DB2. Information Management Software

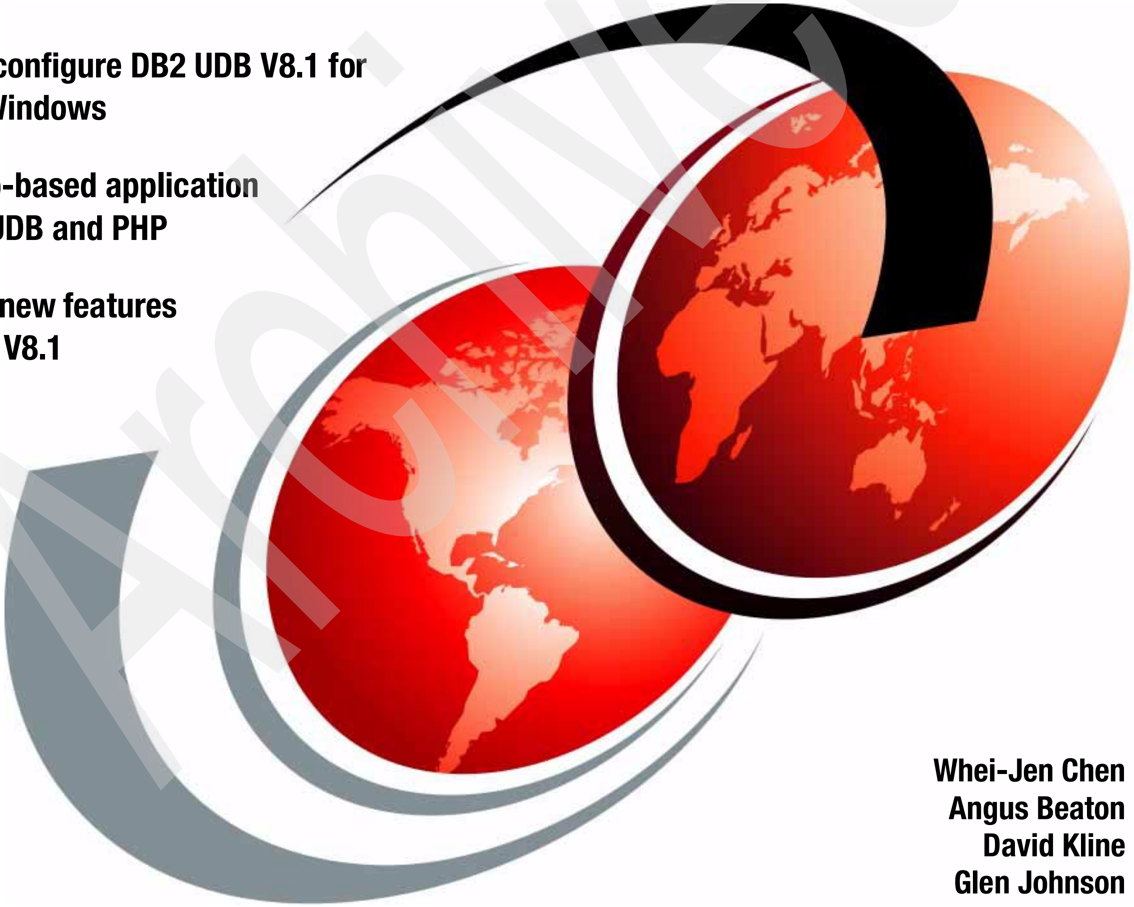
IBM

DB2 UDB Evaluation Guide for Linux and Windows

Install and configure DB2 UDB V8.1 for Linux and Windows

Build a Web-based application using DB2 UDB and PHP

Explore the new features of DB2 UDB V8.1



Whei-Jen Chen
Angus Beaton
David Kline
Glen Johnson

Redbooks

ibm.com/redbooks

International Technical Support Organization

DB2 UDB Evaluation Guide for Linux and Windows

April 2003

SG24-6934-00

Note: Before using this information and the product it supports, read the information in "Notices" on page xvii.

First Edition (April 2003)

This edition applies to DB2 Universal Database Version 8.1 for use with SuSE Linux Version 8.1, Red Hat Linux Version 8.0, and Windows 2000 operating systems.

Contents

Figures

Tables

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:

eserver™
eserver™
eServer™
iSeries™
z/OS™
z/VM™
zSeries™
AIX®
AS/400®
CICS®
Database 2™
DataJoiner®
DB2 Connect™
DB2 Extenders™
DB2 Universal Database™
DB2®
DRDA®
Enterprise Storage Server™
Everyplace™
Informix®
Intelligent Miner™
IBM®
IMS™
Lotus®
Multiprise®
MQSeries®
Notes®
OS/2®
OS/390®
OS/400®
PartnerWorld®
Perform™
QMF™
Redbooks™
Redbooks (logo) ™
S/390®
Tivoli®
VisualAge®
WebSphere®
1-2-3®

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 helps you to become quickly familiarized with the features and functions of DB2® for the Linux and Windows Distributed Platforms Version 8. We cover DB2 product installation, application development, use of DB2 graphical tools, database management and DB2 product selection and licensing. This redbook is organized into these topics:

Chapter 1. Introduction to DB2

In this chapter, we introduce DB2 UDB for Multiplatforms V8.1. We present a high level overview of the key features of DB2 UDB as well as the new features included with Version 8.1. We also provide some DB2 terminology, which is useful for those readers who are not familiar with DB2 UDB.

Chapter 2. Installation

In this chapter, we guide you through the process of obtaining and installing the DB2 V8.1 trial code. The installation covers both the Linux and Windows platforms. The emphasis is on using the graphical DB2 Setup wizard to install and configure DB2. We also cover the additional methods available to install DB2, namely the db2_install command using DB2 response files and native OS commands.

Chapter 3. Navigating DB2 with GUIs

This chapter describes the various graphical tools packaged with DB2. The GUIs provide a much more intuitive approach in navigating DB2, allowing you to become familiar much more rapidly with the DB2 components. Each of the DB2 tools provides a specific administrative or development function to increase your productivity. The Control Center, Task Center, Health Center and many more DB2 GUI tools are explained.

Chapter 4. DB2 order entry Web application

In this chapter we describe how to produce a Web-enabled DB2 application using the Apache Web Server and the PHP scripting language. We describe how to build an Apache Web Server with PHP and DB2 connectivity in both Linux and Windows environments. We build a PHP based Web client, which can interact with the DB2 data using traditional SQL DML, such as INSERT, UPDATE, DELETE, and SELECT. This then is extended to include more sophisticated features of DB2, such as stored procedures and triggers.

Chapter 5. Application development

It has become increasingly important for a database to effectively support a wide range of open standards and open source products for application development. DB2 UDB provides accessibility for many standard interfaces and comes with a full suite of development tools. In this chapter we introduce aspects of developing applications for DB2. We will also provide examples of using these development features within our sample application.

Chapter 6. Database management

In today's complex information management environments, database administrators need an extensive set of tools to fully manage the database system efficiently and effectively. With the release of DB2 Version 8.1, a variety of new management features are available to enhance the already impressive set of tools available in Version 7.2. In this chapter, we discuss DB2 Version 8.1 database management functionality. We introduce both existing and new features available to the database administrator.

Chapter 7. Additional DB2 features

This chapter provides information on some additional features included within DB2. These features are placed together in this chapter, because they are extensions to the general usage of DB2. Most of these topics are sufficiently detailed to have warranted redbooks in their own right. This chapter introduces these features, discusses how they can be used, and provides references to additional material. Another facet shared by these features is that they are included as part of the base DB2 product and also the trial code version.

Chapter 8. High availability, scalability, and parallelism

This chapter discusses these three concepts of high availability, scalability, and parallelism. For high availability, we discuss ways to protect your database and log files using DB2 and IBM products, various third party failover products, and some high availability maintenance features in DB2 to reduce the need for database or instance downtime. For scalability, we introduce DB2's partitioning feature to show how a database can run seamlessly across multiple machines while benefitting from each machine's hardware resources. For parallelism, we show how DB2's partitioning feature can take a database task, such as a query, and break it up into smaller pieces to enable workload sharing across all partitions.

Chapter 9. Selection guide

In this chapter we discuss how DB2 is licensed for different requirements. We cover the different versions of DB2 that are available on Linux and Windows, in some detail. IBM's new Data Management Tools for multi platforms are also covered, and they act as an extension to the feature-rich capabilities supplied as standard. Finally, we cover how a new DB2 UDB user can obtain education, certification, and access a wealth of resources from the Internet.

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, San Jose Center.

The team (left to right): David Kline, Angus Beaton, Whei-Jen Chen, Glen Johnson

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.

Angus Beaton is an IT Specialist working in Data Management Technical Pre-sales in IBM United Kingdom. He has 5 years of experience in data management and is an IBM Certified Solutions Expert in DB2 UDB Database Administration and DB2 Application Development. His areas of expertise include business intelligence and competitive migrations from other relational databases to DB2 UDB.

Glen Johnson is a Software Engineer with IBM's Linux Technology Center in Austin, Texas. In his current role as Technical Liaison, he is responsible for facilitating the technical communication between IBM's product development groups and its Linux Distribution partners. He has extensive experience in developing Web-enabled database tools for work flow automation and business process improvement. Prior to joining IBM in 1992, he was CFO of Nova Concepts, Inc. (an IBM Business Partner) and Head of Immunochemistry Research at Kallestad Laboratories. He holds a Ph.D. in Biochemistry from the State University of New York at Buffalo.

David Kline is a Software Engineer in the United States. He works as a Subject Matter Expert for Independent Software Vendors (ISVs) at the IBM Developer Technical Support (DTS) Center in Dallas, within PartnerWorld® for Developers. He has 4 years of experience in the database field and has worked at IBM for 4 years. His areas of expertise include technical support for database administration tasks on UNIX and Windows operating systems. He has written extensively on DB2 topics which can be found at the DB2 Developer Domain Web site:

http://www7b.boulder.ibm.com/dmdd/

Acknowledgement

The authors express their deep gratitude for the help they received from **Lloyd Budd**, **Jessica Escott**, and **Darin McBride**, from the IBM Toronto Laboratory. They would also like to thank the following people for their contributions to this project:

Mike Logan, James Perabo, Vicki Martin Petruch
IBM Data Management and Marketing

Blair Adamache, Rav Ahuja, Anson Kokkat, Darin Mcbride, Dipesh Patel, Jim Stittle, Florbela Vieira, Paul Yip
IBM Toronto Laboratory

Harold T. Morgan
World Wide Consulting - Data management group

Emma Jacobs, Deanna Polm
International Technical Support Organization, San Jose 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

1

Introduction to DB2

DB2, IBM's premier relational database software, is the worldwide market share leader in the industry. The DB2 family spans a wide variety of Linux, UNIX, and Windows platforms and the IBM iSeries™ (OS/400®) and zSeries™ (OS/390®, z/OS™, z/VM™, and Linux) server lines. With strong and innovative technology, DB2 provides high value to customers of all sizes. DBA automation and self-management enhancements combined with low price provides remarkable value to DB2 customers.

In this chapter, we introduce DB2 Universal Database™ (UDB) for Multiplatforms. We present a high level overview of the key features of DB2 UDB as well as the new features included with Version 8.1. We also provide some DB2 terminology, which is useful for those readers who are not familiar with DB2 UDB.

DB2 Universal Database (UDB) is a complete multimedia, Web-ready, relational database management system, strong enough to meet the demands of large corporations and flexible enough to serve medium-sized and small businesses. Combining the power for business intelligence with high performance and reliability, DB2 drives demanding industry solutions. It continues to meet the demands of today's heterogeneous computing environments by providing open solutions that can access and integrate data from multiple geographically separated sources on different platforms.

DB2 UDB is a scalable and industrial strength relational database management system. It is flexible enough to serve both structured and unstructured data management needs of a single user to very large enterprises. It is suitable for a wide range of customer applications, which can be deployed on a variety of hardware and software platforms all the way from palm devices to massive parallel multiprocessor systems. It is a high performance middleware software that provides a secure and reliable system to manage data or content for a wide variety of customer applications, such as e-Commerce, customer relationship management (CRM), enterprise resource planning (ERP), supply chain management (SCM), business intelligence (BI), Web portals, content and knowledge management.

In all there are more than 60 million DB2 users from 400,000 companies worldwide relying on IBM data management technology.

This chapter contains these topics:

- DB2 UDB features
- Version 8 highlights
- DB2 database objects
- DB2 UDB architecture and processes
- Accessing and updating DB2 data

1.1 DB2 UDB features

DB2 is a highly scalable, highly extensible, very easy-to-use and manage database that can be trusted with your most critical database applications. Here we discuss the high level features of DB2 UDB:

- **Scalability**: DB2 UDB can run on everything from Laptops supporting mobile users to massive parallel systems with terabytes of data and/or thousands of users. It is capable of scaling this breadth of systems with the same function. This feature enables you to minimize costs and maximize personnel skills by using a single database for all your application needs, no matter what the scale.
- **Multimedia extensibility**: DB2 UDB allows you to extend the capabilities of the database to meet your specific organizational requirements. This feature includes the ability to support more advanced applications involving multimedia data such as documents, images, audio and video. You can now develop applications that use this technology and extend DB2 to support these new applications. In addition, new XML, spatial and high-speed searching extenders are available, along with object-relational extensions to the SQL language.
- **Web-enablement**: DB2 UDB is integrated with Web technology so that data can be easily accessed from the Internet or from your intranet to provide a more secure environment. This feature allows you to build e-business or Internet applications that can enable competitive advantage, greater customer service, or reduced costs.

- **Partner solutions**: DB2 UDB is capable of supporting a broad range of data management applications with excellent performance and reliability. As a result, a large number of industry solution providers, including SAP, Baan, Peoplesoft, and many more, have adopted DB2 to support their applications; or adapted their tools to support DB2. This feature allows you to more quickly satisfy your requirements by buying rather than developing solutions.
- **Business intelligence support**: DB2 UDB has particular strengths in supporting business intelligence applications such as data warehousing and on-line analytical processing (OLAP). DB2 provides parallel database technology and query optimization resulting in the ability to help customers find competitive advantage, offer better customer service, or reduce costs by mining their data for the knowledge required to make better decisions. Further, this feature does not require the additional expense of a specialized database. DB2 UDB provides a single database that can be used across an enterprise for all data management requirements from OLAP to OLTP.
- **Ease of use and management**: DB2 UDB is an easy database to set up, use and manage. It includes a complete suite of GUI administration tools that allow for easy installation, administration and remote operation. It also includes programmer-friendly tools to get an application up-and-running quickly, and user-friendly tools to help make end-users productive.
- **Universal access**: DB2 UDB can be accessed from many client workstations over many networks. Through built-in data replication and distributed transactions, it provides you with the flexibility of placing data virtually anywhere in your network, therefore providing optimum service and productivity. Further, DB2 provides efficient and seamless integration of data on mainframe and midrange data servers, enabling you to reduce costs and improve cycle-times by leveraging your current investments in data, hardware, software, and skills.
- **Multi-platform support**: DB2 UDB is an open database. It runs on popular UNIX and Intel server platforms. It supports major industry standards relevant to distributed data so that it can be accessed using thousands of existing tools and applications, and can be easily managed within an open, network computing environment. These capabilities enable you to reduce costs and improve cycle-times by leveraging your current investments in data, hardware, software, and skills.
- **Reliability**: DB2 UDB is setting the standard for quality and reliability in the client/server database industry. As more mission-critical applications are implemented on UNIX and Intel platforms, IBM's ability to bring mainframe-level reliability to this environment has become a major factor in choosing DB2. Better reliability and availability can potentially reduce your costs, while scalability both within and across platforms can potentially enhance the productivity of projects.

- **Key values and benefits**: In summary, with DB2 Universal Database you can:
  - Support applications from business intelligence to transaction processing with a single high-value database enabling you to minimize costs, leverage the skills of your support personnel, and maximize your return on investment.
  - Web-enable enterprise data for your intranet or the Internet, enabling you to go to market faster, gain competitive advantage and increase customer satisfaction.
  - Extend applications with new function, or scale them for more users or data, enabling you to increase productivity and provide more useful information.

1.2 Version 8 highlights

With the launch of DB2 UDB Version 8.1, IBM introduced over 400 new features[1]. Here are the new features introduced by Version 8.1:

- **Innovative manageability**: DB2 Version 8.1 provides significant autonomic capabilities including self-configuring, self-optimizing, and self-managing capabilities.
  - Configuration Advisor puts the knowledge of a seasoned DBA at your fingertips.
  - The Health Center/Monitor keeps your database functioning with a minimum of manual intervention. The Health Center allows an administrator to define automatic actions for some problems, or to be sent system generated messages for issues that may need manual intervention.
  - The Memory Visualizer lets you dynamically see and control DB2's memory usage. This graphical display also supplies information to the DB2 monitoring facilities.
  - The enhanced Design Advisor delivers expert advice on index based performance improvements.
  - The new db2support command packages together the most common diagnostic information required by IBM support personnel.
  - The creation and management of large-scale partitioned databases has been simplified by incorporating the partitioning functionality of DB2 Extended Enterprise Edition (EEE) into the base product. With one

[1] For greater detail of the specific enhancements, see *IBM DB2 Universal Database What's New Version 8* available at: http://www-3.ibm.com/software/data/db2/udb/pdfs/db2q0.pdf

Control Center you can now manage multiple flavors of DB2 across multiple platforms.

- **New levels of integrated information**: DB2 Version 8.1 helps solve critical business problems by integrating information across the entire enterprise by leveraging the federated Web Services and XML.
  - DB2 is delivering new federated capabilities that enable customers to integrate information as Web Services.
  - DB2 also delivers new XML enhancements that make it easier for programmers to integrate DB2 and XML information.
- **Robust e-business foundation**: DB2 Version 8.1 performance, scalability, and availability enhancements continue with cross workload and cross platform leadership, improving overall application performance and making information highly available. A key constituent of this is to eliminate 85% of the disruptions normally associated with planned database maintenance.
  - Connection Concentrator provides greater user scalability by allowing individual transactions for multiple clients to run over the same physical database connection. Whereas previously one database agent was created for each connection, now a physical database agent can be used by multiple logical agents, each representing a client connection.
  - Dynamic configuration of many configuration parameters allows most changes to be made without the need to stop the database or the instance.
  - In-place (online) reorganization allows DB2 tables to be reorganized online without the need to create a copy of the table.
  - Online index creation and reorganization uses a copy of the existing index, while the new one is being created/reorganized. At the end of the create/reorg, a switch is made to the new index and the old one is dropped.
  - Online load allows the high speed bulk loading of data into DB2 in combination with the ability to continue to read the existing data from the table. Loading from within an SQL statement is also supported through the use of a new *cursor* file type.
  - Online storage management increases the available options to the DBA who wishes to modify the database storage. The capability now exists to drop containers from the tablespace as well as reduce the size of these containers.
  - Null and default compression can significantly reduce the disk footprint of a large DB2 system.
  - *Type 2* indexes have been introduced in Version 8. These introduce the concept of the *pseudo delete* to DB2 UDB for Multiplatforms. This feature

  means that when a row is deleted, the index key is marked as *pseudo deleted* and will be physically deleted during a period of low database activity.
  - The new client architecture of DB2 greatly simplifies connectivity between DB2 servers and clients. Now one DB2 *common client* can connect to both 32-bit and 64-bit servers. This combines the benefit of a greater memory usage associated with 64-bit servers with the proliferation of 32-bit clients.

- **Integrated business intelligence**: DB2 Version 8.1 allows customers to easily organize stored information to perform faster and more insightful queries.
  - Multidimensional data clustering (MDC) improves performance of complex queries by physically storing data in more than one dimension. This MDC clustering is achieved by placing data rows into page extent blocks based on their dimensional key values. The management, placement and access are facilitated through a new *Block Index* object type. A Block Index is created for each of the dimensions, and although they are similar in structure to a normal index they point to rows to a dimensional data block rather than individual rows.
- **Enhanced application development productivity**: Building upon DB2's leadership in open standards, IBM delivers a new Development Center that makes it easier for developers to build and deploy applications for either Java or Microsoft environments.

1.3 DB2 database objects

In this section we introduce the database objects that are used in a DB2 system. Throughout this book we will refer to DB2 database objects; here we introduce the most common ones.

Instances

An instance (sometimes called the database manager) is DB2 code that manages data. It controls what can be done to the data, and manages system resources assigned to it. Each instance is a complete environment. It contains all the database partitions defined for a given parallel database system. An instance has its own databases (which other instances cannot access), and all its database partitions share the same system directories. It also has separate security from other instances on the same machine (system). This can allow two completely independent development and test environments.

Databases

A relational database presents data as a collection of tables. A table consists of a defined number of columns and any number of rows. Each database includes a set of system catalog tables that describe the logical and physical structure of the data, a configuration file containing the parameter values allocated for the database, and a recovery log with ongoing transactions and archivable transactions.

Database partition groups

A database partition group is a set of one or more database partitions. When you want to create tables for the database, you first create the database partition group where the tablespaces will be stored, then you create the tablespace where the tables will be stored. This allows tables and indexes to be split across multiple database partitions which can be logical (in one SMP machine) or physical (across multiple physical servers). In a partition group with one partition (a non clustered database) this feature is transparent and only one implicit partition group exists.

Tablespaces

A database is organized into parts called tablespaces. A tablespace is a space to store tables. When creating a table, you can decide to have certain objects, such as indexes and large object (LOB) data kept separately from the rest of the table data. A tablespace can also be spread over one or more physical storage devices (containers).

In a partitioned environment tablespaces reside in database partition groups. Tablespace definitions and attributes are recorded in the database system catalog.

These objects are associated with the tablespace:

- **Pages:** Within the tablespace, data is physically stored on pages. These data pages can be 4K, 8K, 16K and 32K. The page size limits the row length and column count of tables that can be placed in the tablespace. Furthermore tablespaces are limited to containing 16384 pages, so choosing a larger page size will increase the capacity of the tablespace.

  > **Note:** Each tablespace can only be one page size and requires a buffer pool of the same page size.

- **Extent size:** Specifies the number of pages that will be written to a container before moving to the next container. The database manager cycles repeatedly through the containers as data is stored. This parameter has effect only when there are multiple containers for the tablespace.

- **Prefetch size:** Specifies the number of pages that will be read from the tablespace when data prefetching is being performed. Prefetching reads in data to the buffer pool prior to its being directly referenced by the query. This allows greater performance as the relevant data should already have been prefetched from disk into the bufferpool.
- **Type of storage:** A tablespace can be either system managed space (SMS), or database managed space (DMS).
  - For an SMS tablespace, each container is a directory in the file space of the operating system, and the operating system's file manager controls the storage space.
  - For a DMS tablespace, each container is either a fixed size pre-allocated file, or a physical device such as a disk, and the database manager controls the storage space.

Tables

A relational database presents data as a collection of tables. A table consists of data logically arranged in columns and rows. All database and table data is assigned to tablespaces. The data in the table is logically related, and relationships can be defined between tables. Data can be viewed and manipulated based on mathematical principles and operations called relations. Tables are grouped together by extents in the tablespaces that they are created in.

Indexes

An index is a set of keys, each pointing to rows in a table. An index allows more efficient access to rows in a table by creating a direct path to the data through pointers. The SQL optimizer automatically chooses the most efficient way to access data in tables. The optimizer takes indexes into consideration when determining the fastest access path to data. Unique indexes can be created to ensure uniqueness of the index key. An index key is a column or an ordered collection of columns on which an index is defined. Using a unique index will ensure that the value of each index key in the indexed column or columns is unique.

DB2 allows the use of clustered indexes to organize data to make processing that table more efficient, this capability is offered in two ways:

- **Clustered indexes**: For a clustered index DB2 maintains the rows in the same sequence as the columns in the index for as long as there is free space. Clustering indexes greatly improves the performance of range queries that have predicates containing one or more keys of the clustering index. With good clustering, only a portion of the table needs to be accessed and, when the pages are sequential, more efficient prefetching can be performed.

- **Multi-Dimensional Clustering (MDC)**: With MDC the benefits of clustering indexes are extended to more than one dimension, or clustering key. In terms of query performance, range queries involving any combination of specified dimensions of the table will benefit from clustering.

  Furthermore, although a table with a clustering index can become unclustered over time as space fills up in the table, an MDC table is able to maintain its clustering over all dimensions automatically and continuously, therefore eliminating the need to reorganize the table to restore the physical order of the data.

Figure 1-1 shows the relationship of the DB2 objects discussed.

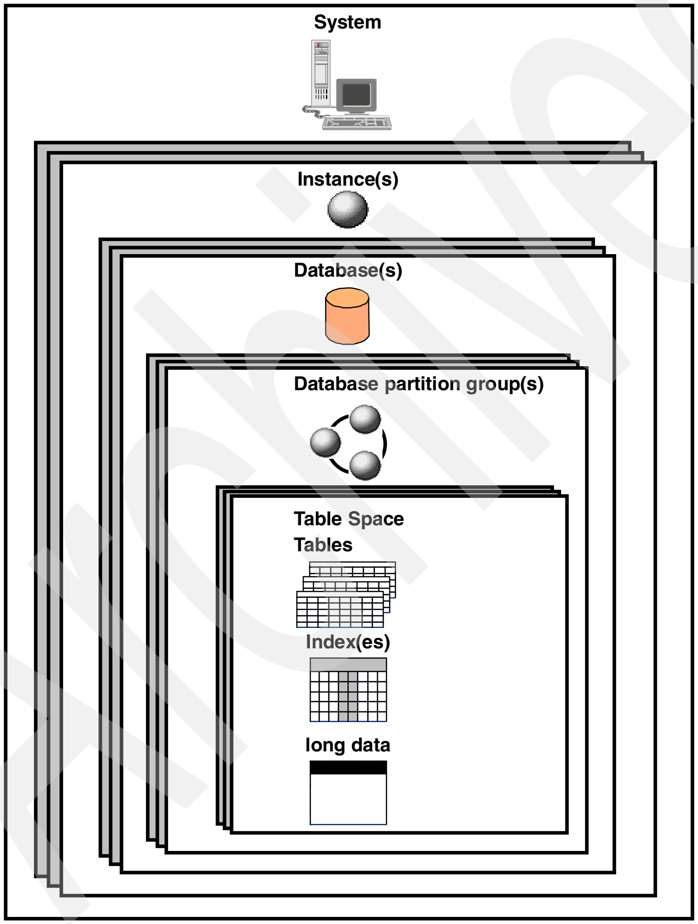


Figure 1-1 Relationship between DB2 database objects

Buffer pools

A buffer pool is the amount of main memory allocated to cache table and index data pages as they are being read from disk, or being modified. The purpose of the buffer pool is to improve system performance. Data can be accessed much faster from memory than from disk; therefore, the fewer times the database manager needs to read from or write to a disk (I/O), the better the performance.

Schemas

A schema is an identifier, such as a user ID, that helps group tables and other database objects. A schema can be owned by an individual, and the owner can control access to the data and the objects within it.

A schema is also an object in the database. It may be created automatically when the first object in a schema is created. Such an object can be anything that can be qualified by a schema name, such as a table, index, or view.

A schema name is used as the first part of a two-part object name. When an object is created, you can assign it to a specific schema. If you do not specify a schema, it is assigned to the default schema, which is usually the user ID of the person who created the object. The second part of the name is the name of the object. For example, a user named ALAN might have a table named ALAN.PARTRIDGE

System catalog tables

Each database includes a set of system catalog tables, which describe the logical and physical structure of the data. DB2 creates and maintains an extensive set of system catalog tables for each database. These tables contain information about the definitions of database objects, such as user tables, views, and indexes, as well as security information about the authority that users have on these objects. They are created when the database is created, and are updated during the course of normal operation. You cannot explicitly create or drop them, but you can query and view their contents using the catalog views.

1.4 DB2 UDB architecture and processes

DB2 UDB is a robust database management system. Figure 1-2 provides an overview of the architecture and processes of DB2 UDB. From a client-server view, DB2 UDB separates the client code and the server code into different address spaces. The application code runs in the client process while the server code runs in separate processes. Each client application links with the DB2 client library, and communicates with the DB2 Server using shared memory (local clients), or a communication protocol such as TCP/IP (remote clients). A description of the key components follows.

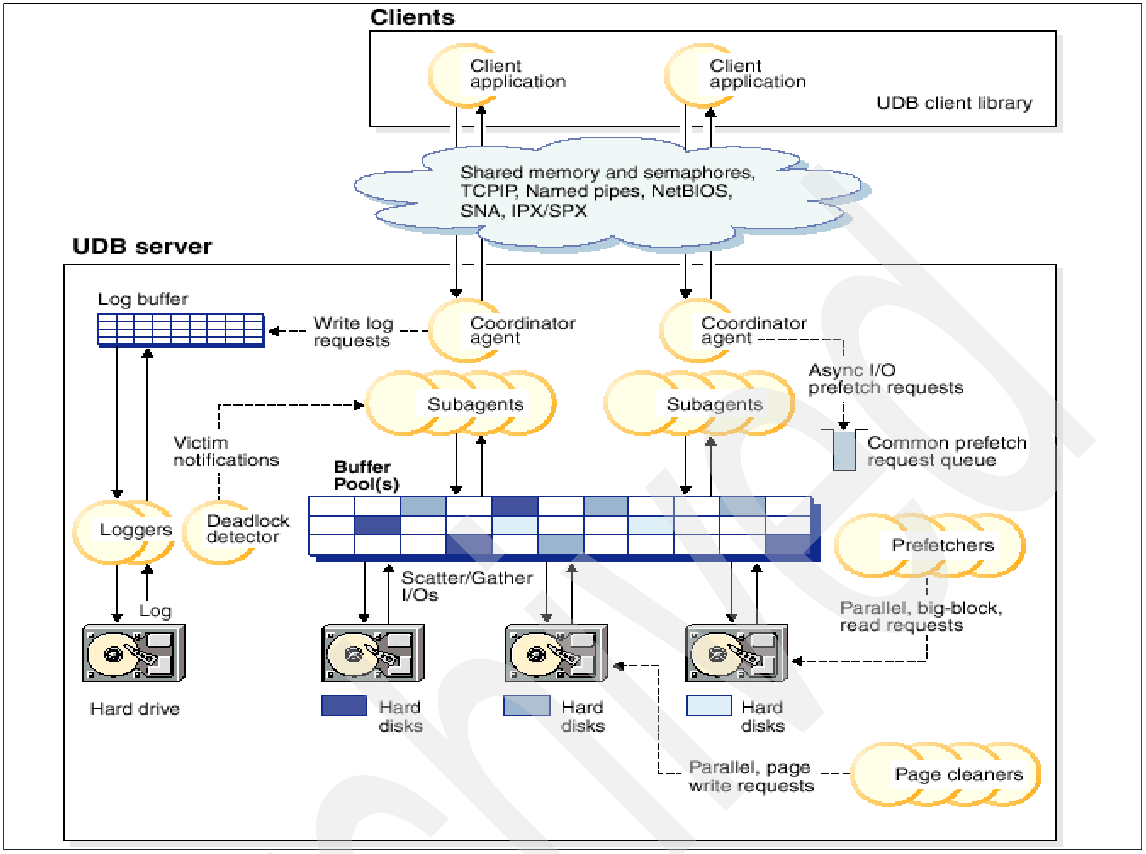


Figure 1-2 DB2 UDB architecture overview

On the server side, activity is controlled by Engine Dispatchable Units (EDUs). In Figure 1-2, EDUs are shown as circles or groups of circles. EDUs are implemented as threads in a single process on Windows-based platforms and as processes on UNIX. DB2 agents are the most common type of EDUs but other examples of EDUs include coordinator agents, subagents, loggers, deadlock detectors, and prefetchers.

- **DB2 agents:** DB2 agents include coordinator agents and subagents, and are the most common type of DB2 processes which carry out the bulk of SQL processing on behalf of applications. DB2 assigns a coordinator agent with an application, and this agent coordinates the communication and processing for this application. Depending on the nature of the application, Online Transaction Processing (OLTP) or Decision Support Systems (DSS), DB2 can be configured to assign subagents to assist the coordinator agents with the application requests.

- **Prefetchers:** Prefetchers are present to retrieve data from disk and move it into the buffer pool before applications need the data. Without prefetchers applications needing to scan through large volumes of data would have to wait for data to be moved from disk into the buffer pool. Their effectiveness is increased by having multiple disks in a DB2 tablespace. The data is striped across multiple disks and enables the prefetchers to access multiple disks in parallel to retrieve data.
- **Page cleaners:** Page cleaners are present to make room in the buffer pool, before agents and prefetchers read pages from disk storage and move them into the buffer pool. For example, if an application has updated a large amount of data in a table, many of the updated data pages in the buffer pool may not yet have been written on to disk storage — such pages are called dirty pages. Since prefetchers cannot place fetched data pages on to the dirty pages in the buffer pool, these dirty pages must first be flushed to disk storage and become “clean” pages, so that prefetchers can find room to place fetched data pages from disk storage.
- **Loggers:** Changes to data pages in the buffer pool are logged. Agent processes updating a data record in the database update the associated page in the buffer pool, and write a log record into a log buffer. To optimize performance, neither the updated data pages in the buffer pool, nor the log records in the log buffer are written to disk immediately. They are asynchronously written to disk by page cleaners, and the logger, respectively. The logger and the buffer pool manager cooperate and ensure that an updated data page is not written to disk storage before its associated log record is written to the log.
- **Deadlock detector:** A deadlock occurs when one or more applications require access to a resource that is currently locked by the other application(s). This can result in interminable waits by all the applications involved in the deadlock. To avoid such a situation, DB2 uses a background process called the deadlock detector to identify and resolve these deadlocks. When the deadlock detector encounters a deadlock situation, one of the deadlocked applications will receive an error code and the current unit of work for that application will be rolled back automatically by DB2. When the rollback is complete, the locks held by this chosen application are released, thereby allowing other applications to continue.

1.5 Accessing and updating DB2 data

Now that we have introduced the DB2 architecture and terminology, here we discuss the operations within DB2 that occur when a select and an insert take place. This will aid your understanding of how the architecture of DB2 services the requirements of applications.

Selecting data from DB2

This is the SELECT scenario:

1. An application issues a connect database statement to connect remotely to a DB2 database through a DB2 client.
2. When this request reaches the server, a coordinating agent is assigned by DB2 which then coordinates the communication and processing for this application.
3. The application then issues a SELECT statement to query data from a DB2 table. The query is checked for accuracy, optimized through the DB2 Query Optimizer and made into an executable statement.
4. The query is `Select * from BRENT.OFFICE` and no indexes exist on this table. This means that a table scan will be necessary to retrieve the appropriate data. With table scans DB2 can easily determine that prefetching the data in sequence will improve I/O performance. In our example DB2 automatically starts to sequentially prefetch the data from the tablespaces into the buffer pool.
5. The database agent performs the necessary operations on the data pages in the buffer and returns the result to the user application.
6. While this is happening the DB2 page cleaners are monitoring the buffer pool and if space is required they write pages back to the tablespaces on the disks.

Inserting data into DB2

This is the INSERT scenario:

1. An application issues a query to connect remotely to DB2 through a DB2 client.
2. When this request reaches the server a coordinating agent is assigned by DB2, which then coordinates the communication and processing for this application.
3. The insert is checked for accuracy, optimized through the DB2 Query Optimizer and made into an executable statement.
4. The insert is being made on a table that has a single clustering index.
5. In this case, the coordinator agent finds the correct data page and attempts to insert records on the same page as other records with similar index key values.
6. If there is no space on the same data page then the agent searches the Free Space Control Records (FSCRs), which are one page control blocks embedded within the data pages, for a page with enough free space to accommodate the row to be inserted.

7. If no space is found after searching a customizable amount of FSCRs then the new row is appended to the end of the table.
8. When the insert updates the associated page in the buffer pool, it writes a log record. The log record contains the information necessary to either redo or undo the change. Neither the page in the buffer pool nor the log record in the log buffer are written to disk immediately to optimize performance. The logger EDU and the buffer pool manager cooperate to implement a Write Ahead Logging (WAL) protocol that ensures that the data page is not written to disk before its associated log record is written to the log. The WAL protocol ensures that there is always enough information in the log to recover from a crash and to restore database consistency.

> **Note:** *DB2 UDB Administration Guide: Performance*, SC09-4821, provides detailed information about DB2 architecture and objects.

2

Installation

In this chapter we guide you through the process of obtaining and installing the DB2 V8.1 trial code. The installation covers both the Linux and Windows platforms. The emphasis is on using the graphical DB2 Setup wizard to install and configure DB2. We also cover the additional methods available to install DB2, namely the db2_install command and using DB2 response files. We install and configure the DB2 trial code on a stand-alone DB2 Server. As part of this installation the Administrative Tools and Development Tools will be installed.

This chapter discusses these topics:

- Obtaining the DB2 UDB trial code
- Installing DB2 on Linux and Windows, concerning:
  - Installation requirements
  - Installation preparation
  - Installation process using the DB2 Setup wizard
  - Post installation tasks
- Advanced installation options for both Linux and Windows

2.1 Obtaining the DB2 trial code

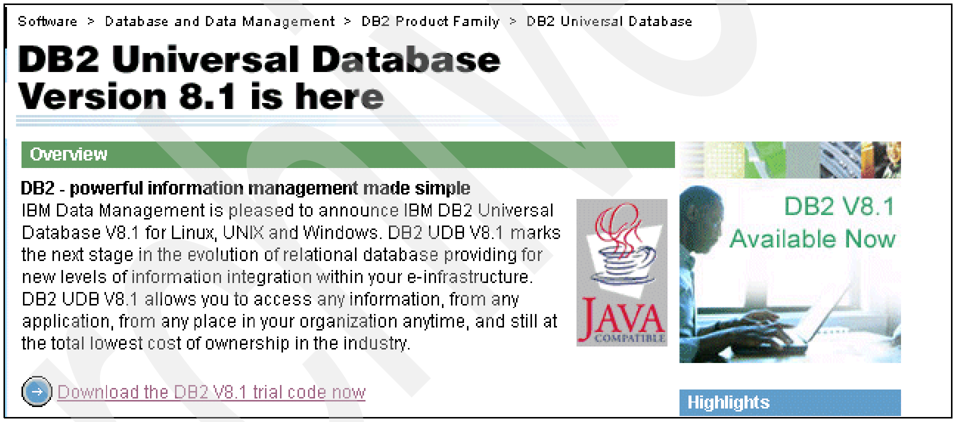
IBM provides free DB2 trial code for the AIX®, Linux, and Windows platforms. This section details how to obtain the Linux and Windows versions of the DB2 trial code. The Web site below offers access to free downloads of DB2 UDB Version 8.1 for UNIX and Windows.

http://ibm.com/db2/v8

2.1.1 Downloading DB2

You want to download either the Linux or the Windows version of DB2. To select the code you want to download:

- Open the DB2 Version 8.1 Web site: http://ibm.com/db2/v8
- Select **Download the DB2 V8.1 trial code now**.



Software > Database and Data Management > DB2 Product Family > DB2 Universal Database

DB2 Universal Database Version 8.1 is here

Overview

DB2 - powerful information management made simple
IBM Data Management is pleased to announce IBM DB2 Universal Database V8.1 for Linux, UNIX and Windows. DB2 UDB V8.1 marks the next stage in the evolution of relational database providing for new levels of information integration within your e-infrastructure. DB2 UDB V8.1 allows you to access any information, from any application, from any place in your organization anytime, and still at the total lowest cost of ownership in the industry.

Download the DB2 V8.1 trial code now

Figure 2-1 Download DB2 trial code

- Select the code you want to download:
  - Linux

    `DB2 Enterprise Server Edition for Linux (32 Bit)`
  - Windows

    `DB2 Enterprise Server Edition for Windows NT/2000 (32/64 Bit)`

  And, select the language of your choice. Then click **Continue.**
- Log in:

  If you already have an IBM download ID:
  - Enter your ID and password.

  - Click **Sign In**.
  - Click **Accept License**.
  - You will be automatically directed to the download page.

  If you don’t have an IBM download user ID:

  - Click the **I have not registered before** hyper link.
  - Enter your details.
  - Click **Accept License**.
  - If you have chosen a unique user name, you will be automatically directed to the download page.
  - If not, you can change to the recommended name, and then click **Continue**.

  If you have forgotten your user ID or password:

  - Click **I have forgotten my user id or password**.
  - This window offers you the chance to use a combination of your ID/e-mail address to send your password or last name/e-mail address to send your user ID.

- Choose to download DB2 via FTP or HTTP or via Download Director. We recommend using the Download Director, because it downloads using multiple connections which should be quicker.
- Choose a location to save the `009_ESE_LNX_32_NLV.tar` file (Linux) or the `DB2_V81_ESE_WIN_32_NLV.exe` (Windows).

  Ensure you have 480 MB to 550 MB available for each of these download images.

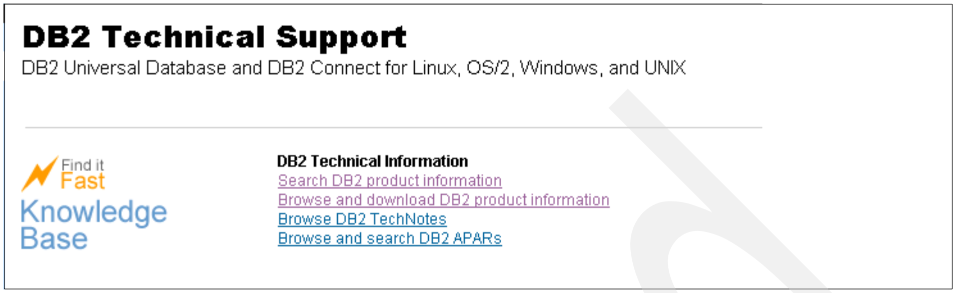
2.1.2 Downloading the DB2 documentation

DB2 Version 8.1 has changed the way that the documentation and manuals are provided. As part of the installation process of the prior version of DB2, the HTML versions of the manuals were included as part of the base install. Now when you buy DB2 Version 8 the documentation is provided on a separate CD-ROM which allows for easier updates of the information.

With the DB2 trial code, the documentation is provided in two ways.

1. When you install DB2, the Information Center is installed with initial documentation that covers:
   - Assistance with the DB2 GUIs
   - DB2 Glossary of terms
   - How to carry out many day-to-day tasks
   - Link to the DB2 online support Web site (Figure 2-2)

`http://www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/index.d2w/report`



DB2 Technical Support

DB2 Universal Database and DB2 Connect for Linux, OS/2, Windows, and UNIX

DB2 Technical Information

- Search DB2 product information
- Browse and download DB2 product information
- Browse DB2 TechNotes
- Browse and search DB2 APARs

Figure 2-2 DB2 Technical Support Web site

2. From the DB2 online support Web site you can either:
   - **Search DB2 product information**: This option allows you to search the entire DB2 V8 Information Center online as well as allowing you to restrict your search by selecting to only search certain groups of publications.
   - **Browse and download DB2 product information**: This option allows you to:
     - Launch and search the Information Center.
     - Download the Information Center for a local installation (subject to having bought the V8 product).
     - Buy the hardcopy DB2 manuals.

     There is also the facility to search or download PDF versions of the DB2 manuals. If you decide to download some of these manuals, but not all, then throughout the book we reference those manuals which are of particular interest. However, it is worth downloading the following manuals, because experience indicates that these are the most widely used:

     - Command Reference
     - SQL Reference, Vol. 1
     - SQL Reference, Vol. 2
     - Administration Guide: Implementation
     - Application Development Guide: Programming Server Applications

2.2 Installing DB2 on Linux

In this section we describe how to install DB2 trial code Enterprise Server Edition (ESE) on a Linux machine. In our lab environment we installed DB2 on Red Hat Advanced Server V8 and on SuSE Linux Enterprise Server V8.

2.2.1 Installation requirements

Before installing DB2 Version 8.1 ESE on Linux, make sure that the system targeted for installation meets the following minimum requirements.

- **Hardware requirements**:
  - DB2 Workgroup Server Edition is supported on Intel 32-bit machines.
  - DB2 Enterprise Server Edition is supported on:
    - Intel 32-bit
    - Intel 64-bit
    - S/390® 9672 Generation 5 or higher, Multiprise® 3000 or eServer™ z-Series
- **Distribution requirements**: For the latest information on supported distribution and kernel levels, point your browser to:
  `http://www.ibm.com/db2/linux/validate`

  We have performed our evaluation using SuSE Linux 8.1 - Professional and Red Hat Linux 8.0 - Professional.
- **Software requirements**: The IBM Developer Kit for Java 1.3.1 Service Release 1 is required for DB2 servers, for using the DB2 Control Center, and for creating and running Java applications, including stored procedures and user-defined functions. During the installation process, if the correct level of the JDK is not already installed, it will be installed.
- **A browser** is required to view online help.
- **Communication requirements**: TCP/IP is required to access remote databases.

  The following configurations are *not* supported:
  - Installing DB2 products or sharing instance directory on NFS. Currently, we do not support the installation of DB2 products on NFS. Installing DB2 on NFS (for example, NFS mounting /usr/opt/db2_08_01 or /opt/IBM/db2/V8.1) can be error prone and these errors can be difficult to diagnose.
  - Setting up an instance on a file system.
  - NFS mounting a file system from multiple computers, and then running DB2 on these machines using that same instance. This configuration can cause file locking and performance problems.
- **Memory requirements**:
  - A minimum of 256 MB of RAM is required.
  - Additional memory may be required based on your specific environment.
- **Disk requirements**:

  - The DB2 code is installed into the `/opt` directory or file system which should have a minimum of 700 MB of free space available for this purpose.
  - Additional space is required by each DB2 instance.

2.2.2 Installation preparation

After placing DB2_v81_ESE_LNX_32_NLV.tar in a convenient location, unpack the contents with:

```
tar -xvf DB2_v81_ESE_LNX_32_NLV.tar
```

The directory *009_ESE_LNX_32_NLV* will be created.

DB2 requires three user IDs and groups:

- The DB2 instance user
- The DB2 fenced user
- The Administration Service user

In a Linux installation, DB2 will create these IDs for you if you use the DB2 Setup installation wizard. User IDs on UNIX-based systems can contain up to eight characters.

You also can create these IDs and groups before installation and use the IDs you created during the installation. This would be necessary if you wanted to use the methods described in 2.4, “Advanced installation” on page 42.

2.2.3 Installing DB2

Now you can begin the DB2 installation.

- Login as the root and change directory to the 009_ESE_LNX_32_NLV directory that was created by tar command in the previous section.
- To use the DB2 Setup install wizard, your desktop environment must be capable of displaying an X windows application. In a Linux terminal window, enter the following command to setup X windows:

  ```
  xhost + host_name
  export DISPLAY=host_name:0
  ```

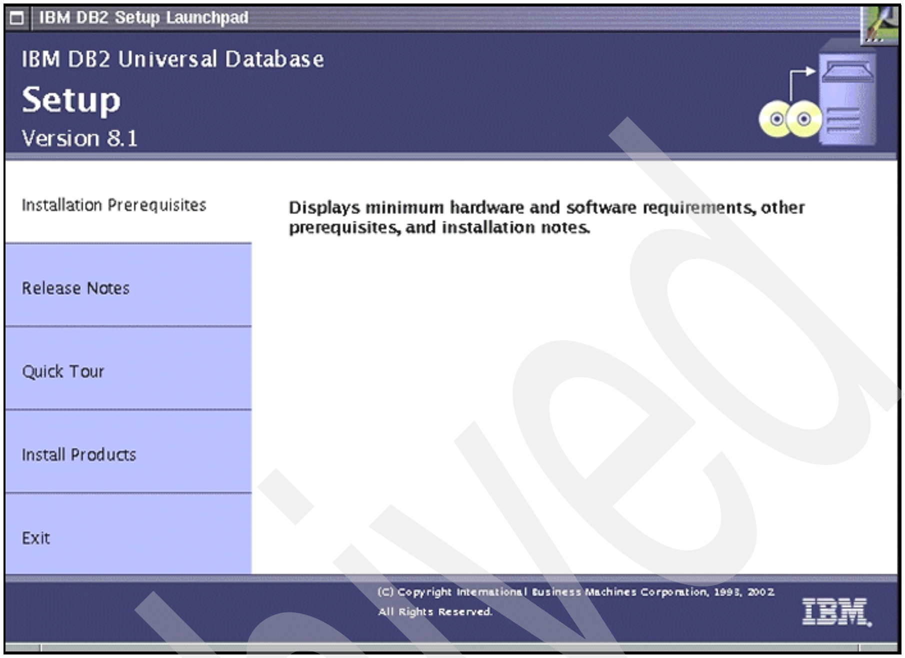
  In our environment:

  ```
  xhost + udbsrv04
  export DISPLAY=udbsrv04:0
  ```

- Launch the installation process by typing in:

  ```
  ./db2setup
  ```

The graphical DB2 Setup wizard appears (Figure 2-3).

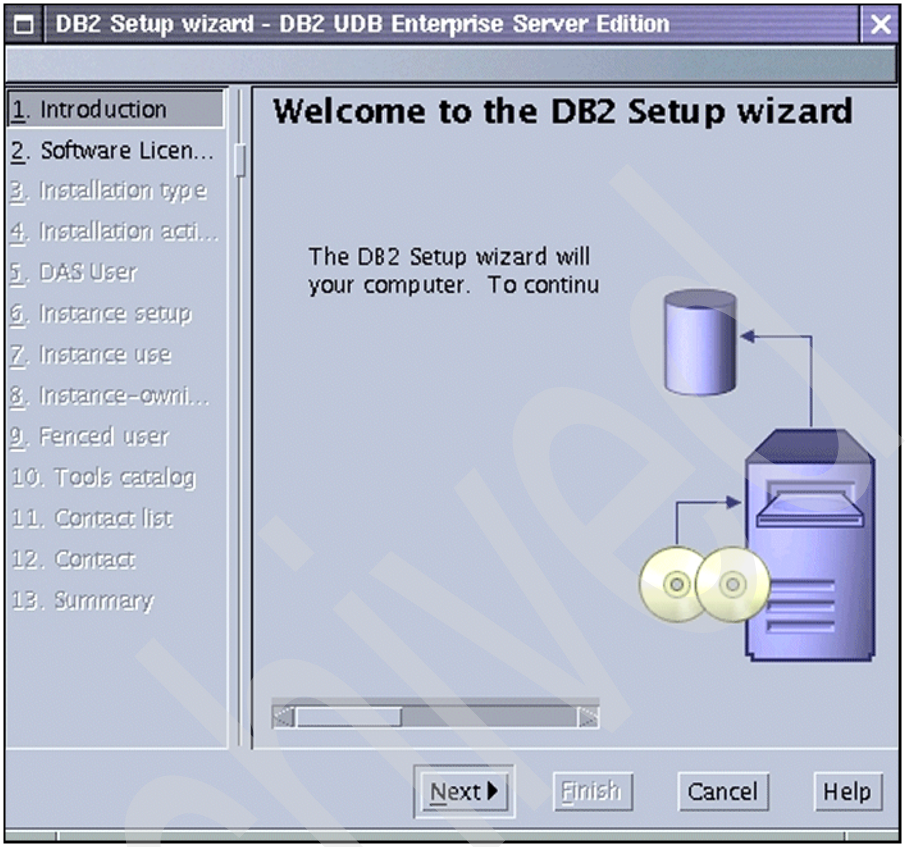


Displays minimum hardware and software requirements, other prerequisites, and installation notes.

(C) Copyright International Business Machines Corporation, 1993, 2002
All Rights Reserved.

Figure 2-3 DB2 Setup wizard on Linux

- Click **Install Products** and then select **DB2 UDB Enterprise Server Edition**.
- The DB2 Setup wizard will appear outlining the installation steps to follow (Figure 2-4).



Welcome to the DB2 Setup wizard

The DB2 Setup wizard will
your computer. To continu

Figure 2-4 Steps to install DB2 UDB Enterprise Server using DB2 Setup wizard

1. **Introduction**

   The wizard will now guide you through a sequence of steps to setup your DB2 server environment. Click **Next** to proceed.

2. **Software license Agreement**

   In this section you will be asked to read and accept the licence for the DB2 program in order to proceed with the installation process. Select **Accept** and click **Next.**

3. **Select installation type**

   You have a choice of three installation types:

   – Typical (480-550 MB) installs most DB2 components except for Application development and Business Intelligence tools.

   – Compact (400-490MB) installs only basic DB2 features and functionality.

- Custom (400-1210MB)

  This allows you install whatever components you want. An advantage of the Custom option is that it allows you to install the Application Development tools with DB2 UDB. Note: With the Typical option, the Application Development tools requires a separate installation.

We chose the **Typical** install for our evaluation. *Do not* select the **Data Warehousing additional function** (this can be installed later by modifying the DB2 install). Click the **View Features** button to see what components will get installed.

Choose an installation type and click **Next** to proceed.

4. **Installation action**

   During the installation you have the option of saving your settings in a response file which can be used to install DB2 at a later time with no further input. This is useful in a partitioned server environment and multiple system deployment.

   Even though we are installing through the DB2 Setup wizard, we suggest you also select **Save your settings in a response file**. This will allow you to examine the response file DB2 created for you later, and also perform a silent install at a later date. Click **Next**.

5. **DAS user**

   The DB2 Administration service runs in the background to enable remote GUI tools to connect to DB2 and also to enable the scheduling of tasks.

   You have the option of having DB2 Setup create a Linux operating system userid, password, and group, or of using an existing user for the DB2 administration server.

   We chose the default *dasusr1* for the DAS user and entered a password of `db24me`.

   Enter a user and password, and click **Next**.

6. **Instance setup**

   An instance (also called the database manager) is an environment where you create and manage DB2 databases. You can choose whether or not to have DB2 Setup create an instance. We chose **Create a DB2 instance**.

   This task can be deferred until the installation is complete and then later use the create instance DB2 command *db2icrt* to create DB2 instance.

   Make your choice and click **Next.**

7. **Instance use**

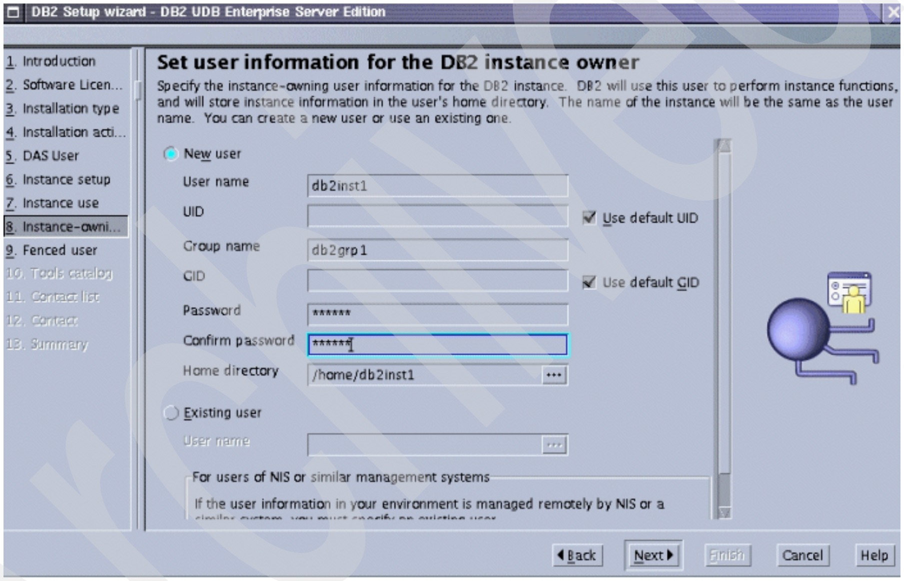
   We do not require the partitioning capability of DB2, because our application does not span multiple physical machines and does not contain vast data volumes.

   Choose **Single-partition instance** and click **Next.**

8. **Instance-owning user**

   This is the user who will have full control over the DB2 instance. You can choose to have DB2 Setup create a user ID, password and group for the instance or use an existing user.

   We chose to accept all the defaults for user name, UID, group name, and GID (Figure 2-5).



DB2 Setup wizard - DB2 UDB Enterprise Server Edition

Set user information for the DB2 instance owner

Specify the instance-owning user information for the DB2 instance. DB2 will use this user to perform instance functions, and will store instance information in the user's home directory. The name of the instance will be the same as the user name. You can create a new user or use an existing one.

● New user

User name: db2inst1

UID: ☑ Use default UID

Group name: db2grp1

GID: ☑ Use default GID

Password: ******

Confirm password: ******

Home directory: /home/db2inst1

○ Existing user

User name:

For users of NIS or similar management systems

If the user information in your environment is managed remotely by NIS or a

Figure 2-5 Creating an instance owner

   We entered a password of `db24me`.

   Enter a User name, UID, Group name, GID, and password, and click **Next**.

9. **Fenced user**

   You can choose to have DB2 Setup create a user ID, password, and group to allow functions and stored procedures to run securely in a separate process from the DB2 engine.

   We let DB2 Setup create the default `db2fenc1` user and `db2gfrp1` group. Again, we entered `db24me` as our password.

Enter a user, group, and password, and click **Next** to proceed.

10. **Tools catalog**

    Before you can use certain tools, such as the Task Center you must create the DB2 tools catalog.

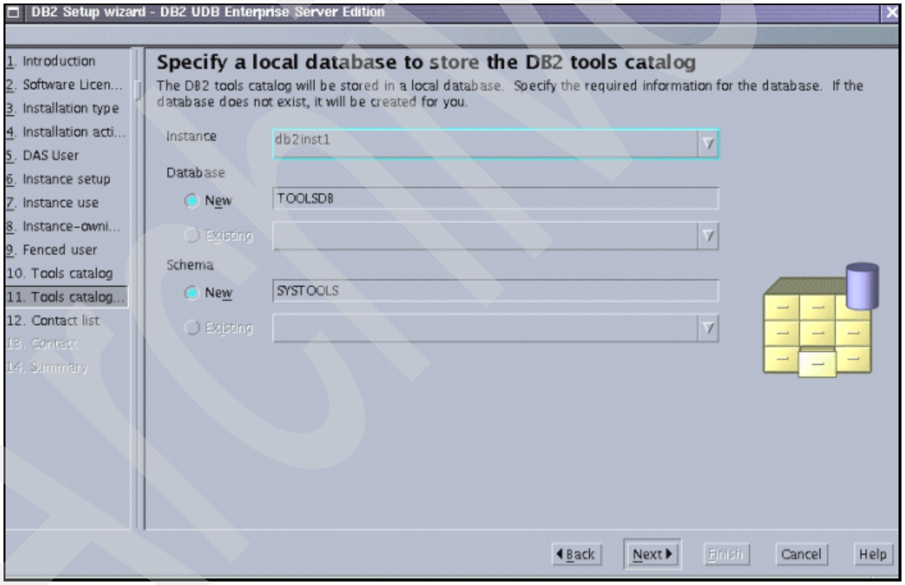
    We will create a local tools catalog.

    So select **Use a local database** and click **Next**.

    > **Note:** In a multi database environment you may select **Do not prepare the tools catalog** on this computer, if you have already defined a system that you want to use for central scheduling.

11. **Tools catalog database**

    We chose to use the default local database *TOOLSDB* in the *SYSTOOLS* schema for this purpose (Figure 2-6). Specify a database and click **Next**.



DB2 Setup wizard - DB2 UDB Enterprise Server Edition

Specify a local database to store the DB2 tools catalog

The DB2 tools catalog will be stored in a local database. Specify the required information for the database. If the database does not exist, it will be created for you.

Instance: db2inst1

Database
- ● New: TOOLSDB
- ○ Existing

Schema
- ● New: SYSTOOLS
- ○ Existing

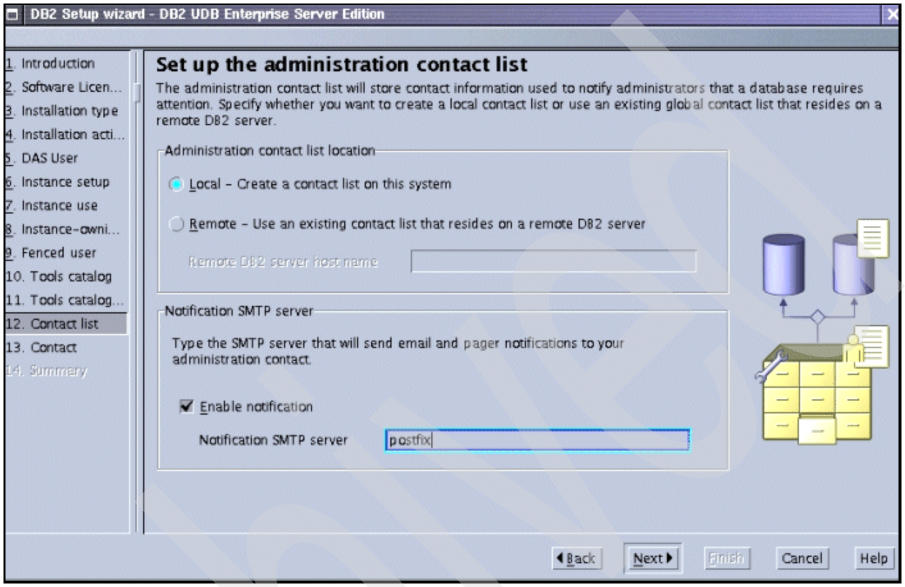
Figure 2-6 Creating a local DB2 tools database

12. **Contact List**

    You can create an administration contact list on this server or use one which resides on a remote DB2 server and specify the SMTP server which will send out email and pager notifications.

We selected to use a **Local** contact list location and **postfix** as our Notification SMTP (Figure 2-7).

If you don't have an SMTP server then just select **Local** for the administration contact location. Click **Next**.



Set up the administration contact list

The administration contact list will store contact information used to notify administrators that a database requires attention. Specify whether you want to create a local contact list or use an existing global contact list that resides on a remote DB2 server.

Administration contact list location

● Local - Create a contact list on this system

○ Remote - Use an existing contact list that resides on a remote DB2 server

Remote DB2 server host name

Notification SMTP server

Type the SMTP server that will send email and pager notifications to your administration contact.

☑ Enable notification

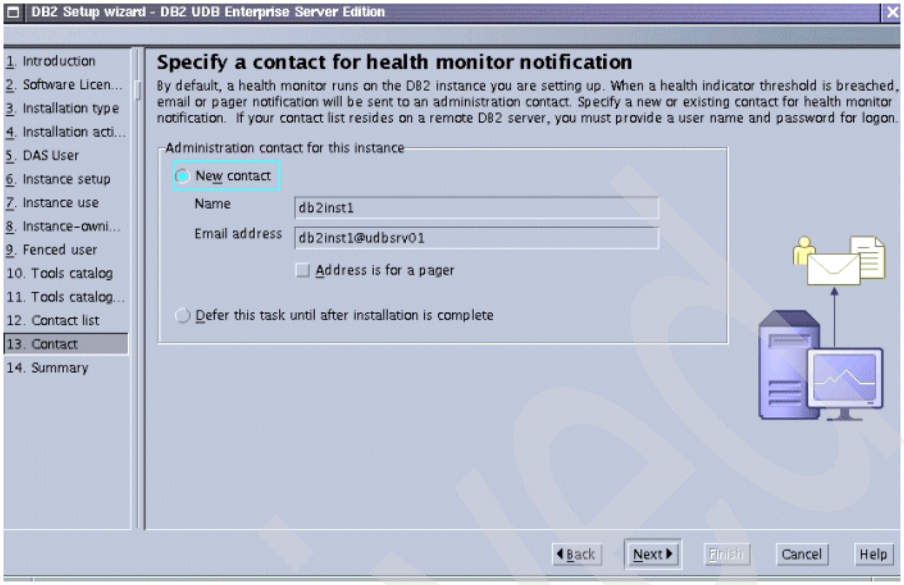
Notification SMTP server: postfix

Figure 2-7 Configuring the administration contacts

13. Contact

You may choose a name and email address for Health Monitor notification. We entered the information as in Figure 2-8.

If you don't wish to set this up during the installation then click **Defer task until after installation is complete.** Click **Next.**



Specify a contact for health monitor notification

By default, a health monitor runs on the DB2 instance you are setting up. When a health indicator threshold is breached, email or pager notification will be sent to an administration contact. Specify a new or existing contact for health monitor notification. If your contact list resides on a remote DB2 server, you must provide a user name and password for logon.

Administration contact for this instance

● New contact

Name: db2inst1

Email address: db2inst1@udbsrv01

□ Address is for a pager

○ Defer this task until after installation is complete

Figure 2-8 Specifying contact information for Health Monitor notification

14.Summary

You will see a window summarizing the installation options that you have selected. Click **Finish** to start the installation.

- As DB2 is installing, another window will be launched indicating the progress of the installation process.
- When DB2 has finished installing, a **Setup is complete** window appears, which allows you to view the installation log. Click **Finish** to exit the DB2 Setup installation process.

2.2.4 Installing the DB2 Application Development Client

We now show you how to install the DB2 Application Development Client, because it provides some of the libraries that enable us to make DB2 calls in our application. Please note that by choosing the Custom installation type, both DB2 server code and the Application Development Client can be installed together in one step. We install the Application Development Client separately to show you the installation steps in case you need to install the Application Development Client in a separate machine.

- Login as the root and change directory to the 009_ESE_LNX_32_NLV directory that was created by the tar command in 2.2.2, “Installation preparation” on page 20.

- To use the DB2 Setup install wizard, your desktop environment must be capable of displaying an X windows application. In a Linux terminal window, enter the following command to setup X windows:

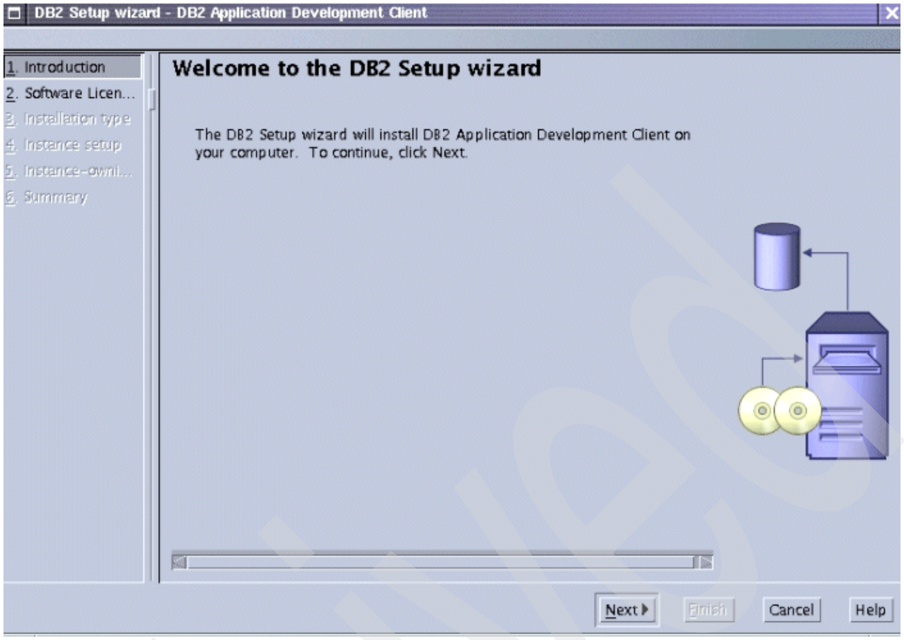
  ```
  xhost + your_localhost_name
  export DISPLAY=your_localhost_name:0
  ```

> **Important:** Because you will be updating the db2inst1 instance, stop the instance before installing the Application Development Client.
>
> - Change to the db2inst1 user (**`su - db2inst1`**).
> - Issue the **`db2stop force`** command. (This will stop the instance and force off any users, this is quite safe in our environment but should not generally be used in a production environment)
> - **`exit`** back to the root user.

- Launch the installation process by typing in:

  ```
  ./db2setup
  ```

  The graphical DB2 Setup wizard will start as in Figure 2-3 on page 21.
- Click **Install Products** and then select **DB2 UDB Application Development Client** and click **Next**.
- The DB2 Setup wizard will appear as shown in Figure 2-9 outlining the installation steps to follow.


DB2 Setup wizard - DB2 Application Development Client

Welcome to the DB2 Setup wizard

The DB2 Setup wizard will install DB2 Application Development Client on your computer. To continue, click Next.

Figure 2-9 Steps to install DB2 UDB Administration Client using DB2 Setup wizard

1. **Introduction**

   The wizard will now guide you through a sequence of steps to setup your DB2 development environment. Click **Next** to proceed.

2. **Software License Agreement**

   In this section you will be asked to read and accept the licence for the DB2 program in order to proceed with the installation process.

   Select **Accept** and click **Next**.

3. **Installation type**

   This section allows you to customize your installation. Also you can click **View Features** to view what is, and what isn't being installed.

   Select **Typical** and click **Next**.

4. **Instance setup**

   This section allows you to define what instance you will add the development capability to. You can use an existing instance use or DB2 Setup can create a new DB2 environment.

   We will use our **db2inst1** instance.

Select **Configure new function for an existing DB2 instance** and select **db2inst1** from the drop-down list. Click **Next**.

5. **Summary**

   This is a summary of what will be installed. Click **Finish** to begin the installation.

- When the setup completes, you can click the **Status report** tab to view the installation log. Click **Finish** to exit from DB2 Setup.

> **Important:** You now need to restart the DB2 instance:
>
> - Change to the db2inst1 user (`su - db2inst1`).
> - Issue the `db2start` command.

