature 6, 10
- Database population 94

- Database shared memory 106
- database structure 229
- Database transaction volume 93
- Database variables, listing 87
- Database variables, setting 89
- Database, 64-bit configuration 105
- Database, APP_CTL_HEAP_SZ 107
- Database, APP_GROUP_MEM_SZ 107
- Database, APPLHEAPSZ 108
- Database, CATALOGCACHE_SZ 108
- Database, DATABASE_MEMORY 107
- Database, ESTORE_SEG_SZ 109
- Database, LOCKLIST 109
- Database, LOGBUFSZ 109
- Database, MAXAPPLS 123
- Database, NUM_QUANTILES 125
- Database, PCKCACHESZ 110
- Database, SHEAPTHRES_SHR 107
- Database, SORTHEAP 106
- Database, STAT_HEAP_SZ 110
- Database, STMTHEAP 111
- Database, UTIL_HEAP_SZ 111
- DATABASE_MEMORY 106
- DataPropagator 162
- DB2 Administration Server 57
- DB2 Clients 53
- DB2 configuration 133
- DB2 product type 53
- DB2 Registry 75
- DB2 registry 75
- DB2 Universal Database 8
- DB2 Universal Database for Windows 8
- DB2, 32-bit version 48
- DB2_AWE 78, 132
- DB2_PARALLEL_IO 165
- DB2_STRIPED_CONTAINERS 166
- DB2_USE_PAGE_CONTAINER_TAG 166
- db2batch 128
- DB2BPVARS 78
- db2ckmig 42
- db2ckmig, command syntax 42
- db2cmd 82
- db2dasrrm.exe 134
- db2empfa 154
- db2icrt 72, 221–222
- db2idrop 46, 74
- db2ilist 221
- db2jds.exe 134
- db2licd 42

- db2licd.exe 134
- db2listvolumes 236
- db2listvolumes.exe 160
- db2ncrt 222
- db2nodes.cfg 222
- DB2NTMEMSIZE 79
- DB2NTNOCACHE 77, 158
- DB2NTPRICLASS 79
- DB2NTWORKSET 79
- db2ReadLog API 162
- db2relocatedb 37, 39
- db2relocatedb, execution example 40
- db2relocatedb, steps 38
- DB2RHIST.ASC 229
- db2rspgn 63
- db2sampl 209
- db2sec.exe 134
- db2set 75, 221
- db2set -all 75
- db2start 208, 221
- db2start addnode nodenum 223
- db2stop 42
- db2syscs.exe 133
- DB2TSCHNG.HIS 229
- dbpartitionnum 209
- Delete pending state 173
- Design Considerations 24
- Development Center 114
- device_size 161
- dft_prefetch_sz 163
- direct attach 147
- Disable Pending State 174
- Disk space 47
- Disk space required 47
- disks assignments 36–37
- DLFM 63
- dministrator 67
- DML 154
- DMS 155
- DMS rebalancer is active 175
- DMS Table Space 171
- DMS, design considerations 157
- DNS name 189
- Documentation 30
- Documentation updates 31
- Documentation, online 48
- Domain 68, 193
- Domain Administrator Group 192
- Domain Administrators group 201

- Domain Controller 47, 70, 184, 188
- Domain controller server 186
- Domain Controller, installing 186
- Domain NetBios Name 189
- Domain types 188
- domain user account 195
- DPF 6, 10
- DPROP 162
- DPS 174
- drop database 210
- drop pending state 175
- DSS instance confirmation 144

E

- Enterprise Server Edition 9–10
- Enterprise X-Architecture 17
- ePlane 19
- ESE 20
- ESE Partitioning Feature 179
- eServer 17
- ESTORE_SEG_SZ 109
- Express Edition 9
- Extended storage segment size 109
- extent size 151
- EXTENTSIZE 164

F

- Fast Communication Manager 206
- Faster transaction performance 93
- FCM buffers 103
- FCM_NUM_BUFFERS 103
- FILE 64
- File system caching 158
- Fixpak 49
- Fixpak 2 48
- Fixpak, readme 50
- Fixpak, required level 49
- Fixpak. identifying 49
- force application 40
- function shipping 182

G

- Gao, Michael 61
- get dbm cfg 80

H

- hashed 232

- Health Monitor 168
- high availability 22
- Host file 207
- hostname 185, 225
- Hot Add 133
- Hot add 130
- Hot-Add Memory 11
- HTML documentation 30