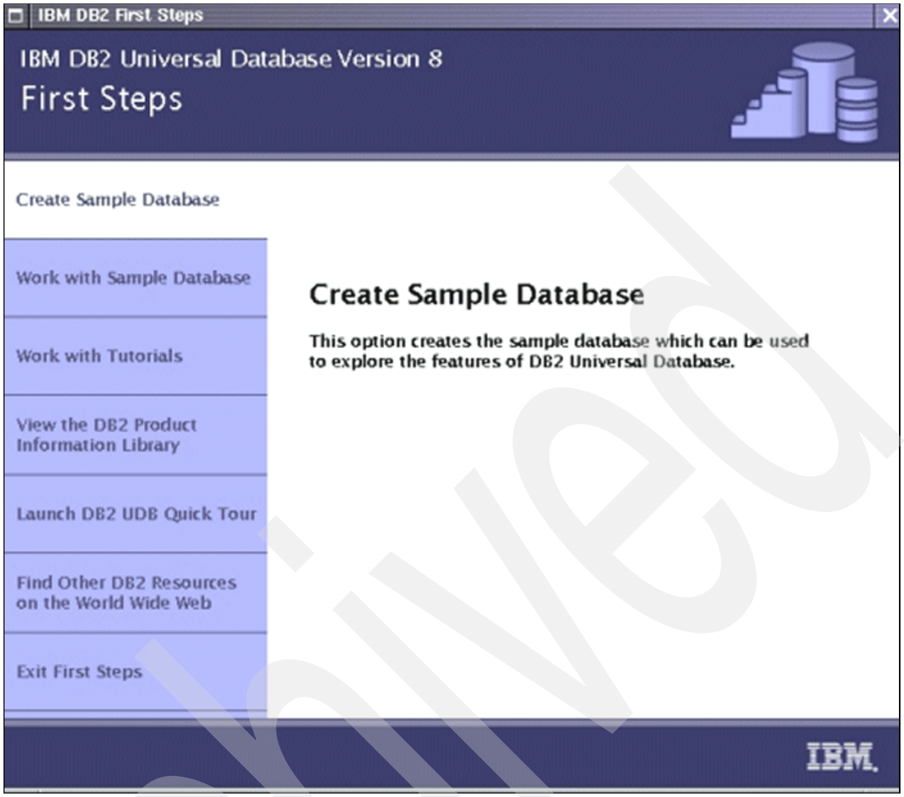
2.2.5 Post installation

DB2 provides you with *First Steps,* a graphical overview of DB2 UDB which acts as an introduction to features and functions of DB2 UDB through the use of sample databases, tutorials, and an interactive presentation. We use First Steps to show you how to create the DB2 sample database.

To use First Steps and set up the sample database:

- Login into Linux as the DB2 instance owner that you have chosen during the installation. In our case, the DB2 instance id is `db2inst1`.
- Make sure you are in an X-window graphical session.
- Enter the command `db2fs`.

Figure 2-10 illustrates the DB2 First Steps window.


IBM DB2 First Steps
IBM DB2 Universal Database Version 8
First Steps
Create Sample Database
Work with Sample Database
Work with Tutorials
View the DB2 Product Information Library
Launch DB2 UDB Quick Tour
Find Other DB2 Resources on the World Wide Web
Exit First Steps
IBM

Create Sample Database

This option creates the sample database which can be used to explore the features of DB2 Universal Database.

Figure 2-10 DB2 for Linux First Steps

- Click **Create Sample database**.
- Select **DB2 UDB sample** and click **OK**.
- The sample database will now be created. Click **OK** when it finishes creating.
- Click **Exit First Steps** to exit.

You can now work with the DB2 sample database.

File locations

There are two distinct file locations you should be aware of:

1. DB2 installation directory (/opt/IBM/db2/V8.1)

   When you install DB2 the executable are extracted to the */opt/IBM/db2/V8.1* directory. Some of the executable can also be manually invoked, so if you want to manually create, update or remove an instance it is done through commands in the */opt/IBM/db2/V8.1/instance* directory.

2. Home directory of the instance owner (/home/db2inst1)

   When you create an instance, an *sqllib* directory is added in the home directory of the instance owner. In our example the sqllib directory is in /home/db2inst1. This directory contains symbolic links to the installation directory, diagnostic logs and control files.

   Also an instance name directory is added into your instance owners home directory. In our example a db2inst1 directory is added to /home/db2inst1. This /home/db2inst1/db2inst1 directory contains all the nodes in the instance. As we chose to install a Single-partition instance the only directory is NODE0000.

   This node directory contains directories for each of the databases we define on the system. The default database creation path is this directory.

   These database directories then contain the default log files, tablespaces, and configuration files. As you will see in Chapter 6, “Database management” we can easily amend these default settings using the DB2 commands or wizards.

DB2 environment

DB2 automatically updates instance owner’s *.profile* or *.bashrc* files adding commands to execute the sqllib/db2profile file. When the db2inst1 user logs in to the system, the DB2 environment will be enabled. The DB2 users in the same system can update their profile in the same way to enable DB2 environment automatically during the login time.

Within the DB2 environment enabled, database commands can be executed directly from a Linux shell. For example:

```
db2 “connect to sample”
db2 “select * from employee”
db2 “connect reset”
```

DB2 graphical environment

Most of the Java based DB2 GUI tools can be invoked from a Linux shell or a Windows command prompt. For example, the command **db2cc** will launch the DB2 Control Center from where most of the other tools can be launched.

2.3 Installing DB2 on Windows

In this section we describe how to install DB2 Version 8.1 trial code Enterprise Server Edition (ESE) on a Windows machine. The Windows operating system used in our lab is Window 2000 Advanced server.

2.3.1 Installation requirements

Before installing DB2 Version 8.1 ESE on Windows, make sure your Windows system meets the following minimum requirements:

- **Operating system requirements**:
  - Windows NT Version 4 with Service Pack 6a or higher
  - Windows 2000 Server (Service Pack 2 is required for Windows Terminal Server)
  - Windows Server 2003 (32-bit and 64-bit)
- **Hardware requirements**:
  - For 32-bit DB2 products, a Pentium or Pentium compatible CPU is required.
  - For 64-bit DB2 products, an Itanium or Itanium compatible CPU is required.
- **Software requirements**: To run DB2's Java-based tools such as Control Center, and to create and run Java applications, it is necessary to have a Java Runtime Environment (JRE) Version 1.3.1. If DB2 installation detects that it is not installed, it will be installed automatically.
- **Memory requirements**: There should be at least 256 MB of physical memory (RAM) available, although more memory may be required depending on the activity on the database system.
- **Disk requirements**: The disk space required for a DB2 Enterprise Server Edition (ESE) depends on the type of installation that will be performed. The DB2 Setup GUI program gives you the following options:
  - Typical installation: approximately 350 - 560 MB
  - Compact: approximately 260 - 400 MB
  - Custom: approximately 260 - 600 MB

2.3.2 Installation preparation

Once verification of system requirements is made, you need to prepare for the installation.

User accounts

During installation two instances are created: the default DB2 instance named DB2, and the Database Administration Server (DAS) instance named DB2DAS00. Unlike DB2 on Linux, DB2 on Windows does not require a user account to own a DB2 instance, but does allow a specified user account to start the service.

The DAS instance is a special Administration Server service which supports the DB2 GUI tools and assists with administration tasks on local and remote DB2 servers. The DAS user account is used to automatically start the DAS service on the computer at boot time. You can create a DAS user account before installing DB2 or you can have the DB2 Setup wizard create it for you. If you allow DB2 to set up the user account, the following user rights will be granted:

- Act as part of the operating system
- Create token object
- Log on as a service
- Increase quotas
- Replace a process level token

Authorizations for installation

To install DB2 on Windows it is necessary to have a local or domain user account that belongs to the Administrator's group. The following user rights should be assigned to the account:

- Act as part of the operating system
- Create a token object
- Increase quotas
- Replace a process level token

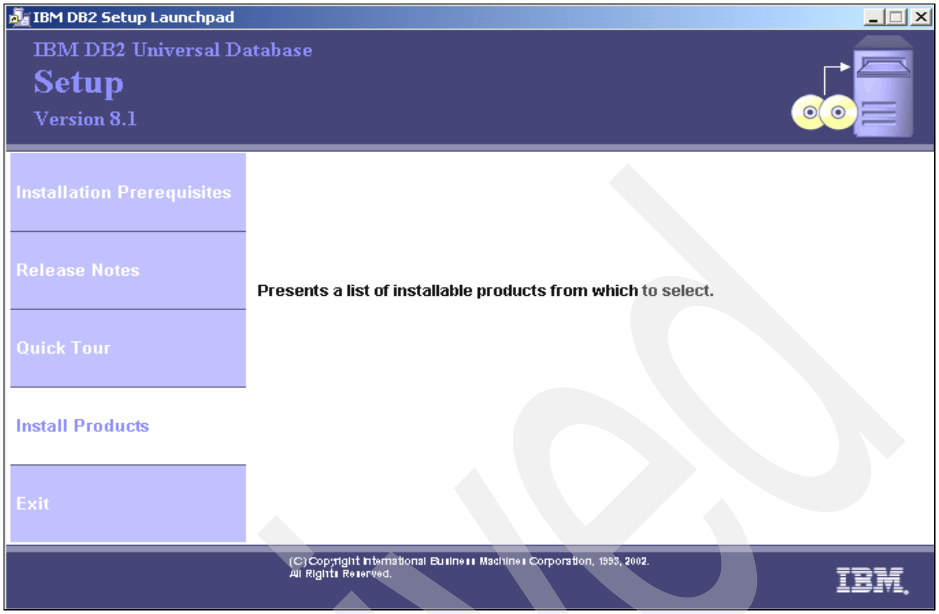
User names on Windows NT, Windows 2000, Windows XP, and Windows .NET currently have a practical limit of 20 characters. For more information about naming rules, refer to Appendix B of the DB2 UDB Version 8 manual, titled *Quick Beginnings for DB2 Servers*.

2.3.3 Installation

In this section, we provide step-by-step installation instructions of DB2 version 8.1 Trial version ESE, using the DB2 Setup wizard. In addition to using the DB2 Setup wizard, you can install DB2 using the response file. We discuss response file installation in 2.4, "Advanced installation" on page 42.

Use the following procedures to install DB2:

1. Once you have unzipped the DB2 trial code download, you will have a folder similar to the naming convention 011_ESE_WIN_32_NLV. Open the directory and look for the installation executable called **setup**. Double-click the file to invoke the DB2 Setup wizard. The first window you will see is the DB2 Setup Launchpad. Click the **Install Products** button (Figure 2-11).



Presents a list of installable products from which to select.

(C) Copyright International Business Machines Corporation, 1993, 2002.
All Rights Reserved.

Figure 2-11 DB2 Setup Launchpad

2. The next window verifies the edition of DB2 that you will be installing. There is only one choice, the DB2 Enterprise Server Edition. Click **Next** to continue.
3. On the DB2 Setup wizard Welcome window, click **Next**.
4. After you read the license agreement on the DB2 Setup wizard License Agreement window, choose **I accept the terms of the license agreement** option and click **Next**.
5. The DB2 Setup wizard Installation Type window now appears requiring the type of installation desired. For our example, we continue with a **Typical** installation which contains most DB2 components.

   The following product options are available:

   - Data Warehousing
   - Satellite administration capability

   We leave both unchecked. Click **Next** to proceed to the next step.

   Please note that the ESE Edition includes the Application Development Client. Since we don't need the Application Development Client in our Windows examples in this book, we did not install the Application Development Client here.
6. A warning dialog may be shown advising you about changes in APPC protocol. Refer to the Release Notes® regarding changes to DB2 UDB and

DB2 Connect™ Version 8 APPC Support documentation if you need more information. Proceed by clicking **OK**.

7. The DB2 Setup wizard installation action dialog is shown. Check the box **Save your settings in a response file**. The details of the response file will be discussed in 2.4, “Advanced installation” on page 42. Make sure that the check box **Install DB2 Enterprise Server Edition on this computer** is checked; otherwise, only the response file will be generated. Click **Next**.
8. The DB2 Setup wizard installation folder window is shown. Select the drive that has enough disk space to install all DB2 files. You can check the amount of free disk space available on each drive of the system by clicking **Disk Space**. You can change the path in the drive where the root directory of DB2 will be placed by clicking **Change**. Click **Next** to advance.
9. DB2 Setup wizard now asks for a user account that will start the DB2 administration service (DAS). Enter the user name and password. Leave the **Use the same user and password for the remaining DB2 services** check box checked. Click **Next**.
10. The DB2 Setup wizard now asks where the contact list should be placed. In our case it will be created locally. For example, select Local - Create a contact list on this system, leave Enable notification unchecked, and proceed by clicking **Next**.
11. A warning is displayed advising that there is no SMTP server specified to send notifications. Click **OK**.
12. Since you haven’t selected an SMTP server yet, leave **Defer the task until the installation is complete** checked. Click **Next**.
13. A list of products to install will be displayed. After reviewing the install list, click **Install** to start the copying of files. At the end of the installation process, click **Finish**. Depending on the system configuration, it may be necessary to reboot the machine. If DB2 Installation wizard detects anything requiring a reboot, it will ask you to do so.

For information on errors encountered during installation, see the db2.log file, which stores general information and error messages resulting from install and uninstall activities. By default, the db2.log file is located in the \MyDocuments\DB2LOG directory. The location of the MyDocuments directory will depend on the settings on your computer.

2.3.4 Post installation

After the installation of DB2 is complete, there are some basic tasks you will want to perform to verify the DB2 installation. For our example, we create the SAMPLE database, verify the location of DB2 file structures, and determine what DB2 services are created.

Using the SAMPLE database

DB2 provides a database called SAMPLE. In addition to the SAMPLE database, a variety of sample programs are included that may be run against the SAMPLE database to assist developers in writing their own applications. We discuss the sample applications in 5.4.1, “DB2 APIs” on page 179.

The SAMPLE database can be created from the First Steps GUI tool or an executable file called `db2sampl`. The `db2sampl` file is located in the *drive:\Program Files\SQLLIB\BIN* directory, but can be executed from anywhere inside the DB2 Command Window environment by typing `db2sampl`.

The following steps show you how to create the SAMPLE database using the First Steps facility:

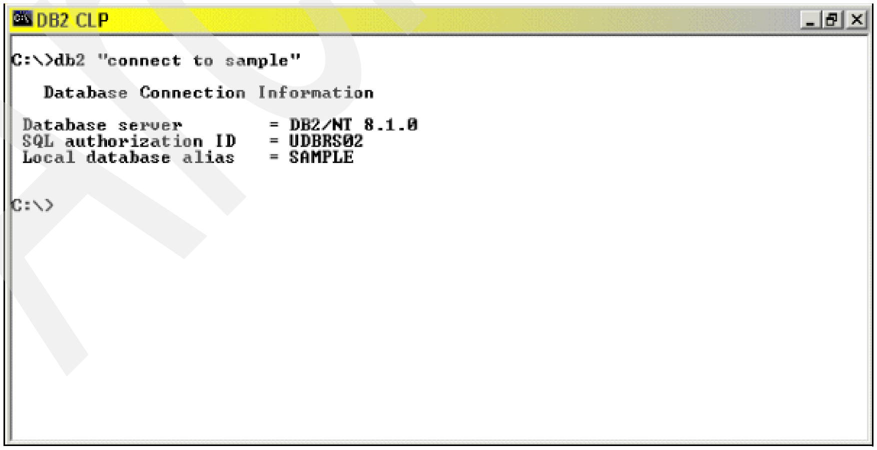
1. Open the First Steps dialog from your Windows programs menu. The path is **Start —>Programs —>IBM DB2 —> Set-up Tools —> First Steps.**
2. Click the **Create Sample Database** tab.

Connect to the SAMPLE database and retrieve data

After the SAMPLE database is created, you can now connect to it to verify that the tables have been created.

1. Open the DB2 Command Window from your Windows programs menu. The path is **Start —>Programs —> IBM DB2 —> Command Line Tools —>Command Window.**
2. A DOS looking screen will appear with the DB2 environment initialized. Type the following statement at the command line and press Enter (Figure 2-12):

   ```
   db2 connect to sample
   ```



```
C:\>db2 "connect to sample"

   Database Connection Information

 Database server        = DB2/NT 8.1.0
 SQL authorization ID   = UDBRS02
 Local database alias   = SAMPLE


C:\>
```

Figure 2-12   Connect to SAMPLE database

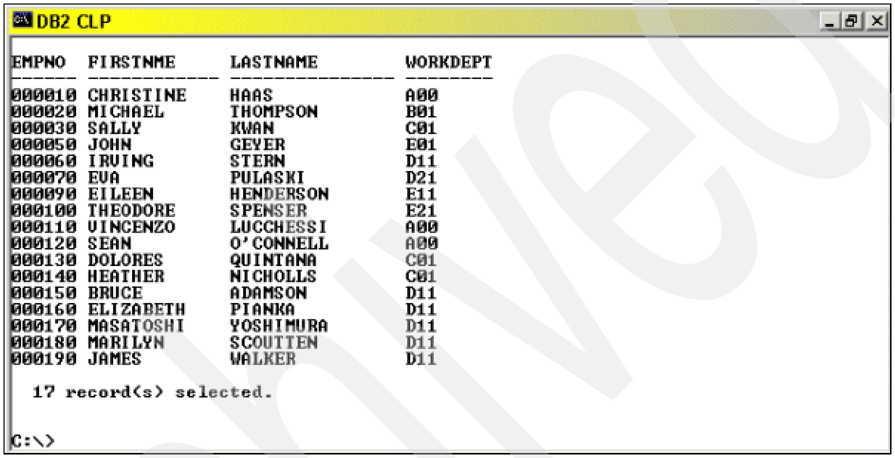
3. Next, list the tables that exist in the SAMPLE database. Type the following command at the command line and press Enter:

   ```
   db2 list tables for all
   ```

   The above statement will return both system tables and user-defined tables.

4. Finally, select data from a user-defined table called EMPLOYEE to determine its contents. Type the following statement at the command line and press Enter (Figure 2-13):

   ```
   db2 “select empno, firstnme, lastname, workdept from employee where empno <
   ‘000200’”
   ```



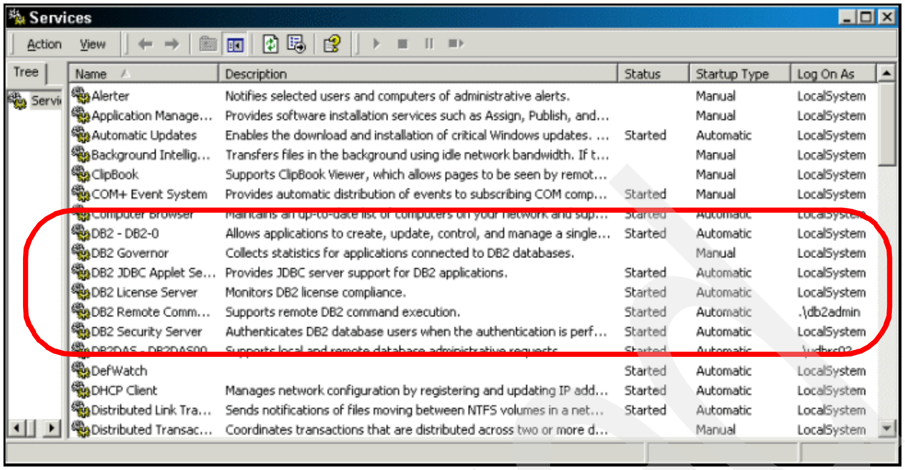
```
EMPNO  FIRSTNME     LASTNAME        WORKDEPT
------ ------------ --------------- --------
000010 CHRISTINE    HAAS            A00
000020 MICHAEL      THOMPSON        B01
000030 SALLY        KWAN            C01
000050 JOHN         GEYER           E01
000060 IRVING       STERN           D11
000070 EVA          PULASKI         D21
000090 EILEEN       HENDERSON       E11
000100 THEODORE     SPENSER         E21
000110 VINCENZO     LUCCHESSI       A00
000120 SEAN         O'CONNELL       A00
000130 DOLORES      QUINTANA        C01
000140 HEATHER      NICHOLLS        C01
000150 BRUCE        ADAMSON         D11
000160 ELIZABETH    PIANKA          D11
000170 MASATOSHI    YOSHIMURA       D11
000180 MARILYN      SCOUTTEN        D11
000190 JAMES        WALKER          D11

  17 record(s) selected.


C:\>
```

Figure 2-13 Select from employee table

Start and stop DB2

For DB2 to function properly, it is necessary to have the DB2 services started. To verify these services are started, check the Windows Services utility. Depending on the type of Windows operating system, the Services utility may be located in a different place on your system. On Windows 2000, you can find the Services utility here: **Start —> Settings —> Control Panel —> Administrative Tools->Services.**

After opening the Services utility, you will notice that there are seven DB2 services with a description of each next to the service name; see Figure 2-14.



Name	Description	Status	Startup Type	Log On As
Alerter	Notifies selected users and computers of administrative alerts.		Manual	LocalSystem
Application Manage...	Provides software installation services such as Assign, Publish, and...		Manual	LocalSystem
Automatic Updates	Enables the download and installation of critical Windows updates. ...	Started	Automatic	LocalSystem
Background Intellig...	Transfers files in the background using idle network bandwidth. If t...		Manual	LocalSystem
ClipBook	Supports ClipBook Viewer, which allows pages to be seen by remot...		Manual	LocalSystem
COM+ Event System	Provides automatic distribution of events to subscribing COM comp...	Started	Manual	LocalSystem
Computer Browser	Maintains an up-to-date list of computers on your network and sup...	Started	Automatic	LocalSystem
DB2 - DB2-0	Allows applications to create, update, control, and manage a single...	Started	Automatic	LocalSystem
DB2 Governor	Collects statistics for applications connected to DB2 databases.		Manual	LocalSystem
DB2 JDBC Applet Se...	Provides JDBC server support for DB2 applications.	Started	Automatic	LocalSystem
DB2 License Server	Monitors DB2 license compliance.	Started	Automatic	LocalSystem
DB2 Remote Comm...	Supports remote DB2 command execution.	Started	Automatic	.\db2admin
DB2 Security Server	Authenticates DB2 database users when the authentication is perf...	Started	Automatic	LocalSystem
DB2DAS - DB2DAS00	Supports local and remote database administrative requests.	Started	Automatic	.\dbrc02
DefWatch		Started	Automatic	LocalSystem
DHCP Client	Manages network configuration by registering and updating IP add...	Started	Automatic	LocalSystem
Distributed Link Tra...	Sends notifications of files moving between NTFS volumes in a net...	Started	Automatic	LocalSystem
Distributed Transac...	Coordinates transactions that are distributed across two or more d...		Manual	LocalSystem

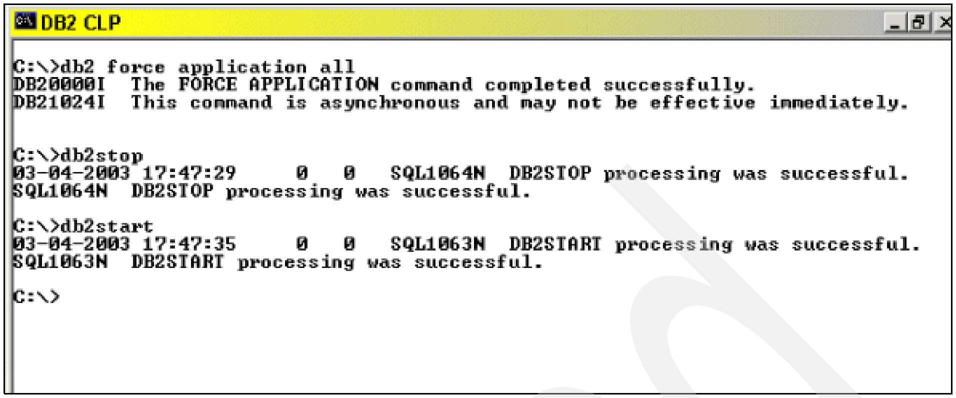
Figure 2-14 DB2 services

As you can see, both the DB2 Instance and DB2DAS00 services are started (column 3) using the user account created during the installation process (column 5). Both the DB2 instance and the DAS instance can be started and stopped either from the Services utility, using DB2 Control Center, or from the DB2 Command Window. To issue a start or stop service request from the DB2 Command Window, type the following:

```
db2start : starts the DB2 instance
db2stop : stops the DB2 instance
```

If applications are still running on the instance, you should stop the applications then perform the **db2stop** command. To stop the application, use the following command:

```
db2 force application all
```



```
C:\>db2 force application all
DB20000I  The FORCE APPLICATION command completed successfully.
DB21024I  This command is asynchronous and may not be effective immediately.


C:\>db2stop
03-04-2003 17:47:29     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.

C:\>db2start
03-04-2003 17:47:35     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.

C:\>
```

Figure 2-15 DB2 commands: force application all, db2stop, db2start

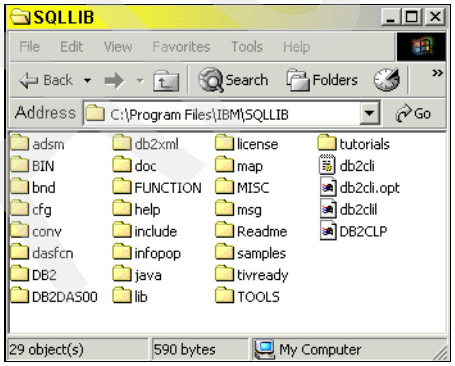
Congratulations! You have installed DB2 Version 8.1 successfully. You have completed the basic verification of the DB2 installation by creating the SAMPLE database and testing the instance by performing the **`db2stop`** and **`db2start`** commands.

DB2 file structure

After the installation is completed and your SAMPLE database is created, the DB2 engine and database file structures will exist on your machine.

DB2 engine directory structure

By default, DB2 is installed in the following directory: <drive letter>\Program Files\IBM\SQLLIB. This directory contains all DB2 executable and is shared by all DB2 instances. Figure 2-16 is a visual representation of the DB2 installation directory structure.



Address: C:\Program Files\IBM\SQLLIB

29 object(s) | 590 bytes

Figure 2-16 Installation directory structure

Database directory structure

On Windows, a database directory is contained in the following path:

```
<drive letter>\DB2\NODE0000\
```

Since we created the SAMPLE database immediately after installing DB2, it will be stored in the path:

```
<drive letter>\DB2\NODE0000\SQL00002
```

The SAMPLE database happens to be the second database entry in the directory (as noted by SQL00002); see Figure 2-17.

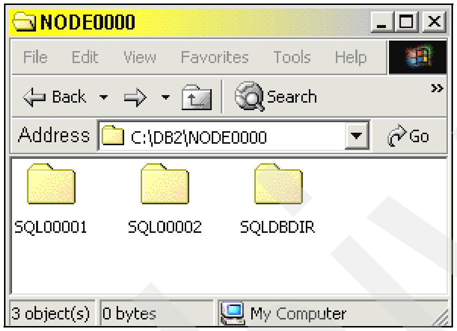


Figure 2-17 Database directory structure

You might be wondering why two databases currently exist. During installation, we asked for the TOOLSDB database to be created so that the Scheduler facility used by Control Center, Command Center, and Task Center will work correctly. More will be discussed about the TOOLSDB database in 6.1.3, "Administration server configuration" on page 204.

The SQLDBDIR directory holds system information regarding the databases. For instance, if you issue the following command in the DB2 Command Window, DB2 system files will be read from the SQLDBDIR directory to retrieve all databases physically residing on the drive letter C:

```
db2 list database directory on c
```

The following output will be printed to the screen (Example 2-1).

Example 2-1 List database directory output

```
Local Database Directory on c
Number of entries in the directory = 2
Database 1 entry:
Database alias                       = TOOLSDB
Database name                        = TOOLSDB
Database directory                   = SQL00001
Database release level               = a.00
Comment                              =
Directory entry type                 = Home
Catalog database partition number    = 0
Database partition number            = 0

Database 2™ entry:
Database alias                       = SAMPLE
Database name                        = SAMPLE
Database directory                   = SQL00002
Database release level               = a.00
Comment                              =
Directory entry type                 = Home
Catalog database partition number    = 0
Database partition number            = 0
```

Note: The database directory structure should not be modified directly, because this may cause database corruption. To change or remove database objects (for example, tables or tablespaces), you will need to use the supported DB2 commands and SQL statements.

2.4 Advanced installation

There are four methods in which you can install DB2 on Linux:

- DB2 Setup wizard
- Using a response file
- The db2_install utility
- Using the Linux command RPM

DB2 on Windows can be installed by one of two methods

- DB2 Setup wizard
- Using a generated response file

Assuming you have the necessary graphical terminal available, installing DB2 using DB2 Setup wizard is our recommended choice, because it installs the code and also configures the DB2 environment.

In this section we discuss the additional installation options available to DB2 users and when you may want to consider using them.

Note: This is a high level discussion of the installation options. The necessary commands as well as other additional details can be found in the following two IBM Redbooks:

- Chapter 2 of *Up and Running with DB2 for Linux*, SG24-6899
- Chapter 2 of *DB2 UDB Exploitation of the Windows Environment*, SG24-6893

2.4.1 Using a generated response file

Here we discuss the DB2 generated response file.

What is it?

Put simply, it is a text file that allows you to install DB2 across multiple machines with consistent installation and configuration settings. It can also be used to create multiple identical partitions in a DB2 Cluster. A response file can install DB2, configure users, create instances, set up SMTP notification, create a tools catalog, and configure a large number of DBM parameters.

Why would you use it?

A response file installation is fast, because it bypasses the graphical wizard and does the configuration for you. This will be the quickest installation method if you already have all the information you need. It is also recommended for use over a non graphical interface (such as a telnet session) or for including in a system build by system administrators users who may not need to be familiar with the DB2 Setup procedure. Another key reason to use a response file is when you are planning a massive deployment or porting the same installation from development to production. Response file installation gives the same configuration on multiple machines.

The article at the following Web site provides details on how to install DB2 transparently with an application:

http://www7b.software.ibm.com/dmdd/library/techarticle/0302gao/0302gao.html

How would you use it?

There are two distinct steps, creating the response file and using the response file.

The creation of a DB2 response file is achieved in one of three ways:

1. As part of the DB2 Setup process (this is covered in 2.2, “Installing DB2 on Linux” on page 18 and 2.3, “Installing DB2 on Windows” on page 32).

2. The DB2 installation directory or CD-ROM contains a ready-to-use sample response file with default entries. This file, called *db2ese.rsp*, is located in: `/009_ESE_LNX_32_NLV/db2/linux/samples` for Linux and `/011_ESE_WIN_32_NLV/db2/linux/samples` for Windows.

   This directory also contains sample response files for the Application Development Client and Administration Client, named *db2adcl.rsp* and *db2admcl.rsp*, respectively.

3. *For the Windows platform only*, a response file generator utility exists which creates a response file from an existing installed and configured DB2 product. You can use the generated response file to recreate the exact setup on other machines. The syntax of the command is:

   ```
   db2rspgn -c:\rspfile\
   ```

   This creates two files, a *DB2.INS* file and a *db2ese.rsp* file. The latter contains the installation instructions and automatically calls the instance and database setup contained within the DB2.INS file.

For all of the methods detailed above the default response file names *(db2ese.rsp, db2adcl.rsp, db2admcl.rsp)* can be changed if you want.

The generated Response File can be used in the following ways:

- **To create additional partitions**: After performing a partitioned installation (not covered in this book, but covered in some detail in the Linux redbook), the DB2 Setup wizard creates two response files: *db2ese.rsp* and *db2ese2.rsp*. The *db2ese.rsp* has the same content that has been discussed above. The *db2ese2.rsp* response file, however, only installs the appropriate DB2 filesets, specifies DAS properties, and specifies languages to install. Only use the *db2ese2.rsp* response file when installing database partition servers on participating computers. This is because when you set up a partitioned DB2 environment, you are using the same instance and databases across the multiple partitions. But a DAS on each server allows clients to use the Control Center to connect to that individual partition.
- **To install an identical DB2 system** based either on an existing install, the sample response files or modified response files.

  For each of the uses above, the command to run is the same:

  ```
  /009_ESE_LNX_32_NL/db2setup -r ./db2/linux/samples/db2ese.rsp
  ```

  This example will install the Linux DB2 trial code according to the sample response file.

2.4.2 Using db2_install

We discuss the db2_installation method which is available on the Linux and UNIX platforms only.

What is it?

The db2_install script installs all DB2 packages on your Linux system using the RPM installation utility. This method is used by experienced users for installing DB2 on larger, complex, and multiple-partition systems. Tasks such as setting up OS userids/groups, creating instances, creating the tools catalog database and setting up notification have to be performed manually after the db2_install installation.

Why would you use it?

This method can be preferable, because it bypasses the configuration performed by DB2 Setup and allows you to initially configure DB2 by your own preferred method. A good example of this is the creation of the tools database. By default, the DB2 Setup wizard creates the tools catalog database in the instance owner's home directory if you choose not to put it in an existing database. The separation of code installation and DB2 configuration that db2_install provides you with will allow you to change the default database directory before creating the tools database. Once you have installed the code and created the instances you can update the necessary configuration parameters before you create the tools database. This DB2 configuration parameter will be discussed as part of 6.5.2, "Using the DB2 configuration parameters" on page 291.

Another key advantage is that unlike the DB2 Setup wizard the db2_install utility does not require a graphical terminal.

There is a language limitation to db2_install in that it only installs support for English. This means that help, messages, and tool interfaces are in English. The DB2 Setup wizard can install support for one or more different languages. Other limitations are that the installation may take longer, as it requires more manual configuration tasks and it cannot create response files. In addition to this, the Java environment needed for the DB2 graphical tools requires manual set up too.

How would you use it?

There are two distinct steps, using db2_install to install the DB2 code and the post installation steps to create instances and configure DB2 ready for use.

- Using db2_install

  a. Run the db2_install command to start the db2_install script:

  ```
  . /009_ESE_LNX_32_NLV/db2_install
  ```

  b. When db2_install prompts you for the product keyword, enter in `DB2.ESE`.

DB2 will be installed in */opt/IBM/db2/V8.1*

- Post-installation tasks:

  After installing DB2 using db2_install, you need to do the following:

  a. Create OS groups and user IDs.

  b. Create a DB2 instance.

  c. Create links for DB2 file (optional).

  d. Configure TCP/IP communications for the DB2 instance.

  e. Create a DB2 Administration Server (DAS).

  f. Create DB2 tools catalog (optional).

  g. Set up SMTP server and notification contact list (optional).

  h. Install the IBM Developer Kit for Java Version 1.3.1.

  As root from the directory `/009_ESE_LNX_32_NLV`/db2/linux/Java-1.3/ run the following command:

  ```
  rpm -ivh IBMJava2-SDK-1.3.1-2.0-i386.rpm
  ```

  Note: For detailed instructions on these post installation steps, consult *Up and Running with DB2 for Linux*, SG24-6899.

2.4.3 RPM command

RPM is the Linux software installation command. Using the RPM command to install DB2 allows you to select specific DB2 files. However, the RPM installation method will only install the DB2 code. You won't be able to create instances, user IDs, or a response file. The RPM installation method is not a supported or recommended installation method.

3

Navigating DB2 with GUIs

GUIs provide a much more intuitive approach in navigating DB2 allowing you to become familiar much more rapidly with the DB2 components. Each of the DB2 tools or wizards provides a specific administrative or development function to increase your productivity. For example, Control Center allows you to visually manage a DB2 instance or database object such as creating a database or dropping a table. The new Task Center enables you to manage scheduling and execution of DB2 command and operating system scripts. The new Health Center is a first step towards the IBM autonomic computing strategy, allowing proactive and self-healing techniques with the concept of *monitoring by exception*.

In this chapter, the following GUI tools and wizards are discussed:

- Control Center and Command Center
- Configuration Advisor and Design Advisor
- Task Center and Journal
- Health Center and Memory Visualizer
- Development Center
- Configuration Assistant
- Replication Center
- Information Center

3.1 Introduction to GUI tools

With the release of each new version of DB2, more and more great features are being added to the DB2 GUI tool set. Version 8.1 is loaded with rich new features to help you manage your database and develop your database applications. Some of the most exciting improvements come with the introduction of Health Center allowing you to *manage by exception* and Replication Center to administer replication between DB2 and non-DB2 relational databases.

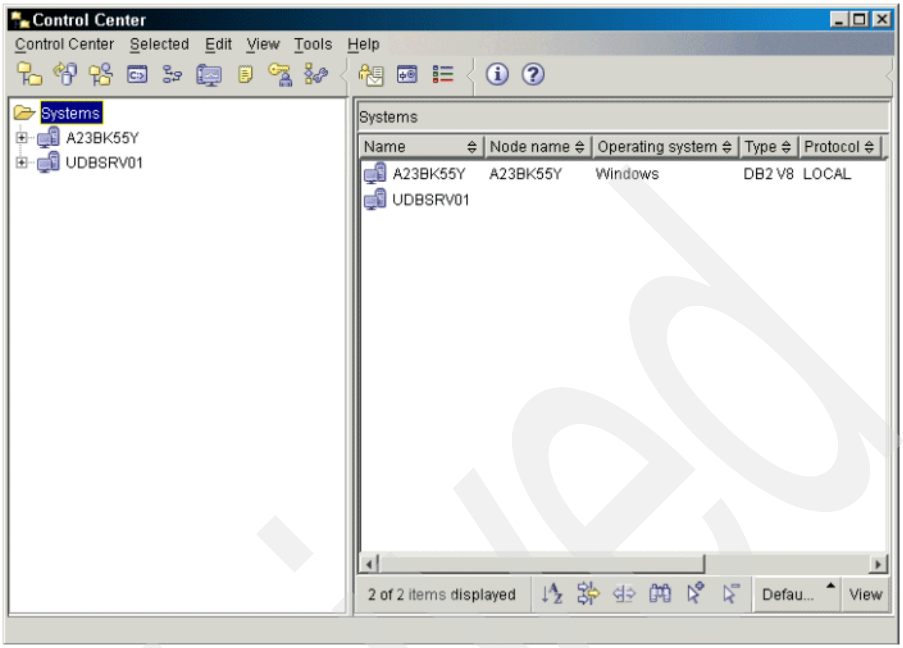
DB2 Version 8.1 introduces the Development Center which incorporates many functions and features to assist the application development. Version 7 Stored Procedure Builder is included in the Development Center. Version 7 Script Center has been enhanced to become Task Center which is used to define and schedule tasks, view the results of completed tasks, and perform other administrative activities on tasks. Client Configuration Assistant is also enhanced to Configuration Assistant.

All of the primary GUI tools discussed in this chapter can be launched as a stand-alone application from the IBM DB2 Universal Database program group or from a DB2 Universal Database Center, such as the Control Center, the Command Center, or the Task Center.

3.2 Control Center

You may consider the Control Center as a window to manage all of your DB2 objects. The Control Center is a good place to start our GUI tour, since we can see a visual of the parent/child relationships between DB2 objects.

To demonstrate some of Control Center's functionality, let's open the Control Center. On Windows, use the following path to open the Control Center's main interface: **Start -> Programs -> IBM DB2 -> General Administration Tools -> Control Center.** On Linux, type `db2cc` at a terminal window. Make sure you are the DB2 instance owner so you can explore all the functions. Figure 3-1 shows the Control Center's main interface.



Name	Node name	Operating system	Type	Protocol
A23BK55Y	A23BK55Y	Windows	DB2 V8	LOCAL
UDBSRV01				

Figure 3-1 Control Center

In Control Center, there are two panels that divide the interface. The left panel shows the DB2 objects in a hierarchical tree starting from the Systems folder. If you were to completely expand the folder tree, you will see folders for the most basic database elements such as tables, triggers, and indexes. The right pane displays the contents of the active folder from the left pane. For instance, if the tables folder is open on the left pane, the right pane will show the list of tables in the folder.

To manage objects, right-click the object. This will reveal a drop-down menu of options.

Let’s connect to the SAMPLE database.

1. Start by expanding the hierarchical tree in the left pane until you see the SAMPLE database icon.
2. Right-click the database icon and select **Connect** from the drop-down menu (Figure 3-2).

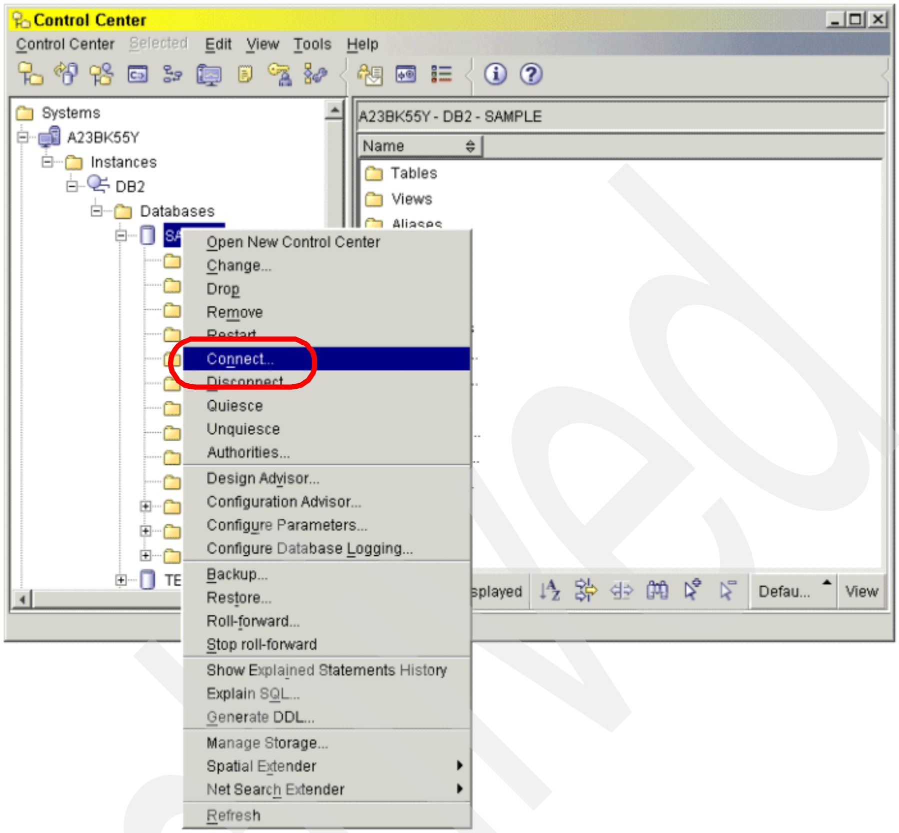


Figure 3-2 Control Center - connect to SAMPLE database

3. You will be prompted with a user ID and password to connect to the database. Enter the user account and password that you created when you installed DB2. You should now be connected to the database. While you are here, you may want to try some of the other functions such as creating a table (Figure 3-3).

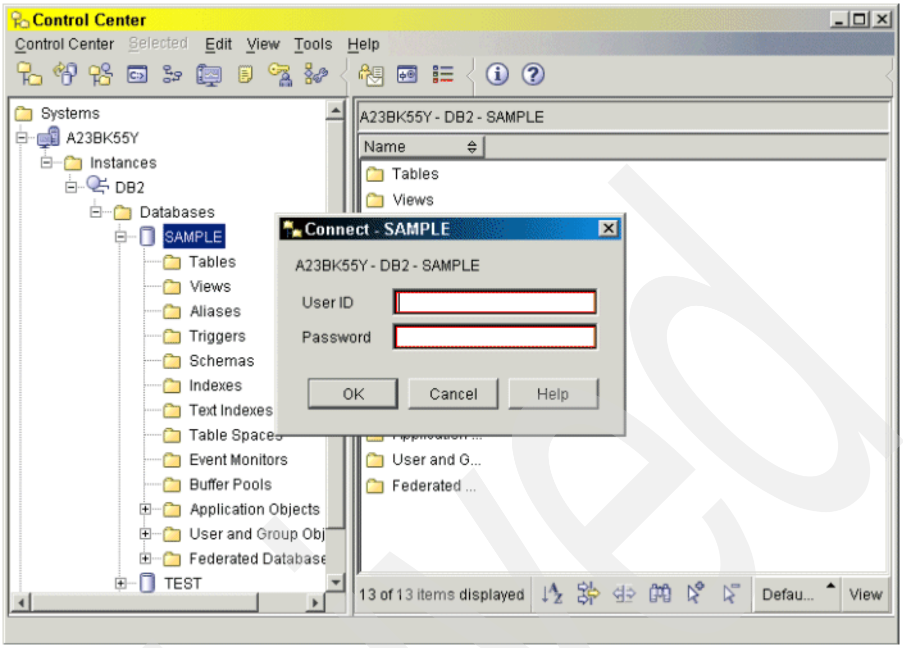


Figure 3-3 Control Center - connect to SAMPLE database

3.3 Command Center

Unlike Control Center which manages objects visually, Command Center allows you to write and execute DB2 commands and SQL statements interactively. It also lets you write scripts containing DB2 commands and/or SQL statements and run them immediately or schedule for later execution.

Let's begin by opening the Command Center using the Control Center. You can launch the Command Center by clicking the **Command Center** icon (Figure 3-4), or left click the Tools menu drop-down list located at the top of the Control Center interface and select Command Center.

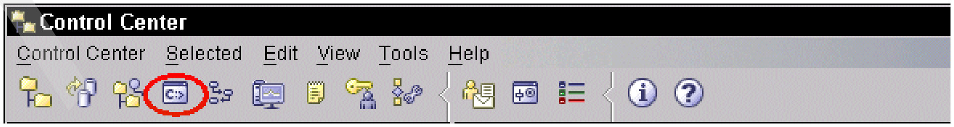
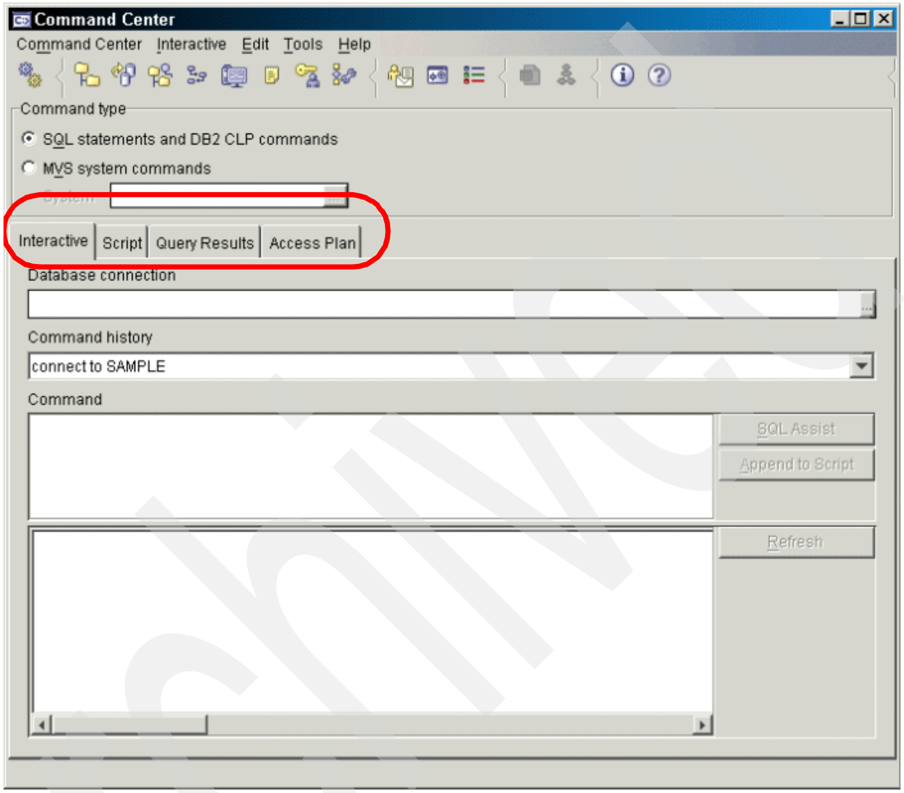


Figure 3-4 Command Center icon

Figure 3-5 shows Command Center’s main interface. In Command Center, there are four tabs in the upper middle left area of the interface that are associated to a specific function. The tabs are: Interactive, Script, Query Results, and Access Plan.



Command type
● SQL statements and DB2 CLP commands
○ MVS system commands
System

Database connection

Command history
connect to SAMPLE

Command

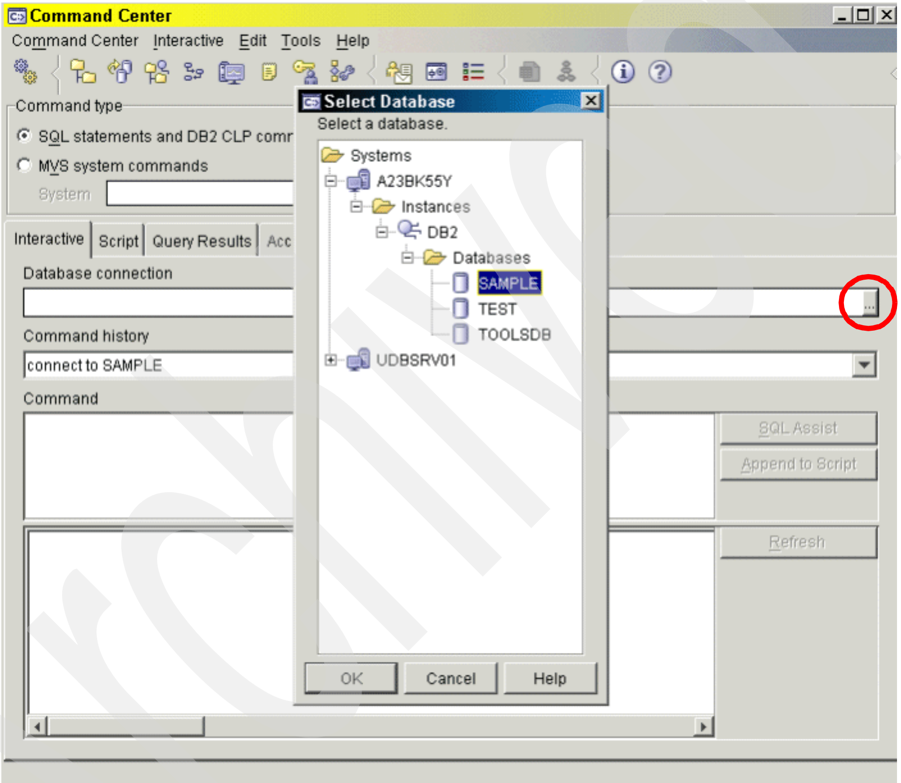
Figure 3-5 Command Center

3.3.1 Interactive and Query Results tabs

The Interactive tab is used for writing and executing DB2 commands and SQL statements. The Interactive tab displays, by default, when invoking Command Center (Figure 3-5). Beneath the Interactive tab, you will see Database Connection field (or box) to insert a database connection. Below the Database Connection field is a Command History field showing your recently executed commands. Below the Command History field is a Command field where you can enter specific DB2 commands and SQL statements. The execution results of DB2 commands will be displayed in the last text field (or box) while the results of SQL statements are displayed in the Query Results tab.

Let's run an SQL statement from the Interactive interface.

1. First, we need to make a connection to the SAMPLE database. To insert a database connection to the SAMPLE database, use the browse function from the Database Connection field. A pop-up folder tree will be displayed. Expand the tree all the way down until you see the SAMPLE database. Select it and press **OK** (Figure 3-6). A connection will be made to the SAMPLE database. The database connect command will show in the Command field and the result will show in the bottom field.



Command type:
● SQL statements and DB2 CLP comr
○ MVS system commands

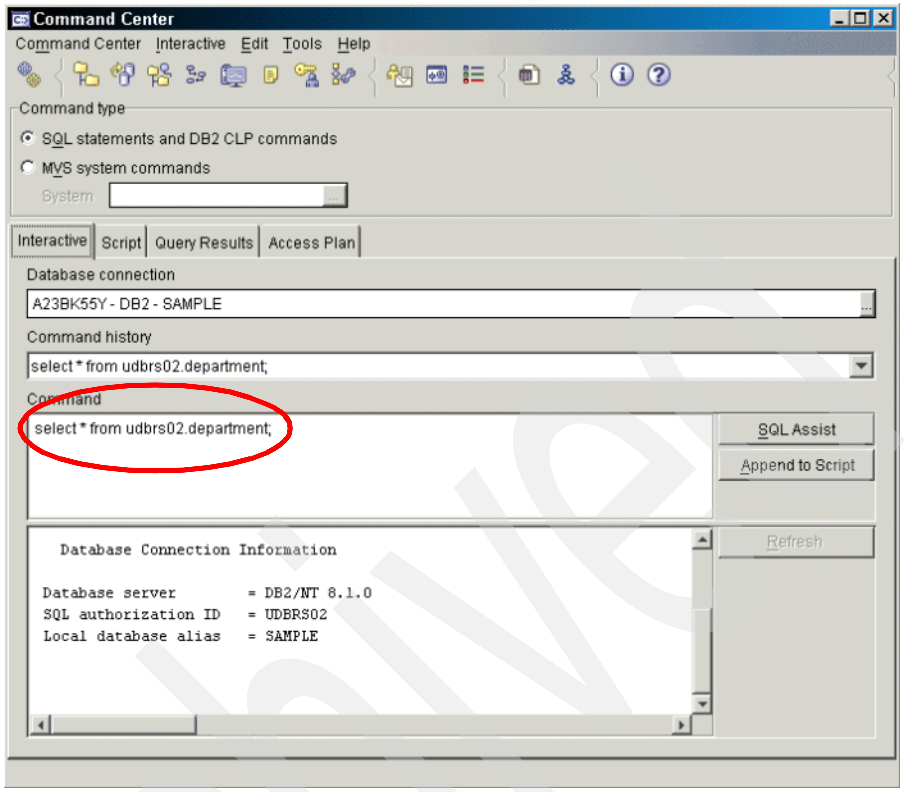
Command history: connect to SAMPLE

Figure 3-6 Command Center - connect to SAMPLE database

2. Next, we will select all rows from the DEPARTMENT table. In the Command field, type:

   ```
   select * from <user name>.department;
   ```

   In the above statement, replace <user name> with the user account that created the SAMPLE database via the First Steps GUI interface (Figure 3-7).



Command type: ● SQL statements and DB2 CLP commands ○ MVS system commands

System

Database connection: A23BK55Y - DB2 - SAMPLE

Command history: select * from udbrs02.department;

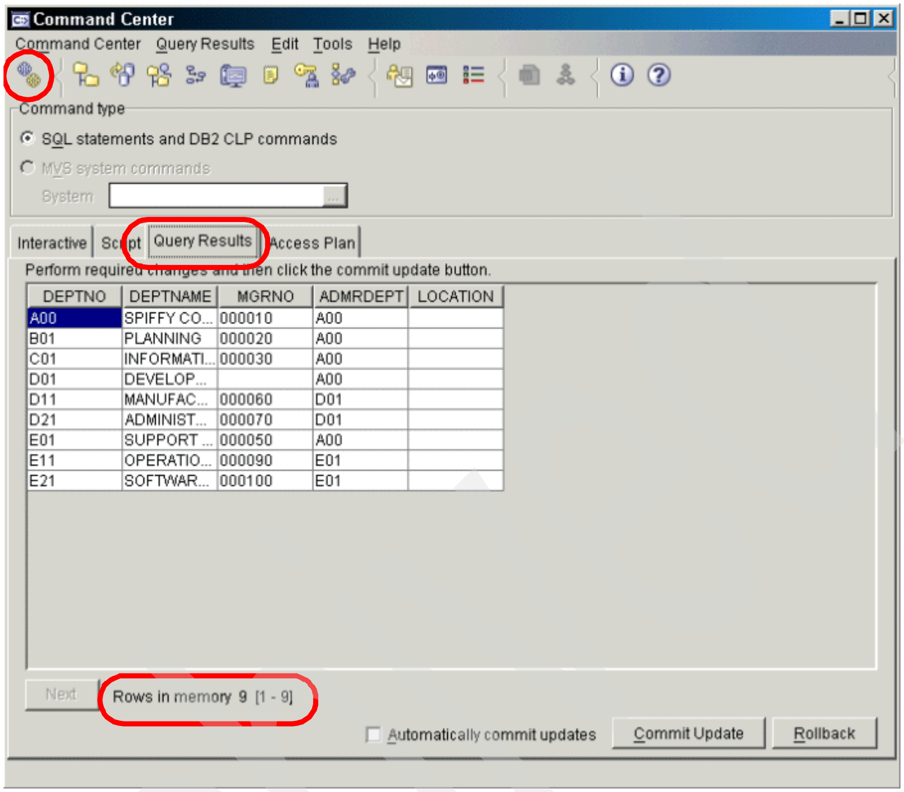
Command:

```
select * from udbrs02.department;
```

```
   Database Connection Information

 Database server        = DB2/NT 8.1.0
 SQL authorization ID   = UDBRS02
 Local database alias   = SAMPLE
```

Figure 3-7 Command Center - Interactive tab

3. Click the Execute icon in the upper left corner of the interface. This button will execute the query. Immediately upon execution, the Query Results tab will become active and display the results of your query. You should see nine rows from the Department table (Figure 3-8).


Command Center
Command Center Query Results Edit Tools Help
Command type
● SQL statements and DB2 CLP commands
○ MVS system commands
System
Interactive Script Query Results Access Plan
Next
□ Automatically commit updates
Commit Update
Rollback

Perform required changes and then click the commit update button.

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT	LOCATION
A00	SPIFFY CO...	000010	A00	
B01	PLANNING	000020	A00	
C01	INFORMATI...	000030	A00	
D01	DEVELOP...		A00	
D11	MANUFAC...	000060	D01	
D21	ADMINIST...	000070	D01	
E01	SUPPORT ...	000050	A00	
E11	OPERATIO...	000090	E01	
E21	SOFTWAR...	000100	E01	

Rows in memory 9 [1 - 9]

Figure 3-8 Command Center - Query Results tab

3.3.2 Script tab

The Script tab is used for writing DB2 commands and SQL statements to a script for later execution. You can also execute DB2 commands and SQL statements immediately using the Execute icon in the tool bar. Upon selecting the Script tab, you will see a Script history list field showing previously executed scripts. Below the Script history list field is a Script field for you to enter your DB2 commands and SQL statements. The field below the Script field will display the results of the script execution (Figure 3-9).

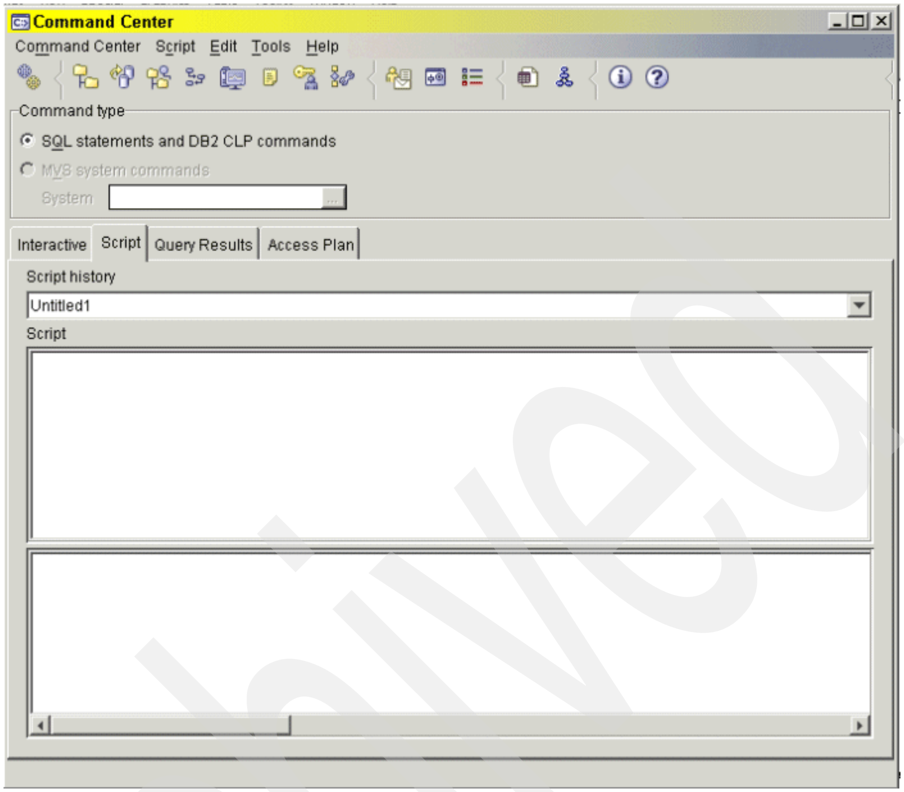
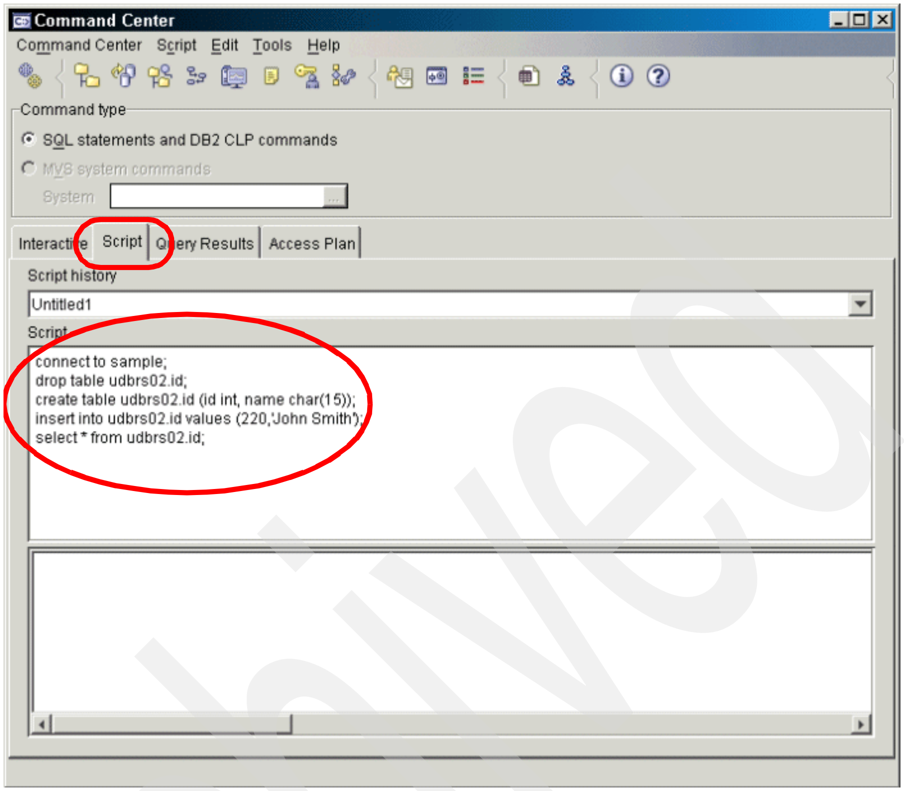


Figure 3-9 Command Center - Script tab

Let's create and execute a script:

1. From the main Command Center interface, select the **Script** tab.
2. Type the following in the Script field (Figure 3-10):

```
connect to sample;
drop table <user name>.id;
create table <user name>.id (id int, name char(15));
insert into <user name>.id values (220,'John Smith');
select * from <user name>.id;
```



Command type

● SQL statements and DB2 CLP commands

○ MVS system commands

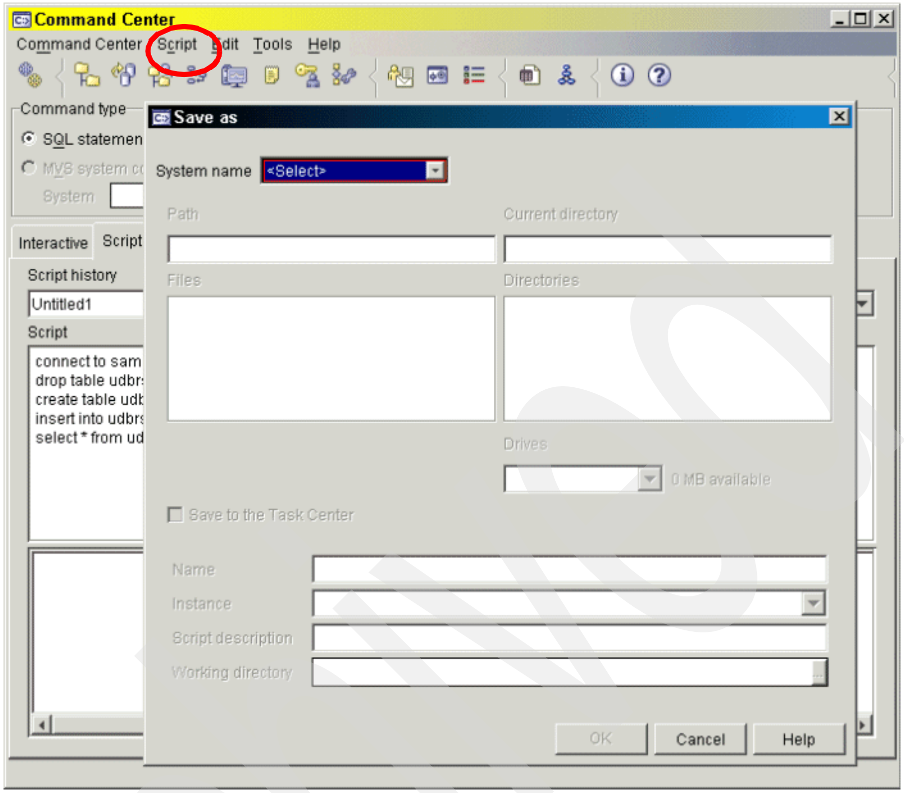
Script history: Untitled1

Script

```
connect to sample;
drop table udbrs02.id;
create table udbrs02.id (id int, name char(15));
insert into udbrs02.id values (220,'John Smith');
select * from udbrs02.id;
```

Figure 3-10 Command Center - enter commands under Script tab

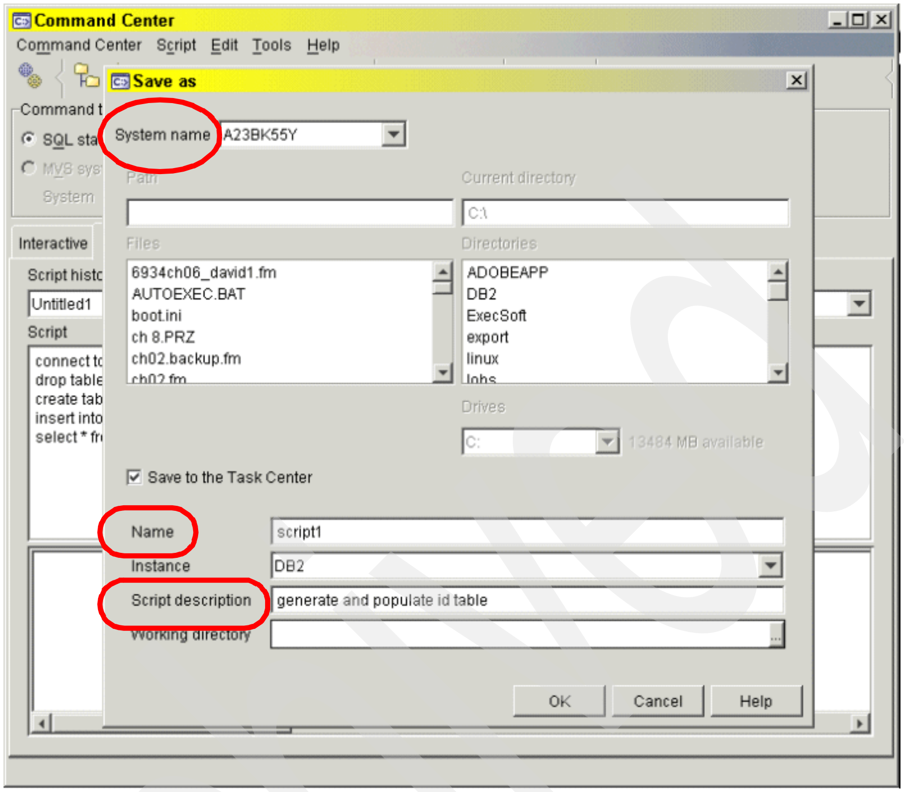
3. After entering the above SQL statements in the Script field, we will save the script to file. Click the **Script** drop-down menu from the menu bar at the top of the interface and select **Save Script As**. A Save as window will appear (Figure 3-11).



```
connect to sam
drop table udbr
create table udb
insert into udbrs
select * from ud
```

Figure 3-11 Command Center - Script tab save commands option

4. Select your system name from the System name list field. Enter the name *script1* for the script in the Name text field and the description *generate and populate id* table in the Script description text field. Leave the *Save to the Task Center* check box checked and select **OK** (Figure 3-12).

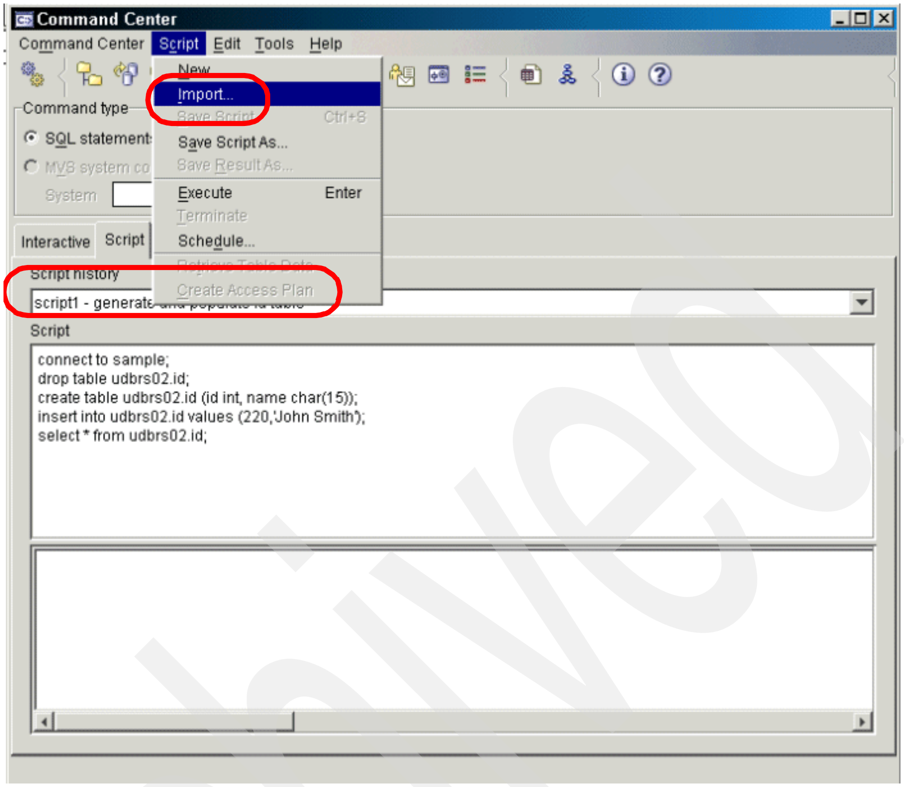


System name: A23BK55Y
Path:
Current directory: C:\
Files:
6934ch06_david1.fm
AUTOEXEC.BAT
boot.ini
ch 8.PRZ
ch02.backup.fm
ch02.fm
Directories:
ADOBEAPP
DB2
ExecSoft
export
linux
lobs
Drives: C: 13484 MB available
☑ Save to the Task Center
Name: script1
Instance: DB2
Script description: generate and populate id table
Working directory:

Untitled1
connect to
drop table
create tab
insert into
select * fr

Figure 3-12 Command Center - Save script to Task Center option under Script tab

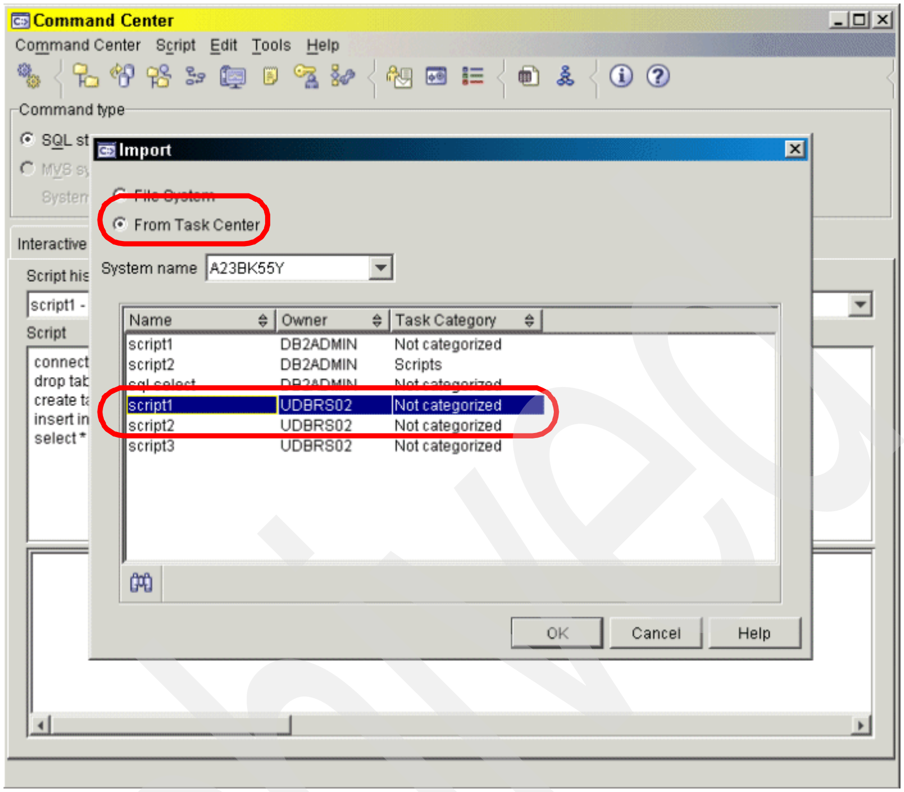
5. You will now see *script1-generate and populate id table* in the Script history field. Even though the script is open, we will describe how you will open the script. Click the Script drop down menu list and select **Import** (Figure 3-13).



```
connect to sample;
drop table udbrs02.id;
create table udbrs02.id (id int, name char(15));
insert into udbrs02.id values (220,'John Smith');
select * from udbrs02.id;
```

Figure 3-13 Command Center - open a script under Script tab

6. A pop-up window will display. Choose the *From Task Center* radio button, click the *script1* file name from the main text field and select **OK** (Figure 3-14).



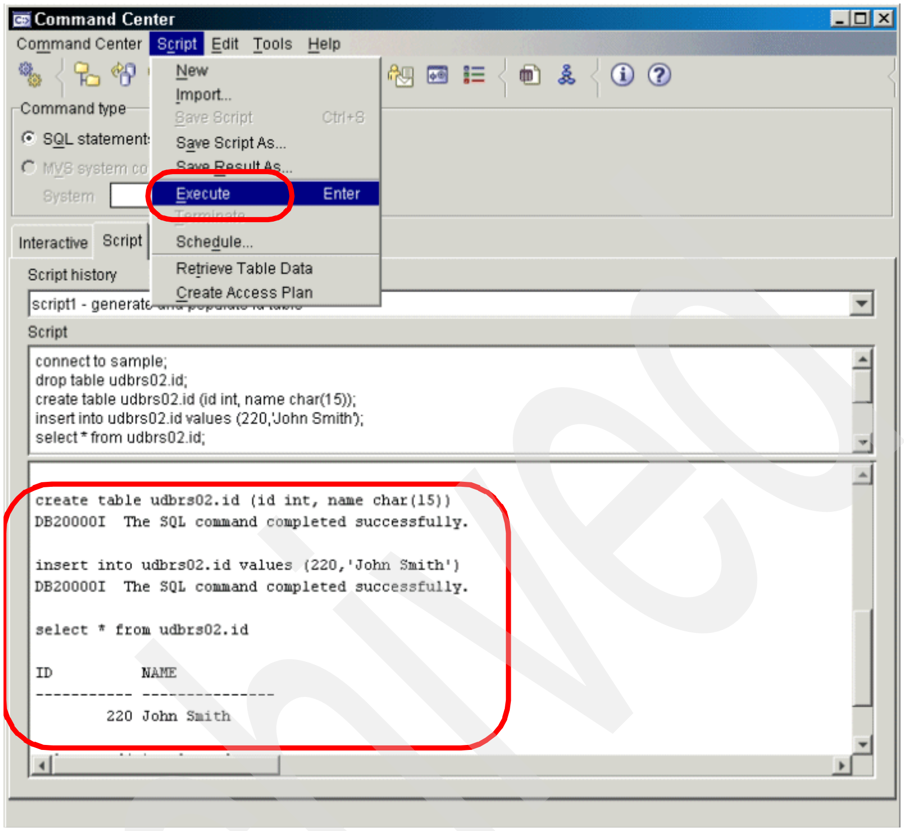
○ File System
● From Task Center

System name: A23BK55Y

Name	Owner	Task Category
script1	DB2ADMIN	Not categorized
script2	DB2ADMIN	Scripts
sql select	DB2ADMIN	Not categorized
script1	UDBRS02	Not categorized
script2	UDBRS02	Not categorized
script3	UDBRS02	Not categorized

Figure 3-14 Command Center - Select a script to open under Script tab

7. Execute the script by choosing Execute from the Script drop down menu list. You will see both the script commands and the results displayed in the text field at the bottom (Figure 3-15).



```
connect to sample;
drop table udbrs02.id;
create table udbrs02.id (id int, name char(15));
insert into udbrs02.id values (220,'John Smith');
select * from udbrs02.id;
```

```
create table udbrs02.id (id int, name char(15))
DB20000I  The SQL command completed successfully.

insert into udbrs02.id values (220,'John Smith')
DB20000I  The SQL command completed successfully.

select * from udbrs02.id

ID          NAME
----------- ---------------
        220 John Smith
```

Figure 3-15   Command Center - execute a script under Script tab

3.3.3 Access Plan tab

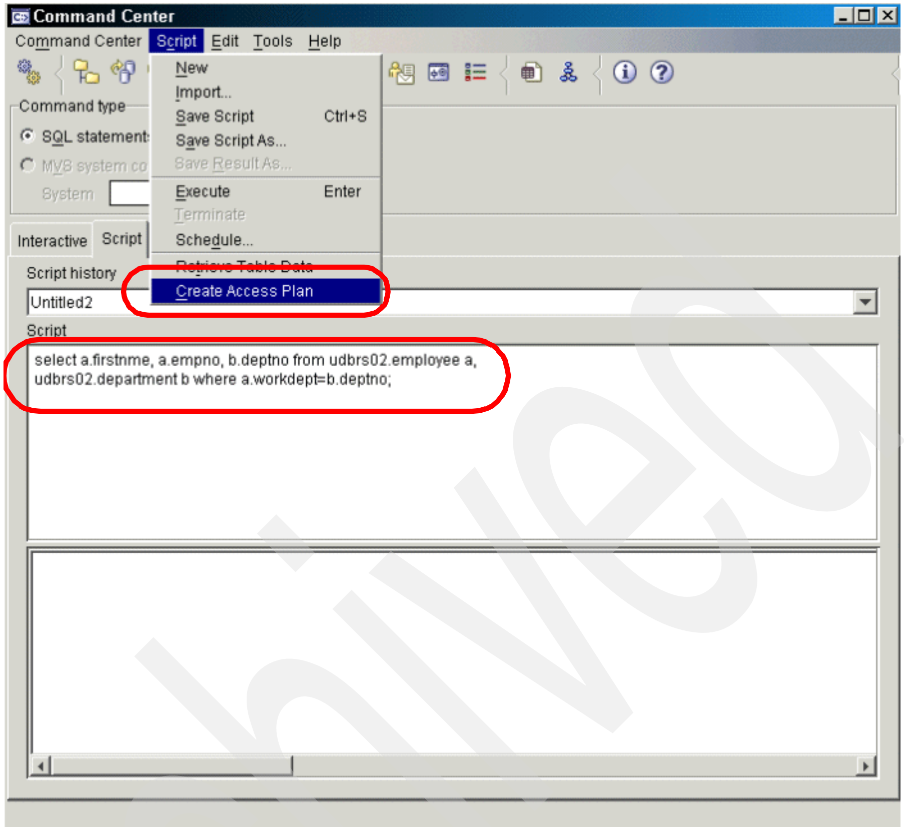
The Access Plan tab is used when you want to determine how a specific SQL statement is optimized by the DB2 optimizer. Both the Interactive and Script tabs can be used to generate an access plan.

Let’s generate and display an Access Plan:

1. From the Script tab, place the following in the Script field (Figure 3-16):

   ```
   select a.firstnme, a.empno, b.deptno from <user account>.employee a,
   <user account>.department b where a.workdept=b.deptno;
   ```

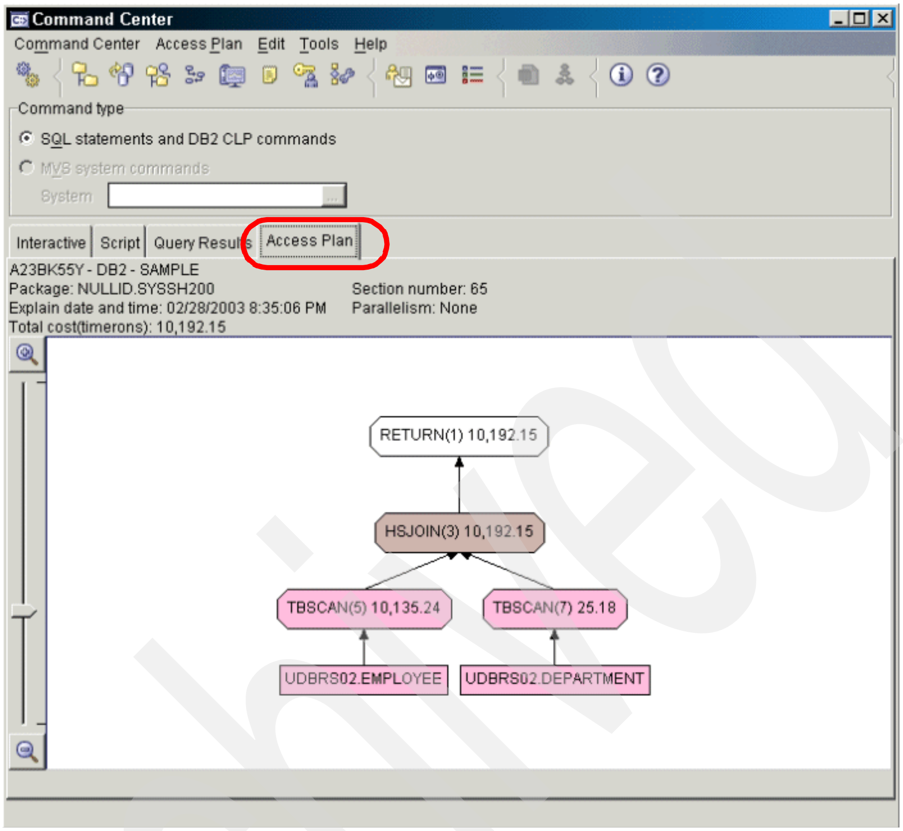
   Then select **Create Access Plan** from the Script drop down menu.



```
select a.firstnme, a.empno, b.deptno from udbrs02.employee a,
udbrs02.department b where a.workdept=b.deptno;
```

Figure 3-16 Command Center - using Script tab to create access plan

2. An access plan will be generated and the Access Plan tab will display with a visual of the access plan (Figure 3-17).



● SQL statements and DB2 CLP commands

○ MVS system commands

System

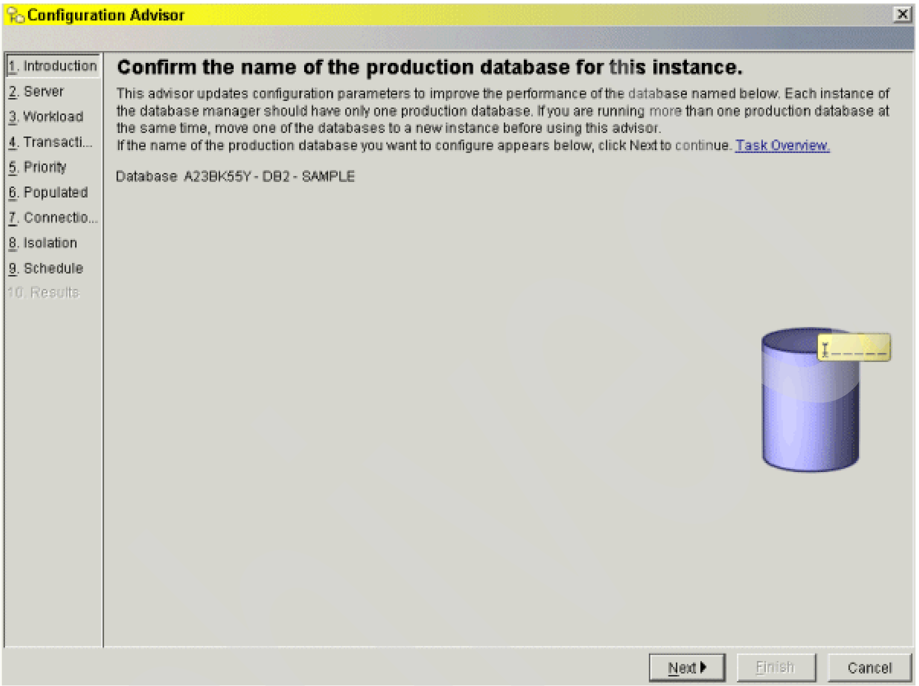
A23BK55Y - DB2 - SAMPLE
Package: NULLID.SYSSH200 Section number: 65
Explain date and time: 02/28/2003 8:35:06 PM Parallelism: None
Total cost(timerons): 10,192.15

Figure 3-17 Command Center - Access Plan tab.

Note: When viewing the access plan, the timerons value is a relative cost measurement to compare against other SQL statements. It is not related to seconds or milliseconds.

3.4 Configuration Advisor

When a database is created, DB2 sets default configuration values for the database. These values may not fit your applications' needs and require modifications. After the database system is running for a period of time, the database administrator will need to evaluate and tune the database system to maximize database performance. DB2 provides a useful graphical tool called Configuration Advisor for DBAs to evaluate and tune a database system. This tool takes you step-by-step to gather information about your day-to-day database requirements. Once you have completed the process, Configuration Advisor displays recommended values for the configuration parameters in your database configuration which can be updated either immediately, or later by specifying the Configuration Advisor to use Task Center.



Confirm the name of the production database for this instance.

This advisor updates configuration parameters to improve the performance of the database named below. Each instance of the database manager should have only one production database. If you are running more than one production database at the same time, move one of the databases to a new instance before using this advisor.
If the name of the production database you want to configure appears below, click Next to continue. Task Overview.

Database A23BK55Y - DB2 - SAMPLE

Figure 3-18 Configuration Advisor

To get to the Configuration Advisor in Control Center, perform the following:

- Navigate down the directory tree in the left pane until you see the SAMPLE database icon.
- Right-click the SAMPLE database icon and select **Configuration Advisor** from the drop-down list.

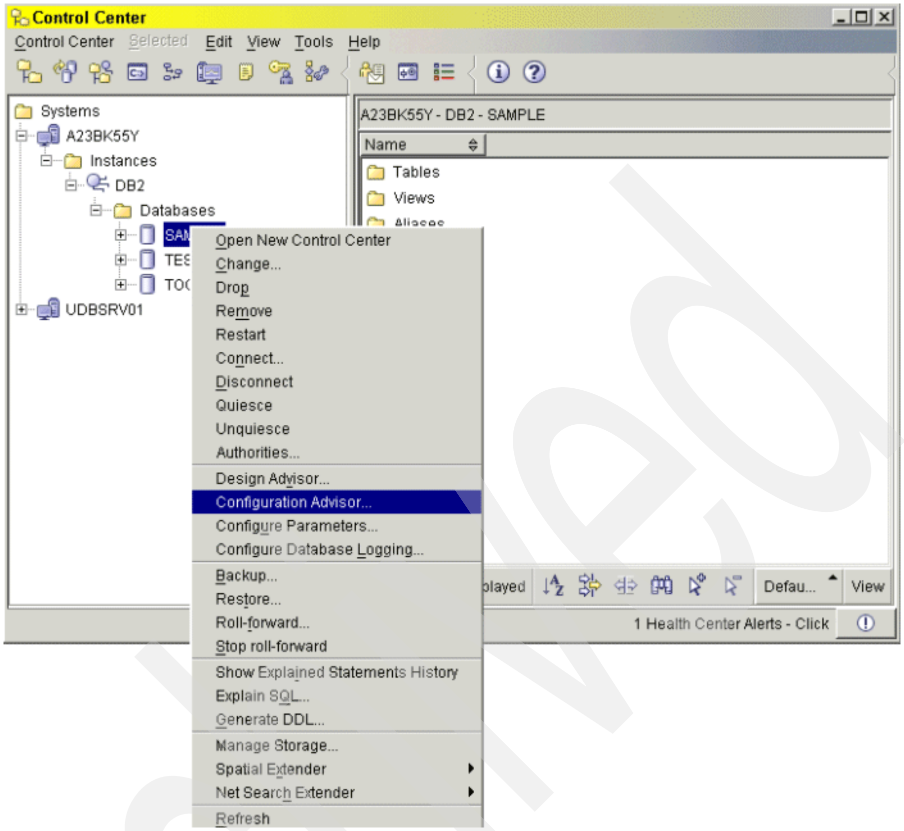


Figure 3-19 Opening Configuration Advisor through Control Center

- Proceed through the ten steps in the left pane of the Configuration Advisor window to configure your database configuration file.

We discuss the Configuration Advisor in more detail in 6.5.2, “Using the DB2 configuration parameters” on page 291.

3.5 Design Advisor

DB2 provides the Design Advisor Wizard, which you can use to determine which indexes to create or drop for a given set of SQL statements. The Design Advisor Wizard will determine the best set of indexes for a given workload. A workload contains a set of weighted SQL statements that can include queries as well as updates. The wizard will recommend which new indexes to create, which existing indexes to keep, and which existing indexes to drop. The Design Advisor is a convenient, useful tool for application development and performance tuning.

To launch the Design Advisor from the Control Center:

- Navigate down the directory tree in the left pane until you see the SAMPLE database icon.
- Right-click the SAMPLE database icon and select **Design Advisor** from the drop-down list.

  Or, select **Tools —> Wizards**, and then select **Design Advisor** (Figure 3-20).

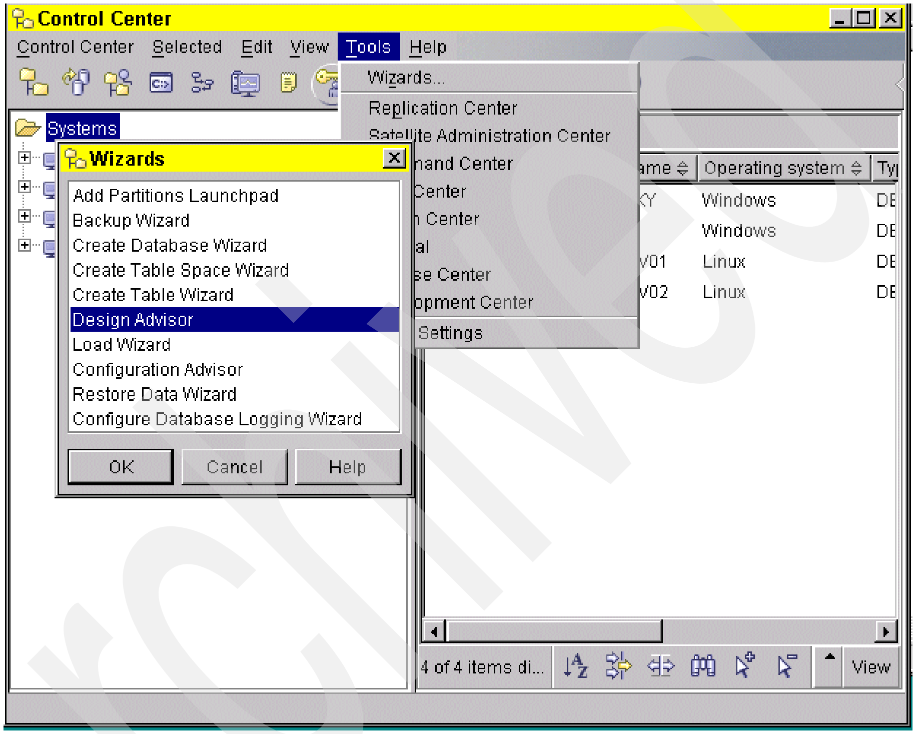
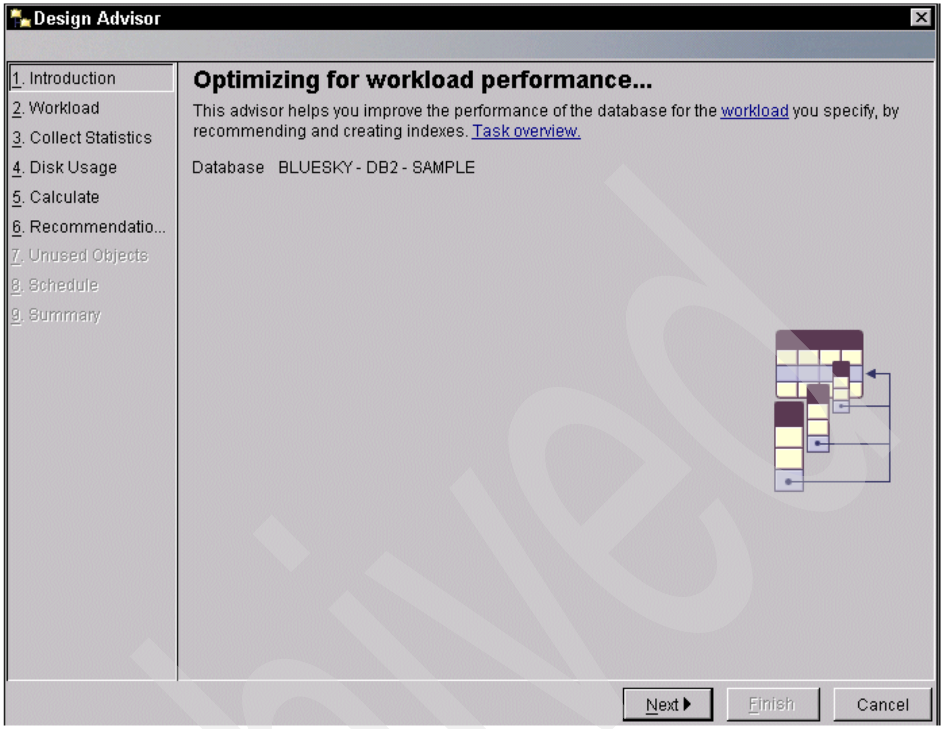


Figure 3-20 Launch Design Advisor

- In the Database Selection window select the system, Instance, and Database
- Proceed through the ten steps in the left pane of the Design Advisor window (Figure 3-21) to configure your database configuration file.



Optimizing for workload performance...

This advisor helps you improve the performance of the database for the workload you specify, by recommending and creating indexes. Task overview.

Database BLUESKY - DB2 - SAMPLE

Figure 3-21 Design Advisor

We discuss Design Advisor in more detail in 6.5.3, “SQL tuning” on page 298. Chapter 3 of the redbook *DB2 UDB Exploitation of the Windows Environment,* SG24-6893, contains detailed information for Configuration Advisor and Design Advisor.

3.6 Task Center

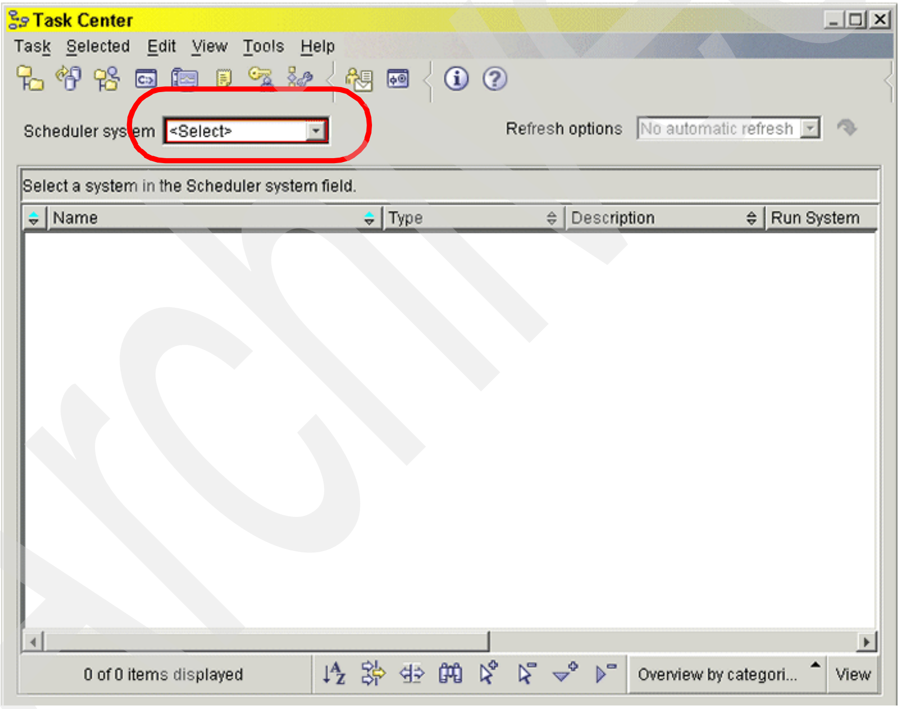
Task Center is a robust tool to manage the scheduling and execution of DB2 Command and operating system scripts. Upon completion of a task, Task Center can create notifications to be sent to your e-mail or pager. A history of task executions is also produced with detailed output, such as start time, end time, duration, return code, and resulting output.

Many options are at your disposal in the Task Center. Perhaps you want to schedule a database backup to run every night at 3 a.m. and receive a notification via e-mail to verify success or failure. Maybe you want to write a new script and execute it immediately. Or possibly, you need to run a batch operation at the end of the day to update rows in a specific table.

Notes:

- It is necessary to have the Database Administration Server (DAS) created and running for the Scheduler to run tasks in the Task Center. In Chapter 2, we created the DAS and verified that it was active.
- It is necessary to have the tools catalog database created for the Task Center to store task information, such as the commands to be run, schedule, notification, and completion actions associated with a task; and run results.

Let's begin by opening the Task Center from the Control Center. Click the **Tools** menu drop-down list located at the top of the Control Center interface and select **Task Center**. Figure 3-22 shows the Task Center's main interface.



Scheduler system <Select>

Refresh options No automatic refresh

Select a system in the Scheduler system field.

0 of 0 items displayed

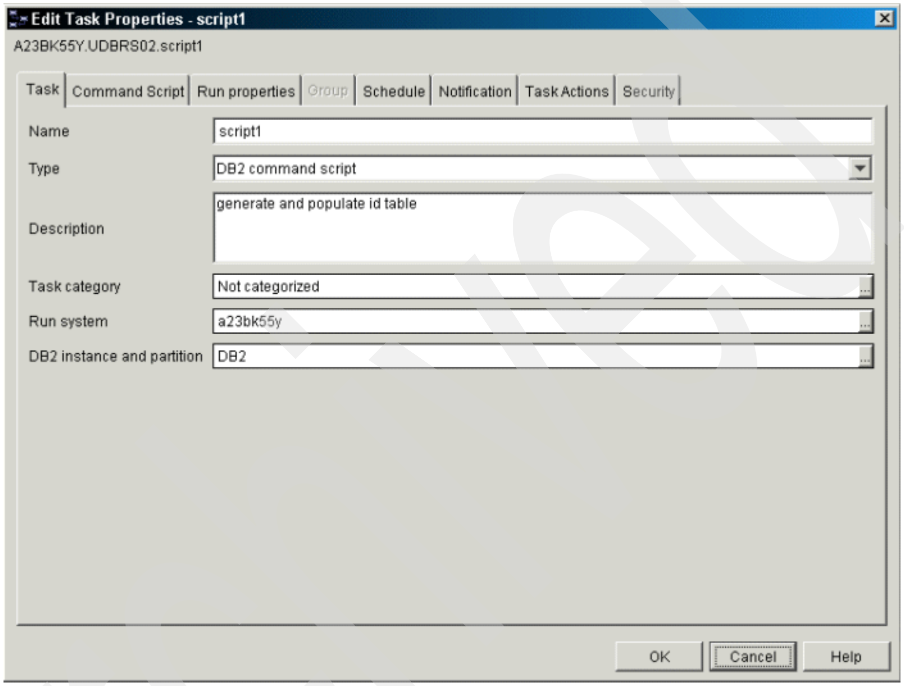
Figure 3-22 Task Center

In the Task Center, the main interface will display all tasks available for execution. To view the scripts available, select a system from the Scheduler system drop down list field. Since we created a script in 3.3, "Command Center" on page 51, there should be a script called *script1* displayed after refreshing the main interface. Recall that we previously saved *script1* by leaving the *Save to the Task*

Center check box checked in “Script tab” on page 55. The *script1* file will not run automatically since it has not been scheduled yet.

Let’s schedule the *script1* file to run at a specified time:

1. From the Task Center’s main interface, double-click the *script1* file to open the Edit Task Properties window (Figure 3-23).

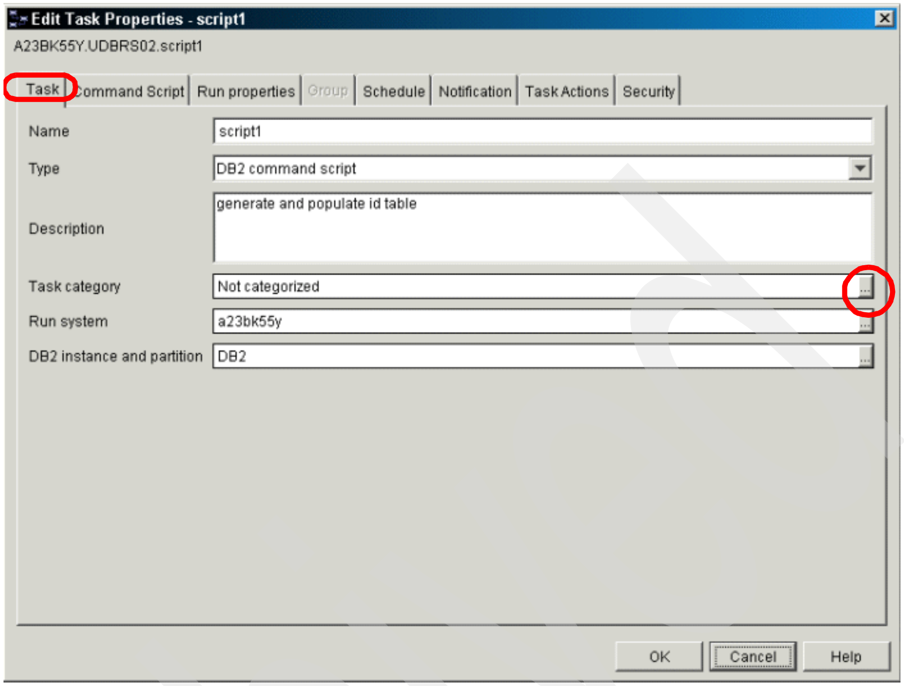


A23BK55Y.UDBRS02.script1

Name: script1
Type: DB2 command script
Description: generate and populate id table
Task category: Not categorized
Run system: a23bk55y
DB2 instance and partition: DB2

Figure 3-23 Task Center - scheduling a task

2. There are eight tabs in the Edit Task Properties window. By default, the first tab, called Task, will be active, displaying information about the task to schedule. Currently the Task Category field reflects `Not Categorized`. Use the browse function associated with the Task Category field to display the Select task Categories pop-up window (Figure 3-24).



A23BK55Y.UDBRS02.script1

Name: script1

Type: DB2 command script

Description: generate and populate id table

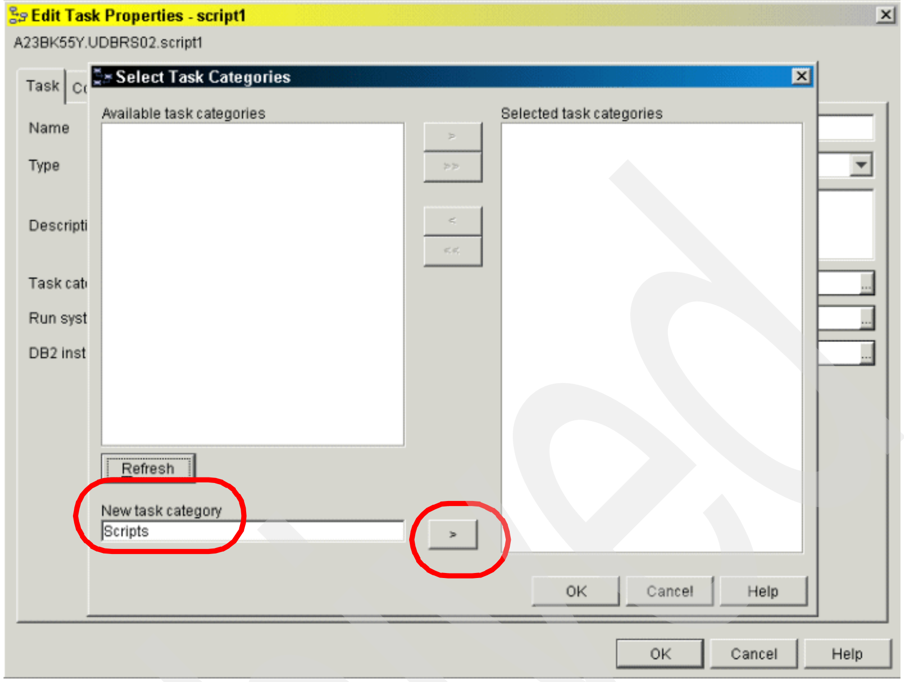
Task category: Not categorized

Run system: a23bk55y

DB2 instance and partition: DB2

Figure 3-24 Task Center - Task category for scheduling a task

3. At the bottom of the Select Task Categories window is a New Task Category text field. Type `Scripts` for the name of the new category in the text field and select the button (arrow) immediately to the right of the New Task Category text field to add the new category to the Selected task categories field. Click **OK** (Figure 3-25).



Edit Task Properties - script1

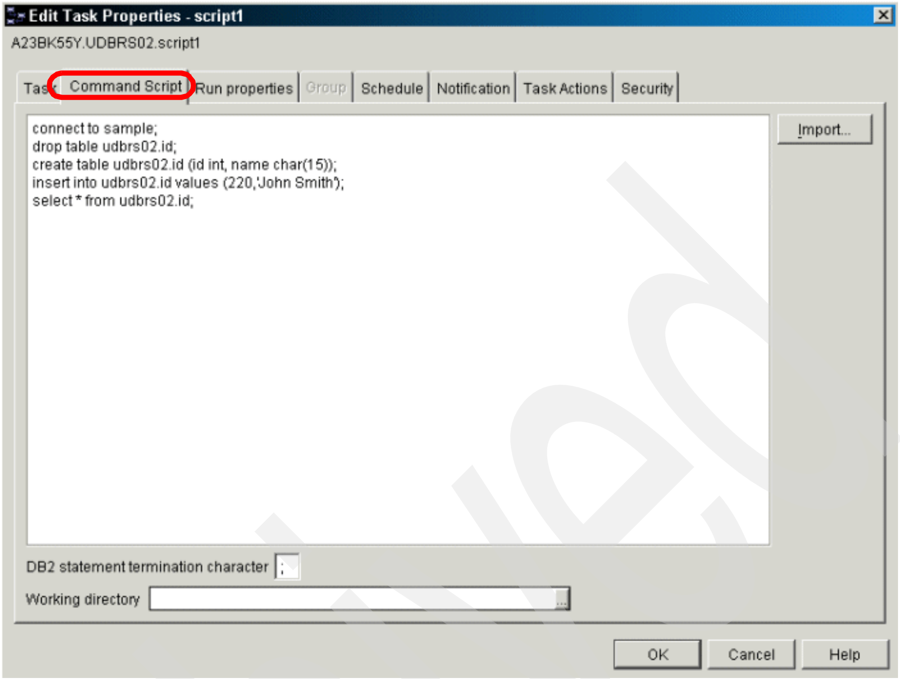
A23BK55Y.UDBRS02.script1

Select Task Categories

New task category: Scripts

Figure 3-25 Task Center - New task category for scheduling a task

4. Select the **Command Script** tab and verify the script to be executed (Figure 3-26).



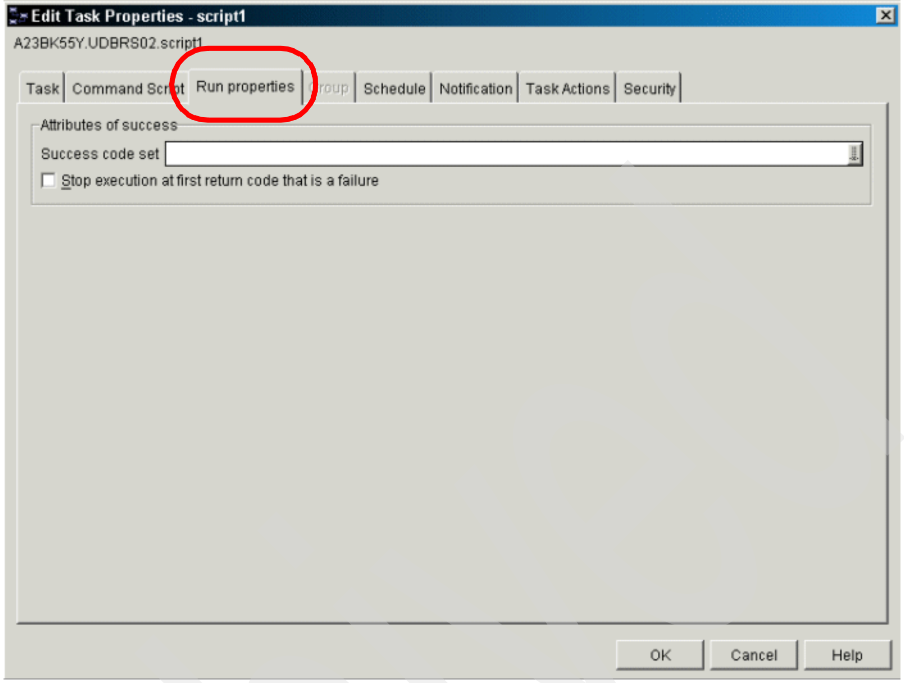
```
connect to sample;
drop table udbrs02.id;
create table udbrs02.id (id int, name char(15));
insert into udbrs02.id values (220,'John Smith');
select * from udbrs02.id;
```

DB2 statement termination character: ;

Working directory:

Figure 3-26 Task Center - Command Script for scheduling a task

5. Select the **Run Properties** tab and leave the **Stop execution at first return code that is a failure** check box unchecked. Since we are running a series of SQL statements, it is possible we may run into an error with the **drop table** statement if the table does not exist. The script should still run if the table does not exist. Click **OK** (Figure 3-27).



A23BK55Y.UDBRS02.script1

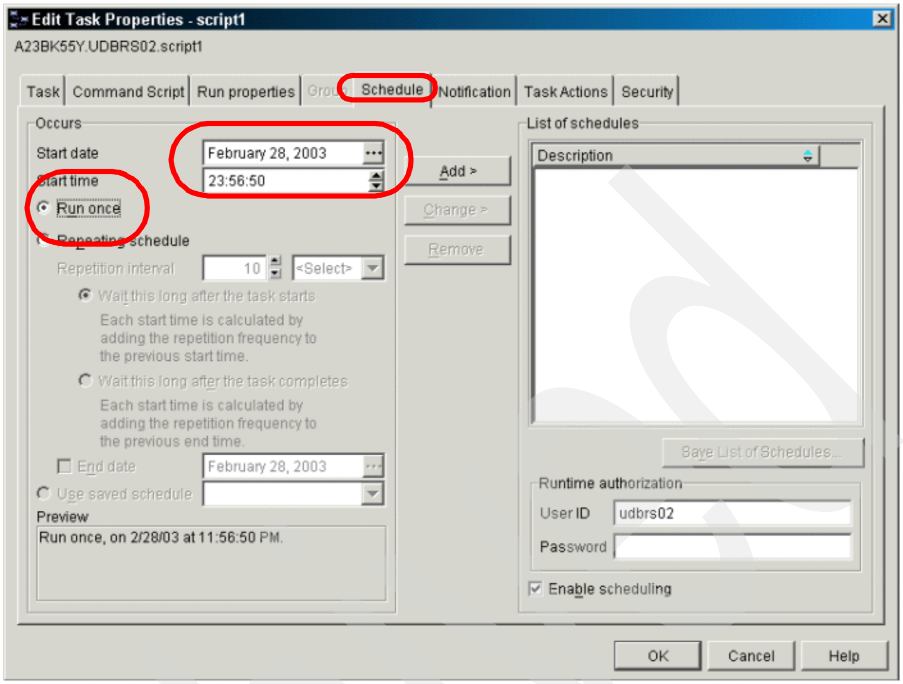
Attributes of success

Success code set

□ Stop execution at first return code that is a failure

Figure 3-27 Task Center - Run properties for scheduling a task

6. Select the **Schedule** tab. You will notice that there are two panes, one on the left and the other on the right. The left pane displays scheduling information while the right pane displays the name of the newly generated schedule for the task. In the left pane enter a start date and start time for your script to run (that is, some time in the near future). Select the radio button **Run once** since the script will be executed only once (Figure 3-28).



A23BK55Y.UDBRS02.script1

Occurs

Start date: February 28, 2003

Start time: 23:56:50

● Run once

○ Repeating schedule

Repetition interval: 10 <Select>

● Wait this long after the task starts

Each start time is calculated by adding the repetition frequency to the previous start time.

○ Wait this long after the task completes

Each start time is calculated by adding the repetition frequency to the previous end time.

□ End date: February 28, 2003

○ Use saved schedule

Preview

Run once, on 2/28/03 at 11:56:50 PM.

List of schedules

Description

Runtime authorization

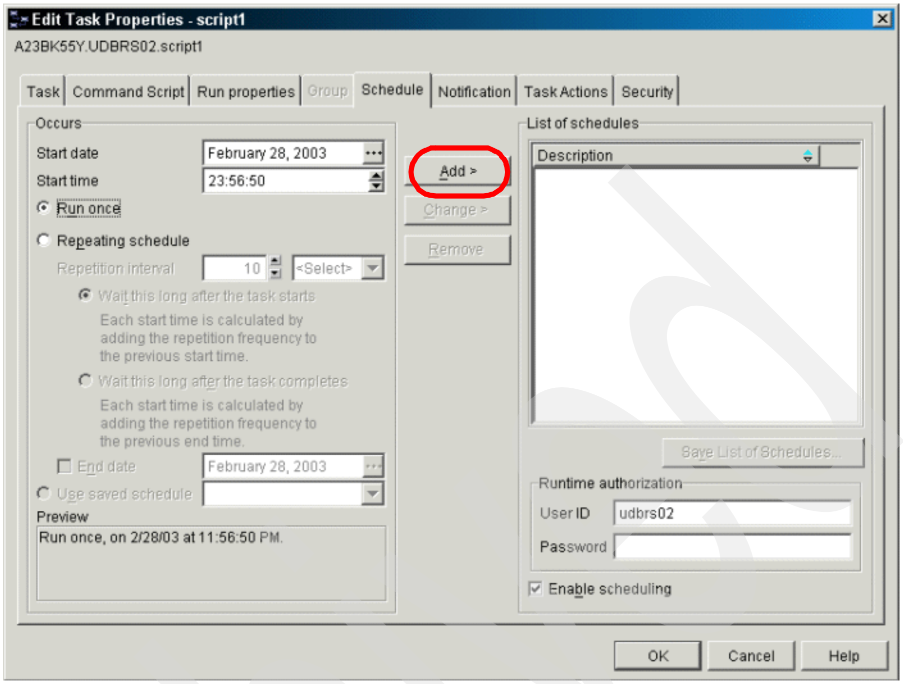
User ID: udbrs02

Password:

☑ Enable scheduling

Figure 3-28 Task Center - schedule a task

7. Select the **Add** button in the center of the Schedule interface to add the task to the Description box in the right pane (Figure 3-29).



Edit Task Properties - script1

A23BK55Y.UDBRS02.script1

Occurs

Start date: February 28, 2003

Start time: 23:56:50

● Run once

○ Repeating schedule

Repetition interval: 10 <Select>

● Wait this long after the task starts

Each start time is calculated by adding the repetition frequency to the previous start time.

○ Wait this long after the task completes

Each start time is calculated by adding the repetition frequency to the previous end time.

□ End date: February 28, 2003

○ Use saved schedule

Preview: Run once, on 2/28/03 at 11:56:50 PM.

List of schedules

Runtime authorization

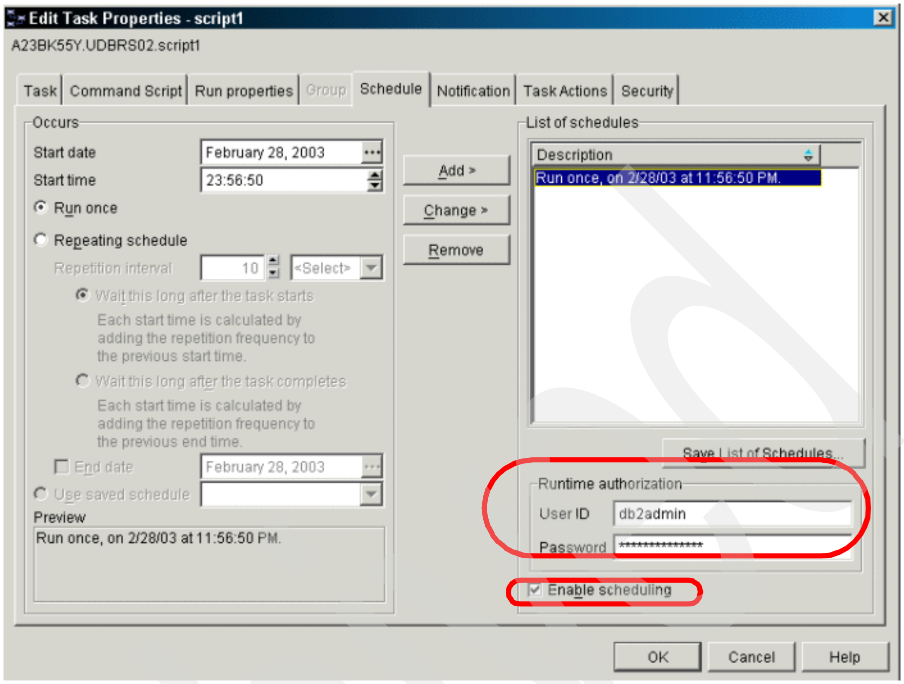
User ID: udbrs02

Password:

☑ Enable scheduling

Figure 3-29 Task Center - Add a task to schedule

8. Below the Description box is a Runtime authorization displaying your user account. The runtime ID is the user ID under which the task will run. The user ID must exist on the system where the script will be run. In our example, we use the **admin server account** and **password**. Leave **Enable scheduling** checked (Figure 3-30).



A23BK55Y.UDBRS02.script1

Occurs

Start date: February 28, 2003

Start time: 23:56:50

● Run once

○ Repeating schedule

Repetition interval: 10 <Select>

● Wait this long after the task starts

Each start time is calculated by adding the repetition frequency to the previous start time.

○ Wait this long after the task completes

Each start time is calculated by adding the repetition frequency to the previous end time.

□ End date: February 28, 2003

○ Use saved schedule

Preview

Run once, on 2/28/03 at 11:56:50 PM.

List of schedules

Description
Run once, on 2/28/03 at 11:56:50 PM.

Runtime authorization

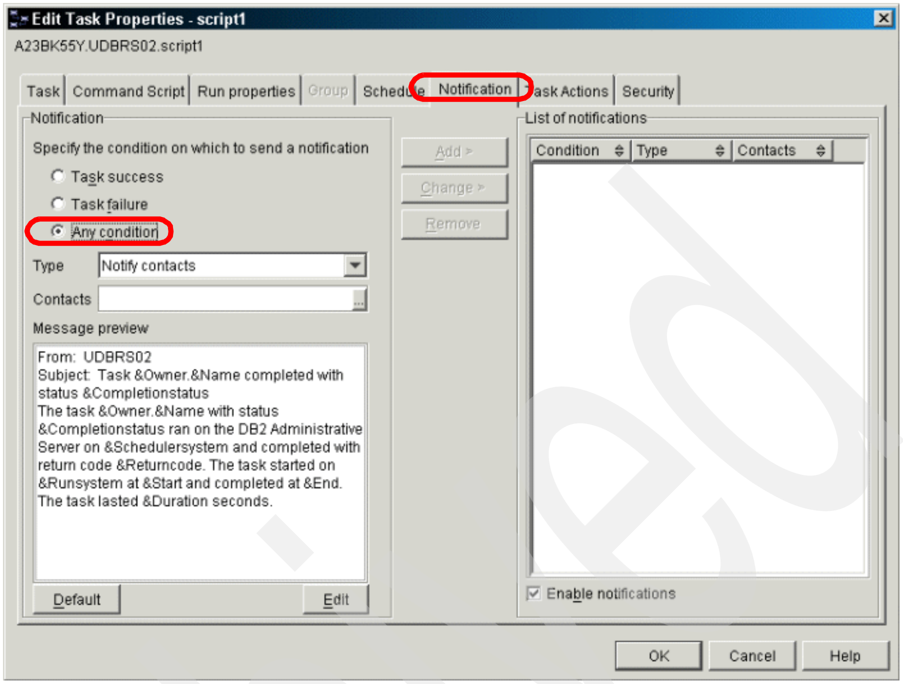
User ID: db2admin

Password: ***************

☑ Enable scheduling

Figure 3-30 Task Center - specify user id for scheduling a task

9. Select the **Notification** tab. In the left pane, select the **Any condition** radio button so that success or failure conditions will trigger the Task Center to send a notification (Figure 3-31).



Notification

Specify the condition on which to send a notification

○ Task success

○ Task failure

● Any condition

Type: Notify contacts

Contacts:

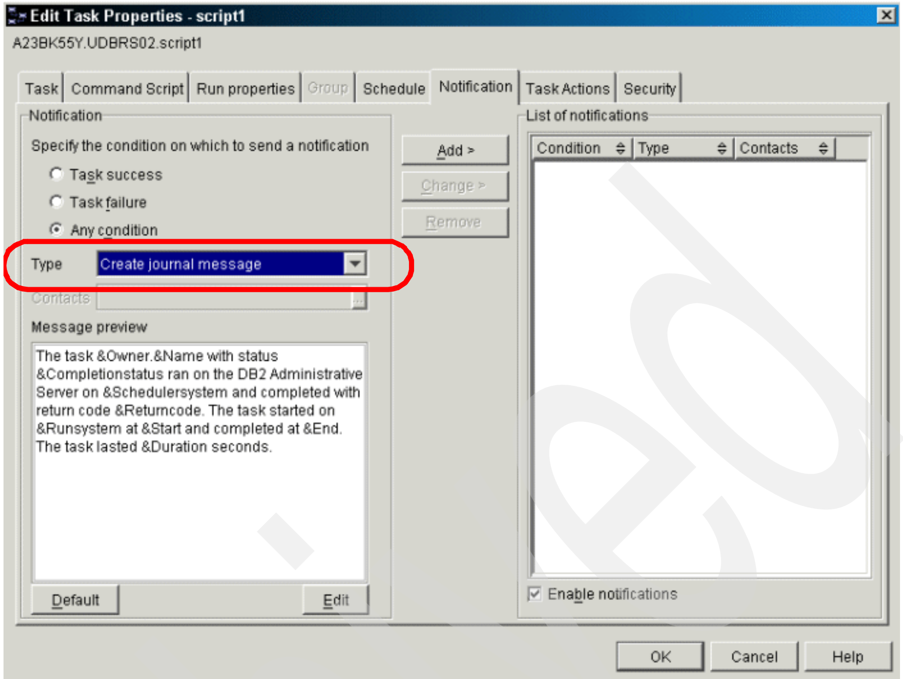
Message preview

From: UDBRS02
Subject: Task &Owner.&Name completed with status &Completionstatus
The task &Owner.&Name with status &Completionstatus ran on the DB2 Administrative Server on &Schedulersystem and completed with return code &Returncode. The task started on &Runsystem at &Start and completed at &End. The task lasted &Duration seconds.

☑ Enable notifications

Figure 3-31 Task Center - set notification for scheduling a task

10. In the Type drop down list field, select **Create journal message** for the type of notification (Figure 3-32).



Specify the condition on which to send a notification

○ Task success

○ Task failure

● Any condition

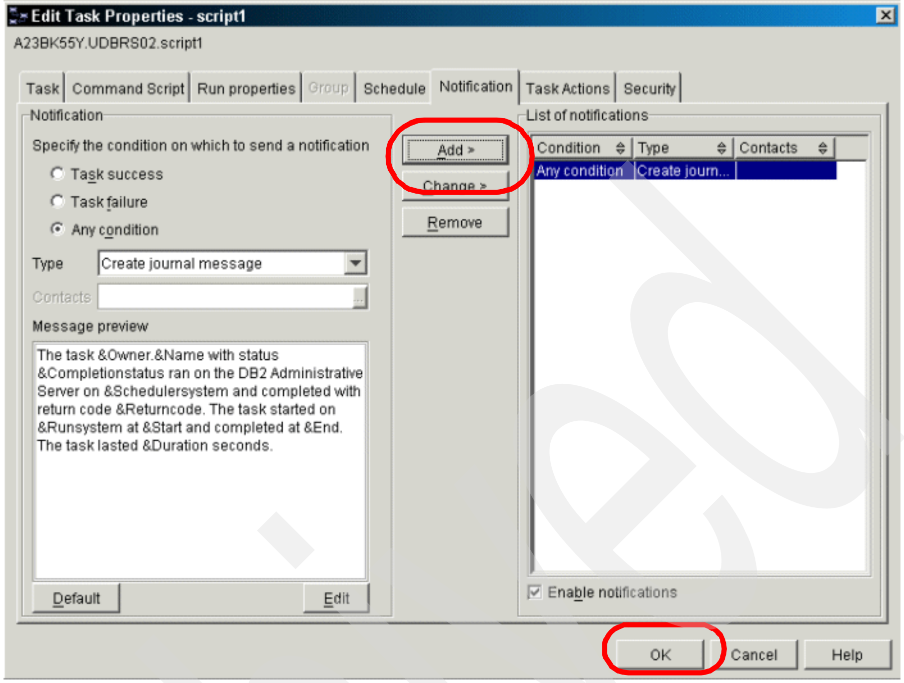
Type: Create journal message

☑ Enable notifications

The task &Owner.&Name with status &Completionstatus ran on the DB2 Administrative Server on &Schedulersystem and completed with return code &Returncode. The task started on &Runsystem at &Start and completed at &End. The task lasted &Duration seconds.

Figure 3-32 Task Center - Notification type of scheduling a task

11.Select the **Add** button in the center of the Notification interface to add the notification task to the List of notifications box in the right pane. Leave the **Enable notification** check box checked. Click **OK** (Figure 3-33).



A23BK55Y.UDBRS02.script1

Notification

Specify the condition on which to send a notification

○ Task success
○ Task failure
● Any condition

Type: Create journal message

Contacts:

Message preview:

The task &Owner.&Name with status &Completionstatus ran on the DB2 Administrative Server on &Schedulersystem and completed with return code &Returncode. The task started on &Runsystem at &Start and completed at &End. The task lasted &Duration seconds.

List of notifications

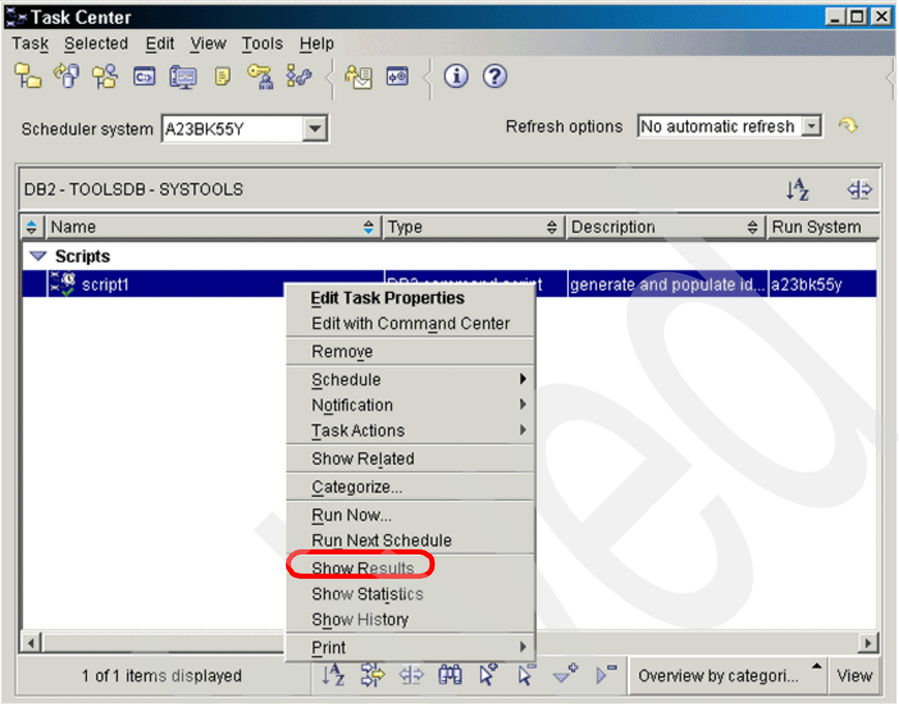
Condition	Type	Contacts
Any condition	Create journ...	

☑ Enable notifications

Figure 3-33 Task Center - Add task to a schedule

You should now be returned to the main Task Center interface where you can monitor the task that was just created. In the upper right corner of the Task Center interface, there is an automatic refresh drop down box, which will allow you to refresh frequently to view the most current standings of your task.

12. Once the task completes, right-click the task name and select **Show Results** from the list (Figure 3-34).



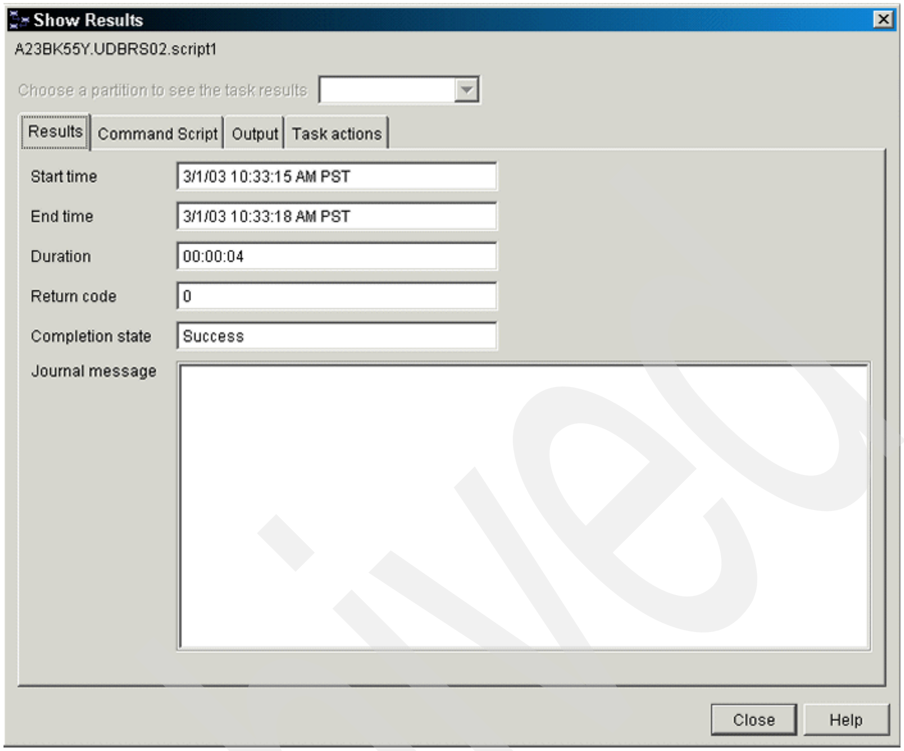
Scheduler system: A23BK55Y

Refresh options: No automatic refresh

Name	Type	Description	Run System
Scripts			
script1	DB2 command script	generate and populate id...	a23bk55y

Figure 3-34 Task Center - show task result for a scheduled task

13. The Show Results window will appear and show the task execution summary (Figure 3-35).



A23BK55Y.UDBRS02.script1

Choose a partition to see the task results

Start time: 3/1/03 10:33:15 AM PST

End time: 3/1/03 10:33:18 AM PST

Duration: 00:00:04

Return code: 0

Completion state: Success

Journal message:

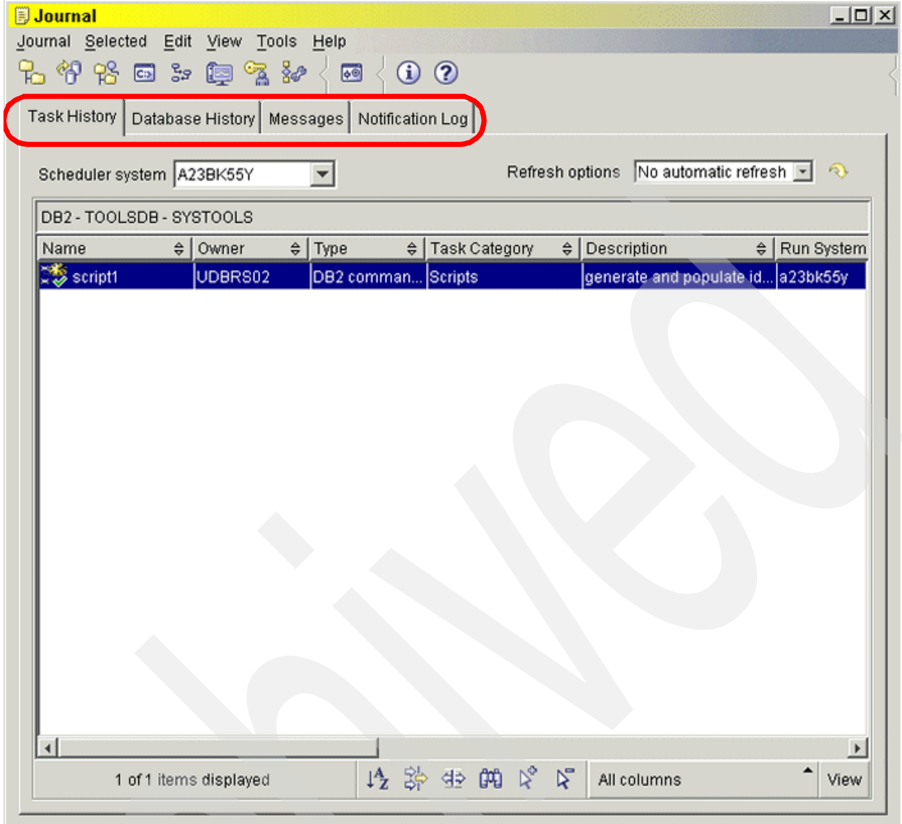
Figure 3-35 Task Center - schedule a task

A helpful paper regarding the Task Center is “Administration made easier: Scheduling and Automation in DB2 Universal Database”. This paper can be located at this Web site:

http://www7b.software.ibm.com/dmdd/library/techarticle/0301gartner/0301gartner.html

3.7 Journal

The Journal keeps a running history of all interactive command and script executions submitted using the DB2 GUI tools. Since we have the Task Center opened from our previous section, open the Journal from there. From the Task Center, select **Tools —> Journal** from the drop-down menu list at the top of the interface.



Scheduler system A23BK55Y

Refresh options No automatic refresh

DB2 - TOOLSDB - SYSTOOLS

Name	Owner	Type	Task Category	Description	Run System
script1	UDBRS02	DB2 comman...	Scripts	generate and populate id...	a23bk55y

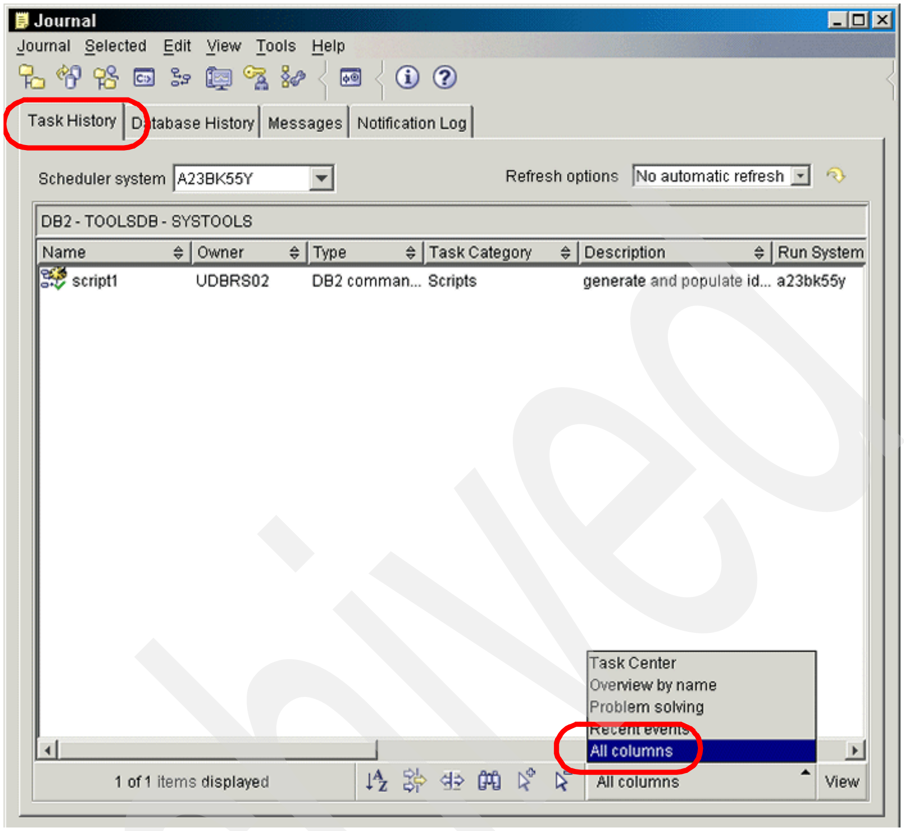
1 of 1 items displayed

Figure 3-36 Journal

As shown in Figure 3-36, informative messages are categorized into four types: Task History, Database History, Messages, and Notification Log.