I

- IBM xSeries Rack Servers 18
- IBM xSeries Scalable Servers 19
- IBM xSeries Universal Servers 18
- IBMCATGROUP 152
- IBMTEMPGROUP 153
- Indexes 25, 170
- Information Center 30
- INSTALL_TYPE 64
- Installation image, identifying 48
- Installation image, incorrect 48
- Installation images, 32 and 64-bit 48
- Installation overview 183
- Installation requirements 46
- Installation, 64-bit 51
- Installation, choosing components 56
- Installation, considerations 46
- Installation, contact list 58
- Installation, DB2 partitioned environment 194
- Installation, instance options 59
- Installation, instance user 59
- Installation, partitioned requirements 183
- Installation, partitioning options 56
- Installation, physical database partition 203
- Installation, references 50
- Installation, starting 54
- Installation, steps 51
- Installation, type 54
- Installation, users and security 194
- Installation, using a response file 61
- Instance creation, authority needed 67
- Instance memory 102
- Instance Owning Database Partition 181
- Instance Owning Node 195
- Instance Owning Node, installation 196
- instance owning partition 195
- Instance parameters 98
- Instance user 59
- Instance, 64-bit parameter list 98

- Instance, AGENT_STACK_SZ 102
- Instance, ASLHEAPSZ 102
- Instance, AUDIT_BUF_SZ 102
- Instance, FCM_NUM_BUFFERS 103
- Instance, INSTANCE_MEMORY 102
- Instance, INTRA_PARALLEL 104, 124
- Instance, JAVA_HEAP_SZ 103
- Instance, MAX_CONNECTIONS 121
- Instance, MAX_COORDAGENTS 121
- Instance, MAX_QUERYDEGREE 125
- Instance, MAXAGENTS 121, 123
- Instance, MAXCAGENTS 122
- Instance, MIN_PRIV_MEM 103
- Instance, MON_HEAP_SZ 103
- Instance, PRIV_MEM_THRESH 104
- Instance, QUERY_HEAP_SZ 104
- Instance, removal restrictions 74
- Instance, removing 74
- Instance, SHEAPTHRES 99
- INSTANCE_MEMORY 102
- Instance-owning database partition 198
- Intel Xeon 19
- Inter-partition parallelism 125
- Inter-query parallelism 120
- intra parallel parallelism 124
- Intra parallelism 104
- INTRA_PARALLEL 104, 124, 145
- Intra-partition parallelism 123
- Intra-query parallelism 122
- Is the database scalable? 21
- Isolation level 94

J

- Java interpreter heap 103
- JAVA_HEAP_SZ 103

K

- Key parameter changes 143
- KILL_PROCESSES 65

L

- LANG 64
- LargeSystemCache 130
- list database directory 209
- LOAD 25, 173
- Load 127
- Load pending state 173

- LOB 149, 158
- Local Policies 70
- Lock list 109
- LOCKLIST 109
- Log buffer 109
- LOGBUFSZ 109
- LOGFILSZ 161
- logical unit 29
- logically partitioned database 229
- logprimary 161
- LOGRETAIN 162
- logsecond 161
- LUN 29

M

- Manage Storage 169
- MAX_CONNECTIONS 121
- MAX_COORDAGENTS 121
- MAX_QUERYDEGREE 125
- max_time_diff 185
- MAXAGENTS 121, 123
- MAXAPPLS 123
- MAXCAGENTS 122
- maximum degree of parallelism 125
- maximum number of agents 121, 123
- maximum number of applications 123
- maximum number of concurrently executing coordinator agents 122
- maximum number of connections 121
- maximum number of coordinating agents 121
- Maximum size of the application group memory 107
- Memory requirements 47
- Memory Visualizer 138
- Memory, adding 129
- Memory, cold add 130
- Memory, hot add 130
- Memory, optimizing 130
- Memory, physically adding 130
- Memory, scaling 26
- Microsoft Cluster Service 15, 178
- Microsoft Clustering Service 4
- Microsoft Windows Cluster 184
- migrate command 43
- migrate, command syntax 43
- Migrating between versions of DB2 41
- Migrating, back to 32-bit 45
- Migrating, back to DB2 V7 or V6 46

- Migrating, between versions of DB2 41
- Migrating, database steps 42
- Migrating, DB2 34
- Migrating, from 32-bit to 64-bit 43
- Migrating, illegal database states 41
- Migrating, Restrictions 41
- Migrating, versions of Windows 34
- Migrating, Windows and DB2 43
- migration 34
- migration process 34
- Migration Restrictions 41
- Migration steps 42
- MIN_PRIV_MEM 103
- Minimum private memory 103
- MLN 7
- MLP 7, 27
- MON_HEAP_SZ 103
- Monitor heap size 103
- Moving disks 36
- MSCS 4, 12, 178
- MSCS, restrictions 179
- multipage file allocation 154
- multipartition database partition group 230
- Multiple Logical Nodes 7
- multiple logical partitioned database 7
- Multiple logical partitions 28
- My Computer 51