3.7.1 Task History tab

The Task History tab displays records of completed tasks. You can view the executed tasks in various ways. **All columns** is the default view. To display tasks in a different view, select the **All columns** button in the lower right hand corner of the Journal interface. A pull up list will provide you with different options for displaying previously executed tasks (Figure 3-37).



Scheduler system A23BK55Y

Refresh options No automatic refresh

DB2 - TOOLSDB - SYSTOOLS

Name	Owner	Type	Task Category	Description	Run System
script1	UDBRS02	DB2 comman...	Scripts	generate and populate id...	a23bk55y

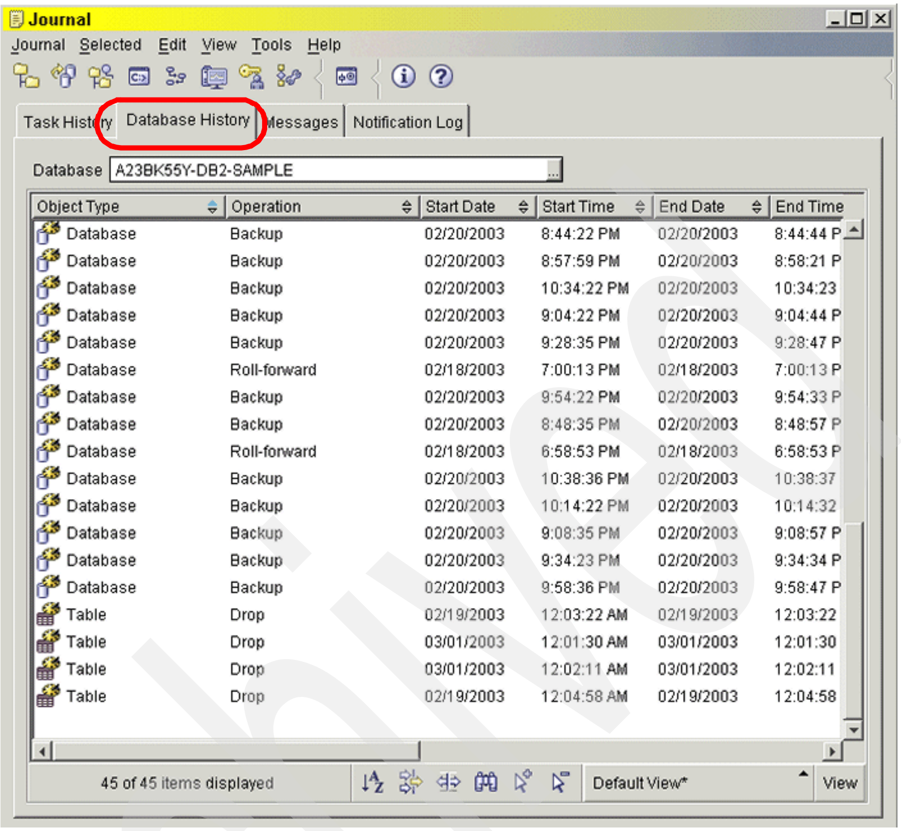
1 of 1 items displayed

All columns

Figure 3-37 Journal - Task History tab

3.7.2 Database History tab

The Database History tab displays records associated with tasks affecting databases or tablespaces. For instance each database backup performed via the DB2 GUI tools will be displayed here. This is a useful benefit for the database administrator who manages database backups. The Restore Data Wizard is also available by right-clicking on the database backup record (Figure 3-38).



Database A23BK55Y-DB2-SAMPLE

Object Type	Operation	Start Date	Start Time	End Date	End Time
Database	Backup	02/20/2003	8:44:22 PM	02/20/2003	8:44:44 P
Database	Backup	02/20/2003	8:57:59 PM	02/20/2003	8:58:21 P
Database	Backup	02/20/2003	10:34:22 PM	02/20/2003	10:34:23
Database	Backup	02/20/2003	9:04:22 PM	02/20/2003	9:04:44 P
Database	Backup	02/20/2003	9:28:35 PM	02/20/2003	9:28:47 P
Database	Roll-forward	02/18/2003	7:00:13 PM	02/18/2003	7:00:13 P
Database	Backup	02/20/2003	9:54:22 PM	02/20/2003	9:54:33 P
Database	Backup	02/20/2003	8:48:35 PM	02/20/2003	8:48:57 P
Database	Roll-forward	02/18/2003	6:58:53 PM	02/18/2003	6:58:53 P
Database	Backup	02/20/2003	10:38:36 PM	02/20/2003	10:38:37
Database	Backup	02/20/2003	10:14:22 PM	02/20/2003	10:14:32
Database	Backup	02/20/2003	9:08:35 PM	02/20/2003	9:08:57 P
Database	Backup	02/20/2003	9:34:23 PM	02/20/2003	9:34:34 P
Database	Backup	02/20/2003	9:58:36 PM	02/20/2003	9:58:47 P
Table	Drop	02/19/2003	12:03:22 AM	02/19/2003	12:03:22
Table	Drop	03/01/2003	12:01:30 AM	03/01/2003	12:01:30
Table	Drop	03/01/2003	12:02:11 AM	03/01/2003	12:02:11
Table	Drop	02/19/2003	12:04:58 AM	02/19/2003	12:04:58

45 of 45 items displayed

Figure 3-38 Journal - Database History tab

To display all of the records associated with database tasks:

1. Select the database field browse function (...) in the upper middle area of the Database History interface (Figure 3-39).

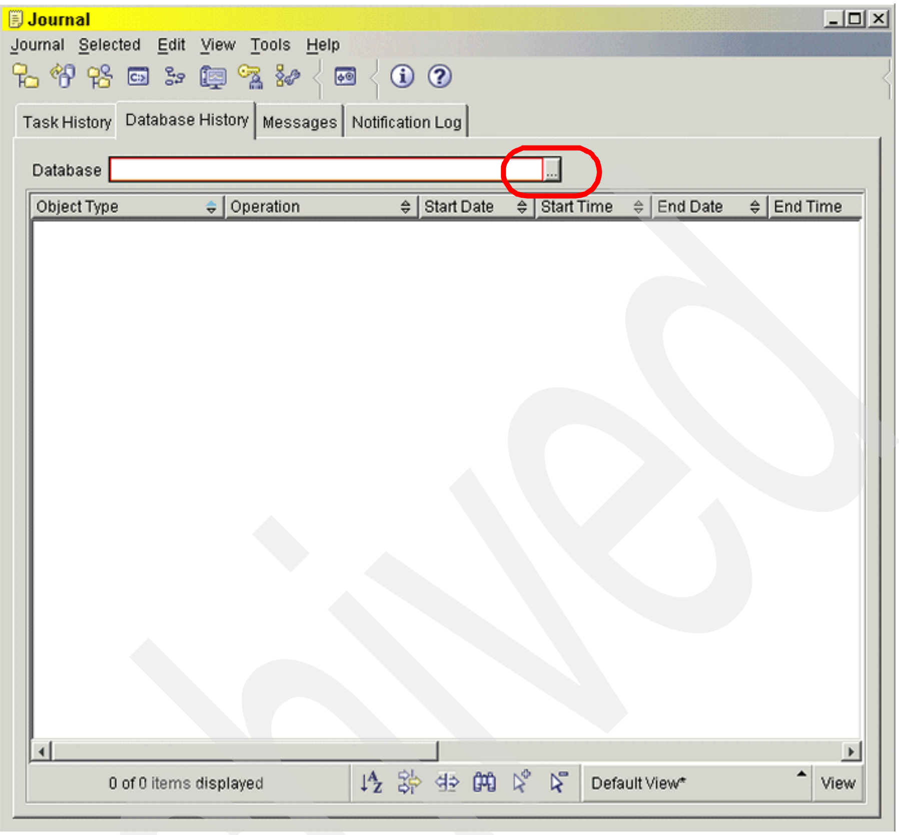
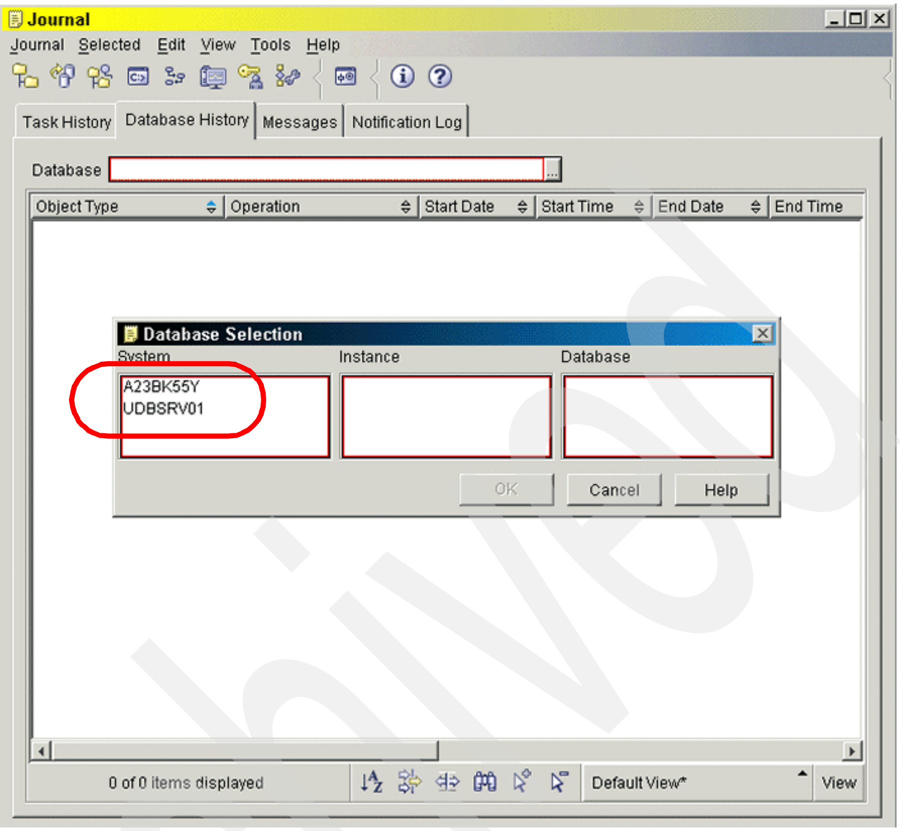


Figure 3-39 Journal - browse history under Database History tab

2. A Database Selection window will appear with three fields: System, Instance, and Database. Starting with System, select a system name inside the System field (Figure 3-40).



Database

Object Type	Operation	Start Date	Start Time	End Date	End Time

0 of 0 items displayed

System	Instance	Database
A23BK55Y UDBSRV01		

Figure 3-40 Journal - Database selection under Database History tab

3. An instance (`DB2`) should appear in the Instance field. Select it (Figure 3-41).

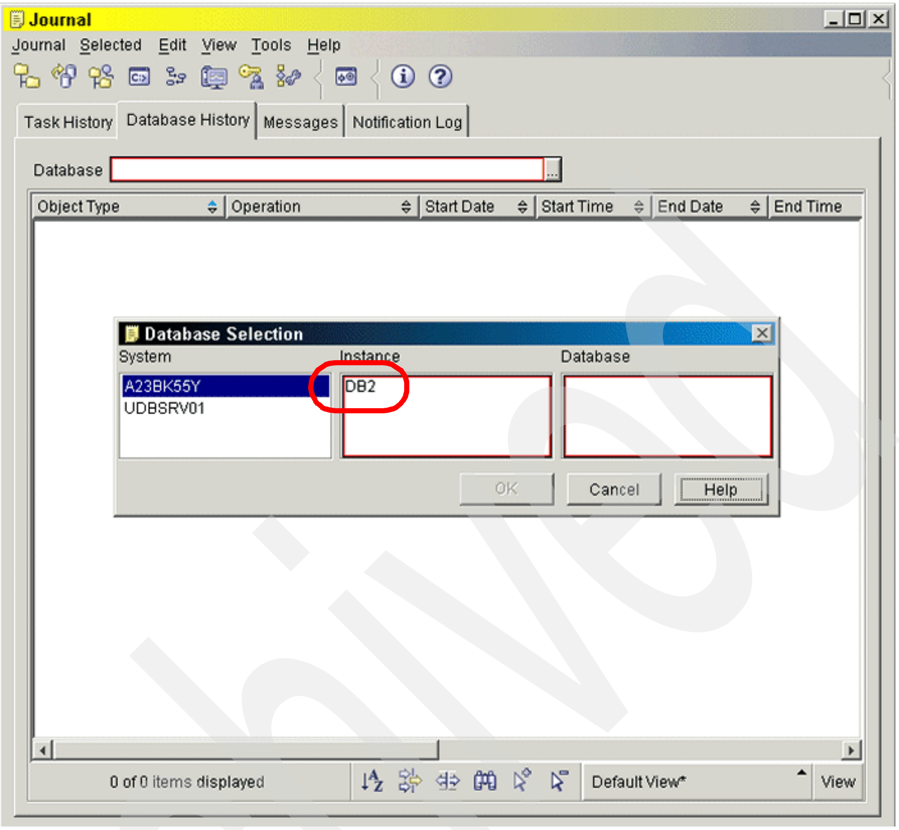


Figure 3-41 Journal - select a instance under Database History tab

4. Two databases will be available for selection in the Database field. Select the SAMPLE database. Click **OK** (Figure 3-42).

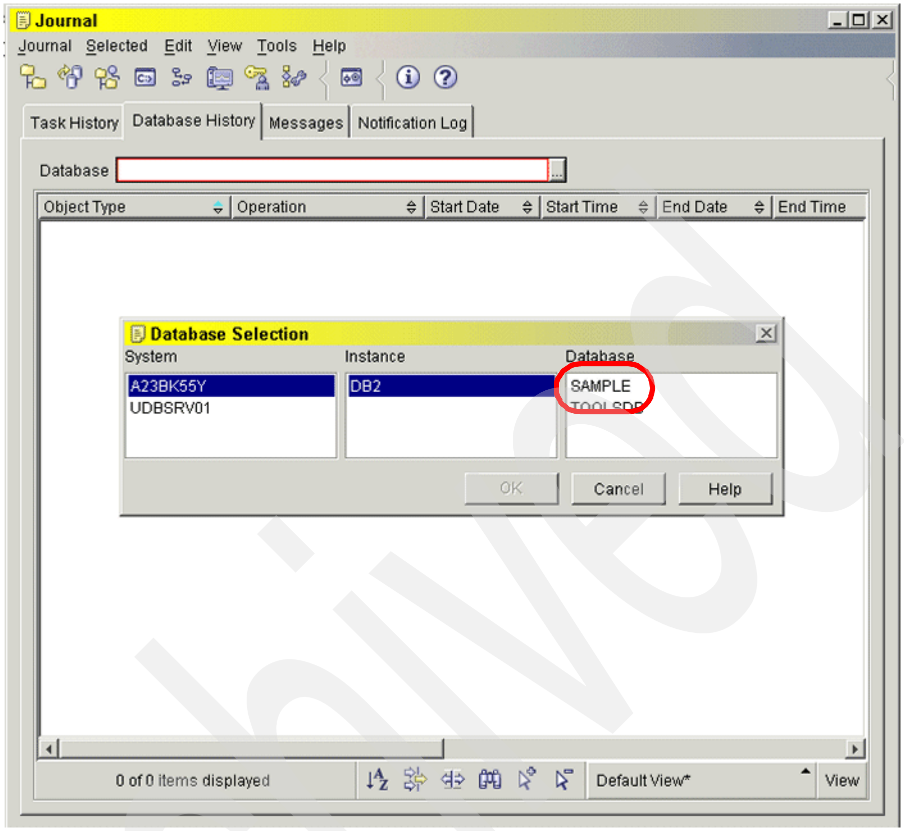
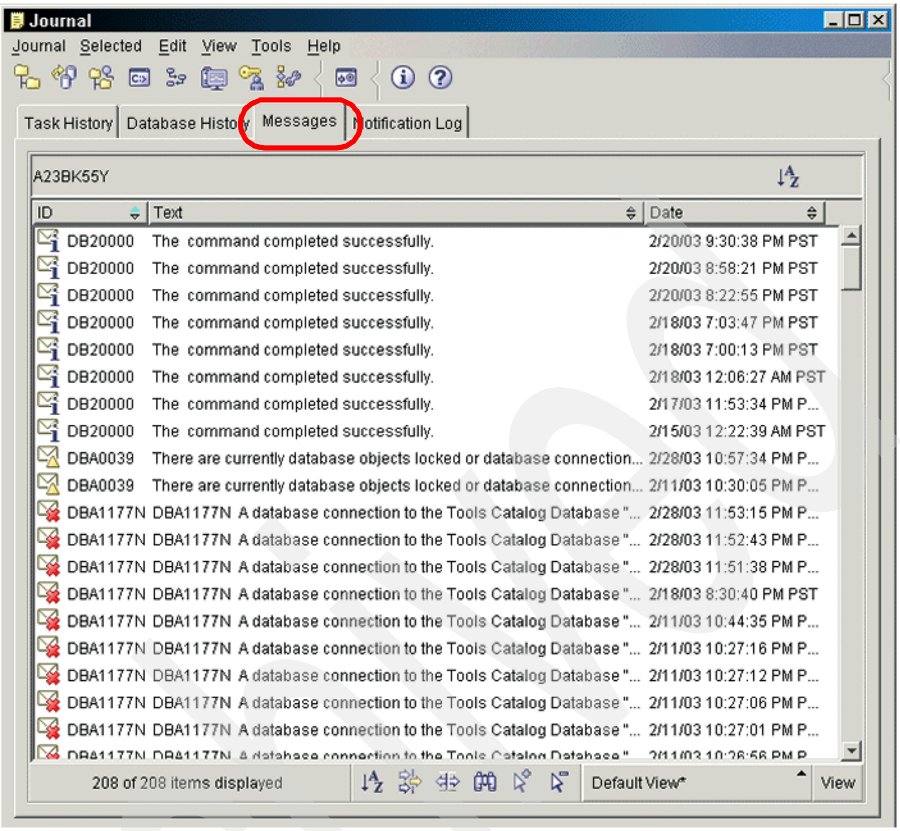


Figure 3-42 Journal - select a database under Database History tab

3.7.3 Messages tab

The Messages tab displays the informational, warning, and error messages associated with all interactive command and script executions submitted using the DB2 GUI tools (Figure 3-43).



ID	Text	Date
DB20000	The command completed successfully.	2/20/03 9:30:38 PM PST
DB20000	The command completed successfully.	2/20/03 8:58:21 PM PST
DB20000	The command completed successfully.	2/20/03 8:22:55 PM PST
DB20000	The command completed successfully.	2/18/03 7:03:47 PM PST
DB20000	The command completed successfully.	2/18/03 7:00:13 PM PST
DB20000	The command completed successfully.	2/18/03 12:06:27 AM PST
DB20000	The command completed successfully.	2/17/03 11:53:34 PM P...
DB20000	The command completed successfully.	2/15/03 12:22:39 AM PST
DBA0039	There are currently database objects locked or database connection...	2/28/03 10:57:34 PM P...
DBA0039	There are currently database objects locked or database connection...	2/11/03 10:30:05 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/28/03 11:53:15 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/28/03 11:52:43 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/28/03 11:51:38 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/18/03 8:30:40 PM PST
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/11/03 10:44:35 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/11/03 10:27:16 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/11/03 10:27:12 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/11/03 10:27:06 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/11/03 10:27:01 PM P...
DBA1177N	DBA1177N A database connection to the Tools Catalog Database "...	2/11/03 10:26:56 PM P...

Figure 3-43 Journal - Messages tab

Since the Messages tab is meant to be a short summary of possible problems, they may not be descriptive enough to troubleshoot a problem. You can take the code provided in the ID field and use Command Center to gather more information about the code. For instance, if you find SQL00204 in the ID field, simply open the Command Center and under the Interactive tab type the command below:

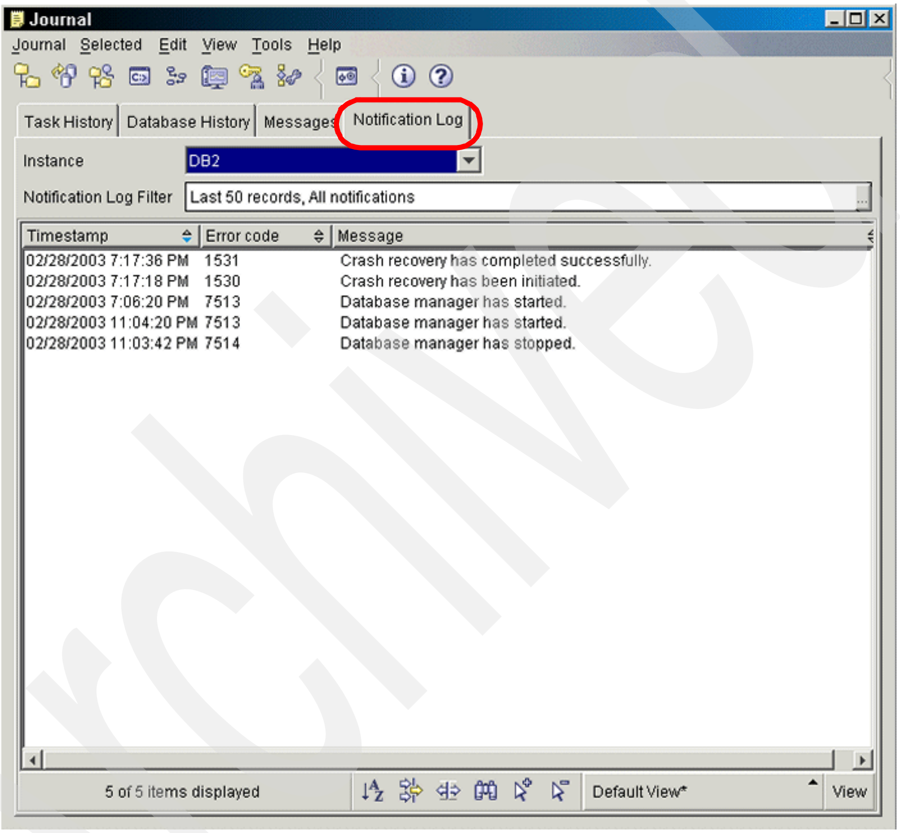
```
? SQL00204;
```

Then execute the command using Execute icon in the upper left corner of the Command Center interface. The error message will display allowing you to troubleshoot the problem more efficiently.

3.7.4 Notification Log tab

The Notification Log tab displays all alert notifications including Health Monitor alerts. These alerts are geared more towards the health of your DB2 instance and should be monitored regularly for potential problems with the DB2 instance.

Figure 3-44 shows the Notification Log tab.



Instance: DB2

Notification Log Filter: Last 50 records, All notifications

Timestamp	Error code	Message
02/28/2003 7:17:36 PM	1531	Crash recovery has completed successfully.
02/28/2003 7:17:18 PM	1530	Crash recovery has been initiated.
02/28/2003 7:06:20 PM	7513	Database manager has started.
02/28/2003 11:04:20 PM	7513	Database manager has started.
02/28/2003 11:03:42 PM	7514	Database manager has stopped.

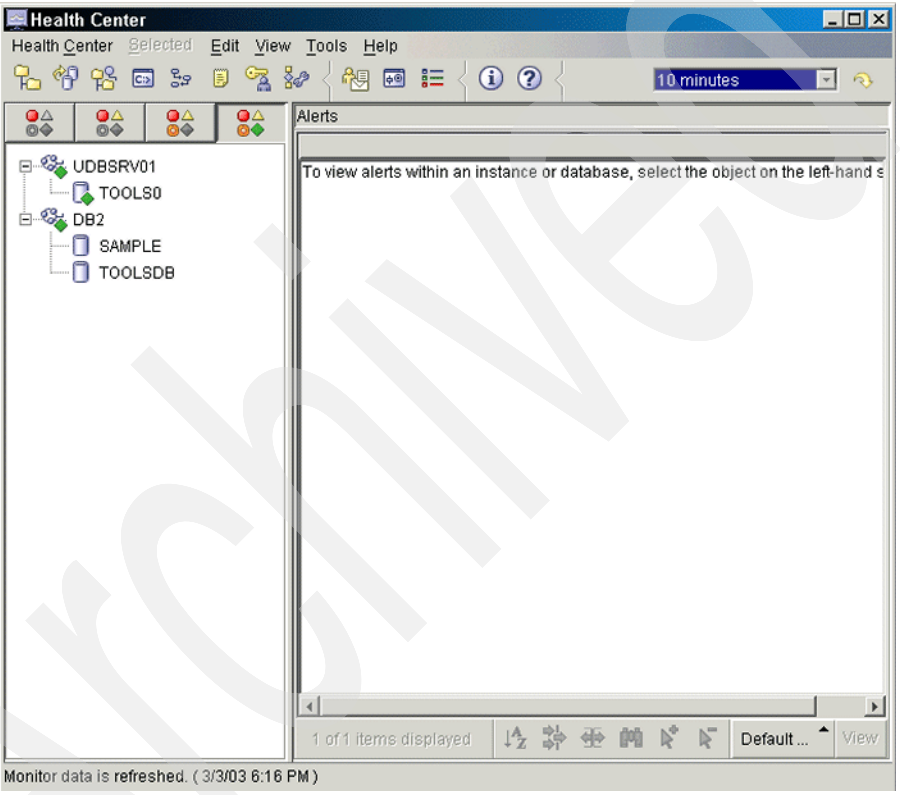
Figure 3-44 Journal - Notification Log tab

3.8 Health Center and Memory Visualizer

The Health Center and Memory Visualizer are the newest monitoring tools that have joined the DB2 tools collection. With Version 8.1, DB2 expands its autonomic computing foundation with new self-healing capabilities. Both of these tools have been designed with the autonomic technology concept in mind. To accomplish this functionality, DB2 is bundled with many tools and special commands to allow a broad analysis of the DB2 environment.

DB2 implements proactive and self-healing techniques with the concept of *monitoring by exception*. Now on DB2, when an alert condition detected, DB2 recommends some options for what you can do, and offers tools to do what you choose, immediately.

Figure 3-45 shows Health Center's main interface.

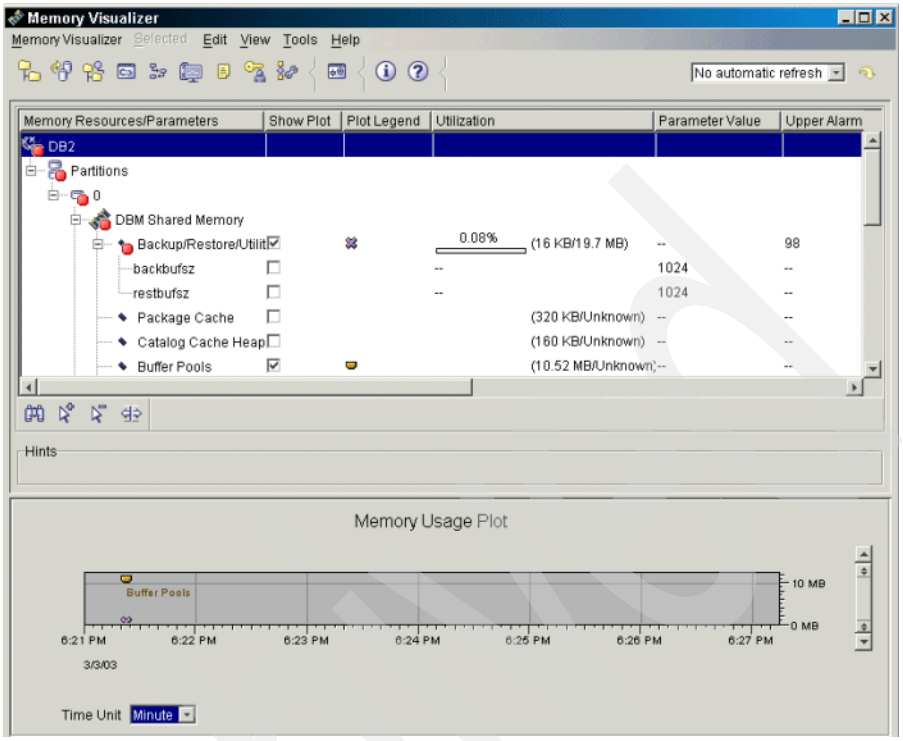


To view alerts within an instance or database, select the object on the left-hand s

Monitor data is refreshed. (3/3/03 6:16 PM)

Figure 3-45 Health Center

Figure 3-46 shows Memory Visualizer's main interface.



No automatic refresh

Memory Resources/Parameters	Show Plot	Plot Legend	Utilization	Parameter Value	Upper Alarm
DB2					
Partitions					
0					
DBM Shared Memory					
Backup/Restore/Utilit	☑	✖	0.08% (16 KB/19.7 MB)	--	98
backbufsz	□		--	1024	--
restbufsz	□		--	1024	--
Package Cache	□		(320 KB/Unknown)	--	--
Catalog Cache Heap	□		(160 KB/Unknown)	--	--
Buffer Pools	☑	●	(10.52 MB/Unknown)	--	--

Time Unit Minute

Figure 3-46 Memory Visualizer

3.8.1 Health Center

The Health Center monitoring tool can be used to monitor the state and utilization of many parts of the database manager, the databases.

In this section, we show a step-by-step utilization of DB2 Health Center. If you need more information on health indicators, refer to Appendix A in the *DB2 UDB Version 8 manual System Monitor Guide and Reference*, SC09-4847.

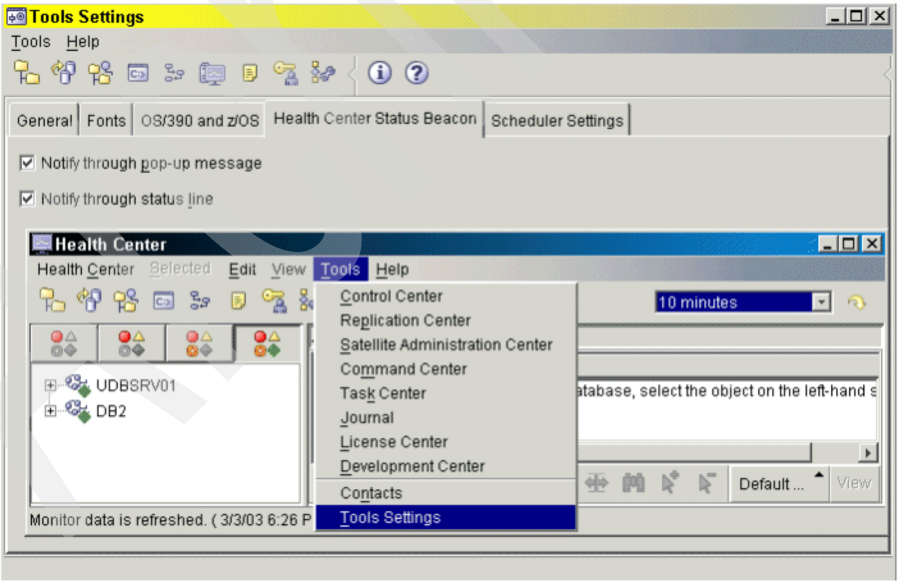
Health indicators are used by the Health Monitor to evaluate specific aspects of database manager or database performance. A health indicator is a specific measurement that gauges the healthiness of some aspect of an object (for example, a database). Criteria are applied to the measurement to determine healthiness, where the criteria applied depends on the type of health indicator. A determination of unhealthiness based on the criteria generates an alert.

The health indicators returned by the Health Monitor fall into the following types:

- **Threshold-based indicators** are a measurement that represents a statistic (on a continuous range of values) of the behavior of the object. Warning and alarm threshold values define boundaries or zones for normal, warning, and alarm ranges. Threshold-based health indicators have three valid states: Normal, Warning, or Alarm.
- **State-based indicators** are a measurement that represents a finite set of two or more distinct states of an object that defines whether or not the database object or resource is operating normally. One of the states is normal and all others are considered non-normal. State-based health indicators have two valid states: Normal or Attention.

Depending on the configuration of the DB2 instance, the Health Monitor, and the client tools, the following actions can occur when the Health Monitor generates an alert:

- An entry is written in the administration notification log, which you can read from the Journal.
- The Health Center status beacon appears in the lower right corner of the DB2 GUI Tools window. You can set the status beacon options opening on any DB2 GUI tool by clicking the menu **Tools —>Tools Settings**, and then clicking the tab Health Center Status Beacon (Figure 3-47).



☑ Notify through pop-up message

☑ Notify through status line

atabase, select the object on the left-hand s

Monitor data is refreshed. (3/3/03 6:26 P

Figure 3-47 Status Beacon configuration options

- A script or task is executed.

- An e-mail or pager message is sent to the contacts that you specify for this instance.

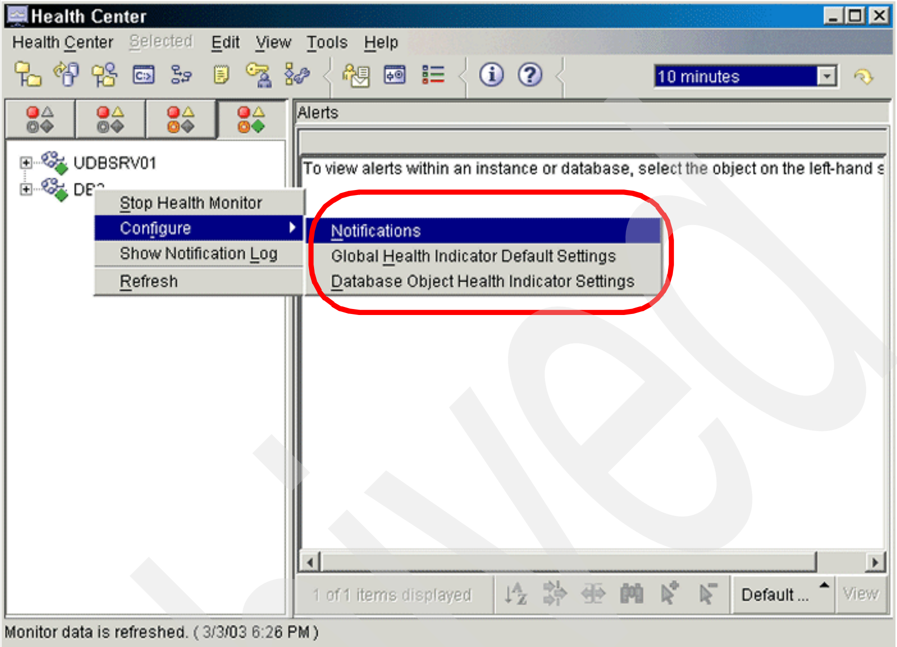
These are some of the key tasks that you can perform with the Health Center:

- View the status of the database environment: Beside each object in the navigation tree, an icon indicates the most severe alert for the object (or for any objects contained by that object). For example, a green diamond icon beside an instance means that the instance and the databases contained in the instance do not have any alerts.
- View the alerts for an instance or a database: When you select an object in the navigation tree, alerts for that object are shown in the pane to the right.
- View detailed information about an alert, and recommended actions: When you double-click an alert, a notebook appears. The first page shows the details for the alert. The second page shows any recommended actions.
- Configure the Health Monitor settings for a specific object, and the default settings for an object type or for all objects within an instance.
- Select which contacts will be notified by an e-mail or pager message.
- Review the history of alerts for an instance: You can use the toggle buttons at the top of the navigation bar to filter the alerts according to their severity:
  - Displays only alarms (the most severe type of alert)
  - Displays alarms and warnings
  - Displays alarms, warnings, and attentions

Now, to demonstrate some of the functionality, we start a common Health Center instance and set some values. But before you begin, be sure that you have defined one instance and have created the scheduler or tools database server or created locally the TOOLSDB.

1. To start DB2 Health Center, go to **Start —> Programs —> IBM DB2 —> Monitoring Tools —> Health Center**. The Health Center will appear.
2. To see all the DB2 servers at your network, right-click the small button with the four icons highlighted or colored.
3. Right-click the server name that you want to monitor. A menu pop-up will be displayed (Figure 3-48):
   - The first option on the pop-up is to control the execution of Health Monitor. Its values change depending on the running status of Health Center monitor. If it is running, a Stop Health Monitor option will appear. And if it is stopped, a Start Health Monitor option appears. A small green diamond (beacon) is placed on the server icon where the Health Monitor is running.
   - The notification log of the events inside the Journal tool can be seen by clicking the option Show Notification Log.

- Positioning the mouse over the Configure option will open a second menu item.



To view alerts within an instance or database, select the object on the left-hand s

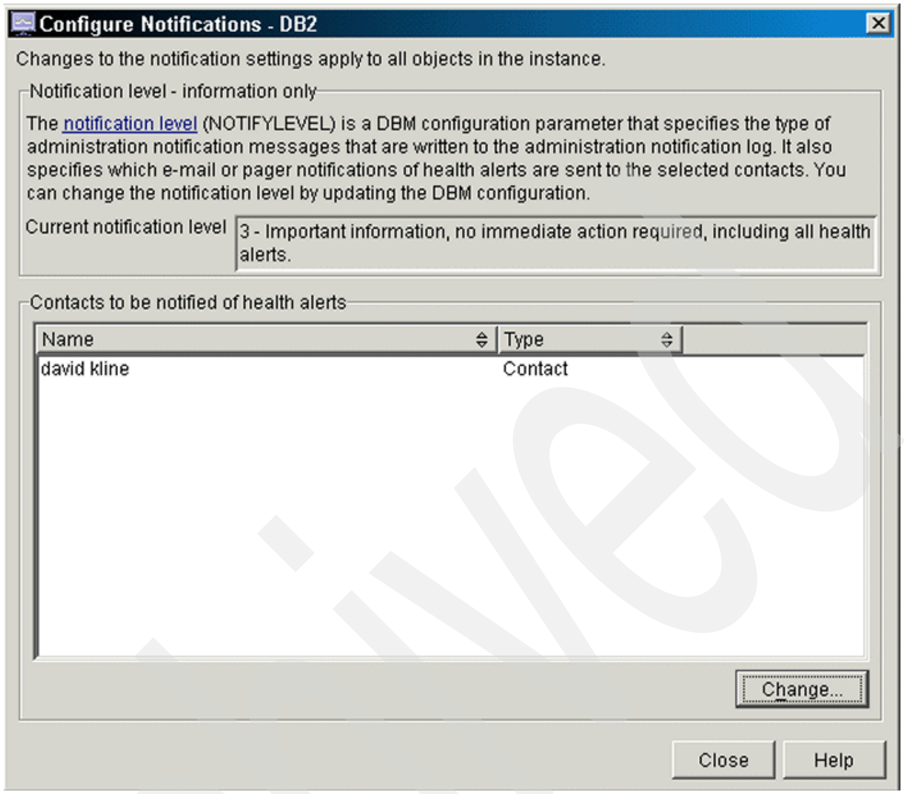
Monitor data is refreshed. (3/3/03 6:26 PM)

Figure 3-48 Health Center - server options

- Figure 3-48 shows Health Center configuration options.
- Select the option Notification, if you want to configure notifications via e-mail to an administrator or a group of administrators. A window will open showing the list of the contacts. In our case, there is a setting to send an e-mail to the administrator (Figure 3-49).

  You can set the contact list or change it by clicking the **Change** button. Two options show:

  - On the **Global Health Indication Default Settings** option, you can set the database manager indicators threshold values as well as the default settings to be used for any database, tablespace, or tablespace container.
  - On the **Database Health Indicator Settings** option, you can set the database indicators threshold values for each database, tablespace and tablespace container on the server instance.


Configure Notifications - DB2

Changes to the notification settings apply to all objects in the instance.

Notification level - information only

The notification level (NOTIFYLEVEL) is a DBM configuration parameter that specifies the type of administration notification messages that are written to the administration notification log. It also specifies which e-mail or pager notifications of health alerts are sent to the selected contacts. You can change the notification level by updating the DBM configuration.

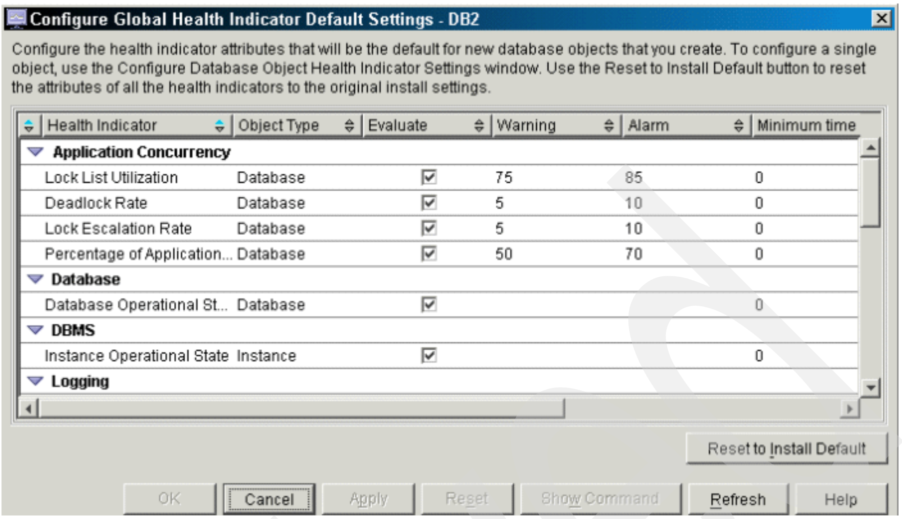
Current notification level: 3 - Important information, no immediate action required, including all health alerts.

Contacts to be notified of health alerts

Name	Type
david kline	Contact

Figure 3-49 Health Center - Configure Notifications

4. Click the **Global Health Indication Default Settings** option. The window to Configure Global Health Indication Default Settings appears (Figure 3-50).



Configure the health indicator attributes that will be the default for new database objects that you create. To configure a single object, use the Configure Database Object Health Indicator Settings window. Use the Reset to Install Default button to reset the attributes of all the health indicators to the original install settings.

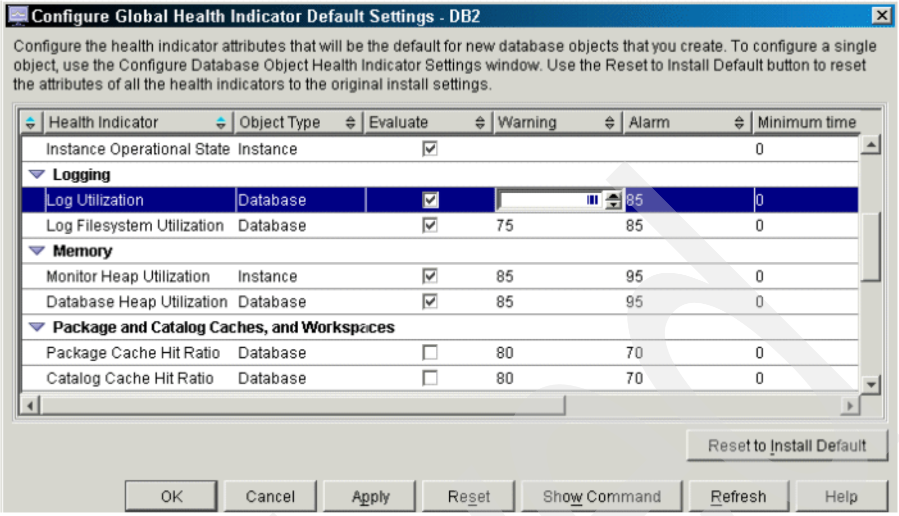
Health Indicator	Object Type	Evaluate	Warning	Alarm	Minimum time
Application Concurrency					
Lock List Utilization	Database	☑	75	85	0
Deadlock Rate	Database	☑	5	10	0
Lock Escalation Rate	Database	☑	5	10	0
Percentage of Application...	Database	☑	50	70	0
Database					
Database Operational St...	Database	☑			0
DBMS					
Instance Operational State	Instance	☑			0
Logging					

Figure 3-50 Health Center - Configure Global Health Indicator Default Settings

In this window, the columns that you should pay more attention to are:

- **Evaluate**: This option activates and deactivates the monitoring of the indicator itself.
- **Warning and Alarm**: These columns let you change the threshold value for each type of information that triggers notifications and action procedures.
- **Minimum time pre-alert**: Sometimes spike conditions occur, but it doesn't mean that the system is in a special condition. If you want to set a minimum time for the condition before starting notification, change the value inside this column. This value is in minutes.
- **Action**: In this column you can set up actions to do on warning or alert conditions. Clicking the cell corresponding to the action of the desired health indicator will bring up a button. By clicking this button, a dialog appears for you to set up actions.
- **Description**: In this column the name of the element or indicator on database monitor is shown.

5. Now change the warning value of Log Utilization from its original value to zero percent of utilization to generate a warning condition for demonstration purposes (Figure 3-51). Click **OK**, and if a window advises you that the updates are done, click **Close**.

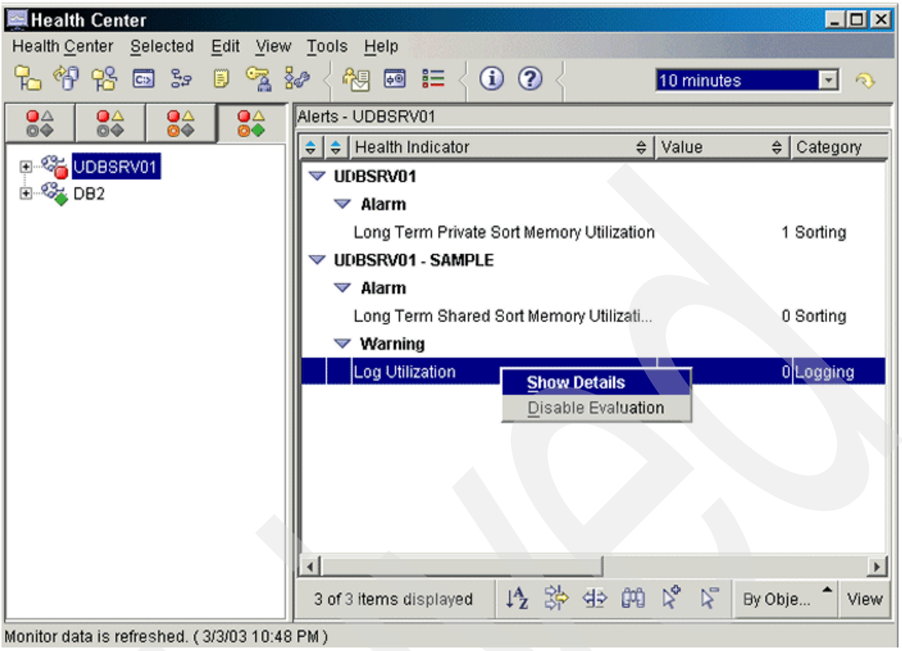


Configure the health indicator attributes that will be the default for new database objects that you create. To configure a single object, use the Configure Database Object Health Indicator Settings window. Use the Reset to Install Default button to reset the attributes of all the health indicators to the original install settings.

Health Indicator	Object Type	Evaluate	Warning	Alarm	Minimum time
Instance Operational State	Instance	☑			0
Logging					
Log Utilization	Database	☑		85	0
Log Filesystem Utilization	Database	☑	75	85	0
Memory					
Monitor Heap Utilization	Instance	☑	85	95	0
Database Heap Utilization	Database	☑	85	95	0
Package and Catalog Caches, and Workspaces					
Package Cache Hit Ratio	Database	□	80	70	0
Catalog Cache Hit Ratio	Database	□	80	70	0

Figure 3-51 Health Center - changing log utilization threshold to 0 percent

6. Now, you should be returning to the Health Center, wait for the period of time of the refresh. After this period, the dialog should display a warning condition. Right-click the warning indicator on the alert list box. If the alert begins flooding the list, you can disable the evaluation of the item by clicking the **Disable Evaluation** item. Click the **Show Details** option to proceed with the process of analyzing the warning condition (Figure 3-52).



Alerts - UDBSRV01

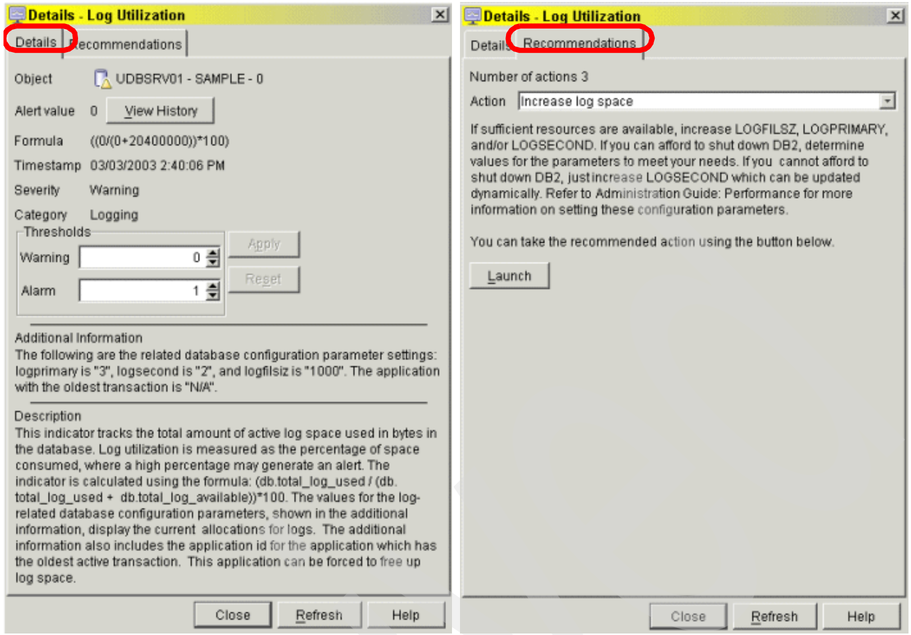
Health Indicator	Value	Category
UDBSRV01		
Alarm		
Long Term Private Sort Memory Utilization	1	Sorting
UDBSRV01 - SAMPLE		
Alarm		
Long Term Shared Sort Memory Utilizati...	0	Sorting
Warning		
Log Utilization	0	Logging

3 of 3 items displayed

Monitor data is refreshed. (3/3/03 10:48 PM)

Figure 3-52 Health Center - Show Details

7. A pop-up menu will display the alert details (Figure 3-53). This information contains most of the collectable data about the warning event itself and an explanation. Click the **Recommendations** tab; DB2 will analyze this event using the autonomic technology and enumerate some actions that you can do to solve the warning condition.



Object UDBSRV01 - SAMPLE - 0

Alert value 0

Formula ((0/(0+20400000))*100)

Timestamp 03/03/2003 2:40:06 PM

Severity Warning

Category Logging

Thresholds

Warning 0

Alarm 1

Additional Information

The following are the related database configuration parameter settings: logprimary is "3", logsecond is "2", and logfilsiz is "1000". The application with the oldest transaction is "N/A".

Description

This indicator tracks the total amount of active log space used in bytes in the database. Log utilization is measured as the percentage of space consumed, where a high percentage may generate an alert. The indicator is calculated using the formula: (db.total_log_used / (db. total_log_used + db.total_log_available))*100. The values for the log-related database configuration parameters, shown in the additional information, display the current allocations for logs. The additional information also includes the application id for the application which has the oldest active transaction. This application can be forced to free up log space.

Number of actions 3

Action Increase log space

If sufficient resources are available, increase LOGFILSZ, LOGPRIMARY, and/or LOGSECOND. If you can afford to shut down DB2, determine values for the parameters to meet your needs. If you cannot afford to shut down DB2, just increase LOGSECOND which can be updated dynamically. Refer to Administration Guide: Performance for more information on setting these configuration parameters.

You can take the recommended action using the button below.

Figure 3-53 Warning information

8. Here is the e-mail output sent to the user about the warning condition (Example 3-1).

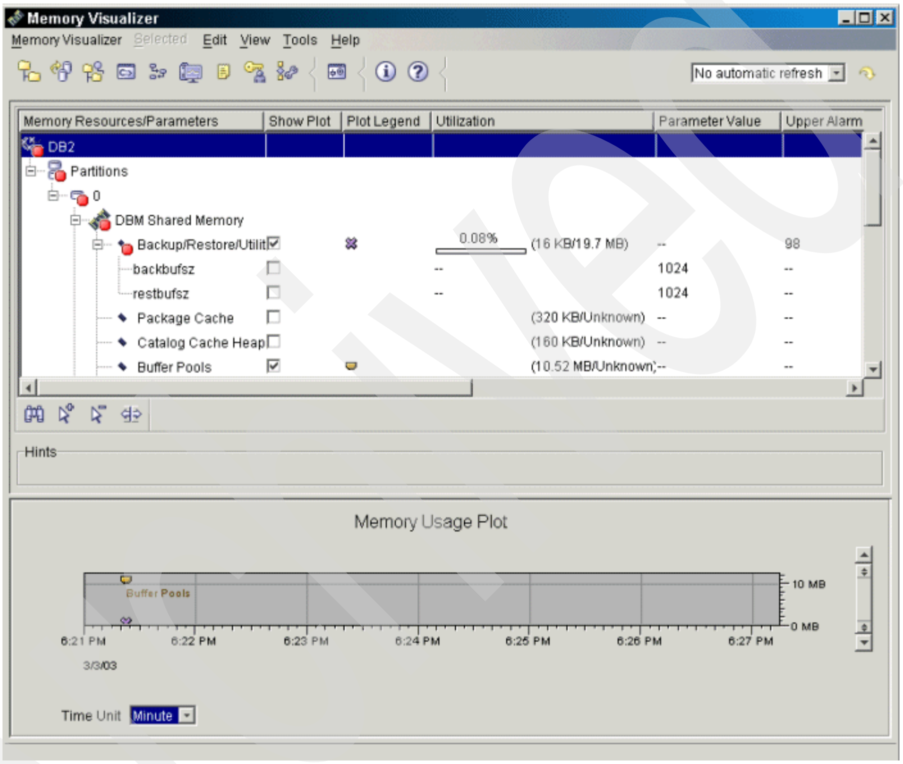
Example 3-1 E-mail output from Health Center

```
ADM10501W  Health indicator "Log Utilization" ("db.log_util") breached
the "upper" warning threshold of "0" with value "0" on "database"
"db2inst1.SAMPLE  ".

Calculation:
"(db.total_log_used/(db.total_log_used+db.total_log_available))*100;" =
"((0/(0+20400000))*100)" = "0".  History (Timestamp, Value, Formula):
",(03-03-2003 11:41:24.676389,0,((0/(0+20400000))*100))"

Please do not respond to this message, as it was generated by DB2
software. Additional help is provided in the following paragraphs.

ADM10501W Health indicator
          "<Health-Indicator-Short-Description>"
          ("<Health-Indicator-Short-Name>") breached the
          "<Threshold-Bound-Name>" warning threshold of
          "<Threshold-Bound-Value>" with value
          "<Health-Indicator-Value>" on
          "<Monitored-Object-Type>"
          "<Monitored-Object-Name>".  Calculation:
          "<Formula-String>"   =  "<Formula-with-Values>" =
          "<Health-Indicator-Value>".  History (Timestamp, Value,
          Formula): "<Health-Indicator-History-List>"

Explanation:

 The health monitor generated an alert because the warning
threshold for this health indicator was breached. This condition
does not necessarily require immediate attention, but rather it
may lead to a degradation in database performance or an
interruption in operation if the condition worsens over time.

User Response:

You can use one of the following tools to obtain recommendations,
and in some cases take actions, to resolve this alert.

Health Center:

If the Control Center tools are installed, you can launch this
tool in one of the following ways:

o   Start Menu on Windows

o   Control Center toolbar

o   command line as "db2hc"

Health indicator description, recommended actions, and links to
tools to effect the actions are provided.

Web Health Center:

If the Web-based tools are installed, then you can launch this
tool from a web browser using the URL provided to you by your web
administrator.

Health indicator description, recommended actions, and some links
to tools to effect the actions are provided.

Command Line Processor (CLP):

From the CLP, you can obtain the Health indicator description and
recommended actions by executing the following commands:

o   GET ALERT RECOMMENDATION FOR
    "<Health-Indicator-Short-Name>"
```

```
o   GET ALERT HEALTH INDICATOR DESCRIPTION FOR
    "<Health-Indicator-Short-Name>"
```

3.8.2 Memory Visualizer

You can use the Memory Visualizer to monitor the memory allocation of a DB2 Version 8 instance (Figure 3-54).



No automatic refresh

Memory Resources/Parameters	Show Plot	Plot Legend	Utilization	Parameter Value	Upper Alarm
DB2					
Partitions					
0					
DBM Shared Memory					
Backup/Restore/Utilit	☑	✖	0.08% (16 KB/19.7 MB)	--	98
backbufsz	□		--	1024	--
restbufsz	□		--	1024	--
Package Cache	□		(320 KB/Unknown)	--	--
Catalog Cache Heap	□		(160 KB/Unknown)	--	--
Buffer Pools	☑	●	(10.52 MB/Unknown)	--	--

Time Unit Minute

Figure 3-54   Memory Visualizer

The top pane of the window shows memory components organized in a navigation tree. The alarm and warning thresholds are shown to the right of each component. The lower pane shows a memory usage plot.

The high-level memory components include:

- Database manager shared memory
- Database global memory
- Application global memory
- Agent/Application shared memory
- Agent private memory

Each high-level component is divided into lower-level components that determine how the memory is allocated and deallocated. For example, memory is allocated and deallocated when the database manager starts, when a database is activated, and when an application connects to a database.

To display a memory usage plot, select the **Show Plot** check box for the desired component or components. The usage plots for different components are shown with a different color and shape, for each change to a configuration parameter. The usage plot also shows the original value, the new value, and the time when the value changed.

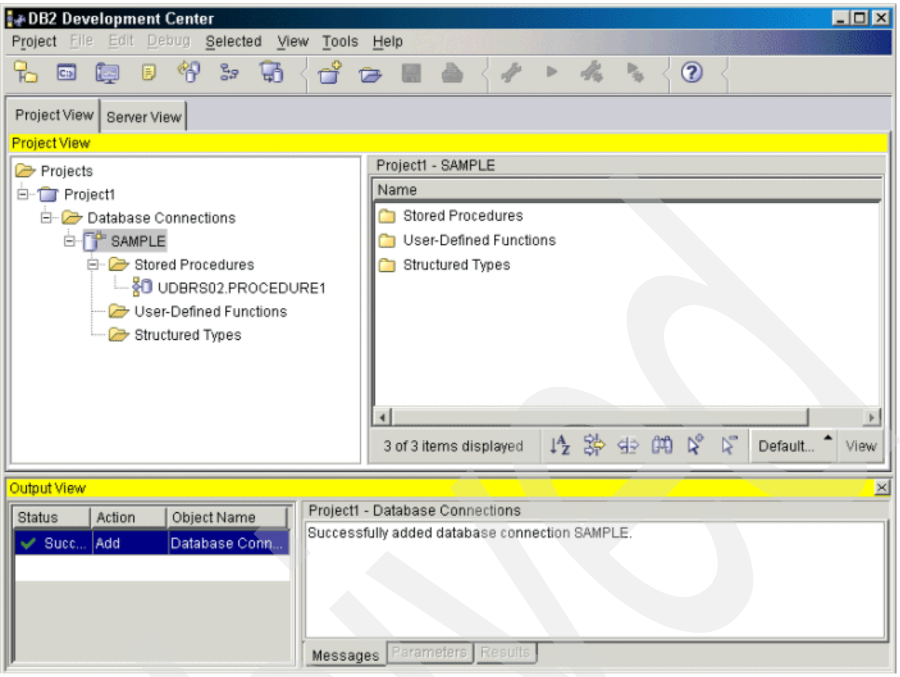
These are some of the key tasks you can perform with the Memory Visualizer:

- View overall memory usage
- Specify which memory information to display and which information to hide for DB2 instance and its databases
- Update the configuration parameters for an individual memory component to prevent it from using too much or too little memory
- Save the memory allocation data
- Load memory allocation data from a file into a Memory Visualizer window

3.9 Development Center

The Development Center is a new application development productivity tool. Built upon DB2's leadership in open standards, the Development Center has features and functions that make it easier for developers to build and deploy applications for either Java or Microsoft environments. The Stored Procedure Builder in Version 7 has been included in the Development Center.

Figure 3-55 shows the Development Center's main interface.



Status	Action	Object Name
✓ Succ...	Add	Database Conn...

Successfully added database connection SAMPLE.

Figure 3-55 Development Center

The Development Center provides an easy-to-use interface for developing routines such as stored procedures and user-defined functions (UDFs). A set of wizards makes it easy to perform your development tasks. The Development Center provides a single development environment.

With the Development Center, you can:

- Create, build, and deploy Java and SQL stored procedures.
- Create, build, and deploy user-defined functions:
  - SQL table and scalar UDFs
  - UDFs that read MQSeries® messages
  - UDFs that access OLE DB data sources
  - UDFs that extract data from XML documents
- Debug SQL stored procedures using the integrated debugger.
- See the contents of the server for each database connection that is in your project or that you have explicitly added to the Server View. You can also view and work with other database objects such as tables, triggers, and views.
- Export and import routines and project information.

The Development Center also provides a DB2 development add-in for each of the following development environments:

- Microsoft Visual C++
- Microsoft Visual Basic
- Microsoft Visual InterDev

With the add-ins, you can easily access the features of the Development Center and other DB2 centers from your Microsoft development environment, making it easy for you to develop and incorporate stored procedures and UDFs into your DB2 application development.

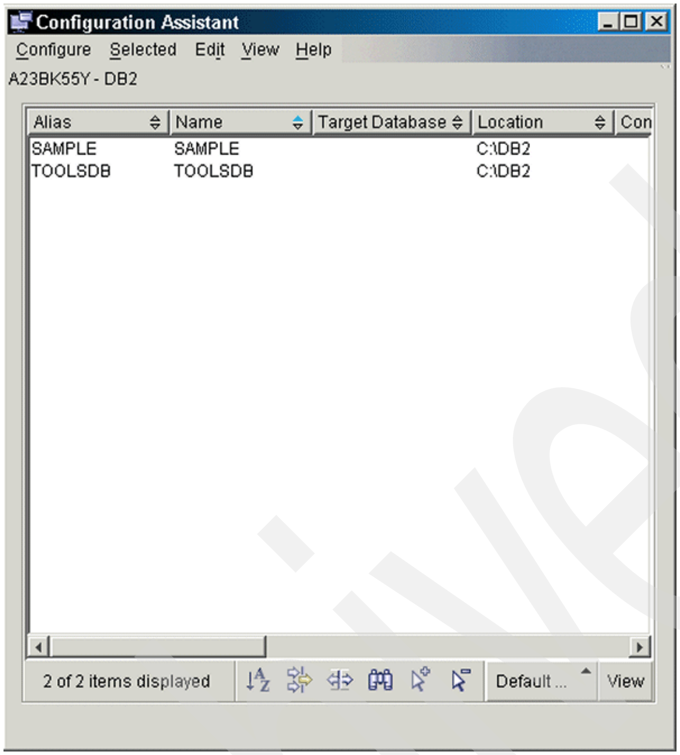
For an example of how to build a stored procedure, refer to 5.3, “Development features within DB2” on page 158.

3.10 Configuration Assistant

Configuration Assistant is a graphical tool to set Database manager configuration parameters, DB2 registry values, and client connections. The Client Configuration Assistant in DB2 Version 7 has been tightly integrated with the Configuration Assistant and enhanced with many new features, such as:

- The ability to invoke the Control Center from the Configuration Assistant.
- The option to configure both local and remote servers, including DB2 Connect servers.
- The ability to create configuration templates without affecting the local configuration.
- Import and export capabilities for exchanging configuration templates with other systems.
- Improved response time for discovery requests along with the option to refresh the list of discovered objects at any time.
- The ability to view and update applicable database manager configuration parameters and DB2 registry variables.

Figure 3-56 shows the main interface for Configuration Assistant.



A23BK55Y - DB2

Alias	Name	Target Database	Location	Con
SAMPLE	SAMPLE		C:\DB2	
TOOLSDB	TOOLSDB		C:\DB2	

Figure 3-56 Configuration Assistant

3.11 Replication Center

The DB2 Replication Center is a new user interface tool that you can use to set up and administer your replication environment. The Replication Center supports administration for DB2-to-DB2 replication environments, and administration for replication between DB2 and non-DB2 relational databases.

Figure 3-57 shows the main interface for Replication Center.

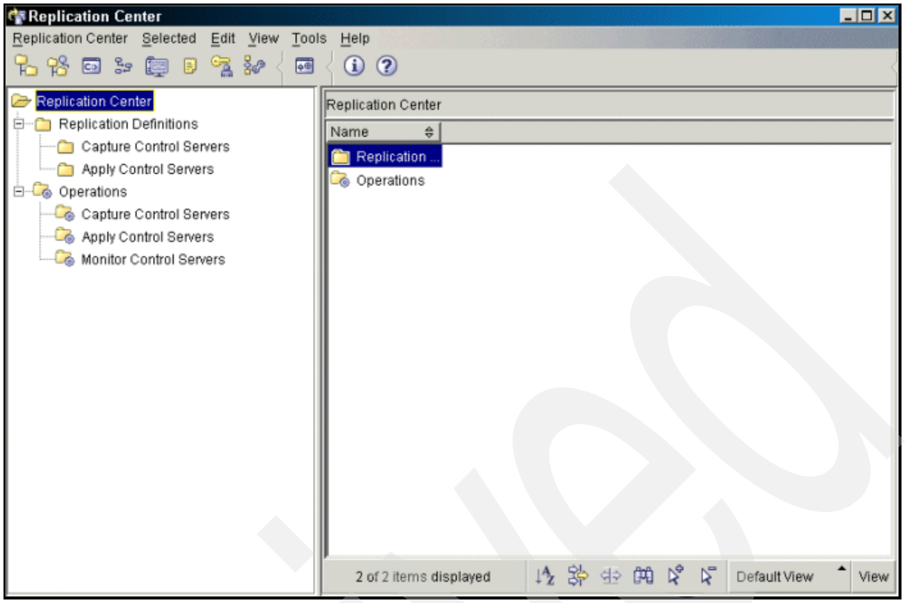


Figure 3-57 Replication Center

The Replication Center includes all of the replication functionality previously available from the DB2 Control Center and the DB2 DataJoiner® Replication Administration (DJRA) tool.

You can use the Replication Center to:

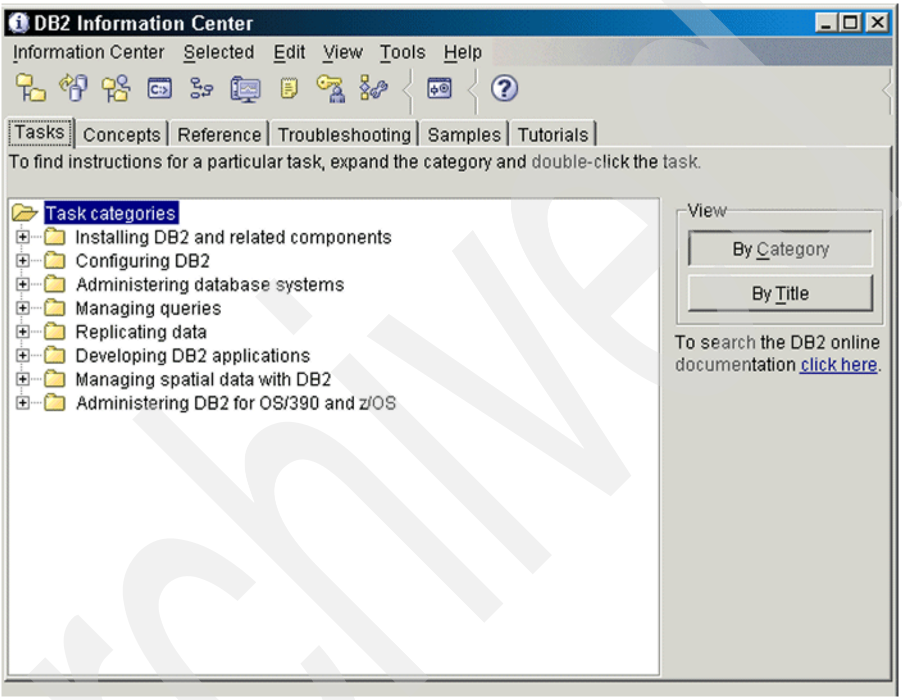
- Define defaults in profiles for creating control tables, source objects, and target objects
- Create replication control tables
- Register replication sources
- Create subscription sets and add subscription set members to subscription sets
- Operate the Capture program
- Operate the Apply program
- Monitor the replication process

You can also use the Replication Center to perform many other replication administration tasks. The Replication Center has a launchpad that allows you to perform the basic functions needed to set up a DB2 replication environment. The launchpad also shows you graphically how the different steps are related to one another.

3.12 Information Center

The DB2 Information Center provides quick access to DB2 product information and is available on all operating systems for which the DB2 administration tools are available.

Figure 3-58 shows the main interface for Information Center.



To find instructions for a particular task, expand the category and double-click the task.

To search the DB2 online documentation click here.

Figure 3-58 Information Center

The DB2 Information Center accessed from the tools provides six types of information.

- **Tasks**: Key tasks you can perform using DB2
- **Concepts**: Key concepts for DB2
- **Reference**: DB2 reference information, such as keywords, commands, and APIs
- **Troubleshooting**: Error messages and information to help you with common DB2 problems
- **Samples**: Links to HTML listings of the sample programs provided with DB2
- **Tutorials**: Instructional aid designed to help you learn a DB2 feature

Prerequisites

Some links in the DB2 Information Center point to the Internet. To display the content for these links, you will first have to connect to the Internet.

Procedure

To find product information by accessing the DB2 Information Center from the tools:

1. Start the DB2 Information Center in one of the following ways:
   - From the graphical administration tools, click the Information Center icon in the tool bar. You can also select it from the Help menu.
   - From the DB2 Command Window, enter **`db2ic`**.
2. Click the tab of the information type related to the information that you want to find.
3. Navigate through the tree and click the topic in which you are interested. The Information Center will then launch a Web browser to display the information.
4. To find information without browsing the lists, click the Search icon to the right of the list.

Once the Information Center has launched a browser to display the information, you can perform a full-text search by clicking the Search icon in the navigation tool bar.

4

DB2 Order Entry Web Application

In this chapter we describe how to produce a Web-enabled DB2 application using the Apache Web Server and the PHP scripting language. We describe how to build an Apache Web Server with PHP and DB2 connectivity in both Linux and Windows environments. We build a PHP based Web client which can interact with the DB2 data using traditional SQL DML, such as INSERT, UPDATE, DELETE, and SELECT. This then is extended to include more sophisticated features of DB2, such as stored procedures and triggers.

We describe these topics:

- Apache Web Server architecture
- Data modeling Web Order Entry application
- Using Apache and PHP to interact with DB2 via the Internet

4.1 Application description

Our goal at the onset of this project is to provide a straight forward yet scalable framework for a client application to exercise the features of DB2. We chose to make the application Web accessible to exploit the largest installed base of desktop environments. To this end, the focal point of our application environment is an Apache Web Server which can interact with a user's Web browser and provide DB2 connectivity via the PHP scripting language.

4.1.1 The Apache Web Server

The Apache Web Server has played a dominant role the explosive growth of the Internet and has therefore fueled the adoption of the Web browser as the universal client. The Apache Web Server is one of the most successful examples of collaborative open source software development. It is developed and maintained by volunteers located around the world who submit code patches to a central repository. The name Apache Server is therefore a kind of homophone for "A patchy server".

It became apparent early on that simple rendering of static HTML would not suffice to meet the business needs of most organizations which required Web applications which could incorporate application logic and access information in relational databases. The Apache Web Server is fortunate to offer the extensibility of modules, such as the PHP (Hypertext Preprocessor) language, which can endow the Web server with the ability to access databases and render Web content dynamically (Figure 4-1). PHP code inserted directly into otherwise static .HTML files allows application logic and database SQL to be dynamically processed by the PHP module of the Apache Server. The PHP language has a syntax patterned after C and Perl and a very shallow learning curve for programmers of all experience levels. PHP is also an open source project of the Apache Group.

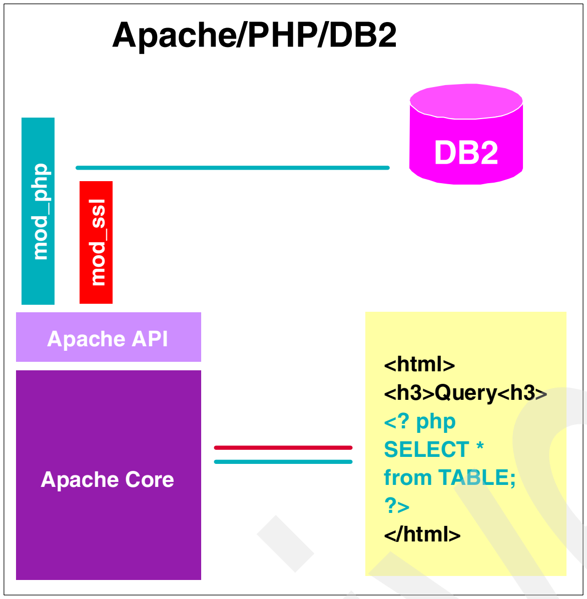


Figure 4-1 Modular nature of Apache Web Server

4.1.2 Redbooks ordering - Web shopping cart application

Developing applications with DB2 is easy. The Redbooks ordering - Web shopping cart application demonstrates that after few simple setup steps, you'll have a running online book order application.

The Redbooks ordering - Web shopping cart application provides the users with book ordering functions, including searching books by keyword, viewing the book list, adding and deleting books from his/her personal shopping cart, check out, and so on. Administrative clerks can view orders which have been submitted using their own set of query tools. Additional administrative functions include DB2 table maintenance, DB2 connectivity testing and PHP setup verification. The application also provides a function for users to generate sample XML code.

We began the data modeling for our application by visiting the "Transaction Processing Council" Web site at `http://tpc.org/tpch/default.asp` and downloading the *dbgen* tool which we used to create tables and populate with randomly generated data. The set of tables provided by this tool provides a good starting point to model a typical order entry application. We modified the DDL for most of the tables created by dbgen to allow DB2 to take on the burden and complexity of generating a unique key for each record as it is added to the table.

We have patterned our build of the Apache Server after that described by Glen Johnson in the Fall issue of DB2 Magazine:

http://www.db2mag.com/db_area/archives/2001/q3/johnson.shtml

This article includes the additional facet of incorporating secure encrypted communication via SSL, but for simplicity's sake is not used in our example.

4.2 Application setup - Linux

For the Linux environment we used the gcc C compiler to build our Apache server with PHP based DB2 connectivity using the code from the Apache and PHP open source projects. Make sure the gcc development tools are installed on your Linux machine before you continue. We used the VIM editor to create the PHP source files and found that turning on the colored syntax highlighting in the vimrc file was very helpful, because it has knowledge of most of the PHP built in functions.

4.2.1 Preparation

In efforts to maintain consistency with the Linux Standards Base and File System Hierarchy, we located our Apache server in the */usr/local/apache* directory. If your system does not have this directory, create the directory using root user ID with the following command:

```
mkdir /usr/local/apache
```

To obtain the sample code for our application, look for the sg246934_sample.tar.gz file either from the book CD or download from the IBM Redbooks Web site:

ftp://www.redbooks.ibm.com/redbooks/SG246934

Select the **Additional Materials** and open the directory that corresponds with the redbook form number, SG246934. Save the file named *sg246934.tar.gz*.

See Appendix A, "Additional material" on page 383 for detailed download information.

This file should be unpacked in the /usr/local/apache directory using:

```
tar -zxvf sg246934_sample.tar.gz
```

(or alternatively)

```
gzip -d -c g246934_sample.tar.gz | tar xvf -
```

You will now have one file and two directories below */usr/local/apache*:

- *ApacheDB2php.sh:* This is the Apache Web Server setup script.
- *htdocs:* This contains the .html and .php files that will be executed by the Web server.
- *dbgen:* This contains the files necessary to create and populate your DB2 database.

We allow the DB2 instance owner db2inst1 to own the files in and below */usr/local/apache.* Enter the following command to change the file ownership:

```
chown -R db2inst1:db2grp1 /usr/local/apache
```

4.2.2 Setup database

The Redbooks ordering online book store application uses a DB2 database RBORDERS. To create this database, log into the DB2 instance owner system account, *db2inst1*. In the Linux command line, enter the following command to create the application database RBORDERS:

```
db2 create database rborders
```

By default DB2 creates three system managed tablespaces, SYSCATSPACE, TEMPSPACE1, and USERSPACE1. SYSCATSPACE is for holding DB2 system catalog data. DB2 uses TEMPSPACE1 as a temporary working space for sorting and so on. The USERSPACE1 can be used to hold the user tables. The database is placed under /home/db2inst1/db2inst1/NODE0000.

The next step is to create the tables and triggers. In the Linux command line, change to the directory that contains the DDL:

```
cd /usr/local/apache/dbgen
```

Issue this command as the DB2 instance owner (*db2inst1)* to create your tables and triggers:

```
db2 -t -v -f rborders.ddl
```

We also use DB2 stored procedures in our book store PHP sample application. To create the DB2 stored procedures, enter these commands in a Linux command line:

```
db2 -td@ -vf create_order.pro
db2 -td@ -vf insert_part.pro
```

The create_order procedure takes the items from the shopping cart and creates a header record in the ORDER table and individual records in the LINEITEMS table for each book that is ordered.

We provide the script for you to populate the tables with data. The data is loaded into the tables by executing:

```
db2 -t -v -f rborders.load
```

Now, the RBORDERS database has been created successfully and data loaded, you can view some data from the database. Enter the following commands from the Linux command line to view the book title on the BOOKS table.

```
db2 connect to rborders
db2 select title from db2inst1.books
db2 connect reset
```

4.2.3 Setup Apache Web Server

The source code for PHP and the Apache Web Server can be acquired from the respective project Web sites.

PHP source code is available at:

http://www.php.net/downloads.php

For this book we used Version 4.3.1 of PHP. The file name is *php-4.3.1.tar.gz*.

Apache Web Server source code is available at:

http://httpd.apache.org/download.cgi

For this book we used Version 1.3.27 of Apache. The file name is *apache_1.3.27.tar.gz*.

Follow these steps to setup Apache Web Server and the Redbooks ordering application:

1. **Setup root to access DB2**

   The httpd program must be started as *root* and have access to the DB2 environment. After it is started, ownership of the httpd process is passed to the *nobody* user. A simple way to ensure that the proper environment has been set up is to place the following lines of code which execute the db2profile in root's .bashrc or .profile.

   ```
   if [ -f /home/db2inst1/sqllib/db2profile ]; then
   ./home/db2inst1/sqllib/db2profile
   fi
   ```

2. **Setup Web server and application**

   We have provided you with the script /usr/local/apache/apacheDB2php.sh which is shown in Example 4-1. This script can be used to download and build the Apache Server all in one step.

To run the script, log in as the root. Enter the commands from Linux command line:

```
cd /usr/local/apache/
./apacheDB2php.sh
```

Alternatively you can type in the lines one at a time to create the Apache server.

If this build process was successful, you should have a copy of the Apache Web Server located at */usr/local/apache/bin/httpd*.

Example 4-1 Apache/PHP download & build script

```
#!/bin/sh
## uncomment next 2 lines to get apache and php source with wget
wget http://apache.towardex.com/httpd/apache_1.3.27.tar.gz
wget http://www.php.net/get/php-4.3.1.tar.gz/from/us2.php.net/mirror
#unpack the source
gzip -d -c apache_1.3.27.tar.gz | tar xvf -
gzip -d -c php-4.3.1.tar.gz | tar xvf -
# pre-configure Apache for PHP's configure step cd apache_1.3.27
./configure \ --prefix=/usr/local/apache
cd ..
# configure PHP and apply it to the Apache source tree
cd php-4.3.1
./configure \
--with-ibm-db2=/opt/IBM/db2/V8.1 \
--with-mysql \
--with-xml \
--with-apache=../apache_1.3.27 \
--enable-sockets
make
make install
cd .. cd apache_1.3.27
./configure \
--activate-module=src/modules/php4/libphp4.a \
--prefix=/usr/local/apache
make
make install
```

When we configure PHP *--with-ibm-db2 option,* the PHP build process will link *ext/odbc/php_odbc.c* with the sqlcli.h sqlcli1.h header files contained in the */opt/IBM/db2/V8.1/include* directory provided by the DB2 Development Client. Therefore we endow Apache with the ability to act as a DB2 client via the CLI interface. At this point you will realize that PHP is using ODBC function names to talk to DB2. In the Linux environment, there is however, no ODBC involved to speak natively to the database using the CLI. The ODBC function names are simply reused for convenience by the DB2 CLI's borrowed semantics of the ODBC.

3. **Setup application user ID and password**

   Change to directory /usr/local/apache/htdocs, and edit two files:

   – **cart.php**: Modify userid and password settings ($user, and $pass) to your DB2 instance owner ID and password.

   – **tableAED.php**: Modify userid and password settings ($user, and $pass) to your DB2 instance owner ID and password.

4. **Edit Apache configuration**

   Before you can start using the Web server, the Apache configuration file *httpd.conf* at */usr/local/apache/conf/* needs to be edited to include instructions on how to respond to requests for .php scripts or phps source files. Edit the httpd.conf file and add the following lines:

   ```
   #next two lines were added here to enable .php and phps scripts
   AddType application/x-httpd-php .php
   AddType application/x-httpd-php-source .phps
   ```

5. **Setup php.ini file**

   The configuration for PHP (php.ini) needs to be copied and modified as follows:

   ```
   cp /usr/local/apache/php-4.3.1/php.ini-dist /usr/local/lib/php.ini
   ```

   Edit php.ini in /usr/local/lib directory and change:

   ```
   register_globals = Off to register_globals = On
   ```

   Note that having **register_globals** turned **On** allows for easier code development, but that code tends to be less secure. If register_globals is set to Off then the ability for any user submitted variable to be injected into your PHP code is turned off; internal variables are effectively isolated from user submitted data. This will become more obvious as we examine the PHP source code later on.

4.2.4 Verify Web Server setup

Now the Apache Server and Redbooks ordering demo application have been set up, under the root user ID, you can start the Web server with the commands:

```
cd /usr/local/apache/bin
./apachectl start
```

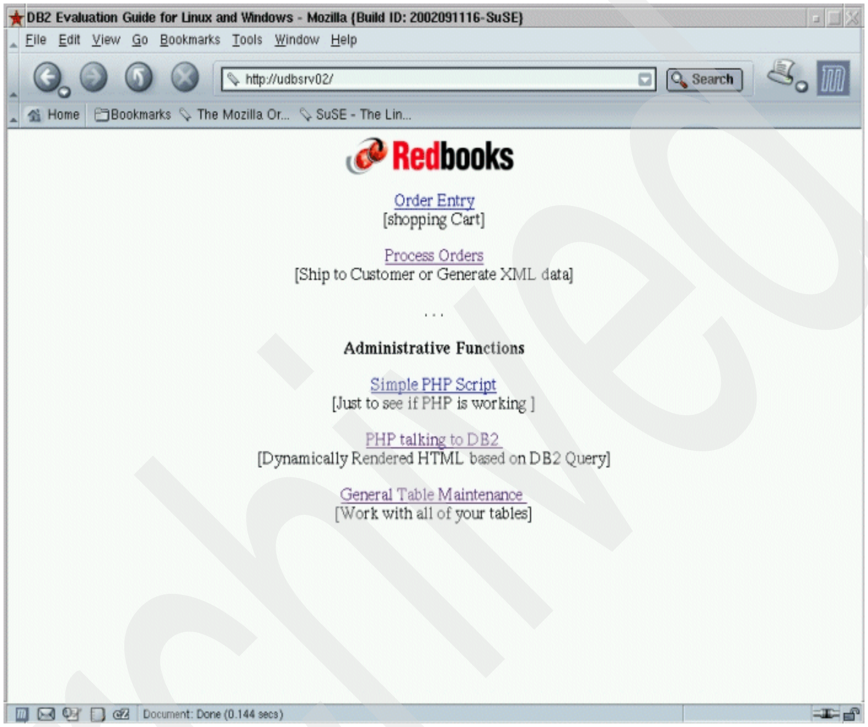
When you start the Apache Server, you should see a message similar to:

```
./apachectl start: httpd started
```

You can now test out your newly constructed Apache Web server. Point your Web browser to:

```
http://yourhostname
```

You should see a window similar to Figure 4-2.



DB2 Evaluation Guide for Linux and Windows - Mozilla (Build ID: 2002091116-SuSE)
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/
Search
Home
Bookmarks
The Mozilla Or...
SuSE - The Lin...
Redbooks
Document: Done (0.144 secs)

Order Entry
[shopping Cart]

Process Orders
[Ship to Customer or Generate XML data]

. . .

Administrative Functions

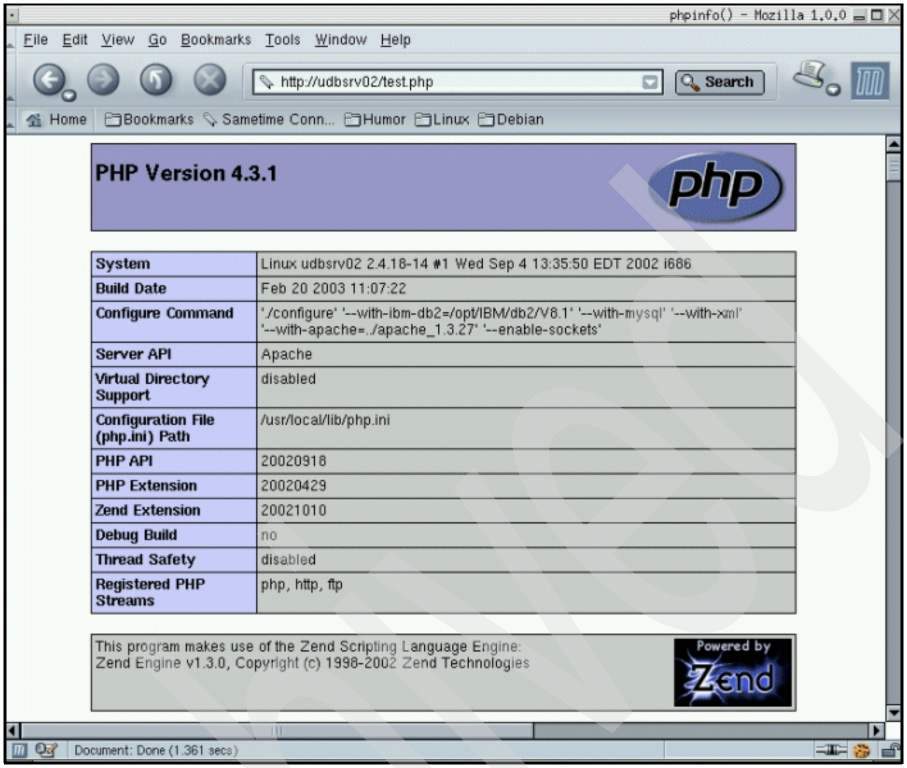
Simple PHP Script
[Just to see if PHP is working]

PHP talking to DB2
[Dynamically Rendered HTML based on DB2 Query]

General Table Maintenance
[Work with all of your tables]

Figure 4-2 Redbooks ordering window

The Redbooks ordering application provides two functions for the administrators to perform a simple verification of PHP functionality and DB2 connectivity. To verify PHP function, click **Simple PHP script**. If you see a window like Figure 4-3, your PHP functions properly.


phpinfo() - Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/test.php
Search
Home Bookmarks Sametime Conn... Humor Linux Debian
php
Powered by Zend
Document: Done (1.361 secs)

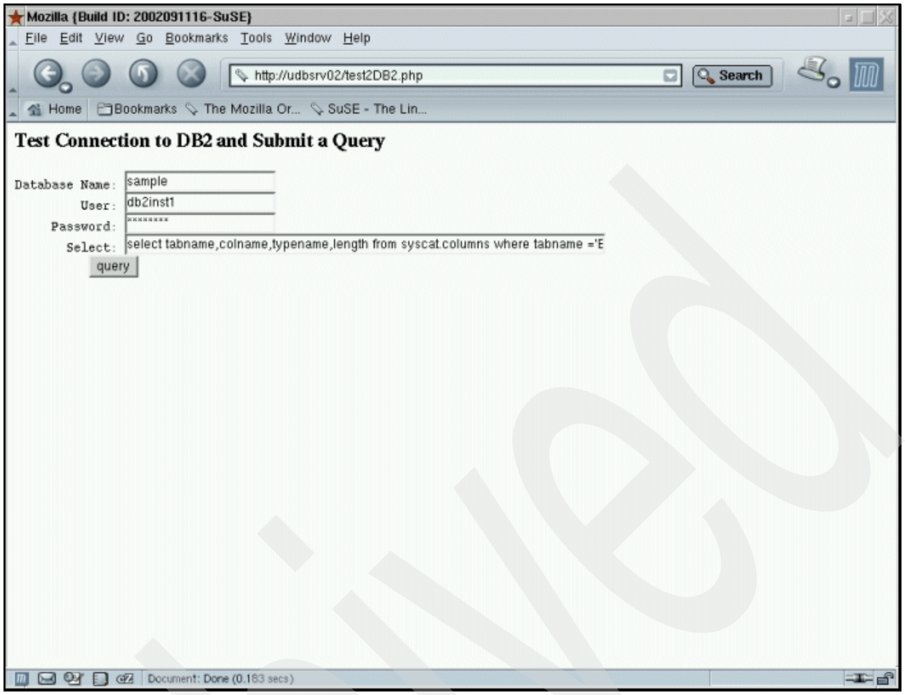
PHP Version 4.3.1

System	Linux udbsrv02 2.4.18-14 #1 Wed Sep 4 13:35:50 EDT 2002 i686
Build Date	Feb 20 2003 11:07:22
Configure Command	'./configure' '--with-ibm-db2=/opt/IBM/db2/V8.1' '--with-mysql' '--with-xml' '--with-apache=../apache_1.3.27' '--enable-sockets'
Server API	Apache
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/lib/php.ini
PHP API	20020918
PHP Extension	20020429
Zend Extension	20021010
Debug Build	no
Thread Safety	disabled
Registered PHP Streams	php, http, ftp

This program makes use of the Zend Scripting Language Engine:
Zend Engine v1.3.0, Copyright (c) 1998-2002 Zend Technologies

Figure 4-3 Test PHP function

To verify DB2 connectivity, click **PHP talking to DB2**. The *Test Connection to DB2 and Submit a Query* window appears (Figure 4-4).

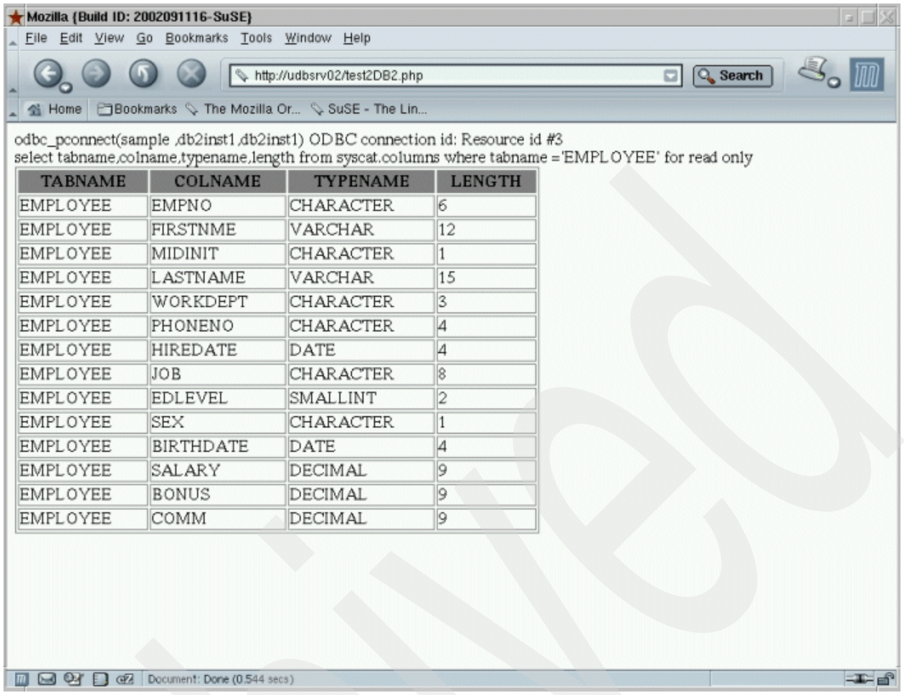

Mozilla {Build ID: 2002091116-SuSE}
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/test2DB2.php
Search
Home
Bookmarks
The Mozilla Or...
SuSE - The Lin...
query
Document: Done (0.183 secs)

Test Connection to DB2 and Submit a Query

Database Name: sample
User: db2inst1
Password: ********
Select: select tabname,colname,typename,length from syscat.columns where tabname ='E

Figure 4-4 Test DB2 connectivity

Verify the user ID and enter the password, click **query** and you should see the data selected from the DB2 catalog table (Figure 4-5). The application connection to DB2 was successful.


Mozilla {Build ID: 2002091116-SuSE}
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/test2DB2.php
Search
Home
Bookmarks
The Mozilla Or...
SuSE - The Lin...
Document: Done (0.544 secs)

odbc_pconnect(sample ,db2inst1,db2inst1) ODBC connection id: Resource id #3
select tabname,colname,typename,length from syscat.columns where tabname ='EMPLOYEE' for read only

TABNAME	COLNAME	TYPENAME	LENGTH
EMPLOYEE	EMPNO	CHARACTER	6
EMPLOYEE	FIRSTNME	VARCHAR	12
EMPLOYEE	MIDINIT	CHARACTER	1
EMPLOYEE	LASTNAME	VARCHAR	15
EMPLOYEE	WORKDEPT	CHARACTER	3
EMPLOYEE	PHONENO	CHARACTER	4
EMPLOYEE	HIREDATE	DATE	4
EMPLOYEE	JOB	CHARACTER	8
EMPLOYEE	EDLEVEL	SMALLINT	2
EMPLOYEE	SEX	CHARACTER	1
EMPLOYEE	BIRTHDATE	DATE	4
EMPLOYEE	SALARY	DECIMAL	9
EMPLOYEE	BONUS	DECIMAL	9
EMPLOYEE	COMM	DECIMAL	9

Figure 4-5 DB2 query output

4.3 Application setup - Windows

For the Windows environment, binaries for the Apache Web server and the PHP modules have been made available because C application development environment is often not available by default. We can effectively skip the build process that we used for the Linux environment.

4.3.1 Setup Apache Web Server

The Apache Web server files can be obtained from this Web site:

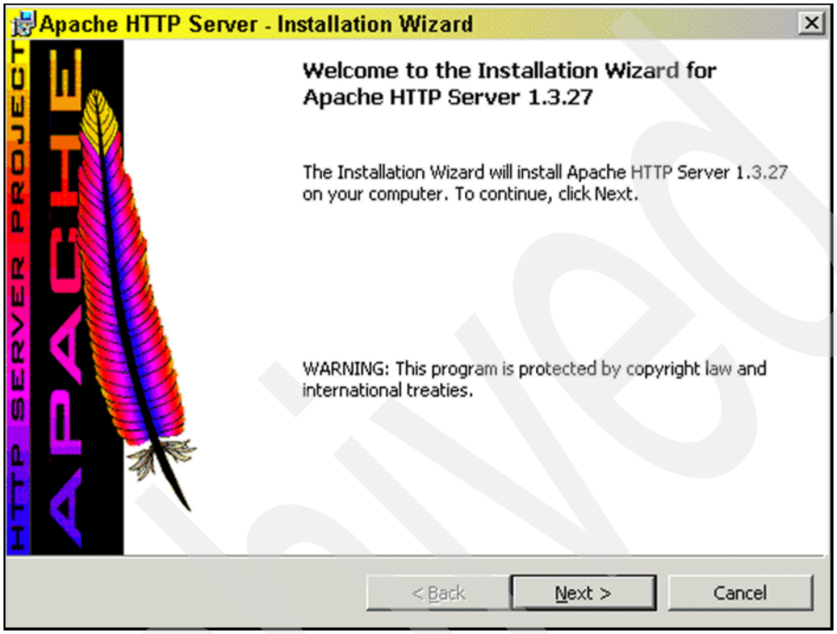
http://httpd.apache.org/download.cgi

We downloaded the MSI Version 1.3.27 of Apache as:

apache_1.3.27-win32-x86-no_src.msi

The apache_1.3.27-win32-x86-no_src is a self-extracting file. To install the Apache Web Server on your machine:

- Double-click the file, or right-click the file and select **Install.** A wizard similar to Figure 4-6 will appear. Click **Next**.



Welcome to the Installation Wizard for Apache HTTP Server 1.3.27

The Installation Wizard will install Apache HTTP Server 1.3.27 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Figure 4-6 Apache installation wizard on Windows

- *License Agreement window*: Click **I accept the items in the license agreement**.
- *Read this first*: Click **Next**.
- *Server information*: Use the default or enter your server information.
- *Setup type*: Select **Complete**.
- *Destination folder*: Use the default **c:/Program File/Apache group**.
- *Ready to install the program*: Click **Install**.

4.3.2 Setup database

Scripts for creating the sample application database, populating it with data and the PHP Web application scripts themselves can be obtained on the book CD-ROM or downloaded from the IBM Redbook Web site:

ftp://www.redbooks.ibm.com/redbooks/SG246934

Select the **Additional Materials** and open the directory that corresponds with the redbook form number, SG246934. Save the file *sg246934_sample.zip*.

See Appendix A, “Additional material” on page 383 for details.

This sg246934_sample.zip should be unpacked in the *c:\Program Files\Apache Group\Apache* directory using a tool such as Winzip (available from http://winzip.com)

You will now have two directories below c:\Program Files\Apache Group\Apache:

- *htdocs:* contains the .html and .php files that will be executed by the Web server.
- *dbgen:* contains the files necessary to create and populate your DB2 database.

The online book store application, Redbooks ordering, uses a DB2 database RBORDERS. To create the database, open the DB2 Command Window by entering **db2cmd** in a MS command prompt window or selecting **Start —> Program —> IBM DB2 —> Command Line Tools —> Command Window**.

Enter the command to create the database:

```
db2 create database rborders
```

Once the database is created, we need to create tables. In the DB2 Command window, locate and execute the DDL file by entering:

```
cd c:\Program Files\apache group\apache\dbgen
db2 -t -v -f rborders.ddl
```

The next step is to create theDB2 store procedures and triggers required for your application.

Note: To create stored procedures, a C compiler needs to be installed. DB2’s environment needs to know the location of the compiler by setting the DB2 environment variable. In the DB2 Command window, enter:

```
db2set db2_sqlroutine_compiler_path="c:\program files\microsoft visual
studio\VC98\bin\vcvars32.bat"
```

To create the create_order procedure, in the DB2 Command window enter:

```
db2 -td@ -vf create_order.pro
```

The create_order procedure takes the items from the shopping cart and creates a header record in the order table and individual records in the LINEITEMS table for each book that is ordered.

Now you have created database, tables, triggers and store procedure successfully, you need to populate the tables with data. The data is loaded into the tables by executing:

```
db2 -t -v -f rborders.load
```

In the Windows environment, an ODBC data source needs to be cataloged.

- Select **Start —> IBM DB2 —> Set-up Tools —> Configuration Assistance**
- Right-click the **RBORDERS** database and select **Change Database**.
- Select **4. Data Source**.
- Click **Register this data source for ODBC**, leave As system separate data source selected. Click **Next**.
- On Specify the security options window, leave Use authentication value in server's DBM Configuration selected, click **Finish**.

For our examples you need to make sure that the SAMPLE and RBORDERS databases are made available by this means.

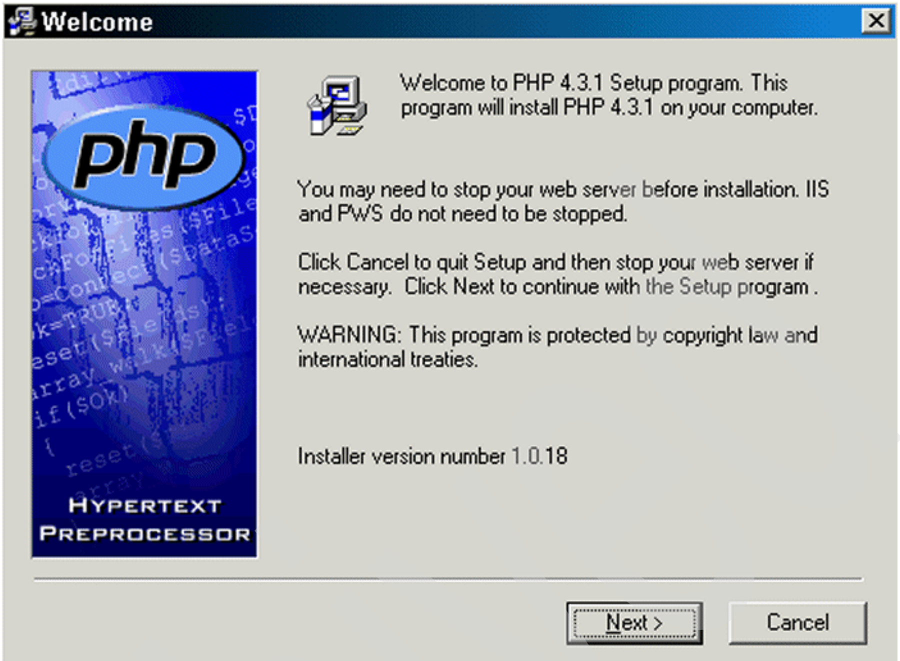
4.3.3 Setup PHP

The sample online book store application is a PHP program. To setup a PHP environment, download *PHP-4.3.1-installer.exe* from this Web site:

http://www.php.net/downloads.php

Follow these steps to install the PHP:

1. Double-click the PHP-4.3.1-installer.exe. You'll see the Welcome window as in Figure 4-7.



Welcome to PHP 4.3.1 Setup program. This program will install PHP 4.3.1 on your computer.

You may need to stop your web server before installation. IIS and PWS do not need to be stopped.

Click Cancel to quit Setup and then stop your web server if necessary. Click Next to continue with the Setup program .

WARNING: This program is protected by copyright law and international treaties.

Installer version number 1.0.18

Figure 4-7 Wizard to install PHP

- *Accept License Agreement*: Select **I Agree**.
- *Install type*: Select **Standard**.
- *Choose Destination Location*: The installation location should be **c:\Program Files\Apache Group**\.
- *Mail Configuration*: Click **Next**.
- *Source type*: Select **Apache**.
- *Start Installation*: Click **Next**.

2. Once the PHP is installed, we need to configure it and setup the application to use it:

   a. **Edit php.ini**

   The C:/WINNT/php.ini file needs to be edited to set:

```
register_globals=on
error_reporting = E_ALL & ~E_NOTICE
```

   b. **Edit httpd.conf**

   To use PHP as a binary, the Apache httpd.conf file at directory c:\Program Files\Apache Group\Apache\Conf needs to be edited manually, or from the Apache Server from the Start Menu to include these lines:

```
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php
Action application/x-httpd-php "/php/php.exe"
```

c. **Recycle Apache Server**

The Apache Server needs to be restarted. In the Windows command prompt, enter these commands:

```
NET START STOP APACHE
NET START APACHE
```

d. **Setup application user ID and password**

Change to directory C:\Program Files\apache group\apache\htdocs, and edit two files (see first few lines at top of file):

- **cart.php**: Modify user ID and password settings ($user, and $pass) to your DB2 instance owner ID and password.
- **tableAED.php**: Modify user ID and password settings ($user, and $pass) to your DB2 instance owner ID and password.

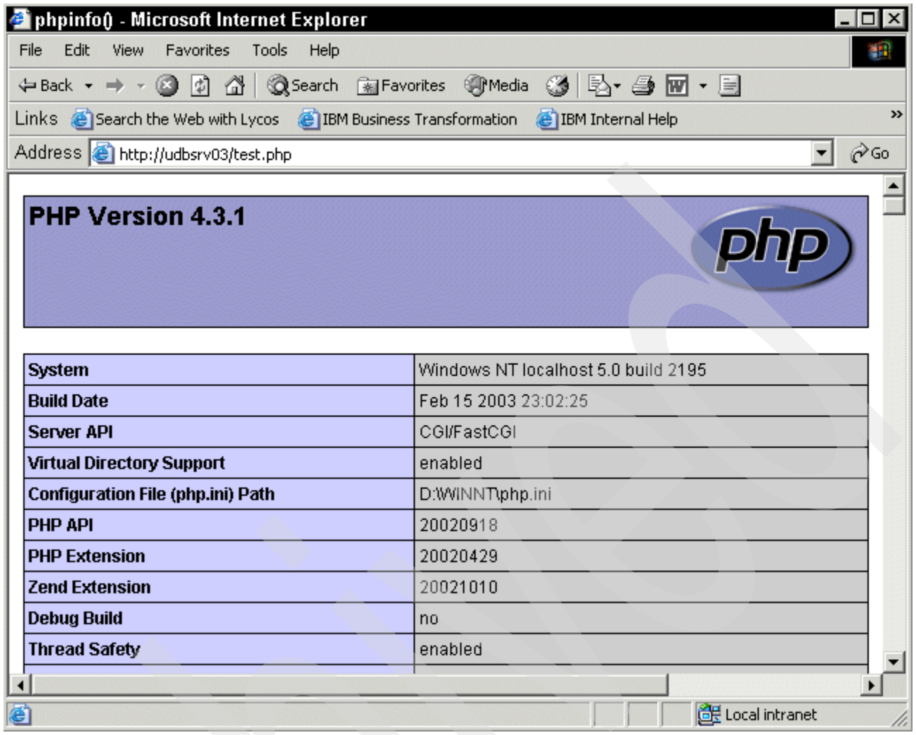
4.3.4 Verify Web Server Setup

You can now test your newly constructed Apache Web server by typing the following into your favorite browser:

```
http://yourhostname
```

You should see a window similar to Figure 4-2 on page 119.

The Redbooks ordering application provides two functions for the administrators to perform a simple verification of PHP functionality and DB2 connectivity. To verify PHP function, click **Simple PHP script**. If you see a window similar to Figure 4-8, your PHP functions properly.

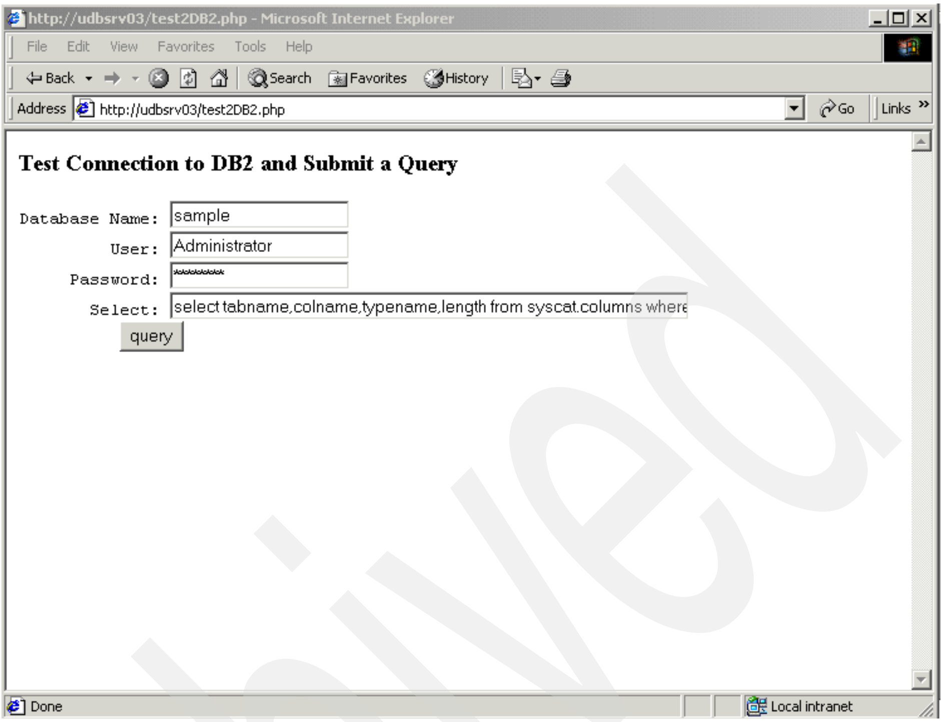

phpinfo() - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Links Search the Web with Lycos IBM Business Transformation IBM Internal Help
Address http://udbsrv03/test.php
Go
Local intranet

PHP Version 4.3.1

System	Windows NT localhost 5.0 build 2195
Build Date	Feb 15 2003 23:02:25
Server API	CGI/FastCGI
Virtual Directory Support	enabled
Configuration File (php.ini) Path	D:\WINNT\php.ini
PHP API	20020918
PHP Extension	20020429
Zend Extension	20021010
Debug Build	no
Thread Safety	enabled

Figure 4-8 Verify PHP setup

To verify DB2 connectivity, click **PHP talking to DB2**. The *Test Connection to DB2 and Submit a Query* window appears (Figure 4-9).


http://udbsrv03/test2DB2.php - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://udbsrv03/test2DB2.php
Go Links
Done
Local intranet

Test Connection to DB2 and Submit a Query

Database Name: sample

User: Administrator

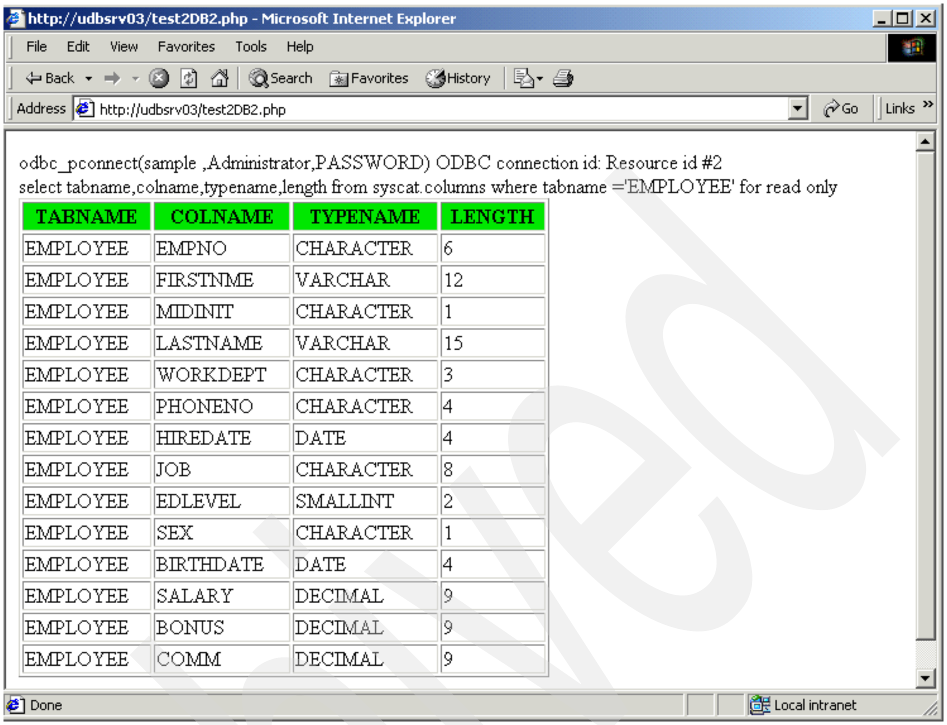
Password: ********

Select: select tabname,colname,typename,length from syscat.columns where

query

Figure 4-9 Verify DB2 connectivity

Verify the user ID and enter the password, and then click **query**. You should see the data selected from the DB2 catalog table (Figure 4-10). The application connection to DB2 was successful.


http://udbsrv03/test2DB2.php - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://udbsrv03/test2DB2.php
Go Links
Done
Local intranet

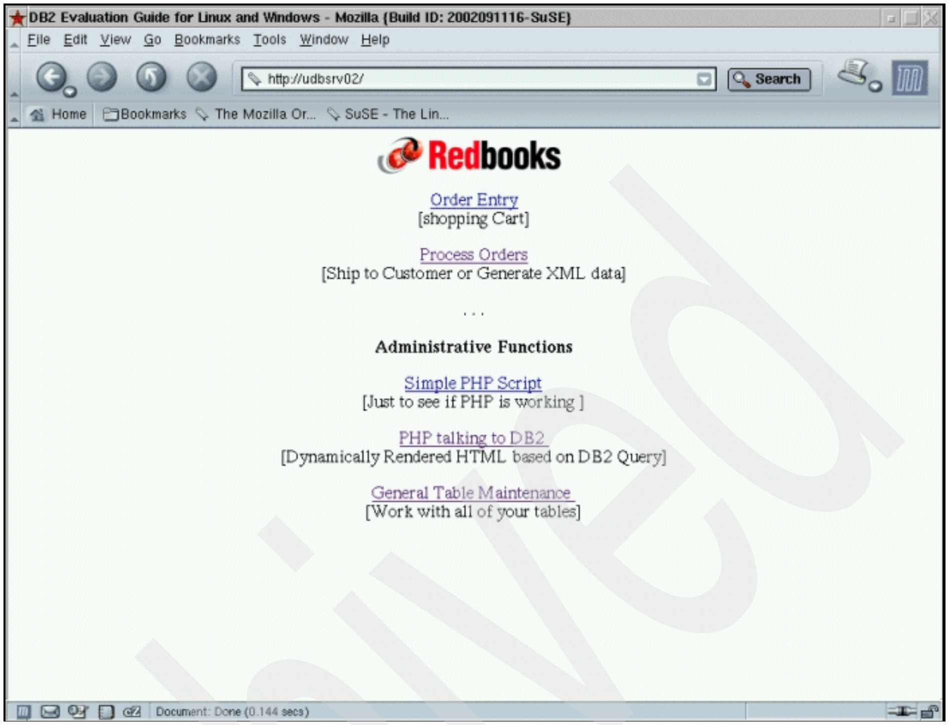
odbc_pconnect(sample ,Administrator,PASSWORD) ODBC connection id: Resource id #2
select tabname,colname,typename,length from syscat.columns where tabname ='EMPLOYEE' for read only

TABNAME	COLNAME	TYPENAME	LENGTH
EMPLOYEE	EMPNO	CHARACTER	6
EMPLOYEE	FIRSTNME	VARCHAR	12
EMPLOYEE	MIDINIT	CHARACTER	1
EMPLOYEE	LASTNAME	VARCHAR	15
EMPLOYEE	WORKDEPT	CHARACTER	3
EMPLOYEE	PHONENO	CHARACTER	4
EMPLOYEE	HIREDATE	DATE	4
EMPLOYEE	JOB	CHARACTER	8
EMPLOYEE	EDLEVEL	SMALLINT	2
EMPLOYEE	SEX	CHARACTER	1
EMPLOYEE	BIRTHDATE	DATE	4
EMPLOYEE	SALARY	DECIMAL	9
EMPLOYEE	BONUS	DECIMAL	9
EMPLOYEE	COMM	DECIMAL	9

Figure 4-10 DB2 query output

4.4 Using the DB2 Web application

Now you have completed the installation and configuration of the Apache Web server and DB2/PHP Web application, and you can try the various functions in this application. The main or home page of your server should appear similar to Figure 4-11.


DB2 Evaluation Guide for Linux and Windows - Mozilla (Build ID: 2002091116-SuSE)
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/
Search
Home
Bookmarks
The Mozilla Or...
SuSE - The Lin...
Redbooks
Document: Done (0.144 secs)

Order Entry
[shopping Cart]

Process Orders
[Ship to Customer or Generate XML data]

. . .

Administrative Functions

Simple PHP Script
[Just to see if PHP is working]

PHP talking to DB2
[Dynamically Rendered HTML based on DB2 Query]

General Table Maintenance
[Work with all of your tables]

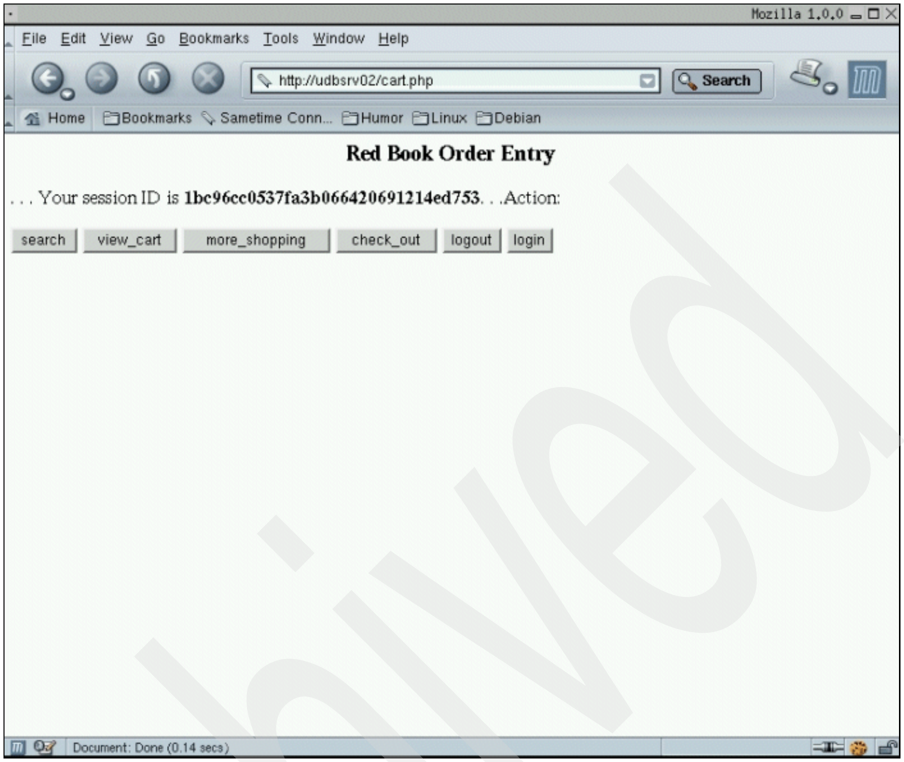
Figure 4-11 Main menu and home page for Apache Server

4.4.1 Order Entry application

The Order Entry - shopping cart example illustrates power and simplicity of DB2 interface with PHP (approximately 350 lines of code) to build an online shopping cart to accumulate book order items and submit the finalized order to the main server. The code is contained in the cart.php file and uses PHP sessions to maintain state with the main Web server. The program also exercises features of DB2 including: table queries, updates and inserts and includes calls to DB2 stored procedures.

Order Entry shopping cart

The Order Entry application is for customers to order IBM Redbooks. It allows the user to search a list of IBM Redbooks related to DB2, select individual items and place them in a virtual shopping cart, login, identify themselves to the system, and send the final order for processing during the check out process. Once you click the **Order Entry,** you'll see the Redbooks Order Entry window (Figure 4-12).



Red Book Order Entry

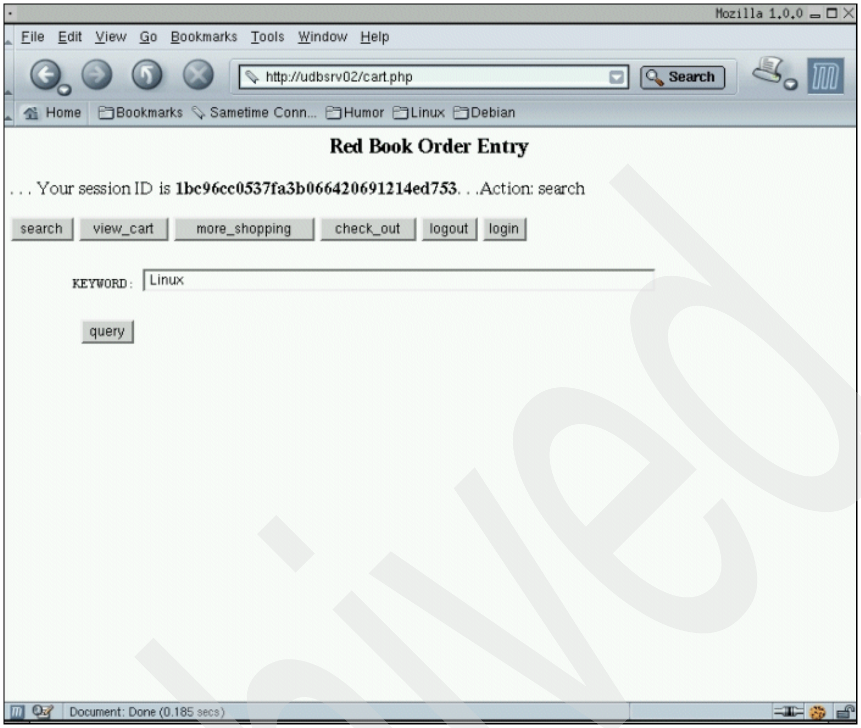
. . . Your session ID is **1bc96cc0537fa3b066420691214ed753**. . .Action:

Document: Done (0.14 secs)

Figure 4-12 Redbooks Order Entry window

The session ID displayed on the top of the window is generated by PHP and stored on the Apache Server. The functions available in this program Search, View Cart, More Shopping, Check Out, Login, and Logout are shown as buttons on the top of the form. The functionality would typically be utilized in sequence from left to right although this is not necessary.

Pressing the **Search** button will produce a window where IBM DB2 Redbooks title keywords can be entered. If you enter Linux in the box and press **query** (Figure 4-13), a list of three books will be returned (Figure 4-14).



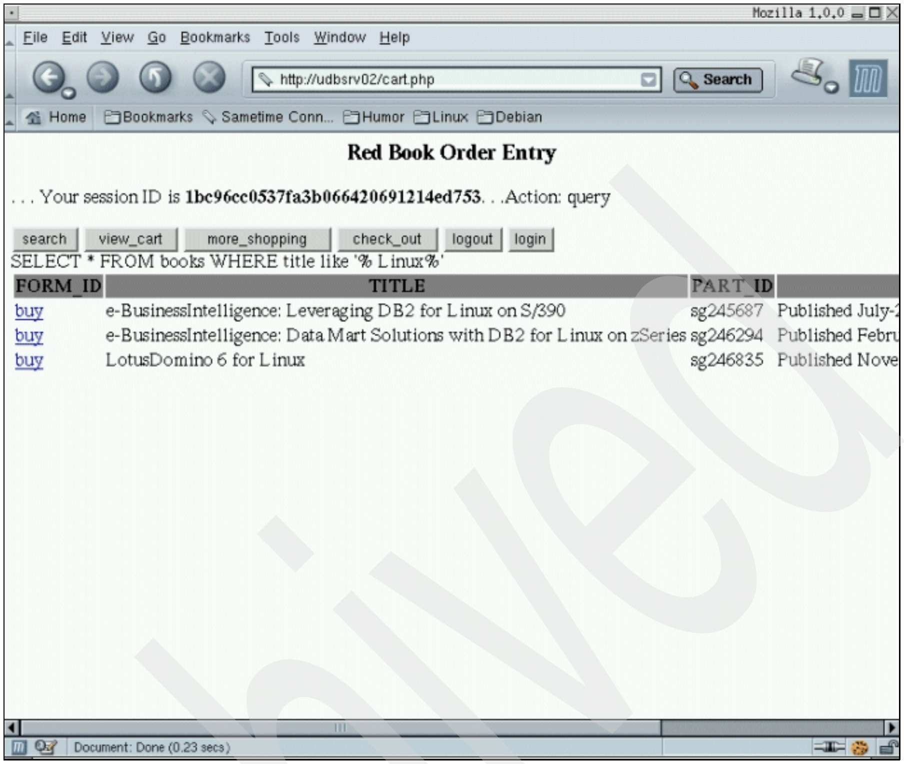
Red Book Order Entry

. . . Your session ID is **1bc96cc0537fa3b066420691214ed753**. . .Action: search

KEYWORD: Linux

Figure 4-13 Querying the catalog for Linux books

A list of three books is returned (Figure 4-14).


Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/cart.php
Search
Home Bookmarks Sametime Conn... Humor Linux Debian
search view_cart more_shopping check_out logout login
Document: Done (0.23 secs)

Red Book Order Entry

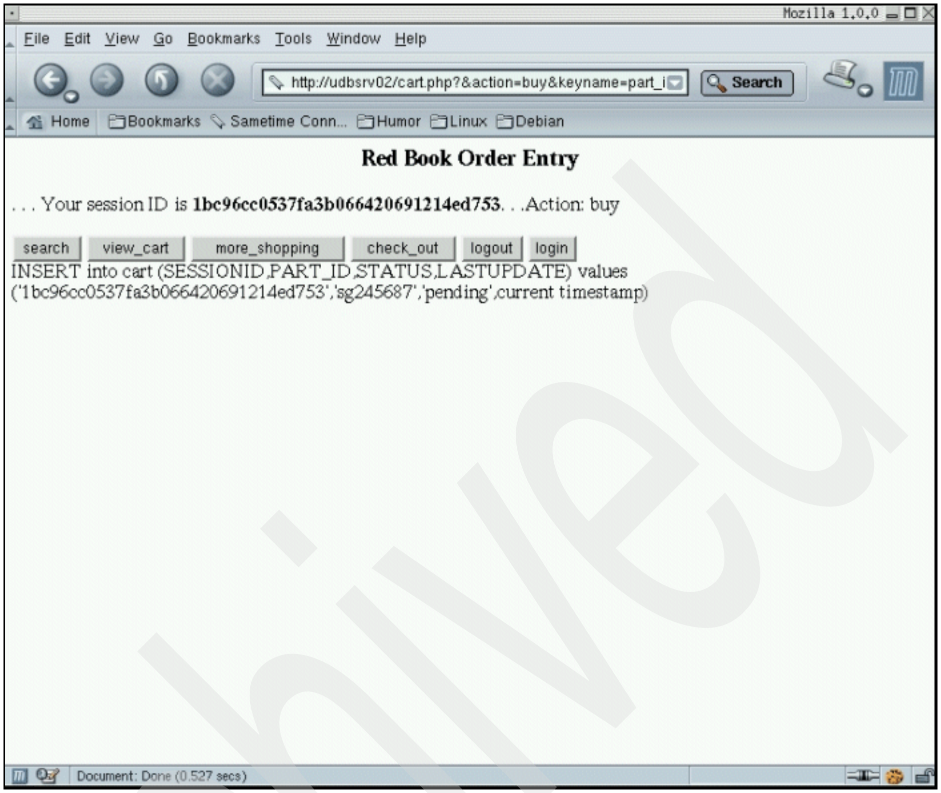
. . . Your session ID is **1bc96cc0537fa3b066420691214ed753**. . .Action: query

SELECT * FROM books WHERE title like '% Linux%'

FORM_ID	TITLE	PART_ID	
buy	e-BusinessIntelligence: Leveraging DB2 for Linux on S/390	sg245687	Published July-2
buy	e-BusinessIntelligence: Data Mart Solutions with DB2 for Linux on zSeries	sg246294	Published Febru
buy	LotusDomino 6 for Linux	sg246835	Published Nove

Figure 4-14 Results of catalog query

You can then click the link labeled **buy** and that particular book will be added to your shopping cart. The DB2 INSERT statement used for placing the item in the shopping cart is shown (Figure 4-15). The process can then be repeated to add as many books as desired to the cart.


Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/cart.php?&action=buy&keyname=part_i
Search
Home
Bookmarks
Sametime Conn...
Humor
Linux
Debian
search
view_cart
more_shopping
check_out
logout
login
Document: Done (0.527 secs)

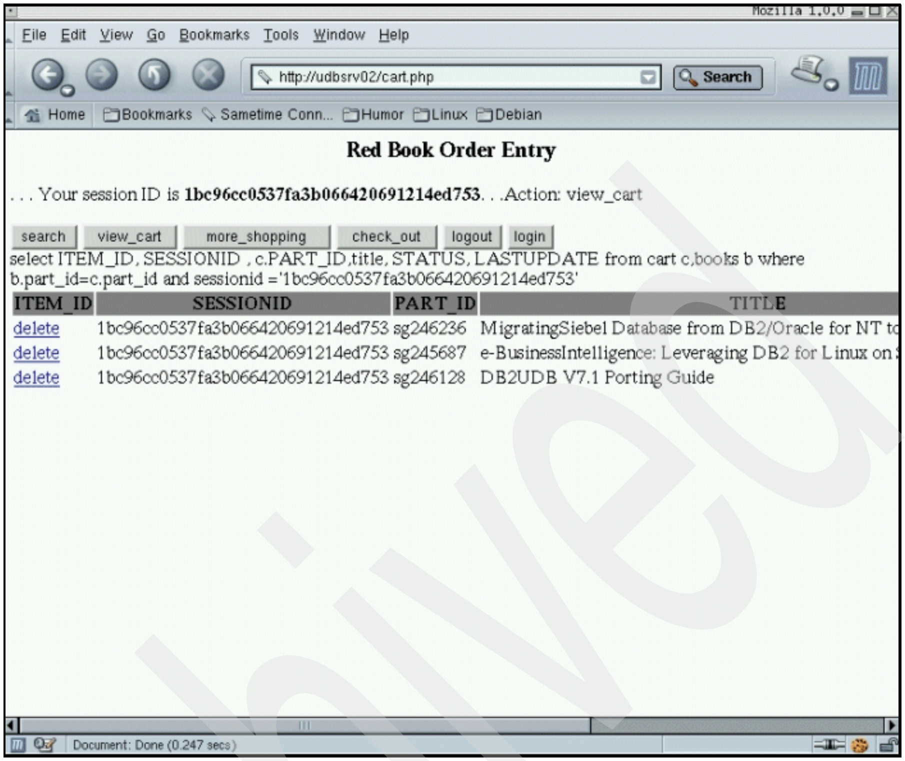
Red Book Order Entry

. . . Your session ID is **1bc96cc0537fa3b066420691214ed753**. . .Action: buy

```
INSERT into cart (SESSIONID,PART_ID,STATUS,LASTUPDATE) values
('1bc96cc0537fa3b066420691214ed753','sg245687','pending',current timestamp)
```

Figure 4-15 Inserting a record into the shopping cart table.

At any point you can press the **view_cart** button to see the contents of the cart (Figure 4-16). If you happen to change your mind about a particular item, then you can click the **delete** link for that item and it will be removed from the cart.



Red Book Order Entry

. . . Your session ID is **1bc96cc0537fa3b066420691214ed753**. . .Action: view_cart

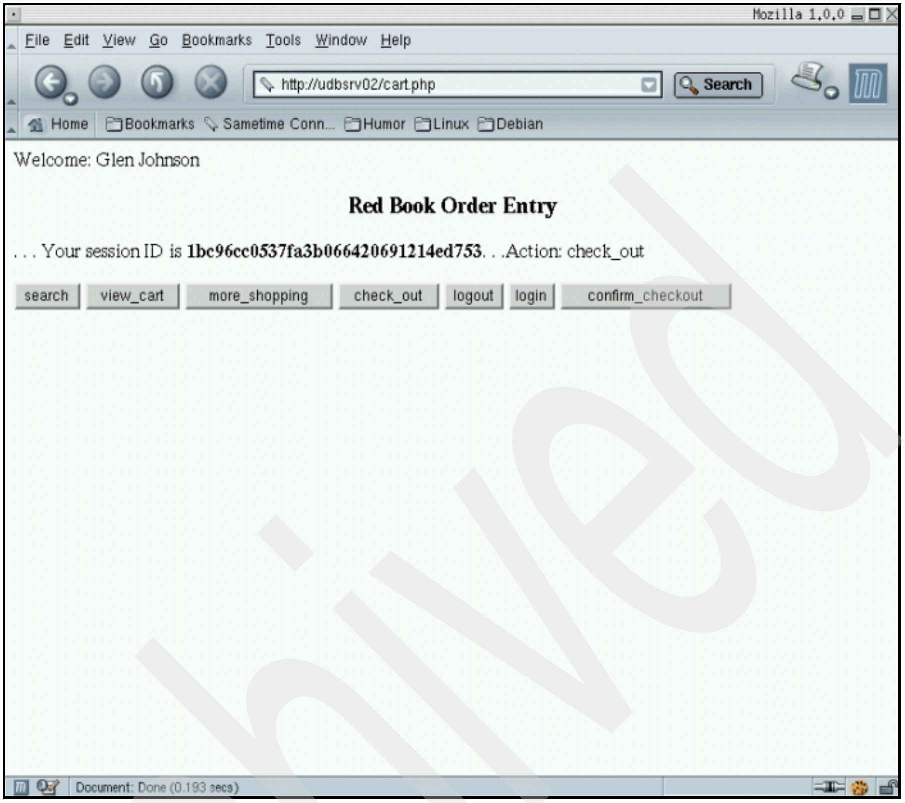
select ITEM_ID, SESSIONID , c.PART_ID,title, STATUS, LASTUPDATE from cart c,books b where b.part_id=c.part_id and sessionid ='1bc96cc0537fa3b066420691214ed753'

ITEM_ID	SESSIONID	PART_ID	TITLE
delete	1bc96cc0537fa3b066420691214ed753	sg246236	MigratingSiebel Database from DB2/Oracle for NT to
delete	1bc96cc0537fa3b066420691214ed753	sg245687	e-BusinessIntelligence: Leveraging DB2 for Linux on
delete	1bc96cc0537fa3b066420691214ed753	sg246128	DB2UDB V7.1 Porting Guide

Figure 4-16 Viewing the shopping contents of the shopping cart.

When you are satisfied with the contents of the cart, it's time to press **check_out** and finalize the order. If you have not logged in to the system, you will be asked to do so at this time. The program will be looking for an entry in the CUSTOMER table as identified by the C_NAME and C_CADDRESS fields. In a real world example, this customer account would probably include preferred payment options and registration of credit card accounts for use with this particular Web site.

Once you are logged into the system and after pressing **check_out**, the **confirm_checkout** button will appear on the far right side (Figure 4-17). When this button is pressed, the DB2 stored procedure create_order is called from the cart.php program. This is important because it performs an insert into the ORDER table, multiple inserts into the LINEITEM table, and an update to the cart table all within one unit of work within the database.



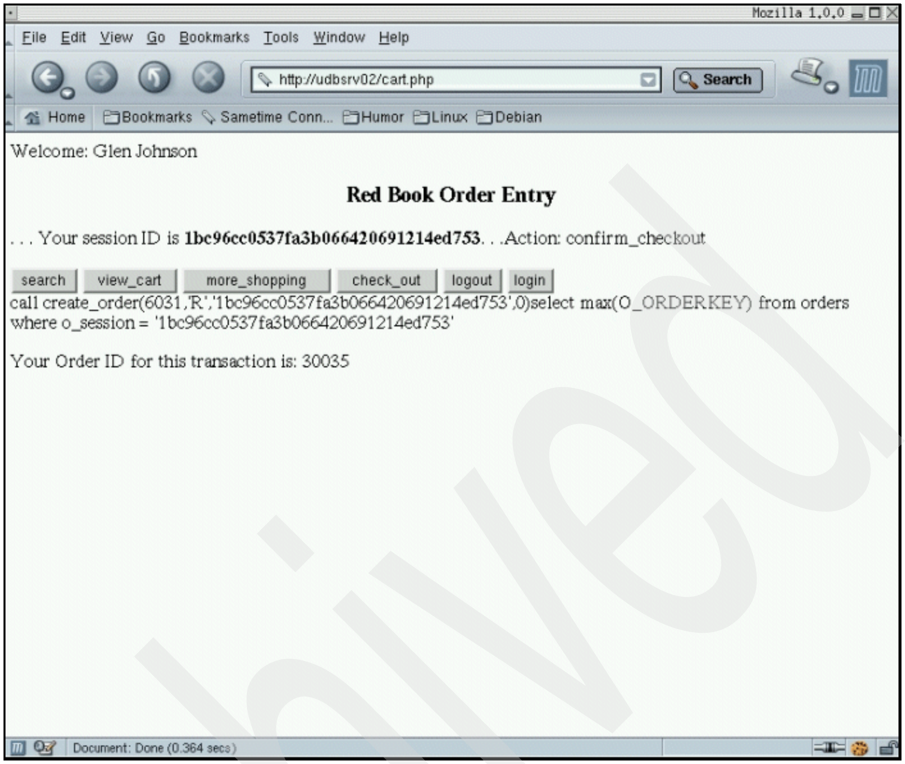
Welcome: Glen Johnson

Red Book Order Entry

. . . Your session ID is **1bc96cc0537fa3b066420691214ed753**. . .Action: check_out

Figure 4-17 Confirming the order

After pressing **confirm_checkout**, you will be notified of the order ID which was generated for this transaction (Figure 4-18).


Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/cart.php
Search
Home Bookmarks Sametime Conn... Humor Linux Debian
search view_cart more_shopping check_out logout login
Document: Done (0.364 secs)

Welcome: Glen Johnson

Red Book Order Entry

. . . Your session ID is **1bc96cc0537fa3b066420691214ed753**. . .Action: confirm_checkout

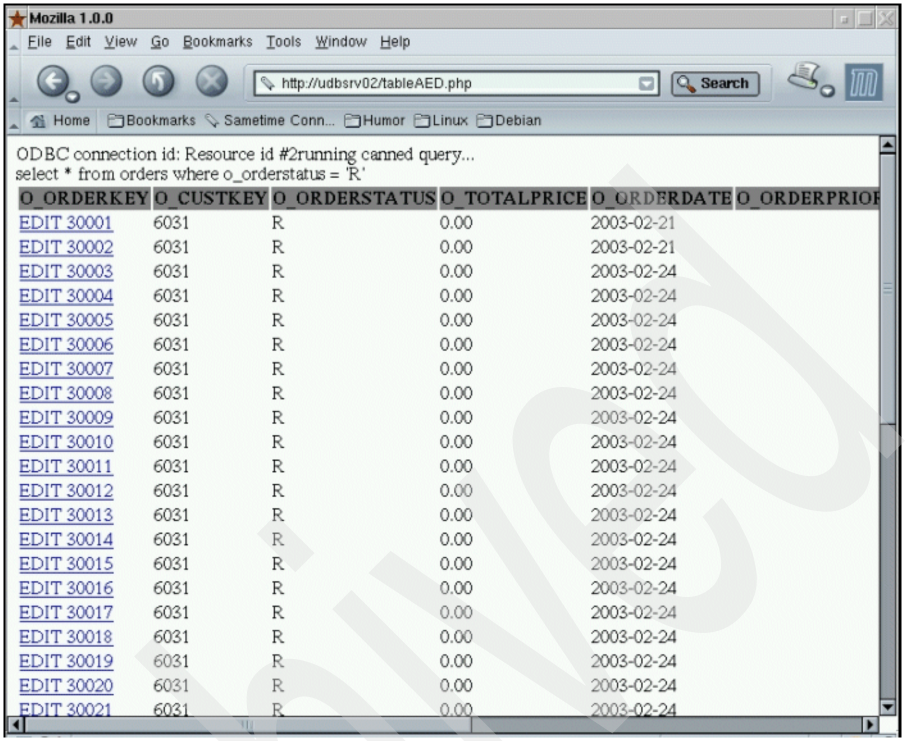
call create_order(6031,'R','1bc96cc0537fa3b066420691214ed753',0)select max(O_ORDERKEY) from orders where o_session = '1bc96cc0537fa3b066420691214ed753'

Your Order ID for this transaction is: 30035

Figure 4-18 Confirmation of the generated order

Process orders

Now that we have our customers flocking to our online book store in droves and submitting orders, for example, we need an efficient way for our fulfillment clerks to find and process them. A quick and simple starting point is to utilize our General Table Maintenance function. To demonstrate this, click the **Process Orders** link, the General Table Maintenance is shown. Click the drop down list button on Query box, and you will then be presented with a list of previously stored queries acting on the ORDER table. Select the one with o_orderstatus='R' (for Received orders) and press **query**. You will then see a dynamically generated HTML report showing all of the orders which have been submitted and are awaiting attention from the appropriate administrative staff (Figure 4-19).


Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/tableAED.php
Search
Home
Bookmarks
Sametime Conn...
Humor
Linux
Debian

ODBC connection id: Resource id #2running canned query...
select * from orders where o_orderstatus = 'R'

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIOR
EDIT 30001	6031	R	0.00	2003-02-21	
EDIT 30002	6031	R	0.00	2003-02-21	
EDIT 30003	6031	R	0.00	2003-02-24	
EDIT 30004	6031	R	0.00	2003-02-24	
EDIT 30005	6031	R	0.00	2003-02-24	
EDIT 30006	6031	R	0.00	2003-02-24	
EDIT 30007	6031	R	0.00	2003-02-24	
EDIT 30008	6031	R	0.00	2003-02-24	
EDIT 30009	6031	R	0.00	2003-02-24	
EDIT 30010	6031	R	0.00	2003-02-24	
EDIT 30011	6031	R	0.00	2003-02-24	
EDIT 30012	6031	R	0.00	2003-02-24	
EDIT 30013	6031	R	0.00	2003-02-24	
EDIT 30014	6031	R	0.00	2003-02-24	
EDIT 30015	6031	R	0.00	2003-02-24	
EDIT 30016	6031	R	0.00	2003-02-24	
EDIT 30017	6031	R	0.00	2003-02-24	
EDIT 30018	6031	R	0.00	2003-02-24	
EDIT 30019	6031	R	0.00	2003-02-24	
EDIT 30020	6031	R	0.00	2003-02-24	
EDIT 30021	6031	R	0.00	2003-02-24	

Figure 4-19 Process orders

Generating XML file

In recent years XML has provided a means of moving toward a standard format for communication of information among trading partners. It is quite easy to dynamically encapsulate the output of our PHP based DB2 queries with XML which in turn can be used in messaging with trading partners. To see how this works, select the query mentioned previously in the **Process Orders** example, but press **xmlout** instead of **query**. An example XML file will be generated and displayed on the screen. This information can be included in a purchase order submitted to a trading partner. The XML file can be transformed into a more browser and human readable HTML using XLST. The XML file could even be mailed directly to a trading partner via PHP's mail functions.

4.5 Understanding the DB2/PHP Web application code

Now that we have seen the Redbooks ordering Web application in action, we will delve deeper into the way the code works.

In our examples we have turned on the register_globals function in the php.ini file:

```
register_globals=On
```

This allows for a more rapid development of PHP application prototype, although it is less secure. The benefit is that variables sent to a PHP program via a HTTP POST or GET functions are implicitly available in that program as global variables. In our example scripts we make extensive use of a variable called $action. We use the value of that variable to determine what we want a particular PHP script to do and therefore what sections of code within that script may need to get executed. With register_globals=On, we can simply reference $action with no further preparation required. If register_globals were set to Off, then we would need to extract the value of $action from the _POST or _GET arrays with something prior to using the $action variable that had been passed into a script, such as:

```
$action = $_POST[“action”];
```

or,

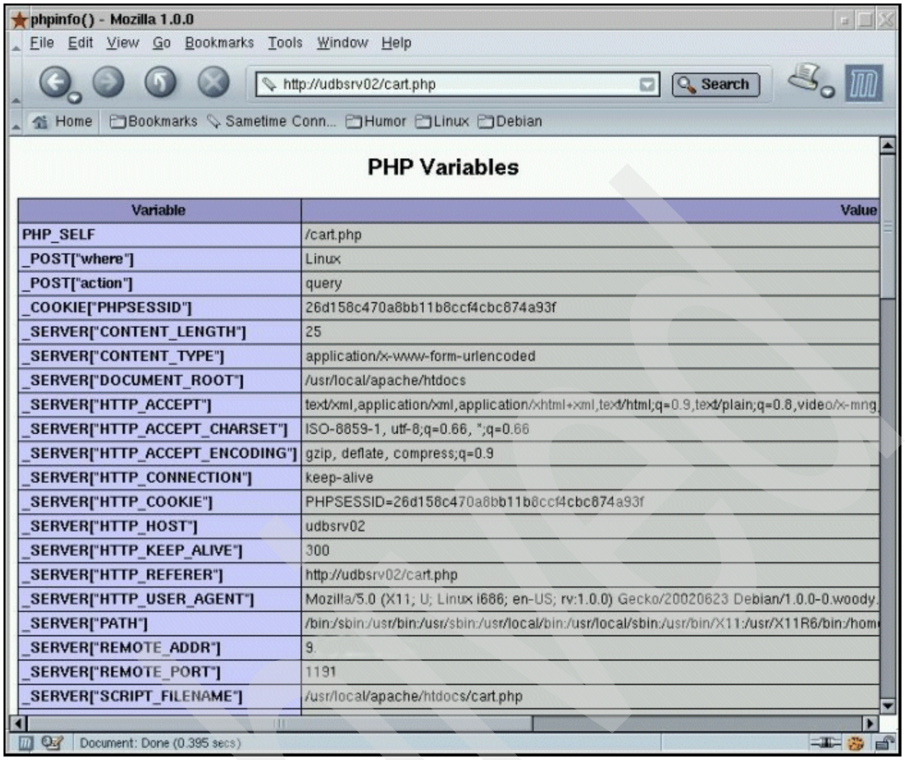
```
$action =$_GET[“action”];
```

Testing PHP functionality

The administrative clerk can determine if the PHP function of the Apache server has been enabled properly by clicking the **Simple PHP Script**, which should open and render the file *test.php*. The contents of the file consists of one line which calls the *phpinfo()* function. Everything between the <? and ?> tags is interpreted by the PHP engine inside the Apache server, whereas that material outside the PHP tags is interpreted in the traditional manner by Apache. For example, execution of the following code should yield a voluminous amount of information about the Apache/PHP environment and configuration:

```
<? phpinfo(); ?>
```

When writing PHP code, the phpinfo() function is an incredibly useful tool. For example, phpinfo(32) will print a list of php variables. In this example you can see the values of the $_POST[“where”] and $_POST[“action”] which have been passed into the cart.php program. If the PHP function in Apache is not working properly, you may just see the PHP source code in your browser, otherwise you will see output similar to that in Figure 4-20.


phpinfo() - Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/cart.php
Search
Home
Bookmarks
Sametime Conn...
Humor
Linux
Debian
Document: Done (0.395 secs)

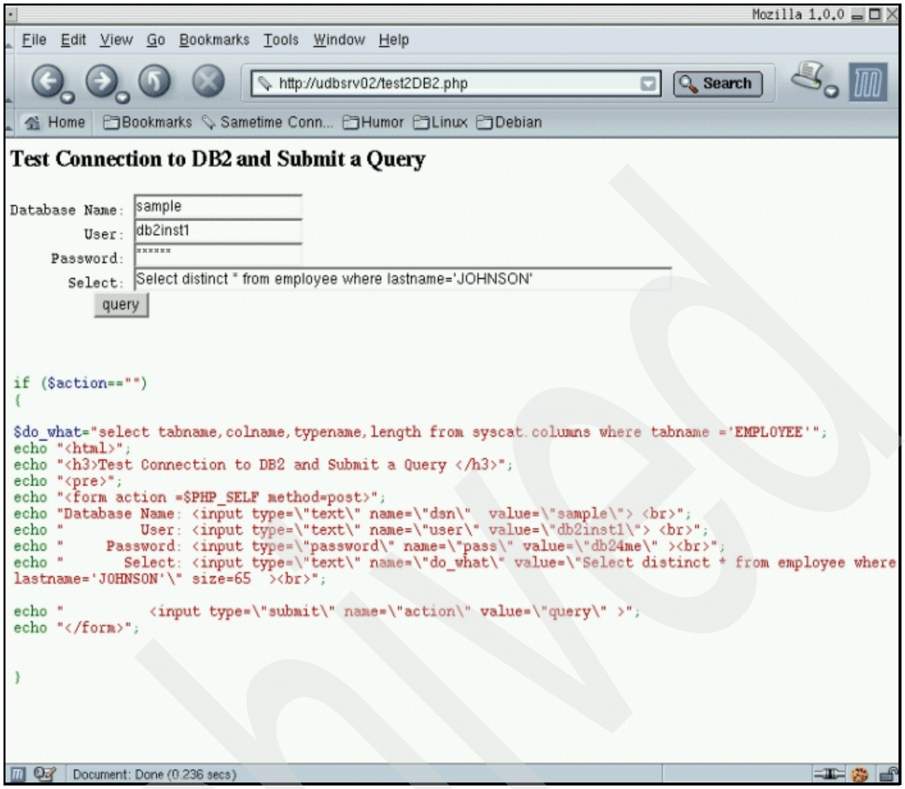
PHP Variables

Variable	Value
PHP_SELF	/cart.php
_POST["where"]	Linux
_POST["action"]	query
_COOKIE["PHPSESSID"]	26d158c470a8bb11b8ccf4cbc874a93f
_SERVER["CONTENT_LENGTH"]	25
_SERVER["CONTENT_TYPE"]	application/x-www-form-urlencoded
_SERVER["DOCUMENT_ROOT"]	/usr/local/apache/htdocs
_SERVER["HTTP_ACCEPT"]	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,
_SERVER["HTTP_ACCEPT_CHARSET"]	ISO-8859-1, utf-8;q=0.66, *;q=0.66
_SERVER["HTTP_ACCEPT_ENCODING"]	gzip, deflate, compress;q=0.9
_SERVER["HTTP_CONNECTION"]	keep-alive
_SERVER["HTTP_COOKIE"]	PHPSESSID=26d158c470a8bb11b8ccf4cbc874a93f
_SERVER["HTTP_HOST"]	udbsrv02
_SERVER["HTTP_KEEP_ALIVE"]	300
_SERVER["HTTP_REFERER"]	http://udbsrv02/cart.php
_SERVER["HTTP_USER_AGENT"]	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.
_SERVER["PATH"]	/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/bin/X11:/usr/X11R6/bin:/hom
_SERVER["REMOTE_ADDR"]	9.
_SERVER["REMOTE_PORT"]	1191
_SERVER["SCRIPT_FILENAME"]	/usr/local/apache/htdocs/cart.php

Figure 4-20 Output of the phpinfo(32) function - Listing PHP variables

PHP talking to DB2

The *test2DB2.php* program consists of two separate stanza's or sections. Only one of which get's executed depending on the value of the variable $action. This is the simplest variation on a theme which will recur in the programs which we will examine later. If the $action variable is not valued then the php script simply prints out an HTML form where the database, user ID, password, and query information can be entered as shown in Figure 4-21. Even though this HTML is dynamically generated, the browser treats it the same as if it were static HTML. One of the submit variables in the form is named *action* and it is valued to **query** when the submit button is pressed. The form action is set to the PHP variable $PHP_SELF, which is the name of the PHP program (test2DB2.php). So when the submit button is pressed, the same program is executed again, but this time the value of $action is now set to **query** and the second stanza is executed instead, as depicted in Figure 4-22. The PHP program uses the odbc_pconnect() function to open up a persistent connection to the database with the ushered and password specified on the form.


Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/test2DB2.php
Search
Home Bookmarks Sametime Conn... Humor Linux Debian
Document: Done (0.236 secs)

Test Connection to DB2 and Submit a Query

```
Database Name: sample
         User: db2inst1
     Password: ******
       Select: Select distinct * from employee where lastname='JOHNSON'
               query
```

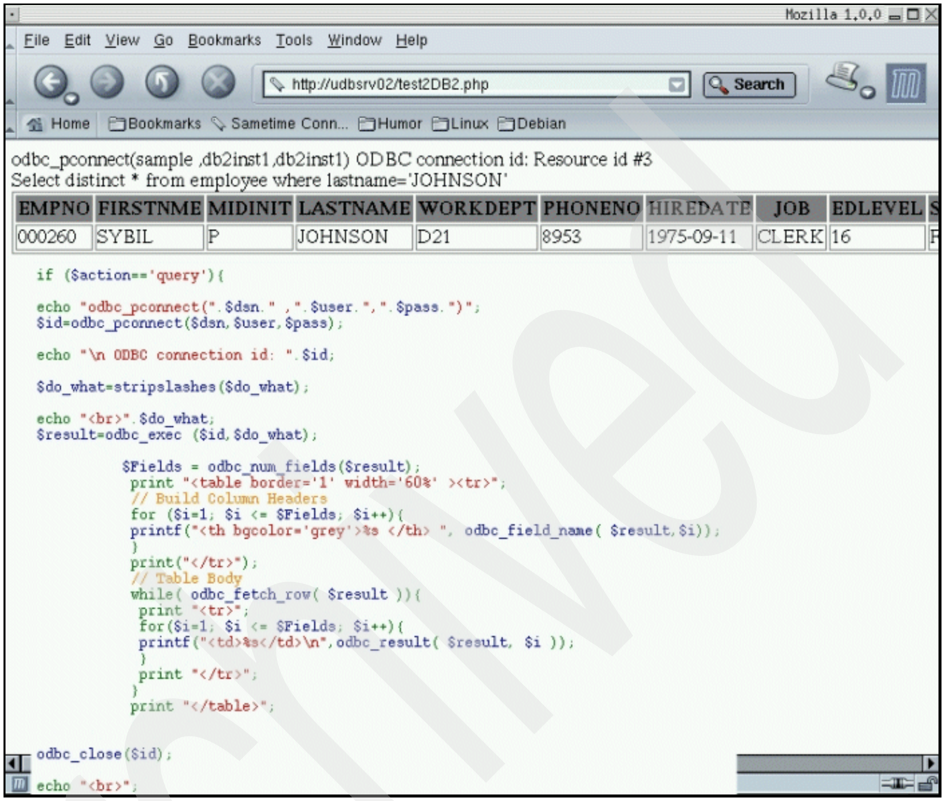
```
if ($action=="")
{

$do_what="select tabname,colname,typename,length from syscat.columns where tabname ='EMPLOYEE'";
echo "<html>";
echo "<h3>Test Connection to DB2 and Submit a Query </h3>";
echo "<pre>";
echo "<form action =$PHP_SELF method=post>";
echo "Database Name: <input type=\"text\" name=\"dsn\"  value=\"sample\"> <br>";
echo "         User: <input type=\"text\" name=\"user\" value=\"db2inst1\"> <br>";
echo "     Password: <input type=\"password\" name=\"pass\" value=\"db24me\" ><br>";
echo "       Select: <input type=\"text\" name=\"do_what\" value=\"Select distinct * from employee where
lastname='JOHNSON'\" size=65  ><br>";

echo "          <input type=\"submit\" name=\"action\" value=\"query\" >";
echo "</form>";


}
```

Figure 4-21 Querying DB2: Input window and corresponding source code

In the Linux environment PHP makes semantic use of ODBC function names via its unified ODBC function calls to the DB2 CLI interface. In the Windows environment, we make use of the ODBC driver manager to create an ODBC data source to be used by PHP. The ODBC function names are identical in each of these two environments.

For purposes of illustration, we are using the SAMPLE database here. You can also use the RBORDERS database which we have previously installed.

In Figure 4-22 we show the output and corresponding source code for the query which is submitted and executed using the *odbc_exec* function. We next dynamically build the column headings for an HTML table by using the odbc_num_fields() and odbc_field_name() functions. The actual results of the query are obtained with the odbc_fetch_row() and odbc_result() functions. It is important to note that the results output in the HTML table is entirely dependent on the table structure and requested fields of the submitted query. So if different fields were requested in the query, that would be reflected automatically in the output table.


Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/test2DB2.php
Search
Home
Bookmarks
Sametime Conn...
Humor
Linux
Debian

odbc_pconnect(sample ,db2inst1 ,db2inst1) ODBC connection id: Resource id #3
Select distinct * from employee where lastname='JOHNSON'

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	S
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11	CLERK	16	F

```
if ($action=='query'){

echo "odbc_pconnect(".$dsn." ,".$user.",".$pass.")";
$id=odbc_pconnect($dsn,$user,$pass);

echo "\n ODBC connection id: ".$id;

$do_what=stripslashes($do_what);

echo "<br>".$do_what;
$result=odbc_exec ($id,$do_what);

          $Fields = odbc_num_fields($result);
           print "<table border='1' width='60%' ><tr>";
           // Build Column Headers
           for ($i=1; $i <= $Fields; $i++){
           printf("<th bgcolor='grey'>%s </th> ", odbc_field_name( $result,$i));
           }
           print("</tr>");
           // Table Body
           while( odbc_fetch_row( $result )){
            print "<tr>";
            for($i=1; $i <= $Fields; $i++){
            printf("<td>%s</td>\n",odbc_result( $result, $i ));
            }
            print "</tr>";
           }
           print "</table>";


odbc_close($id);

echo "<br>";
```

Figure 4-22 Querying DB2: Output and corresponding source code

General Table Maintenance with PHP

We will now build upon our basic theme of the $action variable directing execution of the appropriate stanza as seen in test2DB2.php as we move into the General Table Maintenance section of the main menu and look at tableAED.php. This program provides a dynamic framework for the manipulation of data within any table in your database. When you first start this program, the *From select menu* is dynamically populated with table choices from the system catalog with this query:

```
SELECT tabname as “FORM” form sycat.tables where tabschema ='$schema’ for
read only
```

We use the read only cursor here when we are working with the syscat.tables. For detailed information about using the read only cursor, refer to the article, “Application Development Experiences with PHP and IBM DB2 Universal Database Version” which is available from this Web site:

http://www7b.software.ibm.com/dmdd/library/techarticle/0301liu/0301liu.html

This approach allows the program to manipulate data within a particular table, simply by creating the table; no modifications to the tableAED.php program are necessary.

The program adds the ability to perform INSERT, UPDATE, and DELETE on the set of tables within the database. Queries can be issued in an “ad hoc” manner or stored for later use. In its current state, tableAED.php is restricted to tables in the RBORDERS database, although it would be an easy extension to work with additional databases as well.

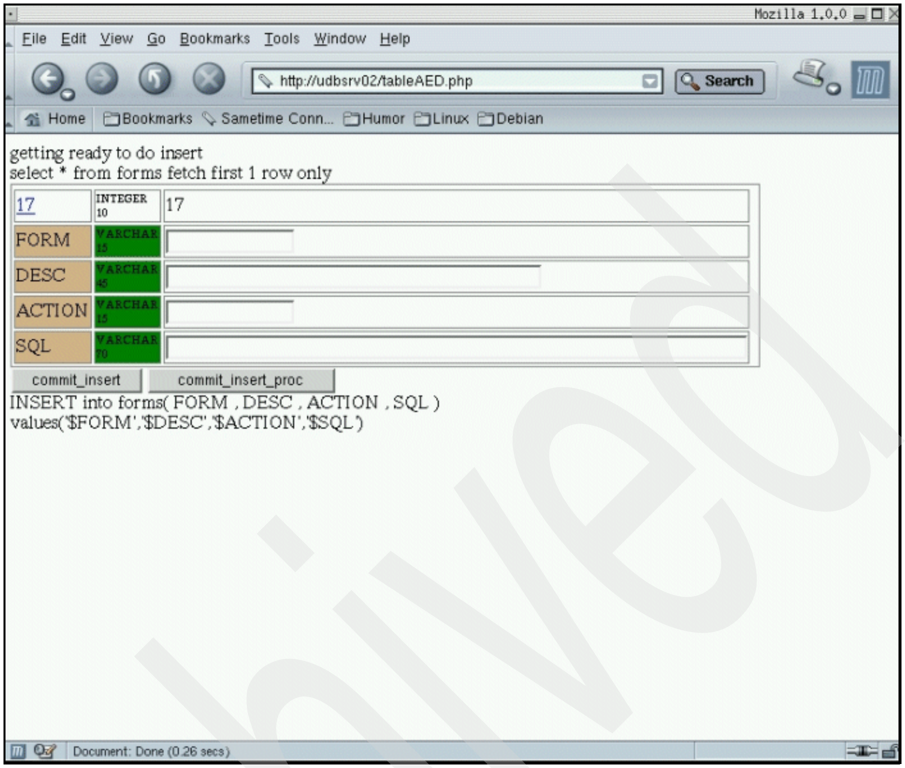
The SELECT and WHERE input box allows free form input of any query and or WHERE clause and execution is achieved by clicking on the **query** button. The query results are returned and written dynamically into an HTML table in a manner similar to test2DB2.php with the major distinction that a hot link is created for each record. The program looks for the first field in the record as the primary key and a link with an action=’edit’ is created.

Another focal point for the design of this program is the FORMS table which contains at least one record for each table that we want to manipulate. If the query is performed on the FORMS table, then individual records in that table can be edited. It is in the FORMS table that interesting queries on the other tables can be stored. Queries are stored in the *action* field of the FORMS table. Other actions can be added to the FORMS table and the PHP code can be extended to take advantage of those actions. Clicking the **get_queries** button will retrieve a list of previously defined queries on a particular table. The desired query can be selected and executed by clicking the **query** button.

Clicking the **insert** button dynamically generates an HTML form for inserting records into a particular table. This routine is able to read the column names and data types from the table. At this point it is useful to note how we have separated the DML UPDATE and INSERT functions of the main tableAED.php into a separate formDML.php file and integrated them using PHP’s include() function:

```
include(‘formDML.hp’)
```

at the top of tableAED.php, formDML.php contains mainly a series of INSERT and UPDATE DML statements corresponding to each table we are working with. In our example the formDML.php file is not dynamically generated, although this would be an good enhancement. The INSERT form (Figure 4-23) does, however, provide you with a very good guess regarding what the INSERT statement should be for a particular table.


Mozilla 1.0.0
File Edit View Go Bookmarks Tools Window Help
http://udbsrv02/tableAED.php
Search
Home
Bookmarks
Sametime Conn...
Humor
Linux
Debian
Document: Done (0.26 secs)

getting ready to do insert
select * from forms fetch first 1 row only

17	INTEGER 10	17
FORM	VARCHAR 15	
DESC	VARCHAR 45	
ACTION	VARCHAR 15	
SQL	VARCHAR 70	

commit_insert
commit_insert_proc

INSERT into forms(FORM , DESC , ACTION , SQL)
values('$FORM','$DESC','$ACTION','$SQL')

Figure 4-23 Performing a table insert using tableAED.php

After you have entered your data for the record, clicking the **commit_insert** button executes the appropriate INSERT stanza in formDML.php with something with an INSERT statement similar to the one in the Figure 4-23, such as:

```
$result=odbc_exec($id,$insertsql);
```

In addition we have the option of using a DB2 stored procedure for performing the insert. We have created an example of a procedure which performs an insert on the PART table and this can be tested by clicking the **commit_insert_proc** button to call the *insert_part* procedure by executing this PHP code:

```
$insertsql="call
insert_part('$P_NAME','$P_MFGR','$P_BRAND','$P_TYPE',$P_SIZE,'$P_CONTAINER',$P_
RETAILPRICE,'$P_COMMENT')";
$result=odbc_exec($id,$insertsql);
```

Note: The current code implements commit_insert_proc for record inserts only into the PART table.

The code to create the insert_part procedure is contained in the /usr/local/apache/db2gen/insert_part.pro file.

While we are talking about the PART table, we want to remind you about the triggers section of the *rborders.ddl* file. The *parthistory* trigger is waiting to insert a record into the PARTHISTORY table after a new record is inserted into the PART table. Even though the trigger is activated by pressing the commit_insert button in the PHP program, it is completely independent of it. This logging to the PARTHISTORY table, which may have to be maintained independently in multiple client applications is handled automatically by the database. The *orderhist* trigger is a similar but slightly more complicated trigger which is activated and inserts a record into the ORDERHIST table when the ORDERSTATUS field in the ORDER table is changed during an update. You can verify that the triggers are working by using tabAED.prg to insert records into PART or update records in ORDER, followed by running queries against ORDERHIST and PARTHIST, respectively.

Order Entry - shopping cart

As we move on to look at cart.php, the heritage of the theme for the function of the code is still quite recognizable, although it has become less generic. Each of the buttons on the top of the form correspond to a value of the $action variable which is inserted back into the program when the button is pressed. You may be wondering about how we keep track of who is pressing those buttons.

Part of the elegance of the http protocol is that it is stateless. A Web browser makes an HTTP connection to a Web server, sends and receives information, and then disconnects. This type of behavior is necessary for a Web site to scale to any reasonable number of users. An order entry transaction on a Web site typically requires several such connections in succession to complete the entire transaction. We need a mechanism to maintain state as that series of HTTP connections is sent from an individual browser. This problem has been partially solved by serializing the transaction and writing that information to the browser's cookie file on the user's local hard drive. Unfortunately the use of cookies is not without various real and imagined security vulnerabilities and as a result some users refuse to allow their use. In our example we have solved the problem in an alternative way using the PHP sessions functions. With PHP sessions, the browser's interaction with the Web server is serialized, but with that information being stored on the Web server and not on the user's local system. In addition to a random serial number, we store the last query executed by the user so that when **more_shopping** is clicked the query can be easily be re-executed. We also keep track of whether or not a valid customer is currently logged in.

5

Application development

It has become increasingly important for a database to effectively support a wide range of open standards and open source products for application development. DB2 UDB provides accessibility for many standard interfaces and comes with a full suite of development tools.

In this chapter we introduce the aspects of developing applications for DB2. We will also provide examples of using these development features within our sample application.

This chapter discusses the following topics:

- Preparing for application development
- DB2 Development Tools
- Development features within DB2
- Programming languages
- DB2 Scripting in Linux and Windows

5.1 Preparing for application development

This section discusses the various application development environment configurations that can be used to connect to DB2.

5.1.1 Creating a development environment

DB2 supports local and remote application development. Although the procedure for configuring remote clients initially requires more setup, once the connection has been created the procedure for creating applications is identical.

Here we illustrate three typical development configurations:

- **Standalone configuration:** Figure 5-1 represents a standalone development environment. In this configuration, the application development and database server are on the same physical machine. By default, DB2 Setup Wizard installs the components necessary for application development on DB2 server while installing DB2. This is the environment we created in Chapter 2 and then developed our application in Chapter 4, and will continue to use in this chapter.

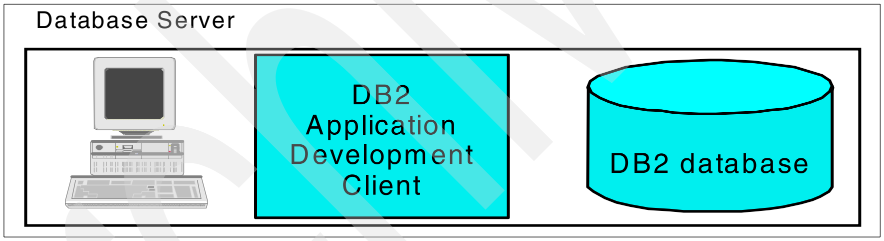


Figure 5-1 Standalone development server

- **Client-server configuration:** Figure 5-2 represents a client server development environment. In this configuration, the DB2 client is installed on a remote machine. The remote client communicates with the database server via the network. On the server side, you need to configure the database server for remote client access. On the client side, you need to setup the database connection. The detailed steps for remote client setup is covered in Chapter 6, “Database management” on page 195.

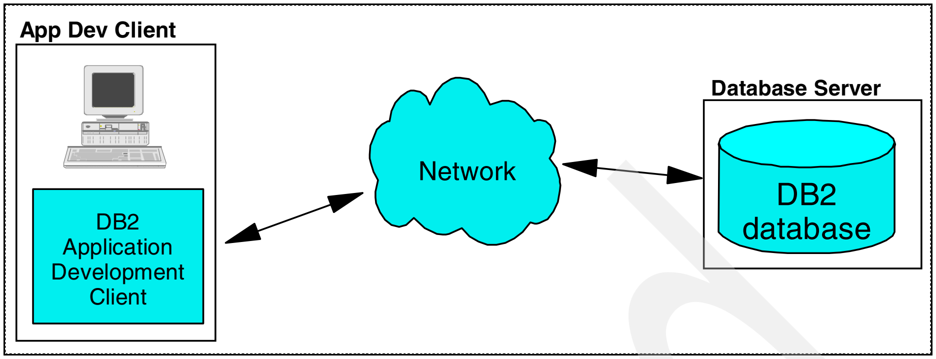


Figure 5-2 Client Server development

- **Web server configuration:** Figure 5-3 is an example of a typical Web Server environment where the Web Application Server and Web Server connect to DB2 as a client, and a separate application development client also remotely connects to DB2. We need to configure these two clients to connect to the DB2 Server.

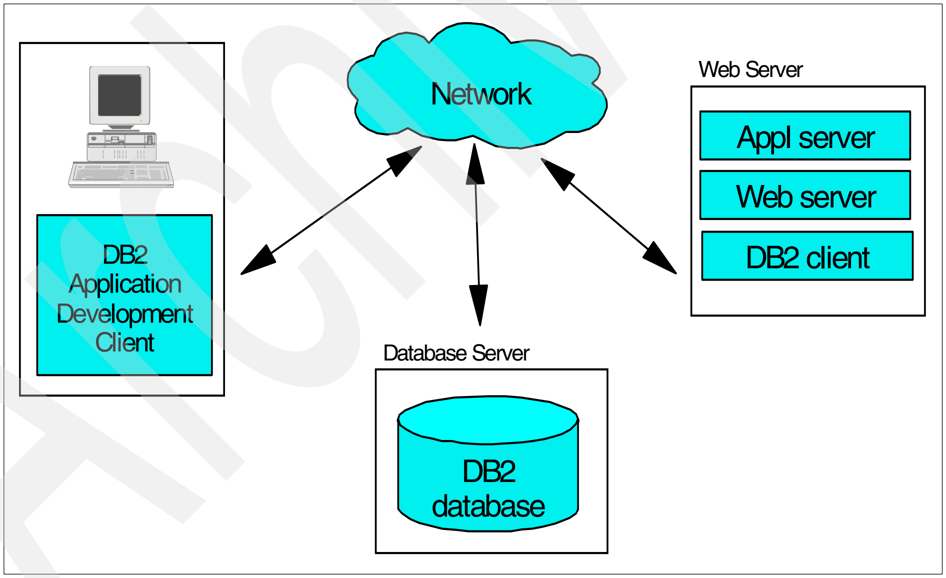


Figure 5-3 Two remote clients connecting to a DB2 Server

5.1.2 Connecting DB2 applications

To access the DB2 data, applications require a connection to DB2, which can be configured in a variety of ways:

- **Connecting to a local DB2 server**: This type of connection requires the applications to run on the same machine where the DB2 server resides. When the application connects to the database locally, the performance should be better since there's no network traffic time. However, performance could also be impacted, due to both DB2 and applications, which are sharing the same resource.
- **Connecting to a remote DB2 server through a client interface**: This is the classic client-server model for database connectivity. The application sits on a desktop machine using a local DB2 client to connect to a remote DB2 server. Any DB2 client can be used for this purpose. The Application Development Client or the Administration Client are both installation options of the trial code version we used in this book.

  The other client alternative is the DB2 runtime client. This is a light-weight client that provides the functionality required for an application to access DB2 servers. It has less GUI facilities than the other clients so configuration is typically done from the DB2 Command Line Processor (CLP) and it requires less of a disk footprint.

  The DB2 8.1 runtime client, as well as the other DB2 clients, can be downloaded from:

  http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report
- **Connecting to a remote DB2 server directly**: This is the capability offered by Java applets and applications, which allows them to connect directly to DB2 without DB2 client code. The connection function is provided by a driver. There are two types of this driver available with DB2. The type 3 driver can be loaded by the Web browser along with the applet, or the applet driver can be used in standalone Java applications. The new type 4 driver is the replacement for the type 3 driver, but still allows remote connections to DB2 without the need for a local client. There are more details on Java connectivity to DB2 in 5.4, "Application interfaces to DB2" on page 178 and 5.5.1, "Java" on page 185.
- **Connecting through a Web server to a DB2 server**: This is the common scenario where a user connects to a browser based application which either accesses DB2 directly or through a Web application server. No DB2 code is required on the user machine. To enable the database connectivity, a DB2 client is required to be configured between the Web server and DB2. If a Web application server is being used then the DB2 client needs to be configured between the Web application server and DB2. Our sample application is an example of the former where the application on the Web server is connecting directly to DB2.

5.2 DB2 Development Tools

Once development clients have been configured we can start to develop applications. This section discusses the tools that are provided as standard in DB2 to assist the application development and maintenance.

5.2.1 DB2 Development Center

The Development Center is a DB2 graphical tool to make the DB2 application developer's job easier. It includes facilities for building user-defined functions, building Java or SQL stored procedures, mapping Enterprise Java Beans (EJBs) to structured types, and incorporates an integrated debugger for SQL stored procedures and UDFs.

In 5.3.1, "Stored procedures" on page 158, we provide an example of using the Development Center to create a project and then build a stored procedure for our sample application. In 5.3.3, "User Defined Functions (UDFs)" on page 168 we also provide an example of building an SQL UDF.

Additional features of the Development Center can be seen in Chapter 8 of *DB2 UDB Exploitation of the Windows Environment*, SG24-6893.

There is also a very useful article on the different features of the Development Center available from:

http://www7b.software.ibm.com/dmdd/library/techarticle/0207alazzawe/0207alazzawe.html

5.2.2 DB2 Project Deployment Tool

The Deployment Tool is used to deploy the routines (stored procedures and UDFs) that are saved in a file to a target database. The target database must be compatible with the database that the object was created for. You normally use the Deployment Tool to deploy routines that are contained in a zip file that you exported using the Export wizard of the Development Center.

These routines can either be exported as source files or as binaries. Binary files can only be imported into the same operating system that they were exported from. This facility is useful if you have a C compiler on one Windows machine which compiles your stored procedures. These stored procedures can then be deployed to other Windows machines.

> **Note:** To deploy the binaries, the Project Deployment Tool uses two supplied DB2 stored procedures, `PUT_ROUTINE_SAR` and `GET_ROUTINE_SAR`.

This GUI tool can be either:

- Called from within the Development Center in Windows or Linux: **Tools —> Deployment Tool**
- In Windows by selecting: **Start —> Programs —> IBM DB2 —> Development Tools —> Project Deployment**

A specific example of using the Project Deployment Tool is contained within Chapter 8 of *DB2 UDB Exploitation of the Windows Environment*, SG246893.

5.2.3 DB2 Command Center

The Command Center was described in 3.3, “Command Center” on page 51. As a general purpose GUI tool we demonstrate its effectiveness in aiding application development by using it to test:

- A stored procedure in 5.3.1, “Stored procedures” on page 158
- A user defined function in 5.3.3, “User Defined Functions (UDFs)” on page 168.

Also we utilize the Command Center to create a script that creates and tests:

- A DB2 trigger in 5.3.2, “Triggers” on page 165

5.2.4 SQL Assist

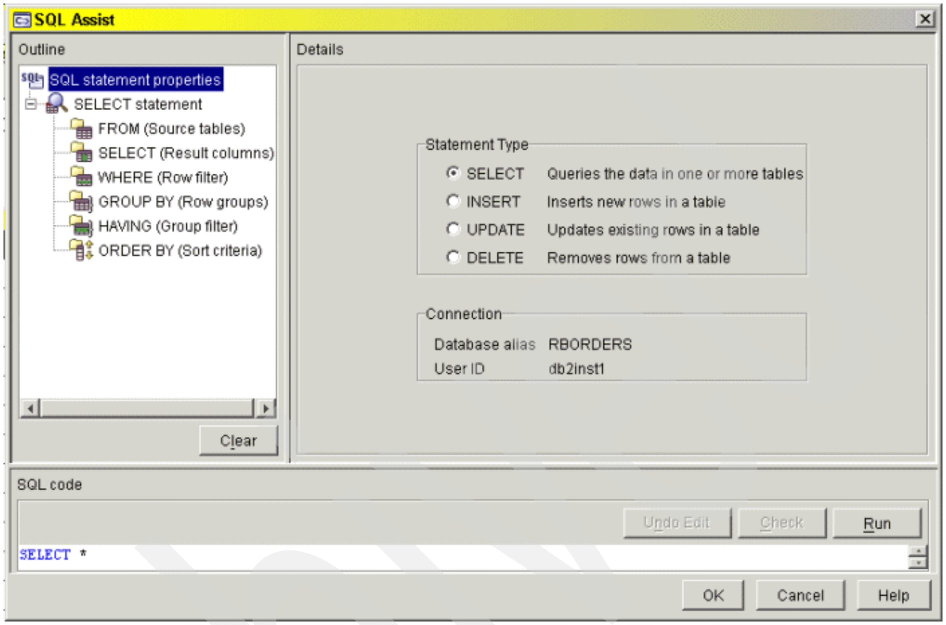
SQL Assist is an easy-to-use GUI tool that will build SQL statements for you in a step-wise fashion. It supports the building of SQL SELECT, INSERT, UPDATE and DELETE statements. SQL Assist has been enhanced in DB2 Version 8.1 to include:

- Assistance for creating table joins
- SQL syntax checking
- The option to copy and paste an existing SQL statement into SQL Assist, and then use the SQL Assist interface to make modifications

SQL Assist can be launched from within the following DB2 GUI tools:

- Control Center
- Command Center
- Replication Center
- Development Center
- Data Warehouse Center

Figure 5-4 is the opening window for the SQL Assist tool, where you can specify what type of statement you want to create. Depending on the type of statement you select, the steps to create the statement will differ.



Statement Type

● SELECT Queries the data in one or more tables
○ INSERT Inserts new rows in a table
○ UPDATE Updates existing rows in a table
○ DELETE Removes rows from a table

Connection

Database alias RBORDERS
User ID db2inst1

```
SELECT *
```

Figure 5-4 Opening window for SQL Assist

For INSERT statements, you can work with the following elements:

- FROM: Choose the tables that will be accessed. If you specify multiple tables they can be joined in this step.
- SELECT: Choose the columns that you want to return values for. This step also allows you to customize the columns using the Expression Builder, such as concatenating two columns.
- WHERE: Add conditions to the result set.
- GROUP BY: To group rows by values in columns.
- HAVING: To filter group by rows.
- ORDER BY: Specifies an ordering of the rows of the result table.

For INSERT statements, you can work with the following elements:

- INSERT INTO: Specifies the table to insert rows into.
- VALUES: Specifies the data to be inserted into the table.

For UPDATE statements, you can work with the following elements:

- UPDATE: Specifies the target table of the update.
- SET: Specifies the values to update.
- WHERE: Specifies the row filtering for the update.

For DELETE statements, you can work with the following elements:

- DELETE FROM: Specifies the table to delete rows from.
- WHERE: Specifies the row filtering for the delete.

The Details area displays the appropriate panel for the element that you selected. You build your SQL statement by working with this panel. The changes that you make are reflected in the SQL code area.

The SQL code area displays the current SQL statement. You can edit the SQL statement or paste an existing SQL statement into this area. If you make any changes in this area, you will not be able to take any actions in the Details area until you validate the syntax of the modified SQL.

To validate the syntax, click the **Check** button. If the syntax is correct, the Details area will be enabled again, and its content will be updated to reflect your changes. To run the SQL statement, click the **Run** button.

Example - Validate and run an existing SQL statement

In this simple scenario we take an existing query and modify it using the SQL Assist tool.

Launch the SQL Assist tool from within Command Center.

- Click the Interactive tab.
- Connect to RBORDERS database.
- Click the SQL Assist button.

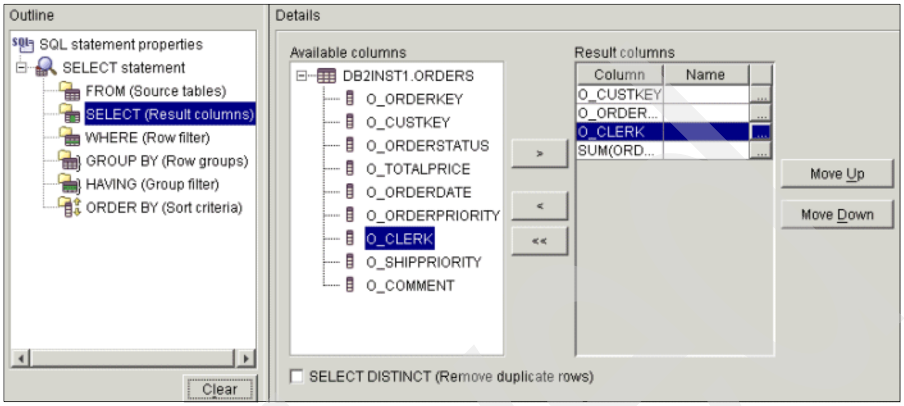
Within SQL Assist:

- Paste the following SQL into the SQL Code box and click **Check**.

  ```
  select o_custkey, o_orderdate, sum(o_totalprice) from
  db2inst1.orders where date(o_orderdate) = '03/03/1997' group by
  o_custkey, o_orderdate
  ```

- Once this completes, DB2 will reformat the query, click **Run**.
- Click **OK** to close the Query Result box.
- Click the **SELECT** button and select column **O_CLERK**.
- Click the > box to add the column to the query.

- Click **O_CLERK** to select the column and then click **Move Up** so that it will be returned to the left of the O_TOTALPRICE column.
- Your window should resemble Figure 5-5.



Column	Name
O_CUSTKEY	
O_ORDER...	
O_CLERK	
SUM(ORD...	

□ SELECT DISTINCT (Remove duplicate rows)

Figure 5-5 Selecting columns using SQL Assist

- Notice that the changes are being reflected in the SQL Code box and that the group by clause is being automatically updated.
- Click **Run** to run our edited query.

We have now shown you how to use SQL Assist to modify and test and existing SQL query.

5.2.5 Visual Explain

Visual Explain translates the information generated by db2explain, stored on explain tables, to a visual graphic. It assists the user to understand the path generated by DB2 optimizer to get the data. When you run a query from within the Command Center and choose to generate an access plan, DB2 will automatically create the necessary explain tables to store the information. This functionality has been demonstrated in 3.3, “Command Center” on page 51 and 6.5.3, “SQL tuning” on page 298.

For more information, You can find an extensive tutorial in the DB2 Version 8 *Visual Explain Tutorial*. You can also find an interesting article on DB2 explain tools and Visual Explain at this Web site:

http://gethelp.devx.com/techtips/db2_pro/10Min1101/gn1101-1.asp

5.2.6 Command-line explain tools

If you don't have Visual Explain installed on your machine, or you need to explain static SQL, you can use the command line tools. The command line tools are:

- db2expln
- dynexpln
- db2exfmt

db2expln

The db2expln tool describes the access plan selected for SQL statements. It can be used to obtain a quick explanation of the chosen access plan when explain data was not captured. For static SQL, db2expln examines the packages stored in the system catalog tables. For dynamic SQL, db2expln examines the sections in the SQL cache. Static SQL and Dynamic SQL will be discussed as part of 5.4.2, "Embedding SQL statements in your applications" on page 180.

dynexpln

The dynexpln tool can also be used to describe the access plan selected for dynamic statements. It creates a static package for the statements and then uses the db2expln tool to describe them. However, because the dynamic SQL can be examined by db2expln, this utility is retained only for backward compatibility.

db2exfmt

The db2exfmt tool can be used to format the contents of the explain tables. This tool is located in the *misc* subdirectory of the instance sqllib directory.

You can find detailed information about these tools in Appendix C of *Administration Guide: Performance*, SC09-4821.

5.2.7 Integration with other tools

Although the use of the Development Center makes the development of DB2 objects more straightforward, DB2 also provides add-ins for two of the most popular application development environments. These are IBM WebSphere® Studio and the Microsoft Visual Studio family of products.

WebSphere Studio

WebSphere Studio is an integrated development environment (IDE) that enables you to build, test, and deploy Java applications to WebSphere Application Server and DB2 Universal Database. The database components are JDBC enabled and allow you to:

- Explore, import design, and query databases.

- Work with either a local copy of an already deployed design, or create an entirely new design to meet your requirement.
- Use SQL wizard and SQL Query builder, which provides a visual interface for creating databases, tables, views, and executing SQL statements.

Microsoft Visual Studio products

The DB2 Development Add-Ins for Visual Studio 6.0 are designed to provide database developers with server-side development features, similar to those provided in the new DB2 Development Center, and client-side development features, such as automatic ActiveX Data Objects (ADO) code generation for your server objects. The features include:

- Integration with Visual Basic, Visual C++, and Visual InterDev 6.0 development environments.
- Wizards that make it easy to create Java and SQL procedures, SQL UDFs, MQSeries table UDFs, OLE DB table UDFs, and XML table UDFs.
- Integration with Visual Source Safe 6.0. to allow source code control management.
- Client-side ADO code generation for DB2 connections and objects.
- Client-side Data Environment object creation for DB2 connections, procedures, and functions.
- Read-only access to stored procedures, UDFs, triggers, tables, and views.

Once you have installed the DB2 client or a server with the application development features you need to register the add-ins by clicking either:

Start —> Programs —> IBM DB2 —> Set-up Tools —> Register Visual Studio Add-Ins

Or, from a command prompt, run:

```
db2vscmd register
```

Microsoft Visual Studio .NET

DB2 UDB also delivers add-ins for Microsoft Visual Studio .NET. These add-ins, designed to greatly simplify the task of building ADO.NET applications, let you use Visual Studio .NET to build DB2 server-side objects, such as stored procedures and user-defined SQL functions (UDFs).

Further information

For more information about using the Microsoft development environment, refer to Chapter 8 of *DB2 UDB Exploitation of the Windows Environment*, SG24-6893.

The following Web site also acts as a good introduction to using add-ins with Microsoft Visual Studio:

http://www7b.software.ibm.com/dmdd/library/techarticle/0208alazzawe/0208alazzawe.html

There is an excellent tutorial provided as part of the DB2 documentation called “Development Center Tutorial for Video Online using Microsoft Visual Basic”.

The home page for IDE add-ins for DB2 can be found at:

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

5.3 Development features within DB2

After discussing the development environment configurations and the tools available, we can now describe the server objects that DB2 includes to make development easier. This section introduces what these objects are, how they can be used, and for some objects an example will be provided.

5.3.1 Stored procedures

A stored procedure serves as an extension to clients that runs on the database server. They can be invoked from a client application or another routine with a CALL statement. Stored procedures and their calling programs exchange data using parameters defined in the CREATE PROCEDURE statement. Stored procedures can also return result sets to their callers.

The *create procedure* statement defines a procedure within an application server. Two basic types of stored procedures can be created in DB2:

- External procedures, which are written in a programming language (such as Java or C)
- SQL procedures, which are written in SQL PL (IBM’s implementation of the ANSI standard SQL Procedural Language)

The benefits of stored procedures include:

- A stored procedure helps to reduce unnecessary data transfer over the network between client and server. This is achieved by enabling multiple SQL statements to be issued by a single invocation from the caller, therefore minimizing data transfer between the client and the database server. The more SQL statements you include in a stored procedure, the lower the data transfer costs for each individual statement, as compared to issuing the same statements from the client.
- Isolating database logic from application logic.

- They can return multiple result sets.
- If invoked from an application, they behave as part of the application.
- The ability to use conditional logic such as `if`, `else`, `for`, `which`.
- The ability to handle exceptions within the stored procedures and return customized error codes back to applications.

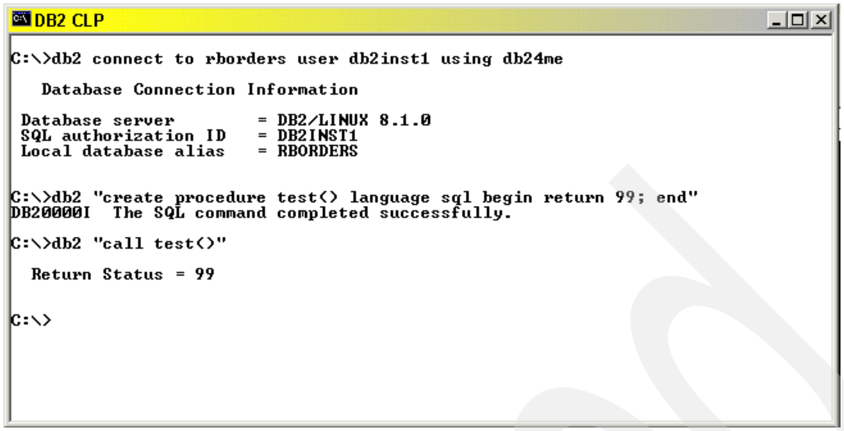
We have used the SQL PL procedures in our sample application and the stored procedure example in this chapter. DB2 provides examples of using external stored procedures in the `sqllib/samples` directory.

> **Note:** Stored procedures are also referred to as routines, a term that also encompasses User Defined Functions and methods. These objects are grouped together as routines, because they all consist of application logic that resides on the database server and can be invoked from a client or other routines.

Configuring the stored procedure environment

When developing a stored procedure using SQL, DB2 transparently generates this into C code and therefore a C compiler is required on the system. The default compile command is included in the Application Development Client. This compile command is based on g++ for Linux, and Microsoft Visual C++ Version 6.0 for Windows.

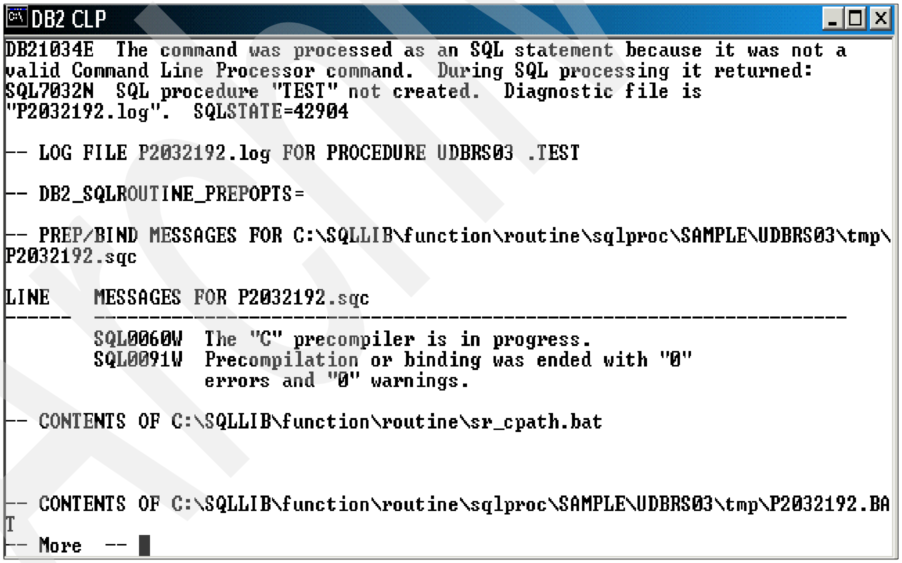
The easiest way to verify if the stored procedure environment has been configured properly is to create a test procedure. Figure 5-6 shows steps to create and call a simple stored procedure which returns a return code of 99. Because the environment has been setup properly in our Lab, the stored procedure *test* is called successfully.



```
C:\>db2 connect to rborders user db2inst1 using db24me

   Database Connection Information

 Database server        = DB2/LINUX 8.1.0
 SQL authorization ID   = DB2INST1
 Local database alias   = RBORDERS


C:\>db2 "create procedure test() language sql begin return 99; end"
DB20000I  The SQL command completed successfully.

C:\>db2 "call test()"

  Return Status = 99


C:\>
```

Figure 5-6 Successful stored procedure creation

Now if the same **`create procedure`** command was run on a system without a C compiler or if the DB2 environment is not configured properly, you would see the error message as shown in Figure 5-7.



```
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL7032N  SQL procedure "TEST" not created.  Diagnostic file is
"P2032192.log".  SQLSTATE=42904

-- LOG FILE P2032192.log FOR PROCEDURE UDBRS03 .TEST

-- DB2_SQLROUTINE_PREPOPTS=

-- PREP/BIND MESSAGES FOR C:\SQLLIB\function\routine\sqlproc\SAMPLE\UDBRS03\tmp\
P2032192.sqc

LINE    MESSAGES FOR P2032192.sqc
------  --------------------------------------------------------------------
        SQL0060W  The "C" precompiler is in progress.
        SQL0091W  Precompilation or binding was ended with "0"
                  errors and "0" warnings.

-- CONTENTS OF C:\SQLLIB\function\routine\sr_cpath.bat



-- CONTENTS OF C:\SQLLIB\function\routine\sqlproc\SAMPLE\UDBRS03\tmp\P2032192.BA
T
-- More  --
```

Figure 5-7 Stored procedure environment requires configuration

If your test result is Figure 5-7 rather than Figure 5-6, you will need to configure DB2 to use the C compiler, this process can be split into the following steps:

1. Install a supported C compiler.

The supported compilers are:

- For Windows: The supported C compiler is Microsoft Visual C++ Version 6.0. The Microsoft Visual C++ Version 5.0 is not supported for the development of SQL stored procedures on DB2 Version 8. The default compile command is based on Microsoft Visual C++ Version 6.0.
- For Linux: The supported compilers are g++ and gcc. The default compile command is based on g++.

2. Configure DB2 to use the C compiler

   Use the **db2set** command to set DB2 environment variables:

   - For Windows:

     From a DB2 Command Window, enter the following command:

     ```
     db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Program Files\Microsoft
     Visual Studio\vc98\bin\vcvars32.bat"
     ```

     > **Note:** You may need to change the path of the compiler if it is not installed in the default installation path.

   - For Linux:

     From a DB2 Command Window or DB2 shell, enter the following command:

     ```
     db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -fpic \
     -I$HOME/sqllib/include SQLROUTINE_FILENAME.c \ -shared -o
     SQLROUTINE_FILENAME -L$HOME/sqllib/lib -ldb2
     ```

More details about this topic are covered in:

- Chapter 2 of *Application Development Guide: Building and Running Applications*

These two redbooks also have information for platform-specific details:

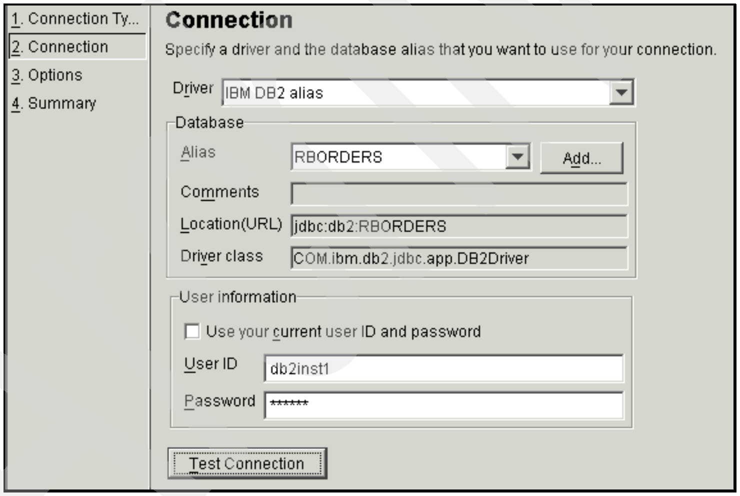
- *Up and Running with DB2 for Linux*, SG24-6899
- *DB2 UDB Exploitation of the Windows Environment*, SG24-6893

Scenario - Build and test a stored procedure to insert data

In our application we have already created a stored procedure that can be used to insert the data into the part table. Here we use the Development Center to create and test another stored procedure to insert the details for a new customer.

1. Start the DB2 Development Center.
   - In a Windows system: **Start —> Programs —> IBM DB2 —> Development Tools —> Development Cente**r, or type `db2dc` at a command prompt.
   - In a Linux system: Log in as the DB2 instance owner *db2inst1,* and type `db2dc` in a graphical shell.
2. Click the **Do not show this again** box for the Launchpad.
3. In the Development Center Launchpad window, click **1. Create Project**.
4. In the Open Project window in the New tab, enter a Project name of **Sample Application** and keep the default Project path. Click **OK**.
5. Click **2. Add Connection**. The Add Database Connection Wizard is opened.
6. Keep the default of **Online** Connection Type and click **Next**.
7. For the Connection you will see a window similar to Figure 5-8. If you are locally connected as the DB2 instance owner, then click to **Use your current user ID and password**.

   Otherwise, enter your DB2 User ID and password.



Connection

Specify a driver and the database alias that you want to use for your connection.

Driver: IBM DB2 alias

Database

Alias: RBORDERS

Comments:

Location(URL): jdbc:db2:RBORDERS

Driver class: COM.ibm.db2.jdbc.app.DB2Driver

User information

☐ Use your current user ID and password

User ID: db2inst1

Password: ******

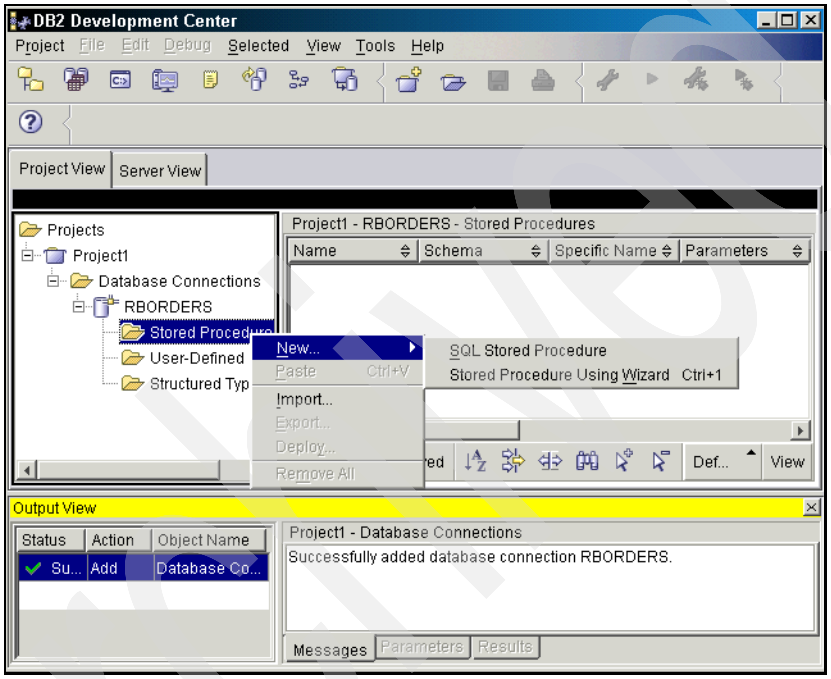
Figure 5-8 Configuring a connection to the DB2 database

8. Click **Test Connection**. Click **OK** if the window returns a success message; if it doesn't, check that you have entered a correct ID and password.
9. Click **Next** to move to the Options window.
10. Click **Next** to move to the Summary window.
11. Click **Finish** to exit the Wizard.

12. Close the Development Center Launchpad window by clicking the **x** at the top right of the window. We show you how to create the objects manually.

Tip: Another option is to click the **Server View** tab and open an existing procedure, edit it, and then save it as a new procedure.

13. You should now be at the main Development Center window (Figure 5-9).



Successfully added database connection RBORDERS.

Figure 5-9 DB2 Development Center

14. Right-click **Stored Procedures** and choose **New —> SQL Stored Procedure**. You won't use the wizard, because we will provide you with the code.

15. A new window is displayed with the CREATE PROCEDURE skeleton code. Select and remove all the text within the window and copy the following text (Example 5-1).

Example 5-1 DB2 stored procedure

```
create procedure db2inst1.insert_customer(v_NAME VARCHAR (25),v_ADDRESS
VARCHAR(40),v_NATIONKEY INTEGER,v_PHONE CHAR(15),v_ACCTBAL
DECIMAL(15,2),v_MKTSEGMENT CHAR(10),v_COMMENT VARCHAR(117))
language sql
```

```
begin

INSERT into db2inst1.customer( C_NAME, C_ADDRESS, C_NATIONKEY, C_PHONE,
C_ACCTBAL ,C_MKTSEGMENT ,C_COMMENT )
values(v_NAME, v_ADDRESS, v_NATIONKEY, v_PHONE, v_ACCTBAL ,v_MKTSEGMENT
,v_COMMENT );
COMMIT;
end
```

16..Click the **Check** icon (green tick) on the tool bar to check the code is correct.

17.Click **File —> Save Object**.

18.You can now close the `DB2INST1.insert_customer` window and return to the to the DB2 Development Center window.

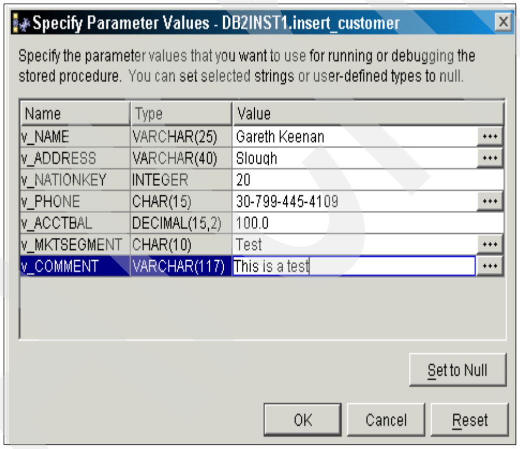
19.Right-click `DB2INST1.insert_customer` and then click **Build**.

20.The bottom right window of the DB2 Development Center window should signify that the build has been successful.

You can now test the procedure within the Development Center.

21.Right-click `DB2INST1.insert_customer` and select **Run**.

22.The Specify Parameter Values window is displayed. Add in the values as displayed in Figure 5-10 and click **OK**.



Specify the parameter values that you want to use for running or debugging the stored procedure. You can set selected strings or user-defined types to null.

Name	Type	Value
v_NAME	VARCHAR(25)	Gareth Keenan
v_ADDRESS	VARCHAR(40)	Slough
v_NATIONKEY	INTEGER	20
v_PHONE	CHAR(15)	30-799-445-4109
v_ACCTBAL	DECIMAL(15,2)	100.0
v_MKTSEGMENT	CHAR(10)	Test
v_COMMENT	VARCHAR(117)	This is a test

Figure 5-10 Data input to a stored procedure in the DB2 Development Center

23.The bottom right of the window should show the following messages:

```
DB2INST1.insert_customer - Run started.
DB2INST1.insert_customer - Calling the stored procedure.
DB2INST1.insert_customer - Run completed.
```

Now check that this data has been successfully added to the database by using the DB2 Command Center.

24. Select **Tools —> Command Center**.

25. Once the Command Center opens, ensure you are in the Interactive tab and enter the following statement to connect to the database. Click the **Execute** icon (gears) at the top left of the window to run the statement:

```
connect to rborders
```

26. Then select and remove the connect statement and replace with the following statement, followed again by clicking the **Execute** icon at the top left:

```
select * from customer where c_name = 'Gareth Keenan'
```

27. The Command Center should have moved to the Query Results tab to display the newly entered row.

28. Close the Development Center using **Project —> Exit** and choose to save the changes to the project.

We have now shown you how to build and test a stored procedure, which could be used from the PHP Web Order application to insert rows into the customer table within DB2.

5.3.2 Triggers

A trigger is a defined set of SQL statements stored as a DB2 object in a DB2 database. This set of SQL statements will be executed when a certain event occurs against a DB2 table. There are three types of events that may invoke triggers; these are: Insert, Update or Delete to a given DB2 table. Triggers may be set up to execute *before*, *after* or *instead of* an insert, update or delete event.

- **No cascade before**: The defined action will be executed before the triggering action is performed. It also specifies that the triggered action of the trigger will not cause other triggers to be activated.
- **After**: All triggered actions will be applied after the triggering action is done.
- **Instead of**: The original action will be replaced by the action defined in the trigger. This is a new addition to DB2 Version 8, which allows these triggers to be used to perform updates, deletes, inserts, and selects transparently against views, even when the view may be too complex to support update operations natively.

Triggers can also be called to fire *for each statement* or *for each row*. If you specify *for each row* the triggered action is to be applied for each row of the table or view that is affected by the triggering SQL. If on the other hand you specify *for each statement* the triggered action is to be applied once for the whole statement.

Scenario - Build and test a trigger to populate a history table

In our sample application we have already created a trigger that updates a history table when we change the status of an order in the orders table. Here we create an additional update trigger that will log any changes to the name of a customer.

Example 5-2 below shows an example of creating the history table and the trigger to populate it.

Example 5-2 Creating a history table and a trigger to populate it

```
connect to rborders

create table db2inst1.custhist (
   custkey integer,
   oldname varchar(25),
   newname varchar(25),
   changed timestamp) @

create trigger db2inst1.custhist
after update on db2inst1.customer
   referencing new as new old as old
   for each row mode db2sql
      begin atomic
      IF new.c_name != old.c_name THEN
      insert into db2inst1.custhist values (old.c_custkey, old.c_name,
new.c_name,current timestamp);
      END IF;
   end@
```

In Example 5-3, we use the @ symbol to terminate the statements rather than the default semi-colon (;). This is because triggers (along with stored procedures and functions) can have multiple lines inside them and these lines are terminated with a semi-colon.

Therefore, when we use routines like triggers we have to use a different statement terminator for DB2 commands.

Here are two ways to execute the sample code to create table and triggers:

- From a DB2 Command Window/shell, enter the command:

  `db2 -td@ -vf rborders.sql`

  This indicates that the @ symbol is being used to terminate the commands rather than the default semi-colon (;).

- From the DB2 Command Center:

  Select **Tools —> Settings** and in the General tab change the **`Use statement termination character`** to **`@`**

  Run the above example from within the Command Center in the Script tab. This will allow you to run multiple commands, as you would in a script.

Now if we update the names of any customers, we fire the trigger and populate the CUSTHIST table. Example 5-3 shows two updates against the CUSTOMER table and against the CUSTHIST table.

Example 5-3 Testing the operation of a trigger

```
update db2inst1.customer set C_NAME='Simon Fisher' where C_NAME = 'Gareth 
Keenan'@
update db2inst1.customer set C_COMMENT='Should not fire trigger' where C_NAME = 
'Simon Fisher'@
select * from db2inst1.custhist@
```

The result of two the updates and the trigger action that inserted a record in the CUSTHIST table is shown in Example 5-4.

Example 5-4 Output of trigger testing

```
update db2inst1.customer set C_NAME='Simon Fisher' where C_NAME = 'Gareth 
Keenan'
DB20000I  The SQL command completed successfully.

update db2inst1.customer set C_COMMENT='Should not fire trigger' where C_NAME = 
'Simon Fisher'
DB20000I  The SQL command completed successfully.

select * from db2inst1.custhist
CUSTKEY   OLDNAME         NEWNAME         CHANGED
--------- --------------- ---------------- --------------------------
763       Gareth Keenan   Simon Fisher    2003-02-20-11.37.37.979682

1 record(s) selected.
```

As you can see only one record has been added to the history table, because although there were two updates to the customer table, the name of the customer was only changed once.

> **Note:** The *create trigger* statement is explained in more detail in Chapter 5 “Statements” in the *SQL Reference*.

5.3.3 User Defined Functions (UDFs)

Built-in functions are provided by DB2. They return a single result value, and are identified as part of the SYSIBM schema. Built-in functions include column functions (such as AVG), operator functions (such as “+”), casting functions (such as DECIMAL), and others (such as SUBSTR).

Recent additions to DB2 built-in functions include:

- Additional XML functions that can allow you to return XML as the result of a query against DB2 tables. These functions will be explained further in Chapter 7, “Additional DB2 features” on page 309.
- New functions that allow DB2 to interact with data from a WebSphere MQ queue. Again this will be covered in more detail in Chapter 7, “Additional DB2 features” on page 309.
- The new *to_date()* and *to_char()* functions that transform between dates and varchars.

In addition to the built-in functions, DB2 also provides capability for users to define their own functions to be used within the database. These User Defined Functions (UDFs) can be written in languages such as C or Java, as well as DB2 SQL.

In this section we concentrate on using SQL based UDFs, but additional detail about the *external* non SQL UDFs can be found in the *Application Development Guide: Programming Server Applications*.

Types of User Defined Functions

There are different types of User Defined Functions available.