N

- Need for High Availability 182
- Need for High Reliability 182
- Need for High Scalability 182
- netname 225
- New database partition 199
- New database partition server 204
- newlogpath 150, 161
- NODE0000 229
- nodes 178
- not accessible state 174
- NUM_QUANTILES 125
- NUMA Memory 12
- Number of applications 94

O

- OLTP case study 141
- OLTP, parameter changes 143
- on nodes 235
- Operating System Compatibility 187

OS security privileges 47

P

Package cache 110
PAE 15
Page cleaning 164
Page Reads 114
page size 152
Page Writes 114
Paging 114
Parallel Processing 25
parallelism 120
PARALLELISM n 126–127
Parallelism, Inter-partition 123, 125
Parallelism, inter-query 120
Parallelism, Intra-query 122
Partition groups and table spaces 234
partition number 200, 205
partition owning server 207
partitioned databases, creating 229
Partitioning 27
partitioning key 232
partitioning map 232
Partitioning, creating 220
Partitioning, installation options 56
partitions 7
Partitions, adding 211
Partitions, assigning screen 213
PCI buses 24
PCKCACHE_SZ 110
PCKCACHESZ 110
PDF documentation 30
perfmon 114, 128, 134
Performance Tuning 143, 240
Performance, observing 127
Permissions 189
Physical Address Extensions 15
Plug & Play 36
plug & play 36
port 225
Port range 60
port range 207
Prefetching 163
PREFETCHSIZE 163
Preliminary Steps 186
PRIV_MEM_THRESH 104
Private memory threshold 104
privileges 70

Processors, adding additional 118
PROD 64
program Menu 67

Q

quantiles 125
Query heap size 104
query optimizer 120
Query time 127
Query workload 92
QUERY_HEAP_SZ 104
Quiesce command 40
quiesce tablespaces 172
quiesce, command syntax 40

R

RAID 147, 164
raw devices 236
raw disk 36
raw I/O 158
Readme 47
real time system analysis 215
REBOOT 64
Recovery 25
Redbooks Web site 241
 Contact us xviii
Redistribution, Staging 216
Redistribute data 214
Redistribution Method 215
Redistribution Strategy 215
Redistribution, Analyze 215
Redistribution, Schedule 217
Redistribution, Strategy 215
Registry variables 77
Registry, DB2_AWE 78
Registry, DB2BPVARS 78
Registry, DB2NTMEMSIZE 79
Registry, DB2NTNOCACHE 77
Registry, DB2NTPRICLASS 79
Registry, DB2NTWORKSET 79
Registry, listing 75
Registry, setting 76
relocation 39
Reorg in progress state 174
Response File 55
Response file, creating 61
Response file, creating with a GUI 62
Response file, database creation 66

- Response file, generator 63
- Response file, installing with 61, 65
- Response file, manually creating 64
- Response file, primary codes 65
- Response file, secondary codes 66
- Response Time 3
- Restore 34, 127
- Restore in progress state 174
- restore pending state 173
- Restore, DB2 V7.2 to V8.1 44
- Rip and Replace 119
- rollforward pending state 173

S

- SAMPLE 209
- sample database 208
- Sample database, creating 208
- SAN 119, 147
- Scalability questions 21
- Scalability, Response Time 3
- Scalability, workload 2
- Scalaibility, throughput 2
- Scaling out 178
- Scaling Overview 2
- Scaling-Up 3
- Scatter read 78
- Scheduling the task 95
- Security issues 112
- Server information 220
- Server Optimization 130
- service level agreement 12
- services 222
- Services file 206
- services file 72, 207
- Services option 72
- set 221
- Set Containers 213
- SET WRITE SUSPEND FOR DATABASE 175
- setup.exe 65
- Shared Nothing Architecture 180
- SHEAPTHRES 99, 145
- SHEAPTHRES_SHR 107
- SLA 12
- SMS 154
- SMS Table Space 168
- Snapshots 140
- Sort heap 106
- sort heap 99