- **Scalar**: Scalar UDFs enable you to extend and customize SQL. They can be invoked in the same manner as built-in DB2 functions (for example, LENGTH). That is, they can be referenced in SQL statements wherever an expression is valid. Scalar UDFs can accept zero or more inputs and return a single value.
- **Row**: This type can be specified in the *return* clause of the CREATE FUNCTION statement. It specifies that the output of the function is a single row. The column-list following that is returned must include at least two columns. A row function can only be used as a transformation function for a structured type. An example of its use would be to return the base types from an object relational structured type (such as address), the function would have one parameter and return multiple base types (such as house number, street, city).
- **Column**: The column UDFs, also known as aggregating functions, receive a set of like values (a column of data) and return a single answer. A

user-defined column function can only be created if it is sourced upon a built-in column function. For example, if a distinct type SHOESIZE exists that is defined with base type INTEGER, you could define a UDF, AVG(SHOESIZE), as a column function sourced on the existing built-in column function, AVG(INTEGER).

- **Table**: Like scalar UDFs, table UDFs enable you to extend and customize SQL, but for the purpose of generating tables. Table UDFs can only be invoked in the FROM clause of an SQL statement. Table UDFs accept zero or more typed values as input arguments and return a table.

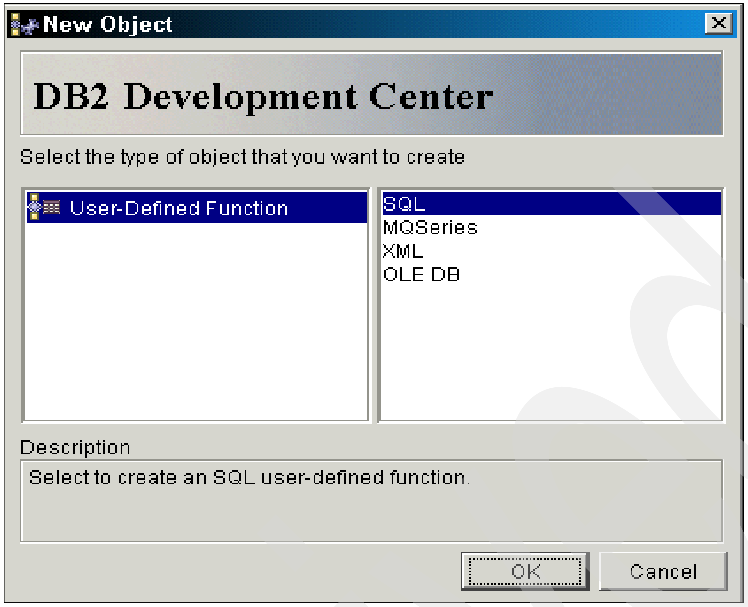
Example - Build and test a UDF

For this example, we have been asked to create a simple scalar function to alter the C_PHONE column in the CUSTOMER table. Although the telephone numbers include the international dialling codes, the users want the 011 code necessary to dial out from the USA included in the phone number. Furthermore, they don’t want to have the 011 prefix added to numbers within the US.

These procedures show you how to create the UDF:

1. Open the DB2 Development Center.
2. Select the existing **Sample Application** project and click **OK**.
3. Right-click **User-Defined Functions** and select **New —> SQL User-Defined Function**.

   Again, the code will be provided so we will not use the wizard. If we had chosen the wizard, we would have seen the following window (Figure 5-11).



Select the type of object that you want to create

Description

Select to create an SQL user-defined function.

Figure 5-11 DB2 Development Center UDF Wizard

This would have allowed us to use the wizard to not only create SQL UDFs, but also UDFs that extract data from the WebSphere MQ Series product, UDFs that extract data from an XML document into DB2 tables, and finally UDFs that can extract data from an OLEDB data source.

4. A new window will be displayed with the skeleton code for a UDF.
5. Select the contents of the window, and replace them with the contents in Example 5-5.

Example 5-5 Sample UDF code

```
CREATE FUNCTION DB2INST1.INTDIAL( v_phone VARCHAR(20) )
RETURNS VARCHAR(20)
DETERMINISTIC NO EXTERNAL ACTION READS SQL DATA

BEGIN ATOMIC
DECLARE v_nation integer default 0;
DECLARE v_str varchar(20) DEFAULT '';

-- First Check what nation the phone number is from
-- We could get this using a substring on the phone number
-- Instead lets use the input phone number to query the customer table and --
return the nationkey
```

```
SET v_nation = (select c_nationkey from db2inst1.customer where c_phone =
v_phone);

IF v_nation = (select  n_nationkey from db2inst1.nation where n_name = 'UNITED
STATES')
THEN RETURN v_phone; -- We don’t needing an international dialling code
END IF;

set v_str = '011-' CONCAT v_phone;
RETURN v_str;
END
```

The elements of the sample User Defined Function are described in Table 5-1.

Table 5-1 Description of sample User Defined Function (UDF)

Language Element	Description
`v_phone VARCHAR(20)`	The UDF reads in the current telephone number.
`RETURNS VARCHAR(20)`	The UDF returns a modified telephone number.
`DETERMINISTIC`	Every time we invoke this function with the same input it will return the same value.
`NO EXTERNAL ACTION`	DB2 can use certain optimizations that assume functions have no external impacts.
`BEGIN ATOMIC`	We want all of the actions contained within the BEGIN..END to either all complete or all fail.
`DECLARE`	Declaration of variables for use within the function.
`SET v_nation`	Set the value of the variable.

`IF v_nation = (select  n_nationkey`
`from nation where n_name = 'UNITED`
`STATES')`
`THEN RETURN v_phone;`
`END IF;`
	If the value of the variable is equal to the nation key for the USA then return the original telephone number back. This means that when v_nation = 24 the function will effectively end here.

`'011-' CONCAT v_phone`	Concatenate 011 with the qualifying telephone numbers.
`RETURN v_str`	Return this value.

Language Element	Description
`-- We don’t needing an international dialling code`	-- Allows you to put comments in.

6. Click the **Check** icon, the green tick, to check that the SQL in the function is correct. There are also debug functions on the second row of the tool bar that could be used in a more complex scenario (Figure 5-12).

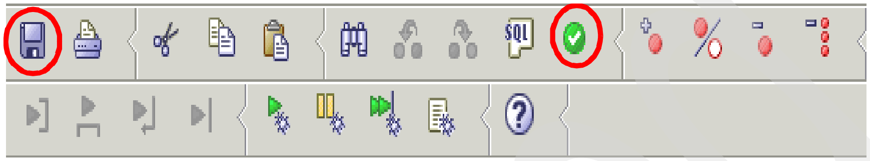


Figure 5-12 Development Center toolbar

7. Click the **Save** icon (disc symbol) to save the function, and then close the function window.
8. Back in the Development Center window, right-click the function and select **Build**.
9. If prompted, enter your DB2 server ID and password.
10. Success should be reported in the bottom right window of the Development Center.
11. Before you run the function, you want to choose some appropriate telephone numbers to test. You should choose at least one that is a US number and one that will have the prefix added.
12. Launch the Command Center.
    a. Connect to the **RBORDERS** database.
    b. Issue the following two statements:

```
select c_phone from db2inst1.customer where c_nationkey <> (select
n_nationkey from db2inst1.nation where n_name = 'UNITED STATES')
fetch first 1 row only

select c_phone from db2inst1.customer where c_nationkey <> (select
n_nationkey from db2inst1.nation where n_name = 'UNITED STATES')
fetch first 1 row only
```

    c. Note the two phone numbers you returned. In our case:

```
34-403-631-3505 (non US number) and
24-989-741-2988 (US number)
```

13. In the Development Center, right-click the function and choose **Run**.

14. Enter the first telephone number you retrieved. This should be returned unchanged as it is a USA number.

15. Run the function again and enter the second number you retrieved. This should be returned with a prefix of 011.

16. Close Development Center saving your changes.

We have now shown you how to create a very simple scalar function that takes in a telephone number and depending on the country (or region) adds a dialing prefix.

> **Note:** The *create function* statement is described in more detail in Chapter 5 “Statements” in the *SQL Reference*.

5.3.4 SQL development

In this section we cover the additional development capability that is available through the use of DB2 SQL. Here we cover the use of constraints, user defined types, materialized query tables, views, and the us of DB2 generated columns.

Constraints

A constraint is a DB2 rule the user can specific in table design to ensure the data integrity. The database manager enforces the constraints. DB2 contains these four types of constraints:

- **Unique constraints**: These are rules that forbid duplicate values in one or more columns within a table. Unique and primary keys are the supported unique constraints. In our example, a unique constraint can be defined on the N_NATIONKEY column in the NATION table to ensure that the same country (or region) identifier is not given to two countries (or regions). The syntax for this would be:

  ```
  alter table db2inst1.nation add constraint db2inst1.nkey primary key
  (n_nationkey)
  ```

- **Referential Integrity (RI) constraints**: This is a logical rule about values in one or more columns in one or more tables. In essence it links the contents of the primary key of one table (parent table) with a column in another table (child table). This linkage ensures that when actions happen on the parent table they are cascaded to the other table(s) that rely on it.

  If we wanted to ensure that the CUSTOMER table only contains customers from countries (or regions) defined in the NATION table, we could define a foreign key constraint referencing the above primary key constraint:

```
alter table db2inst1.customer add constraint db2inst1.c_nation
foreign key (c_nationkey) references db2inst1.nation on delete
cascade
```

The `on delete cascade` is to remove any references to customers in non-existent countries (or regions).

- **Check constraints**: Sets restrictions on data added to a specific table. We use a Check constraints example in our discussion on views, when we use a date based check constraint to limit which months are loaded into which table.
- **Informational Constraints**: These constraints are new in Version 8 of DB2. When you create a referential constraint or a check constraint, you can specify that they be `enforced` or `not enforced`. When they are not enforced, they are not enforced by DB2 during an insert, update or delete. Using `not enforced` can provide quicker results, because there is less checking to be done, but can also result in erroneous results unless the checking of constraints is handled by some other source.

User Defined Types (UDTs)

A user-defined type (UDT) is a data type that you derive from existing data types. UDTs enable you to extend the built-in types already available and create your own customized data types.

There are two classifications of user-defined types:

- **Distinct type**: These share a common representation with built-in data types. The classic example is to create a UDT of pounds and a UDT of dollars. Although both of these UDTs are based on the DB2 decimal built-in, you cannot compare the values of a pounds and dollars without writing a UDF to cast between the two. Much the same way as you cannot directly compare two built-in numerics without using a function.
- **Structured type**: This type enables the representation of a sequence of named attributes that each have a type. One structured type can be a subtype of another structured type defining a type hierarchy. This is an example of the object-relational features of DB2 which also includes methods that return values from the structured types and constructors that initialize these objects. Method outlines enable you to define behavior for these structured types.

  For details on the use of these and other features, we recommend the following article, which is Part 1 of a two part article that covers the object relational capabilities of DB2:

  http://www7b.software.ibm.com/dmdd/library/techarticle/zeidenstein/0108zeidenstein.html

Materialized Query Tables (MQTs)

These are typically added to a DB2 system to increase the performance of queries in certain circumstances. These work in addition to indexes and were known as *summary tables* in Version 7.2 and earlier releases. In many cases for the right workload they can be more efficient than indexes.

A materialized query table (MQT) is a table that is typically defined based on the results of a query. This allows the DB2 optimizer to consider using the contents of the MQT instead of running the query against the base table(s). Changes to the base tables can be immediately propagated to the MQT or the changes can be deferred until the **`refresh table`** command is run.

Enhancements in Version 8 of DB2 allow the existing contents of an MQT to remain online even when the underlying table is being appended. Once the data load has finished at the base table level, the MQT data can be incrementally refreshed, rather than fully refreshed. Version 8 also allows you to maintain your MQTs rather than DB2, this is particularly useful if you plan to have a summary table generated by another means but which you still want the optimizer to consider.

Views

A view provides a customized way of looking at the data in one or more tables. They are typically created as a SELECt statement that is run whenever the view is referenced in an SQL statement. A view has columns and rows just like tables. You can use views to control access to sensitive data, because multiple views allow multiple users to see different presentations of the same data.

Views can also be used within DB2 to allow operations over multiple tables at one time. Example 5-6 shows a UNION ALL view for our sample application, which would split the ORDERS table on the O_ORDERDATE column into quarters.

Example 5-6 Create a Union All View

```
create table db2inst1.ordq1 as (select * from db2inst1.orders) definition only;
alter table db2inst1.ordq1 add constraint db2inst1.c1 check (month(o_orderdate)
between 1 and 3);
create table db2inst1.ordq2 as (select * from db2inst1.orders) definition only;
alter table db2inst1.ordq2 add constraint db2inst1.c2 check (month(o_orderdate)
between 4 and 6);
create table db2inst1.ordq3 as (select * from db2inst1.orders) definition only;
alter table db2inst1.ordq3 add constraint db2inst1.c3 check (month(o_orderdate)
between 7 and 9);
create table db2inst1.ordq4 as (select * from db2inst1.orders) definition only;
alter table db2inst1.ordq4 add constraint db2inst1.c4 check (month(o_orderdate)
between 10 and 12);
```

```
create view db2inst1.orders_v as select * from db2inst1.ordq1 UNION ALL select
* from db2inst1.ordq2 UNION ALL select * from db2inst1.ordq3 UNION ALL select *
from db2inst1.ordq4;

insert into db2inst1.orders_v(o_custkey, o_orderstatus, o_totalprice,
o_orderdate, o_orderpriority, o_clerk, o_shippriority, o_comment) values
(10001,1873,'O',19.1,'05/23/1975','High','Sandy Clerk',77,'Probably an
entertainer third');

select * from db2inst1.orders_v where o_custkey = 1873
select * from db2inst1.ordq1 where o_custkey = 1873
select * from db2inst1.ordq2 where o_custkey = 1873
```

Example 5-6 creates four tables with constraints to ensure that they only contain the correct values. A view is then created over the four tables and data inserted into the view. We can then select data from the view or the individual tables. In our example above the first and third SQL queries would return the row we just inserted. The **orders_v** view will contains data for all four quarters whereas the **ordq2** table has the second quarter data which includes May.

This splitting of the data across multiple tables gives the facility to quickly roll off old tables. This form of partitioning can be used independently or in conjunction with the Data Partitioning Feature (DPF) which is discussed in “Inter-partition parallelism” on page 354.

As demonstrated above, the additional functionality of inserting through the union all view was added in DB2 Version 8.

Another feature added to views in DB2 Version 8 are the inclusion of an *instead of* trigger, which we mentioned in 5.3.2, “Triggers” on page 165. The *DB2 SQL Reference* manual covers the additional examples where views can be inserted, updated, or deleted without the use of an *instead of* trigger.

Other SQL features

As you might expect there are many other SQL features within DB2 and here we discuss how you can ensure the uniqueness of rows and storing application temporary data.

- **Ensuring row uniqueness**: As well as enforcing the uniqueness of a record by using a primary key, DB2 can also automatically populate columns with values which can greatly simplify the process of ensuring that the key field of a column differs from previous records. Here we discuss three ways of achieving this:

- **Generated Identity Columns**: These are specified as part of the CREATE TABLE statement. It allows you to specify that the particular column is not directly inserted into but instead increments its own value by a predefined amount. We used this feature extensively in our application, including in table REGION:

  ```
  R_REGIONKEY INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY ( START
  WITH 0 , INCREMENT BY +1 , CACHE 20 )
  ```

  This identity column will then start at the value of 0 and increment by 1 each time a row is inserted. The identity column will furthermore allocate values 20 at a time.

  When you insert into a row that contains an identity column, you need to explicitly state the columns that you are inserting into. So with our region table the insert command would be:

  ```
  insert into db2inst1.region(r_name, r_comment) values
  ('R8','Test')
  ```

  The r_regionkey is generated automatically.

  There is a built-in function to retrieve the current value of the identity column, *identity_val_local()*. The command to use the function is:

  ```
  values(identity_val_local())
  ```

  This can also be used from within a stored procedure.

- **Sequences**: The **`create sequence`** statement creates a repository of sequential values at the DB2 server. This allows the allocation of unique sequential values across the entire database, identity columns are only unique to a table. A typical invocation of the command would be:

  ```
  create sequence db2inst1.rbord_seq as integer start with 1
  increment by 1 maxvalue 9999 no cycle cache 50
  ```

  Here we have created an integer based sequence that will increment by 1 until it reaches 9999. Upon which point no further values will be generated.

  Assuming we were using the above sequence rather than the identity columns for the REGION table, we would input the values to the database as follows:

  ```
  insert into db2inst1.region (r_regionkey,r_name, r_comment)
  VALUES (NEXTVAL rbord_seq,'R8','Test');
  ```

  We could then use the **`prevval`** function to input the same sequence value to another table if we wanted to ensure that they had the same value.

- **Generating unique values**: DB2 provides the **`generate_unique()`** function which can be used to provide unique values in a table. Each successive value will be greater than the previous value, providing a sequence that can be used within a table. The value includes the partition number where the function executed so that a table partitioned across multiple partitions also has unique values in some sequence. The sequence is based on the time the function was executed.

- **Storing Temporary Data**: There is often a requirement to store temporary data within a transaction or an application. This can be achieved through the use of regular tables but that can mean an extra overhead in terms of cleaning up the table ready for its next use.

  DB2 provides the **`declare global temporary table`** statement which defines a temporary table for the current session.

  > **Note:** A session exists between when a connection is made to DB2 and when it disconnects.

  The declared temporary table description does not appear in the system catalog and cannot be shared with other sessions. Each session that defines a declared global temporary table of the same name has its own unique description of the temporary table. When the session terminates, the rows of the table are deleted, and the description of the temporary table is dropped.

5.4 Application interfaces to DB2

You can use several different programming interfaces to manage or access DB2 objects and databases. These are summarized in Figure 5-13.

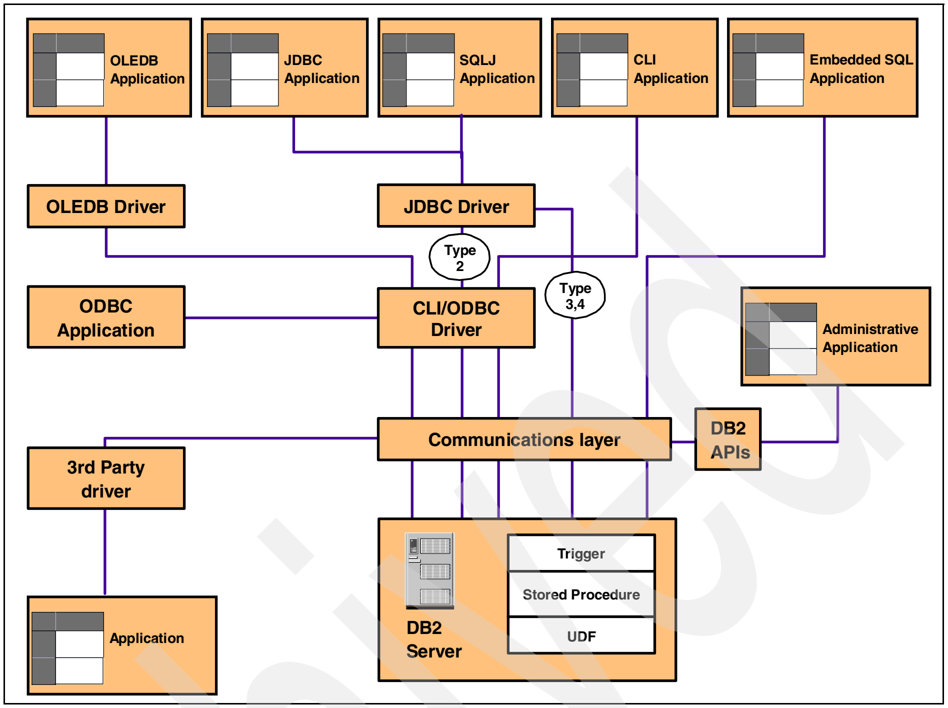


Figure 5-13 How applications connect to DB

There are many different types of applications that can be created against DB2, whether they are ODBC applications ported from other databases, embedded SQL applications written in COBOL, or OLE DB applications that want to use DB2 as a data provider. Therefore, DB2 provides the following interfaces to provide access to objects and database.

5.4.1 DB2 APIs

DB2 provides numerous APIs to allow your applications to perform some database administration tasks, such as creating, activating, backing up, or restoring a database. You can also use APIs to provide facilities to import and export data and modify database manager and database configuration parameter value. In general, applications can use APIs to access database manager facilities that are not available using SQL statements.

These API calls can be included within your embedded SQL and DB2 CLI applications. These APIs allow you to program the same administrative functions, available in Control Center, into your applications.

The DB2 APIs are available in C, C++ or COBOL. If these programs only contain API calls, then they only are required to be compiled. There is no requirement to precompile the program. However, if the programs contain embedded SQL calls, they will need to be precompiled. The precompile process is discussed in 5.4.2, “Embedding SQL statements in your applications” on page 180

Examples of API programs can be found in the DB2 home directory:

- `sqllib/samples/c`
- `sqllib/sample/cpp`
- `sqllib/sample/cobol`
- `sqllib/sample/cobol_mf`

For additional information refer to *DB2 Administrative API Reference*, SC09-4824.

5.4.2 Embedding SQL statements in your applications

An application in which you embed SQL statements is called a host program. The programming language you use to create a host program is called a host language. The program and language are defined this way because they host or accommodate SQL statements. These statements can either be static or dynamic.

Static SQL statements

With static SQL statements, you know before compile time the SQL statement type and the table and column names. The only unknowns are the specific data values the statement is searching for or updating. You can represent those values in host language variables.

As well as compiling and linking your program, you must first precompile and bind it. Basically precompiling converts embedded SQL statements into DB2 run-time API calls that a host compiler can process, and as output creates a bind file. When the precompiler processes a source file, it specifically looks for SQL statements and avoids the non-SQL host language. It can find SQL statements, because they are surrounded by special delimiter.

The bind file contains information on the SQL statements in the application program. The *bind* command creates a package in the database. Optionally, the precompiler can perform the bind step at precompile time. Binding is the process of creating a package from a bind file and storing it in a database. If your application accesses more than one database, you must create a package for each database. This package then contains the SQL operation and the access plan that DB2 will use to perform the operation.

Dynamic SQL

Dynamic SQL statements are those that your application builds and executes at run time. In our sample application where we allow users to dynamically create a query is a good example of dynamic SQL. The application builds the SQL statement while it's running, and then submits the statement for processing.

Examples of embedded SQL (both static and dynamic) can be found in:

- `sqllib/samples/c`
- `sqllib/sample/cpp`
- `sqllib/sample/cobol`
- `sqllib/sample/cobol_mf`

5.4.3 DB2 Call Level Interface (DB2 CLI)

DB2 CLI is a programming interface that your C and C++ applications can use to access DB2 databases. DB2 CLI is based on the Microsoft Open Database Connectivity (ODBC) specification, and the ISO CLI standard. Since DB2 CLI is based on industry standards, application programmers who are already familiar with these database interfaces may benefit from a shorter learning curve.

The DB2 CLI load library can be loaded as an ODBC driver by an ODBC driver manager. Applications written directly to DB2 CLI will link directly to the DB2 CLI load library. DB2 CLI includes support for many ODBC and ISO SQL/CLI functions, as well as DB2 specific functions.

When you use DB2 CLI, your application passes dynamic SQL statements as function arguments to the database manager for processing. As such, DB2 CLI is an alternative to embedded dynamic SQL.

You do not need to precompile or bind DB2 CLI applications, because they use common access packages provided with DB2. You simply compile and link your application. Before your DB2 CLI or ODBC applications can access DB2 databases, the DB2 CLI bind files that come with the DB2 AD Client must be bound to each DB2 database that will be accessed. This occurs automatically with the execution of the first statement.

Typically when you build an ODBC application you need an ODBC driver manager, provided by Microsoft on Windows and third-party vendors on other platforms, such as `http://www.unixodbc.org/` for Linux. However in environments without an ODBC driver manager, DB2 CLI is a self sufficient driver which supports a subset of the functions provided by the ODBC driver. This was the case for our application when we developed it on Linux as we didn't have an ODBC driver manager.

> **Tip:** If you want to prototype CLI calls before placing them in a program, you can use the *db2cli.exe* (Windows) or *db2cli* (Linux) file in the `sqllib/samples/cli` directory. There is also a document, *INTCLI.DOC*, which advises you about how to use the utility.

Examples of C programs making CLI calls can be found in:

- `sqllib/samples/cli`

For additional information refer to *Call Level Interface Guide and Reference, Volume 1 and Volume 2*, SC09-4849 and SC09-4850.

5.4.4 Java Database Connectivity application (JDBC)

DB2's Java support includes JDBC, a vendor-neutral dynamic SQL interface that provides data access to your application through standardized Java methods. Similar to DB2 CLI, you do not have to precompile or bind a JDBC program. As a vendor-neutral standard, JDBC applications offer increased portability. Applications written using JDBC use dynamic SQL although the same application can contain both JDBC and SQLj calls.

JDBC can be especially useful for accessing DB2 databases across the Internet. Using the Java programming language, you can develop JDBC applets and applications that access and manipulate data in remote DB2 databases using a network connection.

The JDBC API, which is similar to the CLI/ODBC API, provides a standard way to access databases from Java code. Your Java code passes SQL statements to the DB2 JDBC driver. The driver then handles the JDBC API calls from your client Java code.

Java's portability enables you to deliver DB2 access to clients on multiple platforms, requiring only a Java-enabled Web browser, or a Java runtime environment.

Available JDBC drivers

DB2 Version 8 offers different ways of creating Java applications, either using a Type 2, Type 3, or Type 4 JDBC driver.

- **Type 2 driver**: With a Type 2 driver, calls to the JDBC application driver are translated to Java native methods. The Java applications that use this driver must run on a DB2 client, through which JDBC requests flow to the DB2 server. This is typically how DB2 is accessed by WebSphere Application Server.

- **Type 3 driver**: The DB2 JDBC Type 3 driver, also known as the applet or net driver, consists of a JDBC client and a JDBC server. The DB2 JDBC applet driver can be loaded by the Web browser along with the applet, or the applet driver can be used in standalone Java applications. When the applet requests a connection to a DB2 database, the applet driver opens a TCP/IP socket to the DB2 JDBC applet server on the machine where the Web server is running.

  After a connection is set up, the applet driver sends each of the subsequent database access requests from the applet to the JDBC server through the TCP/IP connection. The JDBC server then makes corresponding DB2 calls to perform the task. Upon completion, the JDBC server sends the results back to the JDBC client through the connection.

  The use of the Type 3 driver is being deprecated with DB2 Version 8.

- **Type 4 driver**: You can use the JDBC Type 4 driver, which is new for Version 8, to create both Java applications and applets. To run an applet that is based on the Type 4 driver, you only require a Java enabled browser, which downloads the applet and the JDBC driver *(db2jcc.jar)*. To run a DB2 application with a Type 4 driver, all that is required is the JDBC driver in your class path. No DB2 client is required.

  This Type 4 driver provides the initial implementation of the new JDBC driver architecture known as the IBM DB2 JDBC Universal Driver. The Universal Driver is architected as an abstract JDBC processor that is independent of driver-type connectivity or target platform The initial 1.0 release of the Universal JDBC Driver supports all-Java remote connectivity (Type 4), which is based on an open distributed protocol, known as Distributed Relational Database Architecture (DRDA®) for cross-platform access to DB2.

Examples of JDBC calls can be found in:

- `sqllib/samplesjava/jdbc`

For detailed information on the Java support provided by DB2 Version 8, we strongly recommend the manual *Developing Enterprise Java Applications Using DB2 Version 8*.

5.4.5 Embedded SQL for Java (SQLj)

DB2 Java embedded SQL (SQLj) support is provided by the DB2 AD Client. With DB2 SQLj support, in addition to DB2 JDBC support, you can build and run SQLj applets, applications, and stored procedures. These contain static SQL and use embedded SQL statements that are bound to a DB2 database.

SQLj applications use JDBC as a foundation for tasks, such as connecting to databases and handling SQL errors, but also contain embedded static SQL statements in separate SQLj source files. Unlike the other languages that can contain embedded SQL (COBOL, C, C++) the Java code is not precompiled, instead the SQLj translator converts SQLj clauses into JDBC statements.

As SQLj shares its underlying connection with that of JDBC, applications can connect to DB2 using either Type 2, Type 3 or Type 4 drivers.

Examples of SQLj calls can be found in:

- `sqllib/samplesjava/sqlj`

There are some good articles available that cover the areas of JDBC and SQLj, in detail:

- “Developing Enterprise Java Applications Using DB2 Version 8”. You can find this article at this Web site:

  `http://www7b.software.ibm.com/dmdd/library/techarticle/0209hutchison/0209hutchison.html`

- “Considering SQLJ for Your DB2 V8 Java Applications”. You can find this article at this Web site:

  `http://www7b.boulder.ibm.com/dmdd/library/techarticle/0302tsui/0302tsui.html`

5.4.6 ActiveX Data Objects and Remote Data Objects (Windows only)

DB2 supports ActiveX Data Object (ADO) applications that use the Microsoft OLE DB to ODBC Bridge. ActiveX Data Objects (ADO) allow you to write an application to access and manipulate data in a database server through an OLE DB provider.

When you install the client version of DB2 Version 8.1 for Windows, you can optionally install IBMDADB2, the IBM OLE DB 2.0 compliant provider for DB2. The provider exposes interfaces for OLE consumers who want to access data in a DB2 database. With this driver you don’t need to catalog the DB2 database as an ODBC data source.

Remote Data Objects (RDO) provide an information model for accessing remote data sources through ODBC. RDO offers a set of objects that make it easy to connect to a database, execute queries and stored procedures, manipulate results, and commit changes to the server.

As RDO implements a thin code layer over the ODBC API, it requires an ODBC data source to be created for the DB2 database you are connecting to.

Examples of ADO/RDO calls can be found in:

- `sqllib\samples\VC\ADO`
- `sqllib\samples\VB\ADO`
- `sqllib\samples\VB\RDO`

5.4.7 ADO.NET

DB2 UDB supports Microsoft's ADO.NET programming interface via a native managed provider. You can develop high-performing WinForm, WebForm, and mobile WebForm applications using the ADO.NET API. These applications can be developed to work with data managed by DB2 UDB V8.1 servers running on Windows, UNIX, and Linux servers, as well as DB2 for OS/390 and z/OS database servers.

When used in conjunction with stored procedures and the federated database capabilities of DB2 UDB and DB2 Connect servers, this data access can be extended to include a wide variety of other data sources, including non-DB2 mainframe data (such as VSAM, CICS®, IMS™), Informix® Dynamic Server (IDS), Microsoft SQL Server, Sybase and Oracle databases as well as any data source that has an OLE DB Provider available.

The IBM DB2 .NET Data Provider is native provider written in managed C# code to deliver high-performing, secure access to DB2 data. The DB2 .NET Data Provider is developed and tested in-house by the DB2 Development organization, the people who really know DB2, to deliver the best possible exploitation of DB2 features and functions to ADO.NET programmers.

5.4.8 Other interfaces

Applications can be developed using other interfaces provided by DB2. One such example is the Perl’s use of the DBD::DB2 driver to connect to DB2, which will be covered in 5.5.4, “Perl” on page 187 when we discuss the use of Perl.

5.5 Programming languages

In the previous section we discussed the available programming interfaces to DB2. In this section we look at the programming languages that can be used to develop applications using these interfaces.

5.5.1 Java

Java is an ideal language for writing programs that run on multiple platforms. Unlike languages such as C or C++, Java compiles to code that is specific only to the Java Virtual Machine (JVM) not to the underlying operating system.

Therefore, using Java in conjunction with JDBC or SQLj allows you to write portable database applications.

As well as client applications, Java can also be used to create DB2 objects, such as stored procedures and user defined functions.

To build Java applications you need to install and configure the following components on your machine:

IBM Java Developer Kit:

- **For Linux**: IBM Developer Kit and Runtime Environment for Linux, Java 2 Technology Edition, Version 1.3.1, 32-bit version.

  This product gets installed with DB2 if you use DB2 Setup.
- **For Windows**: IBM Developer Kit and Runtime Environment for Windows, Java 2 Technology Edition, Version 1.3.1, 32-bit version.

  This product gets installed with DB2 if you use DB2 Setup.

Refer to the DB2 Java Web site for more information:

http://www-3.ibm.com/software/data/db2/udb/ad/v8/java/

5.5.2 C++

C++ is a very popular and powerful programming language. It also offers a great deal of flexibility in the DB2 environment allowing you to create API applications, created Embedded SQL applications, write applications that use CLI and also applications that call DB2 through the IBM DB2 OLE DB driver.

As well as client applications, C++ can also be used to create DB2 objects, such as stored procedures, triggers and user defined functions.

Currently, DB2 supports the following C/C++ programming languages and compilers:

- **Linux**: GNU/Linux gcc and g++ versions 2.95.3 and 2.96
- **Windows**: Microsoft Visual C++ Version 6.0, Microsoft Visual C++ .NET, Intel C++ Compiler for 32-bit applications Version 6 or later

The DB2 application development Web site contains additional information about developing applications in C/C++:

http://www.ibm.com/software/data/db2/udb/ad

5.5.3 PHP

PHP is a server-side, cross-platform, scripting language that allows you to embed application logic directly into an HTML file. It can be used to access DB2 from Web-based applications to generate customized dynamic content or for capturing database transactions from a Web browser. It supports DB2 access using the Unified-ODBC access method, in which the user-level PHP communicates with DB2 using ODBC calls through the DB2 CLI layer.

PHP is most commonly used with the Apache Web server, but also works well on other Web servers, such as WebSphere, HTTPD, and AOLServer. The Apache server is a commercial grade, freely available HTTP Web server that was created through a collaborative software development effort between Linux and the Apache Web server project. The PHP Apache module runs as part of the Apache (httpd) process.

This was our language of choice for developing our sample application, and it has already been extensively covered in Chapter 4.

5.5.4 Perl

Perl is a popular programming language that is freely available for many operating systems. It was originally developed for text manipulation, but is now used for a wide range of tasks including system administration, Web development, network programming, and GUI development.

Because Perl is an interpreted language and the Perl DBI Module uses dynamic SQL, Perl is an ideal language for quickly creating and revising prototypes of DB2 applications. The Perl DBI Module uses an interface that is quite similar to the CLI and JDBC interfaces, which makes it easy for you to port your Perl prototypes to CLI and JDBC.

Perl is an example of an application that communicates with DB2 using a third-party driver, in this case the DBD::DB2 driver.

There are four things you need to use Perl with DB2:

1. A DB2 Application Development (AD) Client or a local copy of the server with the AD libraries installed.
2. A copy of Perl.

   Perl is typically provided by the Linux distributions, but if you are a Windows user, we recommend that you download ActivePerl from:

   `http://www.activestate.com`

   DB2 currently supports Perl 5.004_04 or later.

3. The Perl DBI module.

   DBI is an open standard API that provides database access for client applications written in Perl. DB2 requires DBI 0.93 or later.

   For Linux you can download this module from:

   `http://www.cpan.org/modules/by-category/07_Database_Interfaces/DBI`

   For Windows, after installing Active Perl, run the following command:

   run **`ppm install DBI`**

4. The DBD::DB2 driver provided by IBM is downloadable from:

   `http://www.ibm.com/software/data/db2/perl`

   The current driver is DBD-DB2-0.76

   Once downloaded and extracted, the driver is required to be compiled using a C compiler to build the DBD::DB2 driver.

Further details about using Perl with DB2 can be found in:

- Chapter 8 of *Up and Running with DB2 for Linux*, SG24-6899
- The IBM DB2 Perl Database Interface Web site: `http://www-3.ibm.com/software/data/db2/perl/`

5.5.5 COBOL (Windows only)

COBOL can be used to create embedded SQL applications as well as DB2 API applications.

DB2 Version 8.1 for Windows supports the following COBOL Compilers:

- Micro Focus COBOL Version 4.0.20
- Micro Focus COBOL Net Express Version 3.1.0
- IBM VisualAge® COBOL Version 2.0

5.5.6 Visual Basic (Windows only)

With Visual Basic, you can connect to DB2 and create applications using the following common libraries:

- CLI
- ODBC
- OLEDB

DB2 supports Microsoft Visual Basic 6.0 Professional Edition. The use of the DB2 plug-in to this environment was discussed in 5.2.7, “Integration with other tools” on page 156. Also consult Chapter 8 of *DB2 UDB Exploitation of the*

Windows Environment, SG24-6893, for details about using Microsoft Visual Basic 6.0 with DB2.

5.5.7 Visual C++ and Visual J++ (Windows only)

DB2 provides add-in tools that integrate with Visual C++. The DB2 UDB Project Add-In is a collection of management tools and wizards that plug into the Visual C++ component of Visual Studio IDE. The tools and wizards automate and simplify the various tasks involved in developing applications for DB2 using embedded SQL. In addition to the visual C++, DB2 also integrates with Visual J++. This support provides the DB2 application developers an additional choice of programming language.

5.5.8 Object REXX (Windows only)

Object REXX contains features typical of an object-oriented language, such as subclassing, polymorphism, and data encapsulation. It is an extension of the IBM REXX language, which has been expanded to include classes, objects, and methods. REXX is an interpreted language which doesn't use a precompiler, compiler, or linker. Instead, three DB2 APIs are used to create DB2 applications in REXX:

- SQLEXEC which supports the SQL language.
- SQLDBS which supports command-like versions of DB2 APIs.
- SQLDB2 which supports a REXX specific interface to the command-line processor.

You can download a trial version of Object REXX for Windows from:

`http://www-3.ibm.com/software/ad/obj-rexx/orx_sys.html`

REXX support stabilized in DB2 Version 5, and no enhancements for REXX support are planned for the future.

Examples of REXX programs can be found in:

- `sqllib\samples\rexx`

Note: Detailed information about DB2 supported programming languages can be found in the DB2 manuals:

- *Application Development Guide: Programming Client Applications*, SC09-4826
- *Application Development Guide: Building and Running Applications*, SC09-4825

5.6 DB2 scripting in Linux and Windows

Throughout this redbook we mainly use GUIs as a means of introducing the features of DB2. Indeed for new users and those wishing to take advantage of many of the new Version 8 features in DB2, GUIs are an invaluable help to a DBA or a developer. Notwithstanding, the ability to script and automate administrative tasks is also invaluable. In this section we briefly introduce:

- Creation and management of scripts within DB2
- operating system scripting

For more information about using scripts with DB2, consult these references:

- Chapter 9 of *DB2 UDB Exploitation of the Windows Environment*, SG24-6893
- *DB2 Command Reference*, SC09-4828

You can also find an excellent article which specializes on scripting within DB2 Version 8 at this Web site:

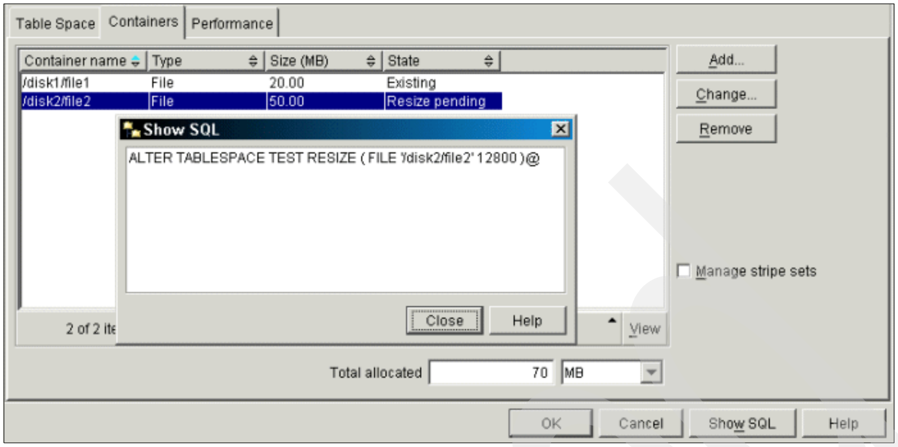
:http://www7b.software.ibm.com/dmdd/library/techarticle/0211yip/0211yip4.html

5.6.1 Creation and management of scripts within DB2

The two main ways that DB2 can assist you in building scripts are by using the Task Center and also the Show SQL function within the GUI tools. We have demonstrated the Task Center function in 3.6, “Task Center” on page 68 as part of the discussion of the DB2 GUI tools. We now briefly discuss the Show SQL function.

Show SQL

This feature of DB2 GUIs shows the SQL that the tool has generated and is about to run. This is very useful not only to understand what exact command is being run, but also as a means of incorporating the command into scripts of your own. These scripts can then be scheduled as part of the Task Center. Figure 5-14 provides an example of the *Show SQL* feature.



Container name	Type	Size (MB)	State
/disk1/file1	File	20.00	Existing
/disk2/file2	File	50.00	Resize pending

```
ALTER TABLESPACE TEST RESIZE ( FILE '/disk2/file2' 12800 )@
```

Total allocated 70 MB

Figure 5-14 Using the "Show SQL" functionality

5.6.2 Operating system scripting

The more traditional method of scripting with DB2 is to create scripts using the operating system facilities. In a Windows system this would typically be through the use of a batch file, and in Linux through the integration into a shell script. Although the platform you are using might be either Linux or Windows, the means of running DB2 commands can still be split into two basic methods:

- Integrating DB2 commands
- Calling DB2 commands

Integrating DB2 commands

In this use of scripting, DB2 commands are directly embedded into an OS script. To access DB2 in this way, it is necessary to prefix the command with `db2` as shown in Example 5-7.

Example 5-7 Using DB2 commands as part of an operating system script

```
db2 "connect to sample"
db2 -stvf part.ddl> part.out
db2 "drop table db2inst1.account"
db2 "create table db2inst1.account (account_id varchar(12) not null primary
key,branch_id char(5),account_balance int default 0,todays_transactions int
default 0)not logged initially"

db2 "drop table db2inst1.branch"
db2 "create table db2inst1.branch (branch_id char(5) not null primary
key,branch_balance int default 0)not logged initially"
```

This file can then be run as an operating system script on either Windows or Linux, subject to the following conditions:

- **Windows**: The script would require to be run from inside a DB2 Command window. The DB2 batch file is simply run as a standard operating system script **`db2script.bat`**.

  To run commands or a script from a standard Windows command prompt, the following format should be used:

  ```
  db2cmd db2 get dbm cfg
  ```

  This command opens a db2cmd window where the command is executed.

- **Linux**: The script would need to be run by a user who has the DB2 environment defined in their `.profile` file.

  ```
  if [ -f /home/db2inst1/sqllib/db2profile ]; then.
  /home/db2inst1/sqllib/db2profile fi
  ```

  The script can then be run as a standard Linux shell script **`./db2script`**.

Calling DB2 commands

As well as prefixing a command with *db2* to enable its use from an operating system script, scripts can also be created that only contain DB2 commands. This type of script is typical of what you would create in the Command Center and consists of commands you can enter interactively in the Command Center or the DB2 Command Line Processor (CLP).

The DB2 CLP can be launched from:

- Windows

  Start —> Programs —> IBM DB2 —> Command Line Tools —> Command Line Processor

  Entering **db2** in a Command Window to start the interactive mode.

- Linux

  After logging on as a user with the DB2 environment settings type **db2** to start the CLP interactive mode.

Example 5-8 is a script file that contains the DDL to create a DB2 table.

Example 5-8 Calling DB2 commands in a file (part.ddl)

```
DROP TABLE DB2INST1.PART2;
CREATE TABLE DB2INST1.PART2( P_PARTKEY     INTEGER NOT NULL,
                          P_NAME       VARCHAR(55) NOT NULL,
                          P_MFGR       CHAR(25) NOT NULL,
                          P_BRAND      CHAR(10) NOT NULL,
                          P_TYPE       VARCHAR(25) NOT NULL,
```

```
                          P_SIZE        INTEGER NOT NULL,
                          P_CONTAINER   CHAR(10) NOT NULL,
                          P_RETAILPRICE DECIMAL(15,2) NOT NULL,
                          P_COMMENT     VARCHAR(23) NOT NULL ) IN DMS1;
```

This script can be called from a Windows command prompt by running:

```
db2cmd db2 -stvf part.ddl
```

Or, this script can be called from a Linux shell by running:

```
db2 -stvf part.ddl
```

Where, options after db2 are:

- -t option means that in the script each line terminates with a semi colon.
- -v option causes the command line processor to echo to the screen.
- -f option tells the command line processor to read input from a specified file.
- -s option causes the command line processor to stop execution and to write error messages to the screen.

This output can also be piped to a file using the standard operating system pipe:

```
db2 -stvf part.ddl > part.out
```

Tip: Operating system commands can be invoked from within the DB2 CLP (or within a DB2 CLP script) by prefixing them with a ! symbol.

Note: Please consult the following product manuals for detailed information about the topics covered in this chapter:

- *Application Development Guide: Building and Running Applications*, SC09-4825
- *Application Development Guide: Programming Client Applications*, SC09-4826
- *Application Development Guide: Programming Server Applications*, SC09-4827

6

Database management

In today's complex information management environments, database administrators need an extensive set of tools to fully manage the database system efficiently and effectively. With the release of DB2 Version 8.1, a variety of new management features are available to enhance the already impressive set of tools available in Version 7.2.

In this chapter, we discuss DB2 Version 8.1 database management functionality. We introduce both existing and new features available to the database administrator. The following topics are described:

- Basic configuration
- Security
- Database backup and recovery
- Task scheduling
- Data movement
- Monitoring
- Performance

Database management is essential for a successful business system. Database management includes securing databases from unauthorized users, moving database objects and the data inside them efficiently, monitoring events, tuning the performance. And most importantly, database management safeguards the integrity of data, one of the most valuable resources in a company

In the following sections, we show how you can secure, move, monitor, tune, and safeguard your database objects. We start with a look at the configuration files used to control specific DB2 settings of your instances and databases. Next, we discuss how to secure your databases and instances followed by database backup and recovery tools. We will also dedicate sections to task scheduling through Task Center DB2 tool and the movement of data with utilities such as **`import`**, **export**, **`load`**, **`db2move`**, and **`db2look`**. Our final two sections focus on monitoring and performance tools.

6.1 Basic configuration

There are three types of configuration parameter files used by DB2. They are:

- Database manager configuration file
- Database configuration file
- Administration configuration file

Each of these files controls a specific part of DB2. The database manager configuration file maintains configuration parameters for a DB2 instance. The database configuration file is specific to a single database. The administration configuration file is associated with the administration server instance. Database administrators can tune a database system by modifying these files through the configuration parameters.

When a DB2 instance and database are created, DB2 automatically sets these configuration parameters with default values. Database administrators can evaluate and tune these parameters through the DB2 tools, such as Control Center or from a command line session.

In addition, there are DB2 environment variables that a database administrator can use to tune the database system. The DB2 environment variables can be modified through the **`db2set`** command. Each of these configuration topics are discussed in the following sections.

In our examples, we use the command line to execute DB2 commands. On Linux, use your instance owner account and type the commands from a terminal window. Make sure you have switched to the instance owner using the **`su`** command with a dash, **`-,`** to execute your user’s .profile. On Windows, use your account ID that was used to install DB2 or a Windows administrator ID; then open the DB2 Command Window from **Start —> Programs —> IBM DB2 —> Command Line Tools —> Command Window.**

6.1.1 Database manager configuration

When you create a DB2 instance, a database manager configuration file gets created with a set of parameters. Some of the configurations that can be modified at the instance level include database monitor switches, system authorization groups, sort heap threshold, maximum number of agents allowed, and many more.

Retrieve database manager configuration file

To retrieve the database manager configuration file, type the following command:

```
db2 get dbm cfg
```

Example 6-1 displays the database manager configuration file output.

Example 6-1 Database manager configuration file output

```
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) = 1.373735e-006
Communications bandwidth (MB/sec)      (COMM_BANDWIDTH) = 1.000000e+000

Max number of concurrently active databases     (NUMDB) = 8
Data Links support                          (DATALINKS) = NO
Federated Database System Support           (FEDERATED) = NO
Transaction processor monitor name        (TP_MON_NAME) =

Default charge-back account           (DFT_ACCOUNT_STR) =

 Java Development Kit installation path       (JDK_PATH) =
C:\PROGRA~1\IBM\SQLLIB\java\jdk

Diagnostic error capture level              (DIAGLEVEL) = 3
Notify Level                              (NOTIFYLEVEL) = 3
Diagnostic data directory path               (DIAGPATH) =

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

```
Monitor health of instance and databases    (HEALTH_MON) = ON

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

Database manager authentication         (AUTHENTICATION) = SERVER
Cataloging allowed without authority    (CATALOG_NOAUTH) = NO
Trust all clients                       (TRUST_ALLCLNTS) = YES
Trusted client authentication           (TRUST_CLNTAUTH) = CLIENT
Use SNA authentication                    (USE_SNA_AUTH) = NO
Bypass federated authentication             (FED_NOAUTH) = NO

Default database path                        (DFTDBPATH) = C:

Database monitor heap size (4KB)           (MON_HEAP_SZ) = 66
Java Virtual Machine heap size (4KB)      (JAVA_HEAP_SZ) = 512
Audit buffer size (4KB)                   (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB)   (INSTANCE_MEMORY) = AUTOMATIC
Backup buffer default size (4KB)             (BACKBUFSZ) = 1024
Restore buffer default size (4KB)            (RESTBUFSZ) = 1024

Agent stack size                        (AGENT_STACK_SZ) = 16
Minimum committed private memory (4KB)    (MIN_PRIV_MEM) = 32
Private memory threshold (4KB)         (PRIV_MEM_THRESH) = 20000

Sort heap threshold (4KB)                   (SHEAPTHRES) = 10000

Directory cache support                      (DIR_CACHE) = YES

Application support layer heap size (4KB)    (ASLHEAPSZ) = 15
Max requester I/O block size (bytes)          (RQRIOBLK) = 32767
DOS requester I/O block size (bytes)      (DOS_RQRIOBLK) = 4096
Query heap size (4KB)                    (QUERY_HEAP_SZ) = 1000
DRDA services heap size (4KB)             (DRDA_HEAP_SZ) = 128

Priority of agents                            (AGENTPRI) = SYSTEM
Max number of existing agents                (MAXAGENTS) = 400
Agent pool size                         (NUM_POOLAGENTS) = 200(calculated)
Initial number of agents in pool        (NUM_INITAGENTS) = 0
Max number of coordinating agents      (MAX_COORDAGENTS) = (MAXAGENTS -
NUM_INITAGENTS)
Max no. of concurrent coordinating agents   (MAXCAGENTS) = MAX_COORDAGENTS
Max number of client connections       (MAX_CONNECTIONS) = MAX_COORDAGENTS

Keep fenced process                         (KEEPFENCED) = YES
Number of pooled fenced processes          (FENCED_POOL) = MAX_COORDAGENTS
Initialize fenced process with JVM      (INITFENCED_JVM) = NO
Initial number of fenced processes      (NUM_INITFENCED) = 0
```

```
Index re-creation time                       (INDEXREC) = ACCESS

Transaction manager database name         (TM_DATABASE) = 1ST_CONN
Transaction resync interval (sec)     (RESYNC_INTERVAL) = 180

SPM name                                     (SPM_NAME) = A23BK55Y
SPM log size                          (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit                 (SPM_MAX_RESYNC) = 20
SPM log path                             (SPM_LOG_PATH) =

NetBIOS Workstation name                        (NNAME) = N00E3D95

TCP/IP Service name                          (SVCENAME) = db2c_DB2
Discovery mode                               (DISCOVER) = SEARCH
Discovery communication protocols       (DISCOVER_COMM) =
Discover server instance                (DISCOVER_INST) = ENABLE

Maximum query degree of parallelism   (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism     (INTRA_PARALLEL) = NO

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) = 4096
Node connection elapse time (sec)         (CONN_ELAPSE) = 10
Max number of node connection retries (MAX_CONNRETRIES) = 5
Max time difference between nodes (min) (MAX_TIME_DIFF) = 60

db2start/db2stop timeout (min)        (START_STOP_TIME) = 10
```

A description of each of these parameters is available in Chapter 13, “Configuration parameters summary” in *Administration Guide: Performance*, SC09-4821.

You also can also use the DB2 Information Center tool to find parameter descriptions.

- To execute Information Center on Windows: **Start —> Programs —> IBM DB2 —> Information —> Information Center.**
- To execute Information Center on Linux, type **db2ic** at the terminal window or invoke the tool from the Control Center.

You can also find online documentation for the database manager configuration parameters using the online DB2 Information Center at:

http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/techlib_v8.d2w/report

Once at the Web site, enter the following search terms in the text box:

```
database manager configuration parameter summary
```

Then select the **Administration** check box and click the button **Search DB2 V8 Information Center**.

Update database manager configuration file

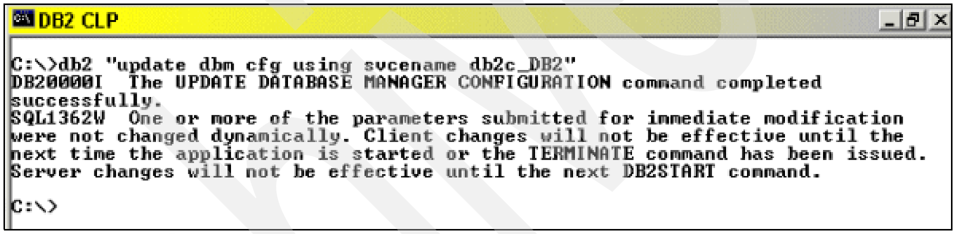
To update the database manager configuration file use the following syntax:

```
db2 update dbm cfg using <parameter name> <parameter value>
```

For example:

```
db2 update dbm cfg using svcename db2c_DB2
```

The above update will change the service name; see Figure 6-1. This SVCENAME parameter is associated with the port in the services file, to listen for incoming TCPIP instance attachment requests.



```
C:\>db2 "update dbm cfg using svcename db2c_DB2"
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:\>
```

Figure 6-1 Update database manager configuration file

> **Note:** Some parameters take effect immediately after the command executes while others may require you to restart the DB2 instance. Figure 6-1 displays a warning message stating that the instance must be restarted before the changes will be effective.

6.1.2 Database configuration file

When you create a database, a database configuration file gets created with a set of parameters. The modification of these parameters enables a database to perform based upon your requirements. For example, LOGRETAIN allows recovery of log files during a roll forward operation, BUFFPAGE sets the default buffer pool size when a buffer pool is created, and SORTHEAP configures sort heap for in-memory sorts, and so on.

Retrieve database configuration file

To retrieve the database configuration file, type the following command:

```
db2 get db cfg for <database name>
```

For example:

```
db2 get db cfg for sample
```

Example 6-2 displays the database configuration file output for the SAMPLE database.

Example 6-2 Database configuration file output for the SAMPLE database

```
Database Configuration for Database sample

Database configuration release level                    = 0x0a00
Database release level                                  = 0x0a00

Database territory                                      = US
Database code page                                      = 1252
Database code set                                       = IBM-1252
Database country/region code                            = 1

Dynamic SQL Query management           (DYN_QUERY_MGMT) = DISABLE

Discovery support for this database       (DISCOVER_DB) = ENABLE

Default query optimization class         (DFT_QUERYOPT) = 5
Degree of parallelism                      (DFT_DEGREE) = 1
Continue upon arithmetic exceptions   (DFT_SQLMATHWARN) = NO
Default refresh age                   (DFT_REFRESH_AGE) = 0
Number of frequent values retained     (NUM_FREQVALUES) = 10
Number of quantiles retained            (NUM_QUANTILES) = 20

Backup pending                                          = NO

Database is consistent                                  = YES
Rollforward pending                                     = NO
Restore pending                                         = NO

Multi-page file allocation enabled                      = NO

Log retain for recovery status                          = NO
User exit for logging status                            = NO

Data Links Token Expiry Interval (sec)      (DL_EXPINT) = 60
Data Links Write Token Init Expiry Intvl(DL_WT_IEXPINT) = 60
Data Links Number of Copies             (DL_NUM_COPIES) = 1
Data Links Time after Drop (days)        (DL_TIME_DROP) = 1
Data Links Token in Uppercase                (DL_UPPER) = NO
Data Links Token Algorithm                   (DL_TOKEN) = MAC0

Database heap (4KB)                            (DBHEAP) = 600
Size of database shared memory (4KB)  (DATABASE_MEMORY) = AUTOMATIC
Catalog cache size (4KB)              (CATALOGCACHE_SZ) = (MAXAPPLS*4)
Log buffer size (4KB)                        (LOGBUFSZ) = 8
Utilities heap size (4KB)                (UTIL_HEAP_SZ) = 5000
Buffer pool size (pages)                     (BUFFPAGE) = 250
Extended storage segments size (4KB)    (ESTORE_SEG_SZ) = 16000
Number of extended storage segments   (NUM_ESTORE_SEGS) = 0
Max storage for lock list (4KB)              (LOCKLIST) = 50

Max size of appl. group mem set (4KB) (APPGROUP_MEM_SZ) = 30000
Percent of mem for appl. group heap   (GROUPHEAP_RATIO) = 70
Max appl. control heap size (4KB)     (APP_CTL_HEAP_SZ) = 128

Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = (SHEAPTHRES)
Sort list heap (4KB)                         (SORTHEAP) = 256
SQL statement heap (4KB)                     (STMTHEAP) = 2048
Default application heap (4KB)             (APPLHEAPSZ) = 256
Package cache size (4KB)                   (PCKCACHESZ) = (MAXAPPLS*8)
Statistics heap size (4KB)               (STAT_HEAP_SZ) = 4384

Interval for checking deadlock (ms)         (DLCHKTIME) = 10000
Percent. of lock lists per application       (MAXLOCKS) = 22
Lock timeout (sec)                        (LOCKTIMEOUT) = -1

Changed pages threshold                (CHNGPGS_THRESH) = 60
Number of asynchronous page cleaners   (NUM_IOCLEANERS) = 1
Number of I/O servers                   (NUM_IOSERVERS) = 3
Index sort flag                             (INDEXSORT) = YES
Sequential detect flag                      (SEQDETECT) = YES
Default prefetch size (pages)         (DFT_PREFETCH_SZ) = 16

Track modified pages                         (TRACKMOD) = OFF

Default number of containers                            = 1
Default tablespace extentsize (pages)   (DFT_EXTENT_SZ) = 32

Max number of active applications            (MAXAPPLS) = AUTOMATIC
Average number of active applications       (AVG_APPLS) = 1
Max DB files open per application             (MAXFILOP) = 64

Log file size (4KB)                         (LOGFILSIZ) = 250
Number of primary log files                (LOGPRIMARY) = 3
Number of secondary log files               (LOGSECOND) = 2
Changed path to log files                  (NEWLOGPATH) =
Path to log files                                       =
C:\DB2\NODE0000\SQL00002\SQLOGDIR\
Overflow log path                     (OVERFLOWLOGPATH) =
Mirror log path                         (MIRRORLOGPATH) =
First active log file                                   =
Block log on disk full                (BLK_LOG_DSK_FUL) = NO

Percent of max active log space by transaction(MAX_LOG) = 0
Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

Group commit count                          (MINCOMMIT) = 1
Percent log file reclaimed before soft chckpt (SOFTMAX) = 100
Log retain for recovery enabled             (LOGRETAIN) = OFF
User exit for logging enabled                (USEREXIT) = OFF

Auto restart enabled                      (AUTORESTART) = ON
Index re-creation time                       (INDEXREC) = SYSTEM (ACCESS)
Default number of loadrec sessions    (DFT_LOADREC_SES) = 1
Number of database backups to retain   (NUM_DB_BACKUPS) = 12
Recovery history retention (days)     (REC_HIS_RETENTN) = 366

TSM management class                    (TSM_MGMTCLASS) =
TSM node name                            (TSM_NODENAME) =
TSM owner                                   (TSM_OWNER) =
TSM password                             (TSM_PASSWORD) =
```

A description of each of these parameters is available in Chapter 13, “Configuration parameters summary” in *Administration Guide: Performance*, SC09-4821.

You can also find the description by using the DB2 Information Center tool.

- To execute Information Center on Windows: **Start —> Programs —> IBM DB2 —> Information —> Information Center.**
- To execute Information Center on Linux, type **`db2ic`** at the terminal window or lunch from Control Center.

You can also find online documentation for the database configuration parameters using the online DB2 Information Center at:

http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/techlib_v8.d2w/report

Once at the Web site, enter the following search terms in the text box:

```
database configuration parameter summary
```

Then select the **Administration** check box and click the button **Search DB2 V8 Information Cente**r. Both the database manager configuration parameter summary and the database configuration parameter summary are located on the same Web page. From the Internet Explorer menu bar, use the **Edit —> Find (on This Page)...** function and search for database configuration parameter summary.

Update database configuration file

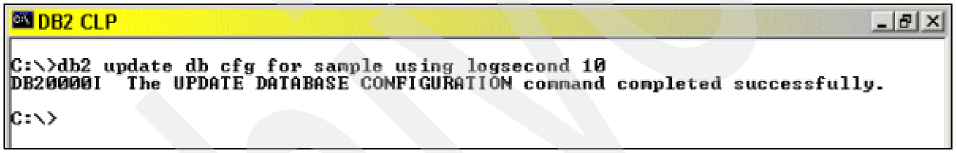
To update the database configuration file use the following syntax:

```
db2 update db cfg for <database name> using <parameter name> <parameter value>
```

For example:

```
db2 update db cfg for sample using logsecond 10
```

The above update will change the logsecond parameter to 10 (Figure 6-2). The *logsecond* parameter is associated with the maximum number of secondary log files that will be created during a unit-of-work. If the logsecond parameter is exceeded, an error will be returned.



```
C:\>db2 update db cfg for sample using logsecond 10
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.

C:\>
```

Figure 6-2 Update database configuration file

Note: Some parameters take effect immediately after the command executes, while others may require the database to be deactivated and reactivated.

6.1.3 Administration server configuration

When you create an administration server instance, an administration configuration file gets created with a set of parameters. The administration configuration file contains parameters affecting the functionality of the DB2 GUI tools. Some of the parameters that can be modified include the *authentication* parameter, which controls how users are authenticated through the DB2 GUI tools, such as Control Center; the *toolscat_db* parameter, which specifies the database that is used by the Scheduler; the *smtp_server* parameter which determines the mail server that will send e-mail notifications.

Retrieve administration server configuration file

To retrieve the administration server configuration file, type the following command:

```
db2 get admin cfg
```

Example 6-3 displays the administration server configuration file output.

Example 6-3 Administration server configuration file output

```
Admin Server Configuration

Authentication Type DAS                (AUTHENTICATION) = SERVER_ENCRYPT

DAS Administration Authority Group Name  (DASADM_GROUP) =

DAS Discovery Mode                           (DISCOVER) = SEARCH
Name of the DB2 Server System               (DB2SYSTEM) = A23BK55Y

Java Development Kit Installation Path DAS   (JDK_PATH) = C:\Program
Files\IBM\SQLLIB\\java\jdk\

DAS Code Page                            (DAS_CODEPAGE) = 0
DAS Territory                           (DAS_TERRITORY) = 0

Location of Contact List                 (CONTACT_HOST) =
Execute Expired Tasks                   (EXEC_EXP_TASK) = NO
Scheduler Mode                           (SCHED_ENABLE) = ON
SMTP Server                               (SMTP_SERVER) =
Tools Catalog Database                    (TOOLSCAT_DB) = TOOLSDB
Tools Catalog Database Instance         (TOOLSCAT_INST) = DB2
Tools Catalog Database Schema         (TOOLSCAT_SCHEMA) = SYSTOOLS
Scheduler User ID                                       =
```

A description of each of these parameters is available in Chapter 13, “DB2 Administration Server” in *Administration Guide: Performance*, SC09-4821.

You can also find the description by using the DB2 Information Center tool.

- To execute Information Center on Windows: **Start —> Programs —> IBM DB2 —> Information —> Information Center.**
- To execute Information Center on Linux, type **`db2ic`** at the terminal window or lunch from Control Center.

You can also find online documentation for the administration server configuration parameters using the online DB2 Information Center at:

http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/techlib_v8.d2w/report

Once at the Web site, enter the following search terms in the text box:

```
das configuration
```

Then select the **Administration** check box and click the button **Search DB2 V8 Information Center**.

Update administration server configuration file

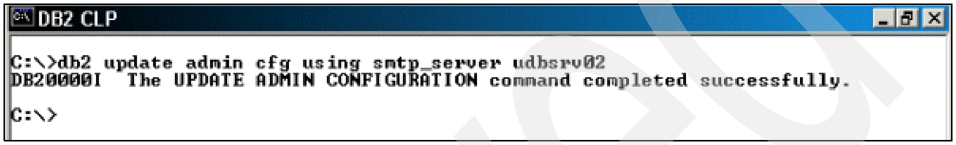
To update the administration server configuration file use the following syntax:

```
db2 update admin cfg using <parameter name> <parameter value>
```

For example:

```
db2 update admin cfg using smtp_server udbsrv02
```

The above update will specify which server to use as the SMTP mail server (Figure 6-3). When notifications are sent, DB2 will look at the *smtp_server* parameter value to determine which machine to search for the active mail server.



```
C:\>db2 update admin cfg using smtp_server udbsrv02
DB20000I  The UPDATE ADMIN CONFIGURATION command completed successfully.

C:\>
```

Figure 6-3 Update administration server configuration file

Note: Some parameters take effect immediately after the command executes, while others may require you to restart the administration server instance. Figure 6-3 does not show a warning that the DB2 instance must be restarted. This implies that the command will be effective immediately.

6.1.4 The DB2 profile registry

The DB2 profile registry contains a number of DB2 environment variables which can be set to modify how DB2 behaves. To set a variable in the profile registry, we use the **db2set** command.

```
db2set <variable>=<parameter value> -<registry profile level> <instance name>
```

For example, in order for a DB2 instance to receive incoming TCPIP attach requests, the DB2COMM profile registry variable must be set:

```
db2set DB2COMM=TCPIP -i DB2
```

The above command will set the variable in the instance level profile registry. If the registry profile level is not specified in the db2set command, the variable will be set using your currently set instance. For example, the following syntax will set DB2COMM to TCPIP in the DB2 instance profile registry since it is the active DB2 instance.

```
dbset DB2COMM=TCPIP
```

You can verify the instance that is active by typing the following command:

```
db2 get instance
```

Or, you can use the **db2set** command to look at the DB2INSTDEF registry variable (Example 6-4).

> **Note:** Once a profile variable is set, you must restart the DB2 instance for it to take effect.

To quickly list the currently set profile registry variables, type the following:

```
db2set -all
```

Example 6-4 Output from db2set -all command in Windows environment

```
[e] DB2PATH=C:\Program Files\IBM\SQLLIB
[i] DB2ACCOUNTNAME=A23BK55Y\db2admin
[i] DB2INSTOWNER=A23BK55Y
[i] DB2PORTRANGE=60000:60003
[i] DB2NBADAPTERS=0
[i] DB2INSTPROF=C:\PROGRA~1\IBM\SQLLIB
[i] DB2COMM=TCPIP
[g] DB2SYSTEM=A23BK55Y
[g] DB2PATH=C:\Program Files\IBM\SQLLIB
[g] DB2INSTDEF=DB2
[g] DB2ADMINSERVER=DB2DAS00
```

As seen in Example 6-4, each variable is pre-fixed with an [e], [g], or [i]. The [e] stands for environment specifying that the variable is set in the environment. The [g] will enforce the variable to be set at a global level while [i] will be set at the instance level.

To quickly list the available profile registry variables, type the following:

```
db2set -lr
```

Example 6-5 Partial output from db2set -lr command

```
DB2CCMSRV
DB2_PARALLEL_IO
DB2ACCOUNT
DB2ADMINSERVER
DB2BQTIME
DB2BQTRY
DB2CHKPTR
DB2CLIENTADPT
DB2CLIINIPATH
DB2CODEPAGE
DB2COMM
DB2COUNTRY
DB2DBDFT
DB2DBMSADDR
```

```
DB2DEFPREP
DB2DMNBCKCTLR
DB2INCLUDE
DB2INSTDEF
DB2INSTPROF
DB2IQTIME
DB2LOADREC
DB2LOCK_TO_RB
DB2NBADAPTERS
DB2NBCHECKUPTIME
DB2NBINTRLISTENS
DB2NBRECVBUFFSIZE
DB2NBRECVNCBS
DB2NBRESOURCES
DB2NBSENDNCBS
DB2NBSESSIONS
```

Each of these profile variables is described in Appendix A in *Administration Guide: Performance*, SC09-4821.

You also can find the profile variables using Information Center tool.

- To execute Information Center on Windows: **Start —> Programs —> IBM DB2 —> Information —> Information Center.**
- To execute Information Center on Linux, type **`db2ic`** at the terminal window or lunch from Control Center.

You can also find online documentation for the db2set profile registry using the online DB2 Information Center at:

http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/techlib_v8.d2w/report

Once at the Web site, enter the following search terms in the text box:

```
declaring registry and environment variables
```

Then select the **Administration** check box and click the button Search **DB2 V8 Information Center**.

6.1.5 Client connectivity

Most likely, your strategy with DB2 will require users to connect to the remote database server from a client machine. This section describes how you can setup client connectivity to the database server.

To connect to the database server from your client machine, each client will need to have DB2's client code.

This code is free and can be downloaded from:

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

Note: With the DB2 Enterprise Server Edition (ESE) trial code product, the client component is automatically included.

After you install the DB2 client, you can set up a client connection using either the Configuration Assistant GUI tool, or use the DB2 CLP. For more information on using Configuration Assistant to create a client connection, refer to Chapter 5, “Configuring client-to-server communications” in *IBM DB2 UDB Quick Beginnings for DB2 Clients V8*, GC09-4832. Alternatively, the following URL provides a PDF of the above reference guide:

ftp://ftp.software.ibm.com/ps/products/db2/info/vr8/pdf/letter/db2ite80.pdf

Using Figure 6-4 on page 210 as a guide, we discuss how to setup a client/server database connection using the DB2 CLP and services files.

In Figure 6-4, there are four boxes which are numbered 1 through 4. Each box depicts a particular role in setting up client connectivity.

Server configuration

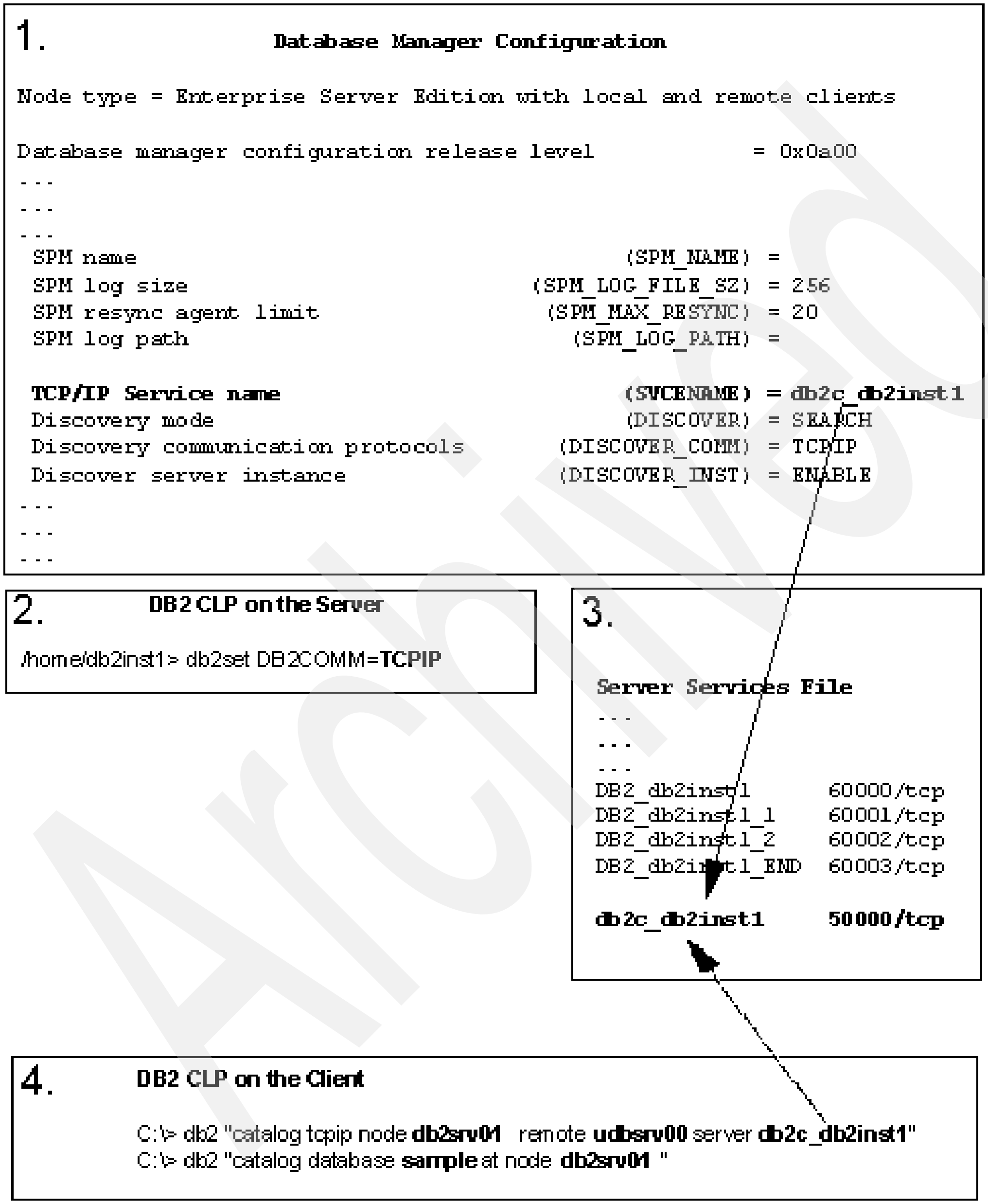
On the database server, it is necessary to configure the DB2 instance to be able to accept incoming database connection requests. DB2 listens for requests at the instance level by listening on ports that you specify in the services file (box 3). On Windows, the services file is located at %WINDIR%\system32\drivers\etc. On Linux, the services file is located at /etc. Depending on the permissions assigned to the services file on Linux, you may need to have root authority when editing the file.

To associate the port in the services file with the DB2 instance, the *svcename* database manager configuration parameter should be set (box 1). Here is the command we used to set the *svcename*:

```
db2 “update dbm cfg using svcename db2c_db2inst1”
```

The *svcename* parameter associates the DB2 instance with the port in the services file. Because there can be many DB2 instances on a server, each instance should have a unique service name with an associated connection and interrupt port in the services file.

After updating the database manager configuration file and services file, the DB2 registry variable DB2COMM should be set so that the DB2 instance knows what protocols to listen for (box 2). Once this variable is set you must restart the DB2 instance.



1\.

```
                    Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

Database manager configuration release level            = 0x0a00
...
...
...
 SPM name                                     (SPM_NAME) =
 SPM log size                          (SPM_LOG_FILE_SZ) = 256
 SPM resync agent limit                 (SPM_MAX_RESYNC) = 20
 SPM log path                             (SPM_LOG_PATH) =

 TCP/IP Service name                          (SVCENAME) = db2c_db2inst1
 Discovery mode                               (DISCOVER) = SEARCH
 Discovery communication protocols       (DISCOVER_COMM) = TCPIP
 Discover server instance                (DISCOVER_INST) = ENABLE
...
...
...
```

2. **DB2 CLP on the Server**

```
/home/db2inst1> db2set DB2COMM=TCPIP
```

3\.

```
Server Services File
...
...
...
DB2_db2inst1         60000/tcp
DB2_db2inst1_1       60001/tcp
DB2_db2inst1_2       60002/tcp
DB2_db2inst1_END     60003/tcp

db2c_db2inst1        50000/tcp
```

4. **DB2 CLP on the Client**

```
C:\> db2 "catalog tcpip node db2srv01   remote udbsrv00 server db2c_db2inst1"
C:\> db2 "catalog database sample at node db2srv01 "
```

Figure 6-4 Client/server configuration schematics

Client configuration

To connect to a database on the server, it is necessary to make the DB2 client aware of the database server, the DB2 instance, and the database via cataloging DB2 instance and database (box 4). The first command allows us to catalog the node (that is, the DB2 server instance). Cataloging the node allows us to specify the host name or IP address of the database server and the service name associated with the ports. By cataloging the node we are effectively referencing the DB2 instance on the server associated with the service name. Note that **db2srv01** is an arbitrary name we assign for later reference when issuing the **catalog database** command.

Cataloging the database on the client allows us to reference a specific database in the DB2 instance on the server (box 4). When cataloging the database, we reference the arbitrarily named node **db2srv01** so that DB2 knows what instance is associated with the database that we are cataloging. Once these updates are made, you must restart the DB2 instance on the client.

After the above steps are taken, you should be able to connect to the database. To connect to the database, type the following command from the DB2 client:

```
db2 connect to <database alias> user <username> using <password>
```

If the database manager configuration parameter *authentication* is set to *server* (the default), you will need to specify a user account that resides on the database server.

Connecting to DB2 on a host server

If your strategy includes remotely connecting to a database residing on a host server such as VM, VSE, or AS400, it is necessary to have the DB2 Connect product installed. The option to install the DB2 Connect product is available during initial installation of your DB2 ESE V8.1 trial code download, or you can install the product with the DB2 Setup wizard at a later time. Note that although DB2 Connect comes bundled with DB2 ESE, it is a separate product and requires its own license.

6.2 Security

DB2 uses a combination of external security services and internal access control information to secure the information kept in the database. To access a database server, a user must be authenticated either on the client or the server. The database manager configuration file contains configuration parameters to define how clients are authenticated. Once a user is authenticated and has access to a DB2 instance, DB2 continues to provide security by restricting the user to the set of authorities and privileges that the user is assigned to.

6.2.1 Authentication

When first setting up your database server, you should consider how users will be authenticated. There are several configuration parameters in the database manager configuration file that have an effect on how users are authenticated. By modifying the parameters, you can customize the security level of your DB2 instance.

The following configuration parameters are available:

- **Database manager authentication (AUTHENTICATION)**

  This parameter specifies whether the client is to be authenticated at the server or client machine; or alternatively, a Kerberos security system. The following parameter values are available CLIENT; SERVER; SERVER_ENCRYPT; KERBEROS; KRB_SERVER_ENCRYPT.

  - SERVER: This is the default value. With this setting, a client's user ID and password are sent to the server for authentication. This means that the user on the client submitting the request must have access to a user ID that is created on the server.
  - CLIENT: Allows the database manager to authenticate at the client. If this setting is used, you may consider two other configuration parameters, TRUST_ALLCLNTS and TRUST_CLNTAUTH.
  - KERBEROS: Means that authentication is performed at a Kerberos server using the Kerberos security protocol for authentication. With an authentication type of KRB_SERVER_ENCRYPT at the server and clients that support the Kerberos security system, the effective system authentication type is KERBEROS. If the clients do not support the Kerberos security system, then the effective system authentication type is equivalent to SERVER_ENCRYPT.

- **Trust All Clients (TRUST_ALLCLNTS)**

  TRUST_ALLCLNTS can be set to YES (default) or NO. If set to YES, all clients will be authenticated at the client regardless of whether or not the client has a security sub-system. A value of NO prevents client machines without a security sub-system from authenticating at the client and forces the user to provide a user id and password for authentication at the server.

- **Trusted Clients Authentication (TRUST_CLNTAUTH)**

  TRUST_CLNTAUTH can be set to CLIENT (default) or SERVER. If a user ID and password are not provided, the client is assumed to have validated the user, and no further validation is performed at the server. If this parameter is set to CLIENT (the default), the trusted client can connect without providing a user ID and password combination, and the assumption is that the operating system has already authenticated the user. If it is set to SERVER, the user ID and password will be validated at the server.

- **Cataloging allowed without authority (CATALOG_NOAUTH)**

  This parameter specifies whether users are able to catalog and uncatalog databases and nodes, or DCS and ODBC directories, without SYSADM authority. The default value NO for this parameter indicates that SYSADM authority is required. When this parameter is set to YES, SYSADM authority is not required.

- **Use SNA authentication (USE_SNA_AUTH)**

  When USE_SNA_AUTH is set to yes and AUTHENTICATION is set to SERVER, inbound connections to the server that use the SNA protocol with security type SAME or PROGRAM are only authenticated at the SNA layer, and not by DB2.

- **Bypass federated authentication (FED_NOAUTH)**

  When FED_NOAUTH is set to YES, AUTHENTICATION is set to SERVER or SERVER_ENCRYPT, and FEDERATED is set to YES, then authentication at the instance is bypassed. It is assumed that authentication will happen at the data source. Exercise caution when FED_NOAUTH is set to YES. Authentication is done at neither the client nor at DB2. Any user who knows the SYSADM authentication name can assume SYSADM authority for the federated server.

6.2.2 Authorization

Once the user is authenticated and attached to the DB2 instance, the user will be able to use DB2 objects based on the authorities and privileges that are given. There are three types of authorities which are assigned to groups by the instance owner (on Linux) or an Administrative user (on Windows). These authorities are available so that a user can maintain database objects at a high level, such as the database or tables. The following authorities are configurable in the database manager configuration file: SYSADM group name (SYSADM_GROUP); SYSCTRL group name (SYSCTRL_GROUP); and SYSMAINT group name (SYSMAINT_GROUP).

- **System administration authority group name (SYSADM_GROUP)**

  System administration (SYSADM) authority is the highest level of authority within the database manager and controls all database objects. This parameter defines the group name with SYSADM authority for the database manager instance.

  SYSADM authority is determined by the security facilities used in a specific operating environment.

For the Windows NT and Windows 2000 operating system, this parameter can be set to any local group that has a name of eight characters or fewer, and is defined in the Windows NT and Windows 2000 security database. If NULL is specified for this parameter, all members of the Administrators group have SYSADM authority.

For UNIX-based systems, if NULL is specified as the value of this parameter, the SYSADM group defaults to the primary group of the instance owner.

If the value is not NULL, the SYSADM group can be any valid UNIX group name.

To restore the parameter to its default (NULL) value, use the command **`update dbm cfg using sysadm_group NULL`**. You must specify the keyword NULL in uppercase. You can also use the Configure Instance notebook in the DB2 Control Center.

- **System control authority group name (SYSCTRL_GROUP)**

  This parameter defines the group name with system control (SYSCTRL) authority. SYSCTRL has privileges allowing operations affecting system resources, but does not allow direct access to data.

  To restore the parameter to its default (NULL) value, use the command **`update dbm cfg using sysctrl_group NULL`**. The keyword NULL must be specified in uppercase. You can also use the Configure Instance notebook in the DB2 Control Center.

- **System maintenance authority group (SYSMAINT_GROUP)**

  This parameter defines the group name with system maintenance (SYSMAINT) authority. SYSMAINT has privileges to perform maintenance operations on all databases associated with an instance without having direct access to the table data.

  To restore the parameter to its default (NULL) value, use the command: **`update dbm cfg using sysmaint_group NULL`**. You must specify the keyword NULL in uppercase. You can also use the Configure Instance notebook in the DB2 Control Center.

  To determine the authorities you have, type the following command after connecting to a database:

  ```
  db2 get authorizations
  ```

  Example 6-6 shows authorization output for an administrator on a Windows operating system.

Example 6-6 List of authorizations for Administrator

```
Administrative Authorizations for Current User

 Direct SYSADM authority              = NO
```

```
 Direct SYSCTRL authority             = NO
 Direct SYSMAINT authority            = NO
 Direct DBADM authority               = YES
 Direct CREATETAB authority           = YES
 Direct BINDADD authority             = YES
 Direct CONNECT authority             = YES
 Direct CREATE_NOT_FENC authority     = YES
 Direct IMPLICIT_SCHEMA authority     = YES
 Direct LOAD authority                = YES

 Indirect SYSADM authority            = YES
 Indirect SYSCTRL authority           = NO
 Indirect SYSMAINT authority          = NO
 Indirect DBADM authority             = NO
 Indirect CREATETAB authority         = YES
 Indirect BINDADD authority           = YES
 Indirect CONNECT authority           = YES
 Indirect CREATE_NOT_FENC authority   = NO
 Indirect IMPLICIT_SCHEMA authority   = YES
 Indirect LOAD authority              = NO
```

Direct authorities are given to a user explicitly by another user, whereas indirect authorities are based on the group to which a user belongs.

6.2.3 Authorities and privileges at the database level

At a more granular level, authorities and privileges can be assigned to and removed from a user through the use of the GRANT and REVOKE statements. Both the GRANT and REVOKE statements can apply to a single user, group, or to everyone. Example 6-7 shows the following types of authorities and privileges available.

Example 6-7 List of authorities and privileges available

```
Database Authorities
Index Privileges
Package Privileges
Schema Privileges
Table, View, or Nickname Privileges
Server Privileges
Table Space Privileges
Sequence Privileges
Routine Privileges
```

Each set of authorities or privileges mentioned in Example 6-7 has a subset of authorities or privileges. Example 6-8 shows a list of authorities that are available within the Database Authorities category.

Example 6-8 List of authorities specific to Database Authorities

```
bindadd
connect
createtab
create_external_routine
create_not_fenced_routine
implicit_schema
dbadm
load
quiesce_connect
```

To use the GRANT or REVOKE statement, your user ID must have a specific level of authority. For example, to grant the DBADM database authority to another user, you must have SYSADM authorities. All other authorities within Database Authorities require you to have DBADM authority. Here is an example of how you can use the GRANT statement after connecting to the database:

```
db2 grant createtab on database to user <user name>
```

Here is an example of how you can use the REVOKE statement after connecting to the database:

```
db2 revoke createtab on database from user <user name>
```

Alternatively, you can perform a grant or revoke statement to affect all users:

```
db2 revoke createtab on database from public
```

> **Note:** Revoking a specific privilege does not necessarily revoke the ability to perform an action. A user may proceed with a task if other privileges are held by PUBLIC or a group, or if the user has a higher level authority, such as DBADM.

6.3 Database backup and recovery

To safeguard the data contained in your databases, DB2 has the **backup**, **restore** and **rollforward** utilities. This section describes how to backup and recover databases, schedule tasks using Control Center, and DB2 data movement functions. For detailed backup and recovery information and strategy, refer to DB2 UDB V8 manual *Data Recover and High Availability Guide and Reference*, SC09-4831.

6.3.1 Database backup

DB2 is flexible in allowing different methods for backing up your database. Before backing up a database, it is important to determine what type of recovery strategy you have in mind. For instance, your strategy may be that you only care about the data you have since the last time you backed up your database. A typical scenario is that of a data warehouse environment where changes to the database are static. It could be today or a week ago that you last backed up your database. In a database failure scenario, any data later than the last backup will be lost. If this is the recovery scenario you are looking for, a simple backup of the database can be performed to save off the currently stored data. The simplest backup can be performed with the syntax **`backup database <database name>`**. This is an offline database backup, meaning all applications must be disconnected from the database. The above backup command will save the database to the current directory.

Most likely, if your data changes daily such as in an OLTP environment, a better backup strategy is required. For those who need a more rigorous backup strategy, a database can be backed up online while users are connected, or incrementally to backup only data pages that have changed since the last backup, or a single tablespace rather than the whole database, or many combinations of the preceding methods.

Before performing a backup

If you want to recover up to the point when a database becomes unusable, it is necessary to tell DB2 to do this. By updating the database configuration file parameter LOGRETAIN to RECOVERY, you are saying all transaction logs are to be preserved so that those log files are re-applied to the database. This type of logging is called *archival logging*. The process of re-applying log files to the database is called *roll forward recovery* and is discussed in 6.3.2, "Database recovery" on page 236. If we keep the LOGRETAIN parameter at the default value, NO, the log files will eventually be overwritten by new transactions, thereby making the log files useless for a full recovery after a database restore. This type of logging is called *circular logging*.

Let's go through an example of how to update the database configuration file through the Control Center to enable recovery logging:

1. From a DB2 CLP or terminal window, type **`db2cc`** to bring up the Control Center.
2. On the left pane of Control Center, expand the directory tree down to the SAMPLE database (Figure 6-5).

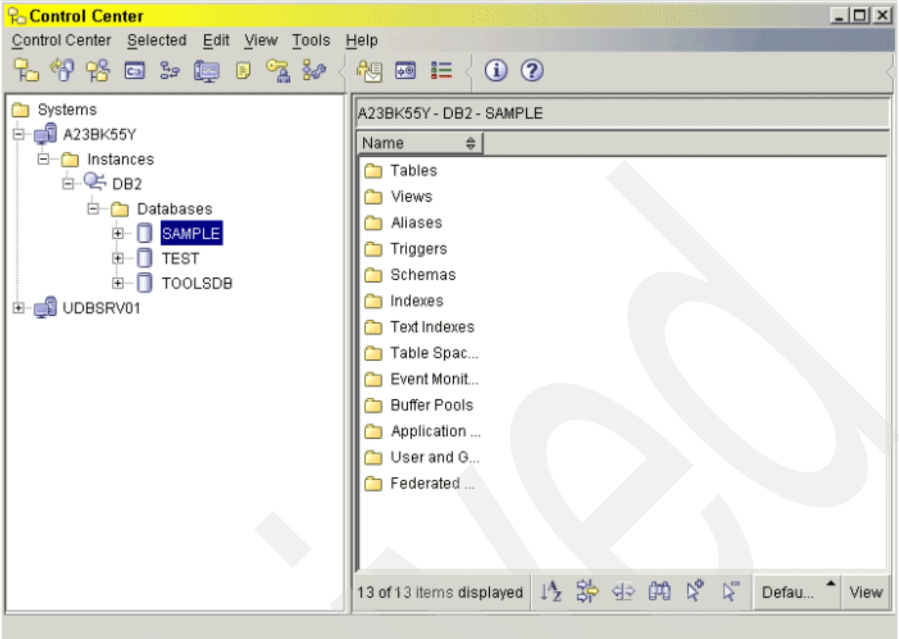


Figure 6-5 Expand directory tree down to the SAMPLE database icon

3. Right-click the **SAMPLE** database icon and select **Configure Parameters...** from the list. A Database Configuration window displays (Figure 6-6).

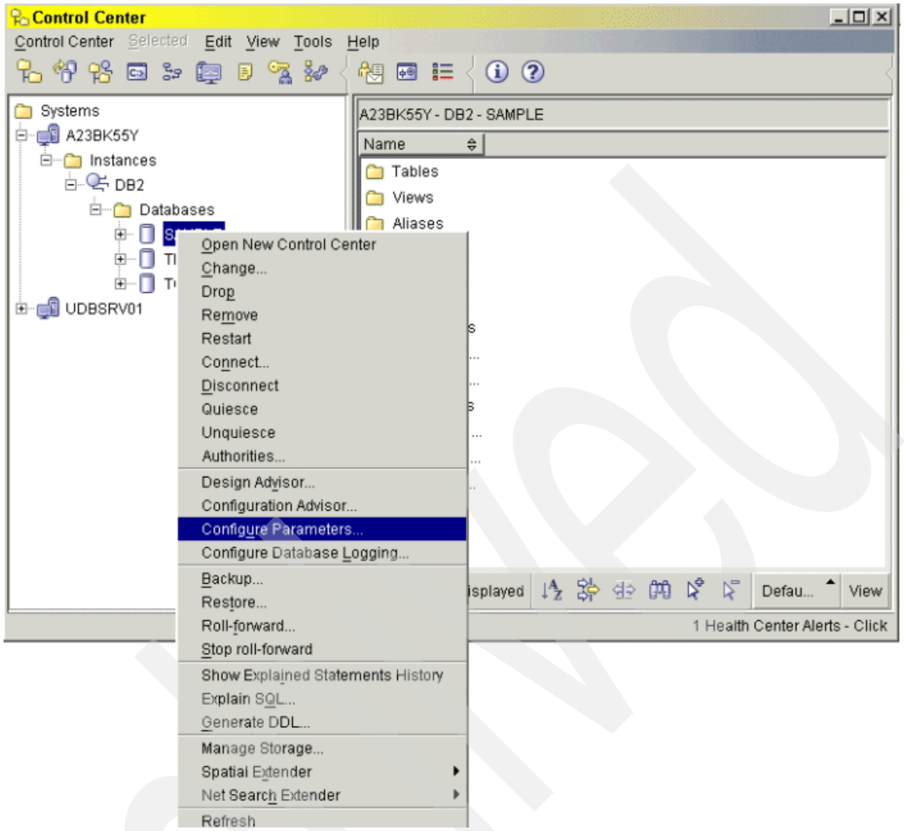
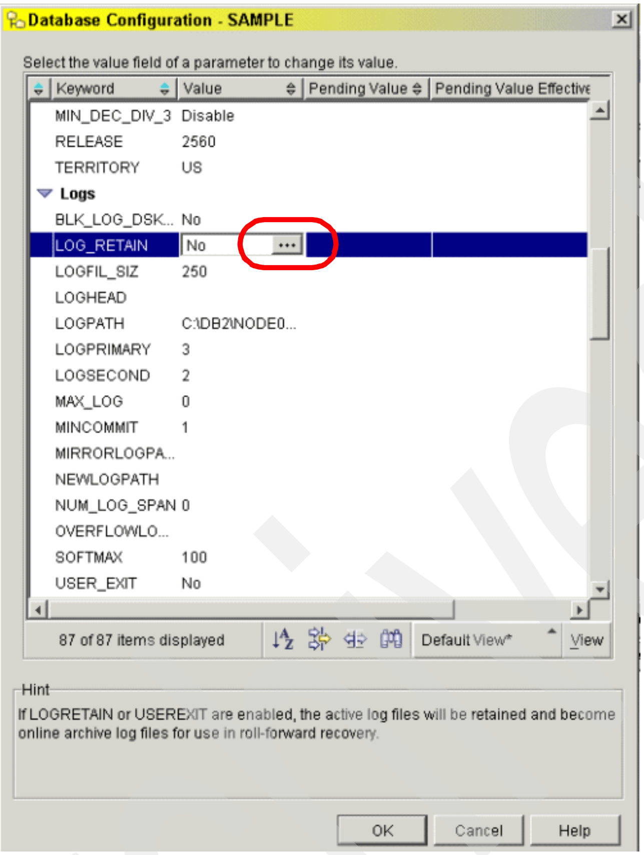


Figure 6-6 Select Configure Parameters

4. Search for LOG_RETAIN in the list of configuration parameters and click **NO**. Then click the**...** button to the right of NO to change the value to Recovery (Figure 6-7).


Database Configuration - SAMPLE

Select the value field of a parameter to change its value.

Keyword	Value	Pending Value	Pending Value Effective
MIN_DEC_DIV_3	Disable		
RELEASE	2560		
TERRITORY	US		
Logs			
BLK_LOG_DSK...	No		
LOG_RETAIN	No ...		
LOGFIL_SIZ	250		
LOGHEAD			
LOGPATH	C:\DB2\NODE0...		
LOGPRIMARY	3		
LOGSECOND	2		
MAX_LOG	0		
MINCOMMIT	1		
MIRRORLOGPA...			
NEWLOGPATH			
NUM_LOG_SPAN	0		
OVERFLOWLO...			
SOFTMAX	100		
USER_EXIT	No		

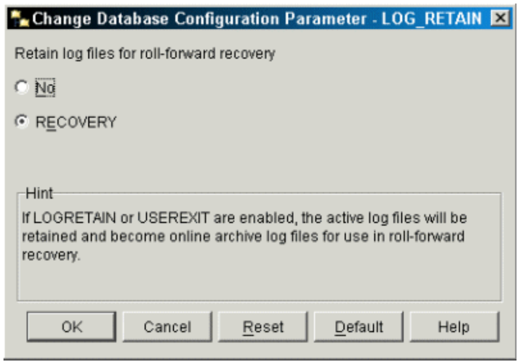
87 of 87 items displayed

Hint

If LOGRETAIN or USEREXIT are enabled, the active log files will be retained and become online archive log files for use in roll-forward recovery.

Figure 6-7 Search for LOG_RETAIN parameter and click NO

5. The Change Database Configuration Parameter.. window appears. Select the **Recovery** radio button and select **OK** (Figure 6-8).


Change Database Configuration Parameter - LOG_RETAIN

Retain log files for roll-forward recovery

○ No

● RECOVERY

Hint

If LOGRETAIN or USEREXIT are enabled, the active log files will be retained and become online archive log files for use in roll-forward recovery.

Figure 6-8 Select RECOVERY radio button

6. Select **OK** from the Database Configuration window and restart the DB2 instance.

Note: After setting LOGRETAIN to RECOVERY, it is necessary for the first database backup to be performed offline. The database cannot be connected until the backup is taken.

Backup database

Two examples of a database backup are provided. Our first backup is an offline backup since it is required immediately after setting LOGRETAIN to RECOVERY. The second backup is an online backup.

Offline database backup

1. From a DOS prompt or terminal window, type **`db2cc`** to bring up the Control Center.
2. On the left pane of Control Center, expand the directory tree down to the SAMPLE database.
3. Right-click the **SAMPLE database icon** and select **Backup** (Figure 6-9).

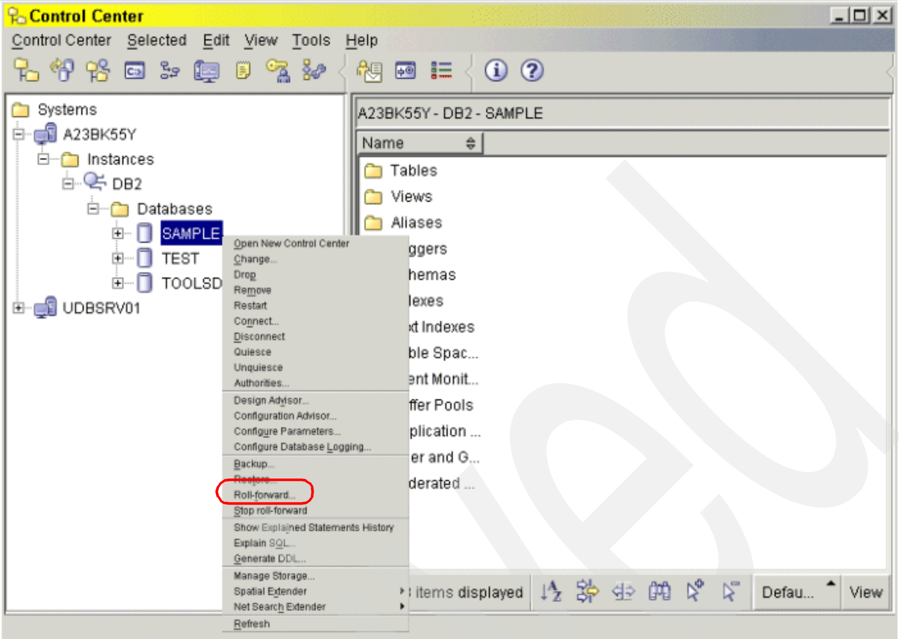
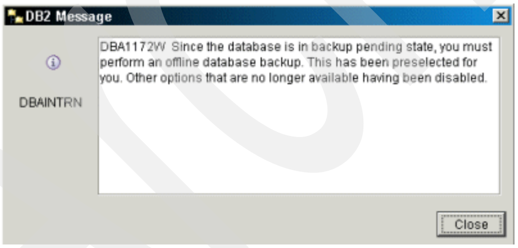


Figure 6-9 Right-click the SAMPLE database icon and select Backup

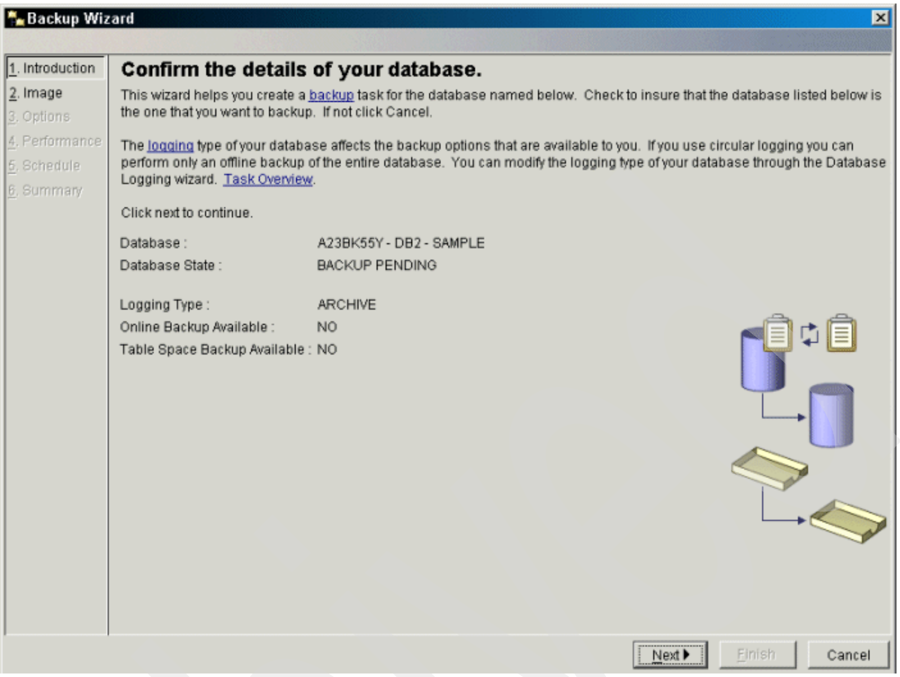
4. A message box pops up stating that an initial offline database backup is necessary after setting LOGRETAIN to RECOVERY (Figure 6-10). Click **Close** to proceed.



DBA1172W Since the database is in backup pending state, you must perform an offline database backup. This has been preselected for you. Other options that are no longer available having been disabled.

Figure 6-10 Click the Close button

5. The Confirm the details of your database window appears. Click **Next** (Figure 6-11).



Confirm the details of your database.

This wizard helps you create a backup task for the database named below. Check to insure that the database listed below is the one that you want to backup. If not click Cancel.

The logging type of your database affects the backup options that are available to you. If you use circular logging you can perform only an offline backup of the entire database. You can modify the logging type of your database through the Database Logging wizard. Task Overview.

Click next to continue.

Database : A23BK55Y - DB2 - SAMPLE

Database State : BACKUP PENDING

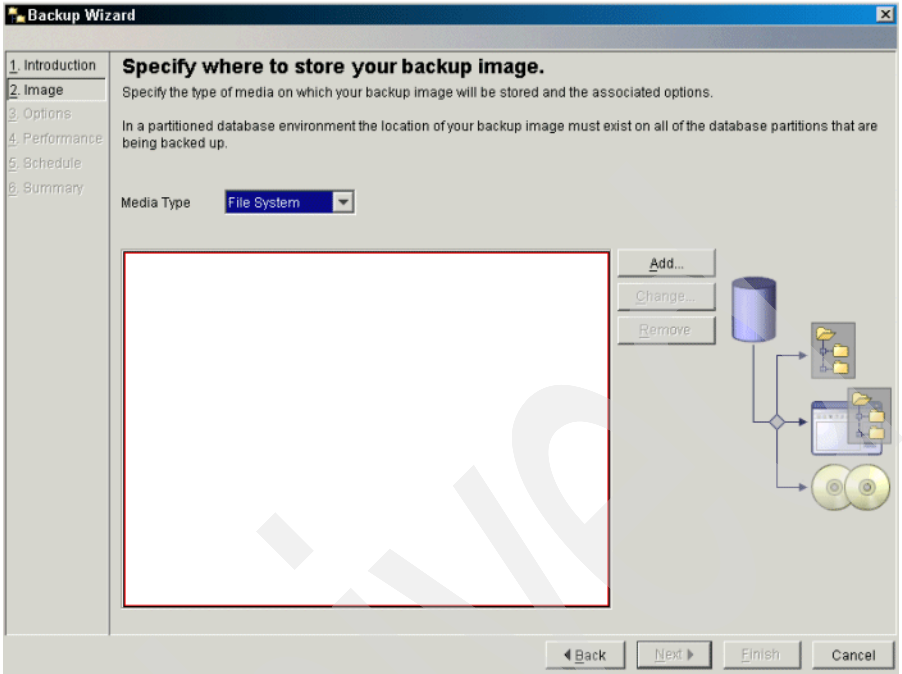
Logging Type : ARCHIVE

Online Backup Available : NO

Table Space Backup Available : NO

Figure 6-11 Confirm the details of your database window

6. The Specify where to store your backup image window appears. Leave the Media Type drop-down box as File System and click **Add** (Figure 6-12).



Specify where to store your backup image.

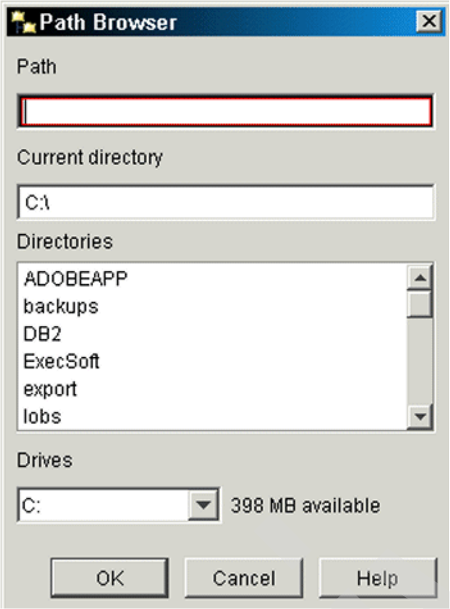
Specify the type of media on which your backup image will be stored and the associated options.

In a partitioned database environment the location of your backup image must exist on all of the database partitions that are being backed up.

Media Type: File System

Figure 6-12 Specify where to store your backup image window

7. A Path Browser window appears (Figure 6-13). You can backup the database to a file or other media such as a tape. The selected backup media has to be local to the server. Select a file system and, optionally, a directory to store the database backup image. Click **OK** to go back to the Specify where to store your backup image window. Click **Next** to continue.



Path

Current directory

C:\

Directories

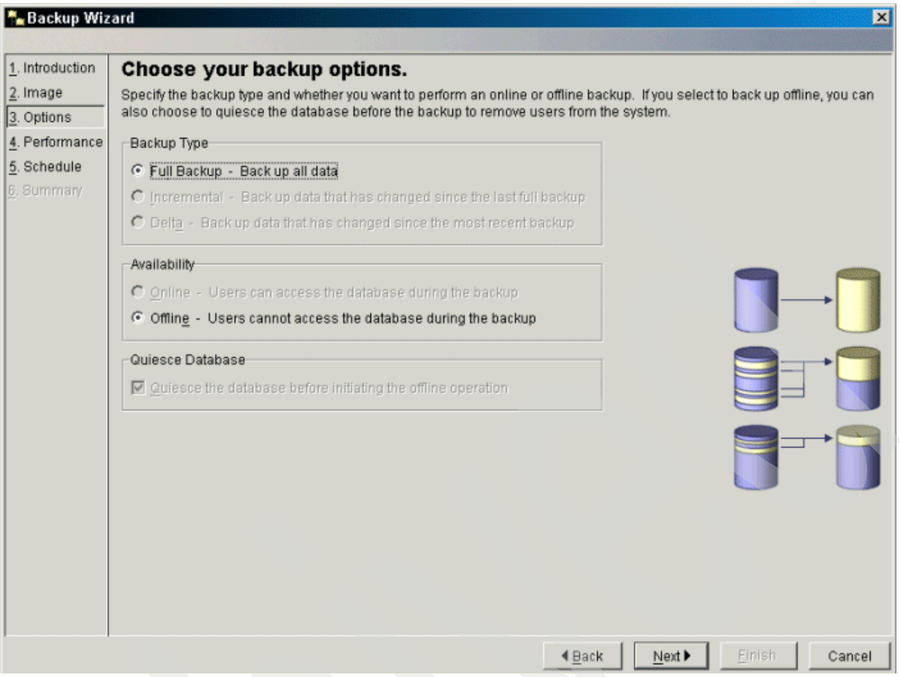
ADOBEAPP
backups
DB2
ExecSoft
export
lobs

Drives

C: 398 MB available

Figure 6-13 The Path Browser window

8. The Choose your backup options window appears (Figure 6-14). Select the **Full Backup** radio button. Select the **Offline** radio button. Leave the *Quiesce the database before initiating the offline operation* check box checked. Click **Next**.



Backup Wizard

Choose your backup options.

Specify the backup type and whether you want to perform an online or offline backup. If you select to back up offline, you can also choose to quiesce the database before the backup to remove users from the system.

Backup Type

● Full Backup - Back up all data
○ Incremental - Back up data that has changed since the last full backup
○ Delta - Back up data that has changed since the most recent backup

Availability

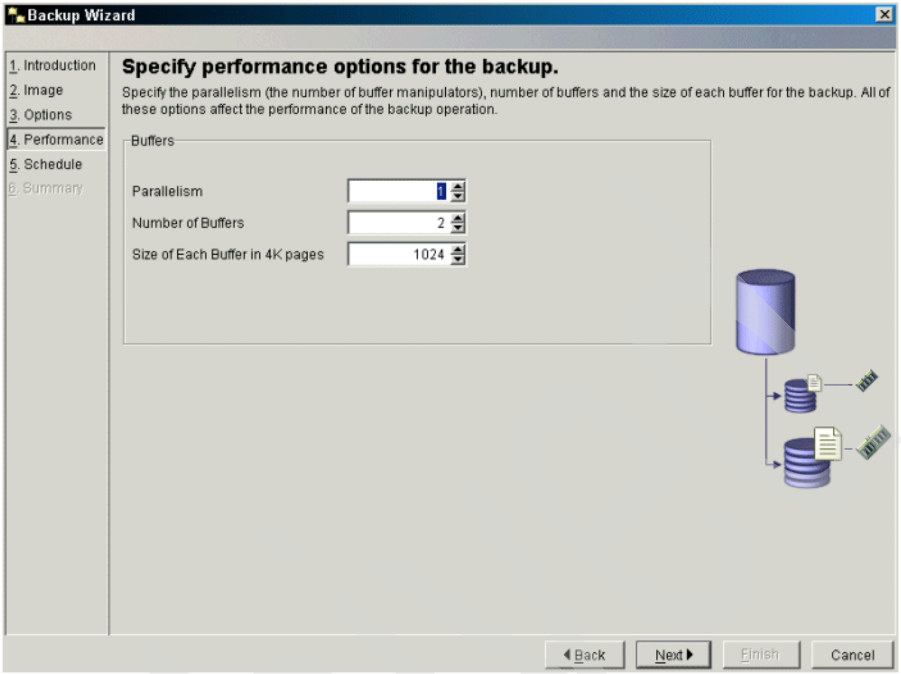
○ Online - Users can access the database during the backup
● Offline - Users cannot access the database during the backup

Quiesce Database

☑ Quiesce the database before initiating the offline operation

Figure 6-14 Choose your backup options window

9. The Specify performance options window appears (Figure 6-15). Leave this as is. Click **Next**.



Specify performance options for the backup.

Specify the parallelism (the number of buffer manipulators), number of buffers and the size of each buffer for the backup. All of these options affect the performance of the backup operation.

Buffers

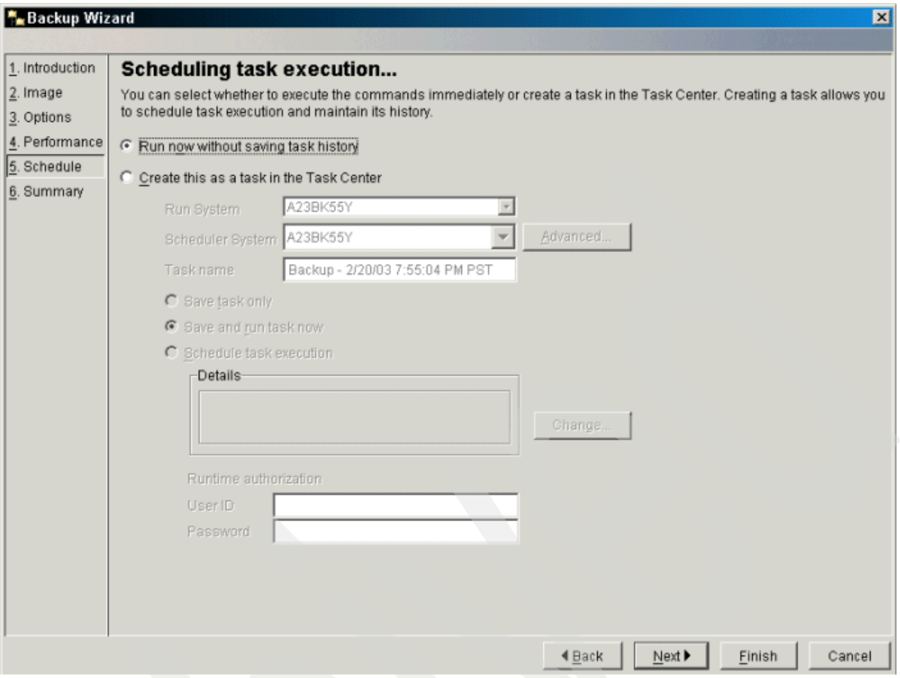
Parallelism: 1

Number of Buffers: 2

Size of Each Buffer in 4K pages: 1024

Figure 6-15 Specify performance options for the backup window

10. The Scheduling task execution window appears (Figure 6-16). Select the **Run now without saving task history** radio button. Click **Next**.



Scheduling task execution...

You can select whether to execute the commands immediately or create a task in the Task Center. Creating a task allows you to schedule task execution and maintain its history.

● Run now without saving task history

○ Create this as a task in the Task Center

Run System: A23BK55Y

Scheduler System: A23BK55Y

Task name: Backup - 2/20/03 7:55:04 PM PST

○ Save task only

● Save and run task now

○ Schedule task execution

Details

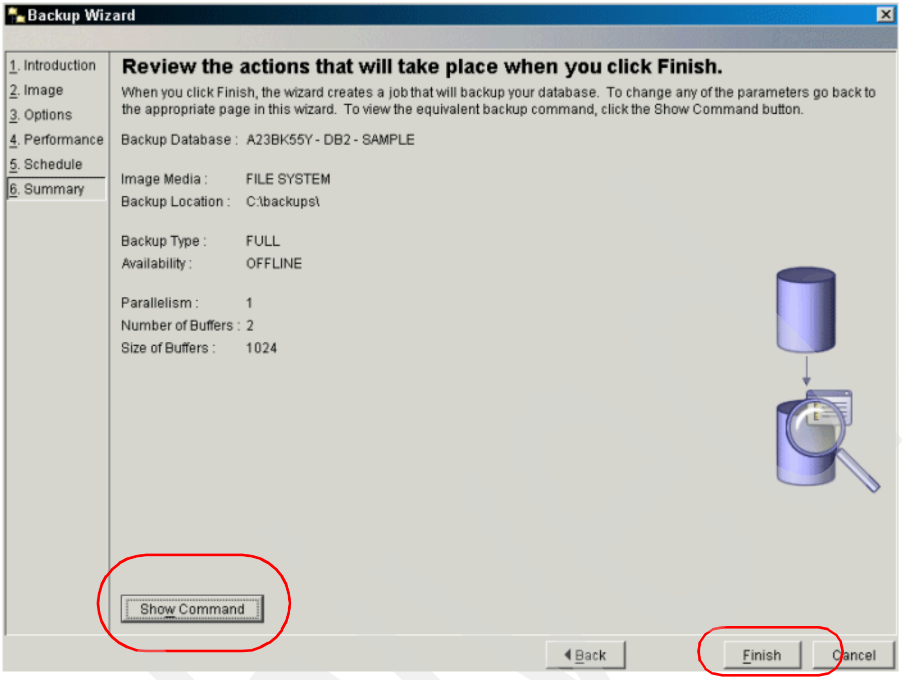
Runtime authorization

User ID

Password

Figure 6-16 Scheduling task execution window

11. A summary of actions for your database backup appears (Figure 6-17). Click the **Show Command** button to view the syntax that will be used to execute the backup. Click **Finish** to immediately execute the backup task.



Review the actions that will take place when you click Finish.

When you click Finish, the wizard creates a job that will backup your database. To change any of the parameters go back to the appropriate page in this wizard. To view the equivalent backup command, click the Show Command button.

Backup Database : A23BK55Y - DB2 - SAMPLE

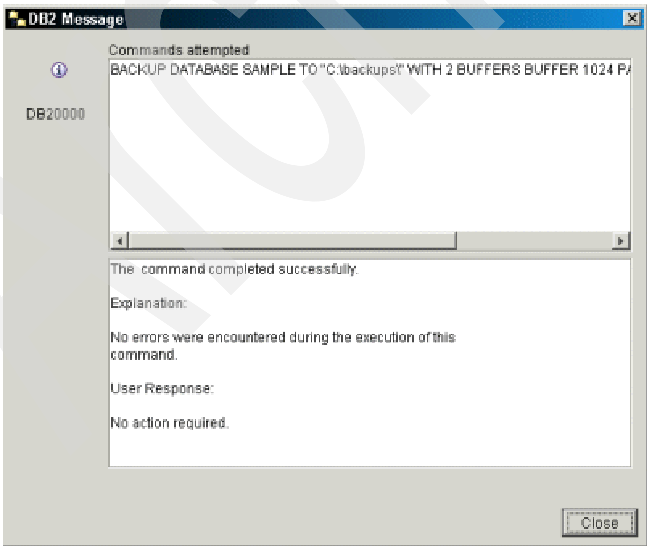
Image Media : FILE SYSTEM
Backup Location : C:\backups\

Backup Type : FULL
Availability : OFFLINE

Parallelism : 1
Number of Buffers : 2
Size of Buffers : 1024

Figure 6-17 Backup wizard: Review the actions window

12.A window appears showing success of the execution (Figure 6-18).



Commands attempted

```
BACKUP DATABASE SAMPLE TO "C:\backups\" WITH 2 BUFFERS BUFFER 1024 PA
```

The command completed successfully.

Explanation:

No errors were encountered during the execution of this command.

User Response:

No action required.

Figure 6-18 Success message window

Online database backup

In the following example, the SAMPLE database is backed up to a specific directory. Since our scenario has users connected to the database, you will perform an online backup. Here are the steps to backup your database online:

1. From a DB2 CLP or terminal window, type `db2cc` to bring up the Control Center.
2. On the left pane of Control Center, expand the **directory tree** down to the SAMPLE database.
3. Right-click the **SAMPLE database icon** and select **Backup** (Figure 6-19).

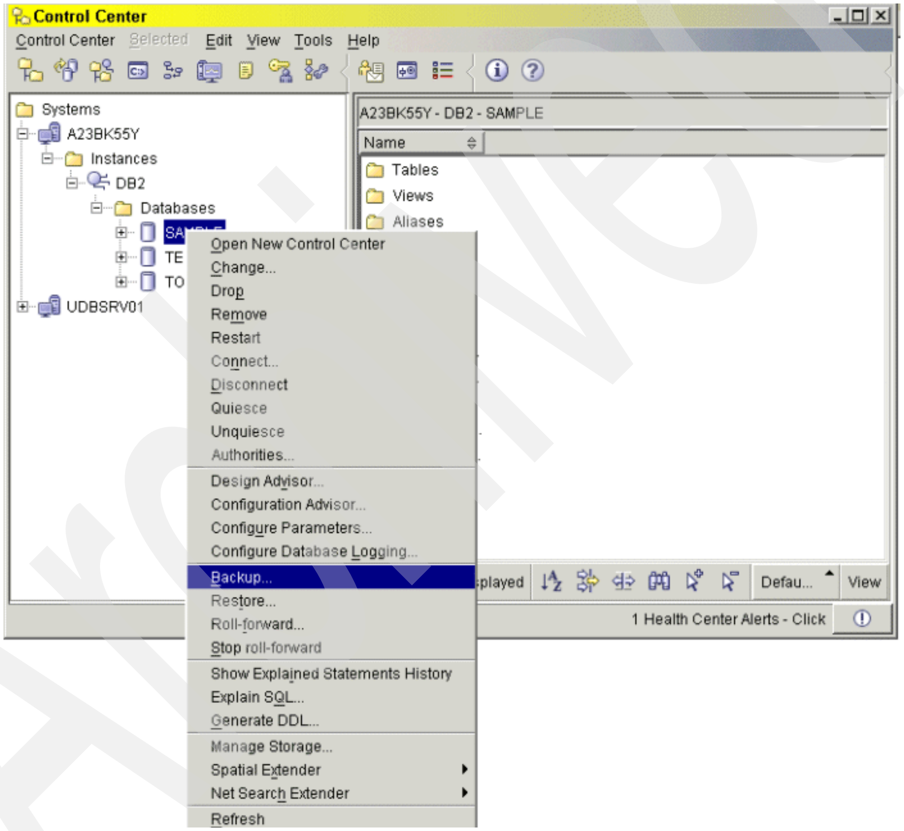
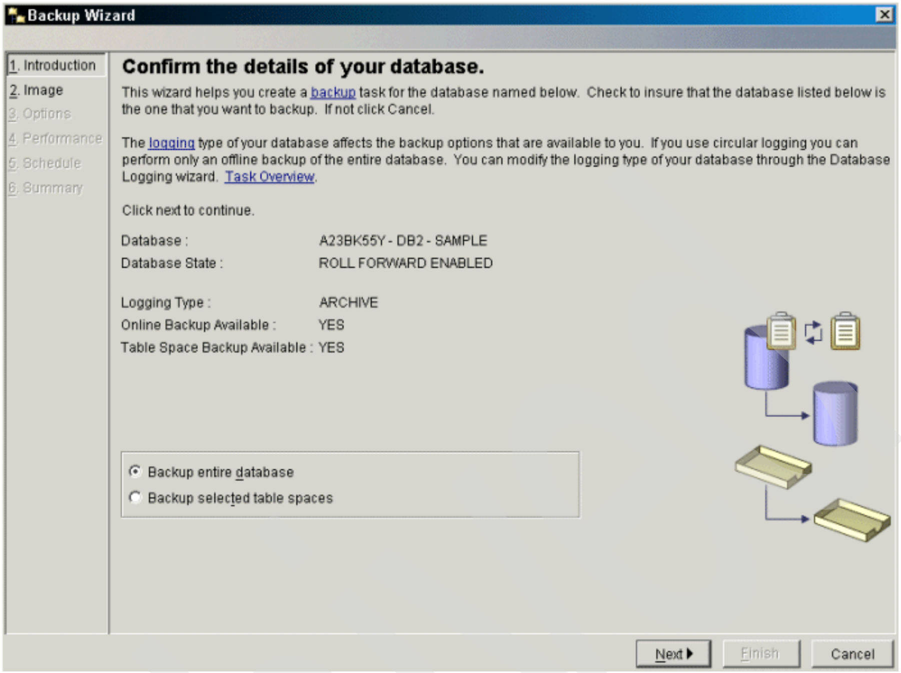


Figure 6-19 Right-click the SAMPLE database icon and select Backup

4. The Confirm the details of your database window appears. Select the **Backup entire database** radio button and click **Next** (Figure 6-20).



Backup Wizard

Confirm the details of your database.

This wizard helps you create a backup task for the database named below. Check to insure that the database listed below is the one that you want to backup. If not click Cancel.

The logging type of your database affects the backup options that are available to you. If you use circular logging you can perform only an offline backup of the entire database. You can modify the logging type of your database through the Database Logging wizard. Task Overview.

Click next to continue.

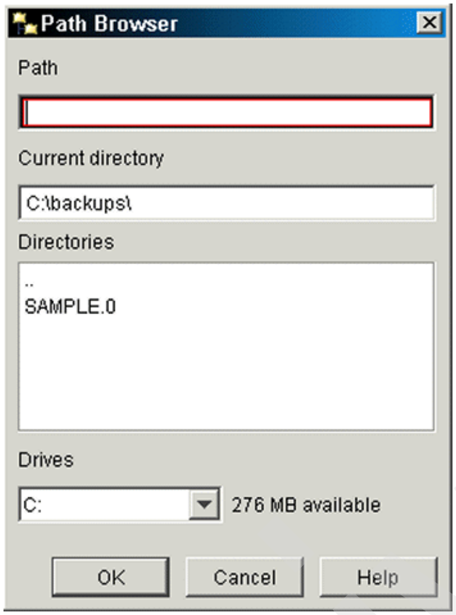
Database : A23BK55Y - DB2 - SAMPLE
Database State : ROLL FORWARD ENABLED

Logging Type : ARCHIVE
Online Backup Available : YES
Table Space Backup Available : YES

● Backup entire database
○ Backup selected table spaces

Figure 6-20 Confirm the details of your database window

5. The Specify where to store your backup image window appears. Leave the Media Type drop-down box as File System and click **Add**.
6. A Path Browser window appears (Figure 6-21). Select a file system and, optionally, a directory to store the database backup image. Click **OK** to go back to the Specify where to store your backup image window. Click **Next** to continue.



Path

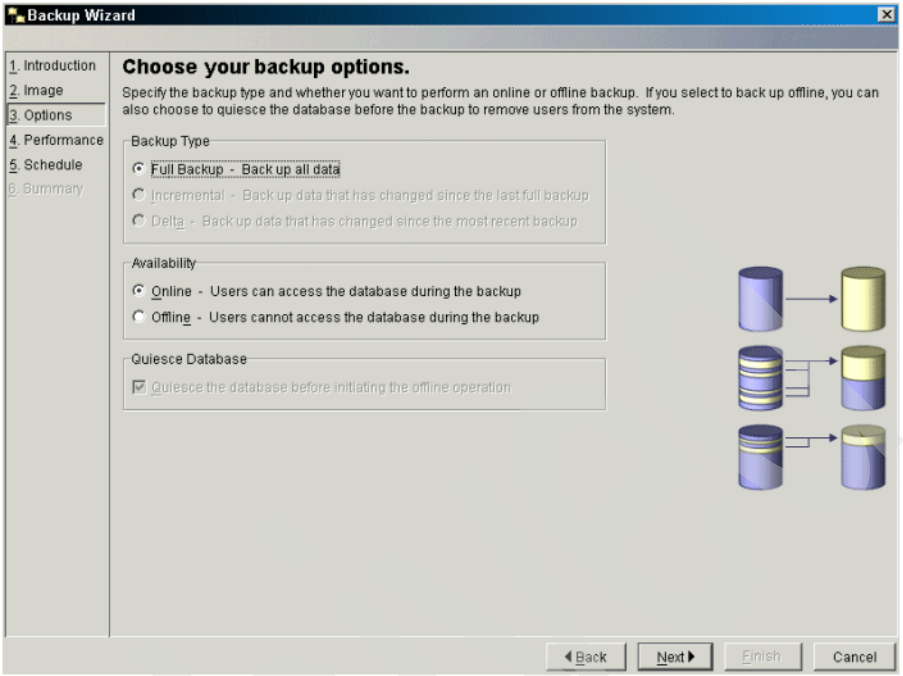
Current directory
C:\backups\

Directories
..
SAMPLE.0

Drives
C: 276 MB available

Figure 6-21 Path Browser window

7. The Choose your backup options window appears (Figure 6-22). Select the **Full Backup** radio button. Select the **Online** radio button. Click **Next**.



Backup Wizard

1. Introduction
2. Image
3. Options
4. Performance
5. Schedule
6. Summary

Choose your backup options.

Specify the backup type and whether you want to perform an online or offline backup. If you select to back up offline, you can also choose to quiesce the database before the backup to remove users from the system.

Backup Type

● Full Backup - Back up all data
○ Incremental - Back up data that has changed since the last full backup
○ Delta - Back up data that has changed since the most recent backup

Availability

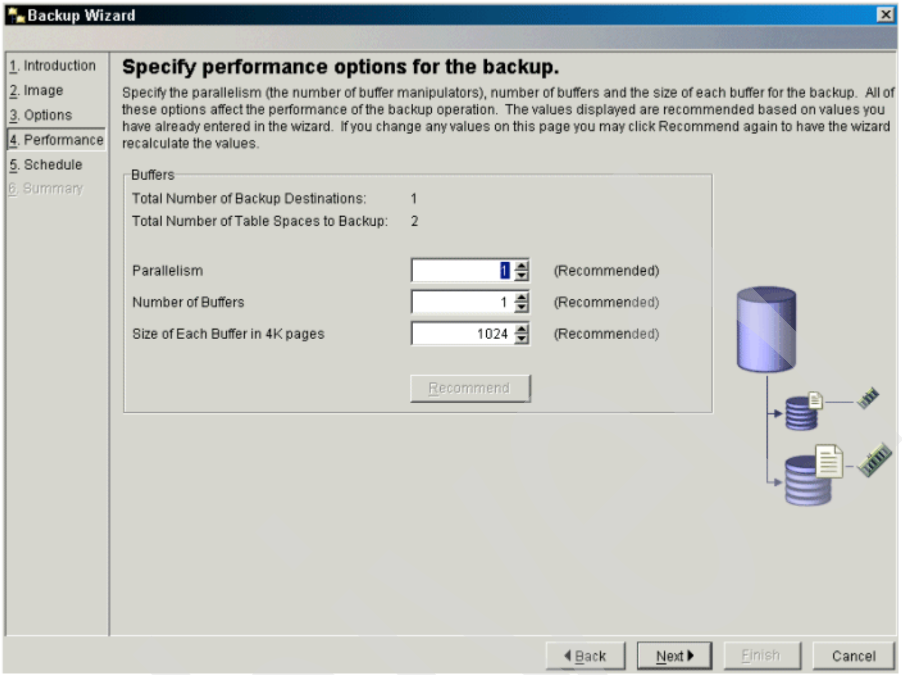
● Online - Users can access the database during the backup
○ Offline - Users cannot access the database during the backup

Quiesce Database

☑ Quiesce the database before initiating the offline operation

Figure 6-22   Choose your backup options window

8. The Specify performance options window appears (Figure 6-23). Leave this as is. Click **Next**.



Specify performance options for the backup.

Specify the parallelism (the number of buffer manipulators), number of buffers and the size of each buffer for the backup. All of these options affect the performance of the backup operation. The values displayed are recommended based on values you have already entered in the wizard. If you change any values on this page you may click Recommend again to have the wizard recalculate the values.

Buffers

Total Number of Backup Destinations: 1

Total Number of Table Spaces to Backup: 2

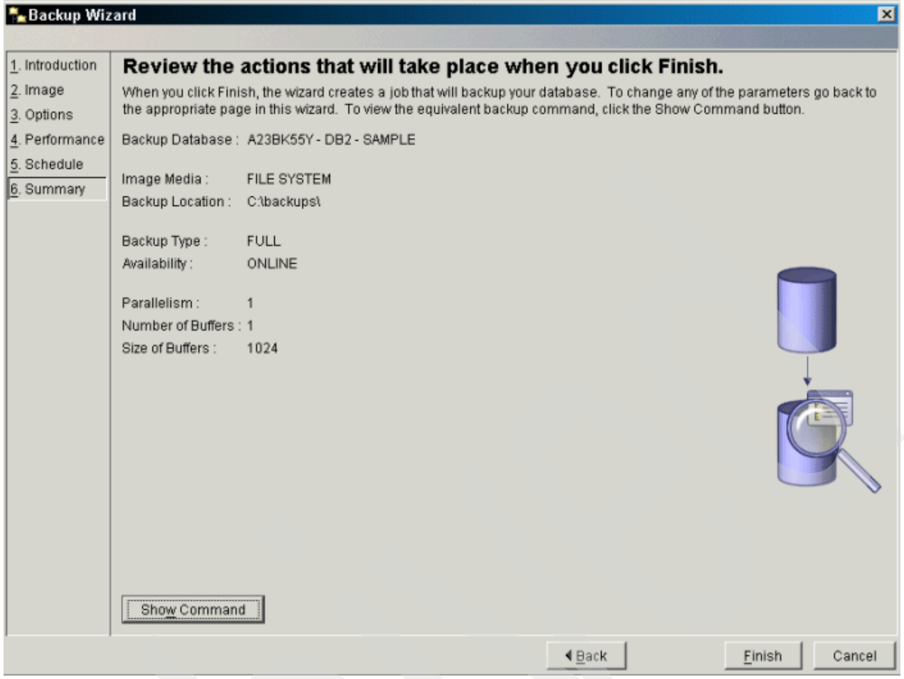
Parallelism: 1 (Recommended)

Number of Buffers: 1 (Recommended)

Size of Each Buffer in 4K pages: 1024 (Recommended)

Figure 6-23 Specify performance options for the backup window

9. The Scheduling task execution window appears. Select the **Run now without saving task history** radio button. Click **Next**.
10. A summary of actions for your database backup appears (Figure 6-24). Click the **Show Command** button to view the syntax that will be used to execute the backup. Click **Finish** to immediately execute the backup task.



Review the actions that will take place when you click Finish.

When you click Finish, the wizard creates a job that will backup your database. To change any of the parameters go back to the appropriate page in this wizard. To view the equivalent backup command, click the Show Command button.

Backup Database : A23BK55Y - DB2 - SAMPLE

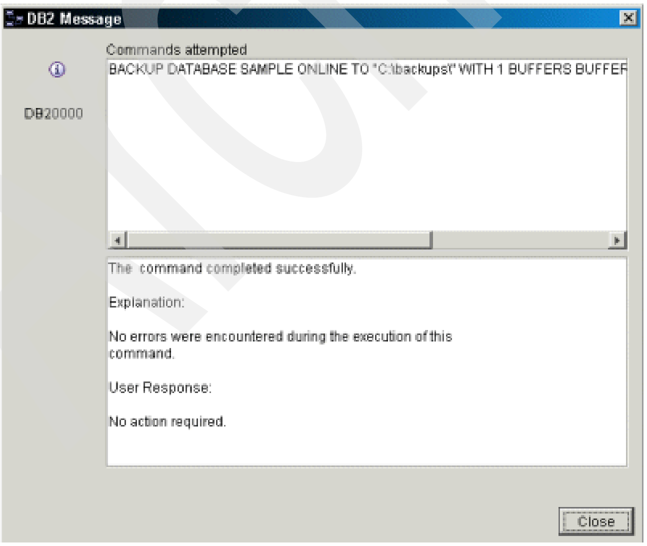
Image Media : FILE SYSTEM
Backup Location : C:\backups\

Backup Type : FULL
Availability : ONLINE

Parallelism : 1
Number of Buffers : 1
Size of Buffers : 1024

Figure 6-24 Review the actions that will take place when you click Finish window

11.A success message appears showing success of the execution (Figure 6-25).



Commands attempted

```
BACKUP DATABASE SAMPLE ONLINE TO "C:\backups\" WITH 1 BUFFERS BUFFER
```

The command completed successfully.

Explanation:

No errors were encountered during the execution of this command.

User Response:

No action required.

Figure 6-25 Success message window

Backup image naming conventions

Once you have backed up your database, a file will be created on Linux with a naming convention similar to the following:

```
SAMPLE.0.db2inst1.NODE0000.CATN0000.20030217161834.001
```

On Windows, a directory structure will be created with the last part of the timestamp containing the image:

```
SAMPLE.0/DB2/NODE0000/CATN0000/20030217/161834.001
```

Other backup considerations

To protect your database backup image, it is always a good idea to keep the database backup image and a copy of your log files on a different machine or removable media such as tape. DB2 is integrated with IBM Tivoli® Storage Management (TSM) system which is an excellent database backup method.

The default location of your log files depends on the location of your database directory. On Linux, the default database directory resides in your instance owner's home directory. On Windows, the default database directory resides on the same file system as your DB2 installation. Your log files are located by default, within your database directory in the following path:

```
<instance name>/NODE0000/SQL0000../SQLOGDIR
```

Where, SQL0000.. is the database directory and SQLOGDIR holds the log files. SQLOGDIR is the active log directory and therefore, you should archive off your log files frequently with a user exit program or manually copy the log files to another location. Log files are required for database recovery if the database is backup online and archiving logging is used.

In addition to backing up the entire database, you can backup tablespaces. Backing up selected tablespaces can save time and storage space.

6.3.2 Database recovery

If your database becomes unusable, you may need to use the **`restore`** and **`rollforward`** utilities to be back in business. If you backup your database offline, a simple database restore will get your database running. If you backup your database online, a roll forward operation of the log files is mandatory. It allows you to recovery your database to the point-in-time of the database failure.

Database restore and roll forward

Restoring a database involves taking the database backup image and applying the database files in the backup image to an existing or new database. Once the restore is successful, log files recreated during or later than the backup image can be re-applied with the **rollforward** command to bring the database to a state near the time of the database failure.

In order to restore a database, you will need the database backup image created from the **backup** command. In a real-life scenario, it is likely that there will be more than one database backup image in your backup directory. If there is more than one backup image in your database directory, it is necessary to specify the backup image to be restored. Since each backup image has a timestamp associated with it, determining the latest backup image is easily identifiable.

The following steps show how to use the Control Center to perform a database restore:

1. From a DB2 CLP or terminal window, type **db2cc** to bring up the Control Center.
2. On the left pane of Control Center, expand the **directory tree** down to the SAMPLE database.
3. Right-click the **SAMPLE database icon** and select **Restore** (Figure 6-26).

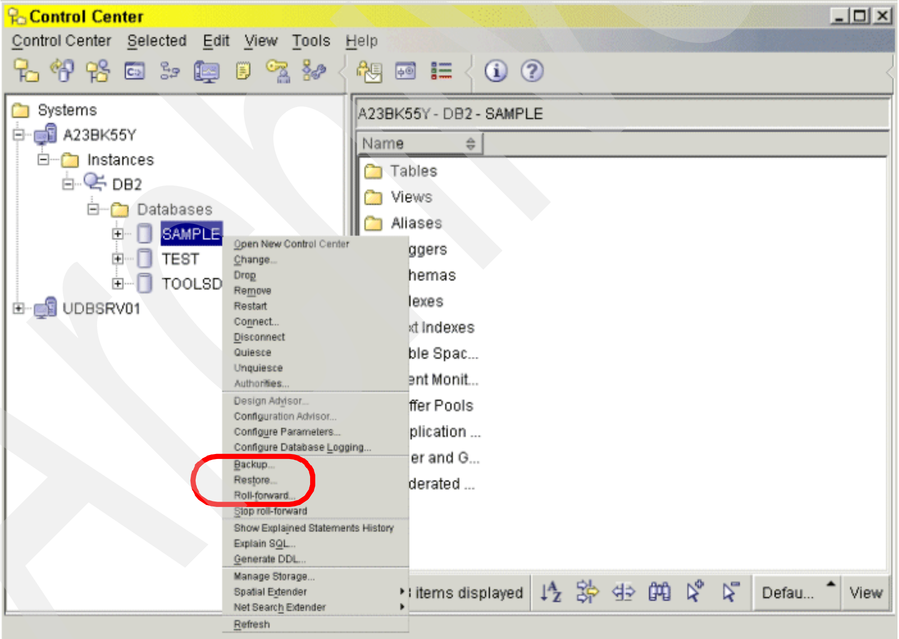