- sort heap overflow 99
- Sort heap threshold 99
- Sort heap threshold for shared sorts 107
- SORTHEAP 106
- SP3 14
- Specify Snapshot Schedule 169
- Specify Snapshot Storage 169
- Specify Threshold Settings 169
- SQL*.DAT 230
- SQL00001 229
- SQLBP.1 229
- SQLDBCON 229
- SQLINSLK 229
- SQLSPCS.1 229
- SQLT* 230
- SQLTAG.NAM 230
- Staged Redistribution Plan 216
- start database manager 225
- STAT_HEAP_SZ 110
- Statement heap 111
- Statistics heap 110
- Steps of using db2relocatedb 38
- STMTHEAP 111
- storage area network 119
- Storage Management GUI 167
- storage management record 215
- Storage management, DB2 167
- Storage may be defined state 175
- Storage must be defined state 174
- Storage snapshot 167
- Storage, DB2 considerations 149
- Storage, scaling 26
- Storage, Windows 147
- Stripe Sets 156
- SYSCATSPACE 152
- system area networks 147
- System Managed Space 154
- System scalability 2
- system scalability 3
- System use 128

T

- Tables 170
- tablespace 26
- Tablespace container, operational state health indicator 176
- Tablespace in Not Accessible state 176
- Tablespace, data distribution 150

- Tablespace, design 150
- Tablespace, health indicator 172
- Tablespace, quiesced exclusive state 172
- Tablespace, quiesced share state 172
- Tablespace, quiesced update state 172
- tablespace.ddl 236
- Task Manager 128, 133
- Task Scheduler 60
- TBS creation in progress state 175
- TBS deletion in progress state 175
- TCP/IP Considerations 206
- TEMPSPACE1 153
- terminate 43
- Throttling 126
- Throughput 2
- Time settings consistent 184
- Tools database 60

U

- UDB 8
- update dbm cfg 42
- User Rights Assignments 70
- User specifications 69
- User tablespace 152
- User, adding to groups 69
- User, adding with Active Directory 68
- User, creating 67
- USERSPACE1 152
- UTIL_HEAP_SZ 111
- UTIL_IMPACT_LIM 126
- Utility heap 111
- Utility parallelism 126
- Utility time 128

V

- V7 64-bit 113
- V7 Clients to a V8 Server 113
- V8 clients to a V7 Server 112
- VLDB 6, 179
- volumes.xml 236

W

- W2K 14
- Web Health Center 168
- White paper 61, 240
- Windows Datacenter Hardware Compatibility List 147

- Windows Datacenter High Availability 17
- Windows domain 47
- Windows kernel 131
- Windows Server 2003 Datacenter Edition 16
- Windows Server 2003 Editions 13
- Windows Server 2003 Enterprise Edition 15
- Windows Server 2003 Standard Edition 14
- Windows Server 2003 Web Edition 14
- Windows Server, 64-bit 53
- Windows Services 71
- Windows services file 72
- Windows System Resource Manager 12
- Windows, 64 Bit Support 11
- Windows, configuring 130
- Windows, scaling issues 120
- Windows, storage 147
- Windows, viewing memory 133
- Winsock Direct 147
- winver 52
- Workgroup Server Edition 9–10
- Workgroup Server Unlimited Edition 9–10
- write resume 175
- write suspend state 175
- WSE) 20
- WSRM 12
- WSUE 20

X

- x360 18
- x440 19
- x445 19, 119
- x450 19, 119
- x455 20, 119
- X-Architecture 17
- xSeries 17
- xSeries information 133
- xSeries x440 Server 19
- xSeries x450 Servers 20



Scaling DB2 UDB on Windows Server 2003



Redbooks

Scaling DB2 UDB on Windows Server 2003

Scaling out on clustered database servers

Scaling up on large SMP servers

Leveraging 64-bit technology

As organizations strive to do more with less, DB2 Universal Database V8.1 on Windows Server 2003 brings unprecedented enterprise-class scalability to 32-bit and 64-bit Intel platforms. Supporting both vertical (scale up) and horizontal (scale out) scaling, DB2 UDB V8.1 represents over a decade of industry-proven scalability with both non-clustered and clustered database architectures.

This IBM Redbook is designed as an informative guide to scaling DB2 UDB V8.1 with Windows Server 2003. It is intended for database and system administrators who need both an introduction and detailed information on scaling DB2 UDB on Windows Server 2003.

We begin with an overview of scalability including the goals, measurements, and common approaches. We introduce the scalability features of DB2 UDB, Windows Server 2003, and the IBM xSeries Servers, followed by design considerations for scaling. We dedicate an entire chapter to running DB2 UDB on 64-bit Windows including installation, configuration, and 64-bit migration techniques. We discuss scaling up and how to leverage large memory support in 32-bit environments. Finally, we include a chapter on scaling out, including adding logical and physical database partitions.

**INTERNATIONAL
TECHNICAL
SUPPORT
ORGANIZATION**

**BUILDING TECHNICAL
INFORMATION BASED ON
PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks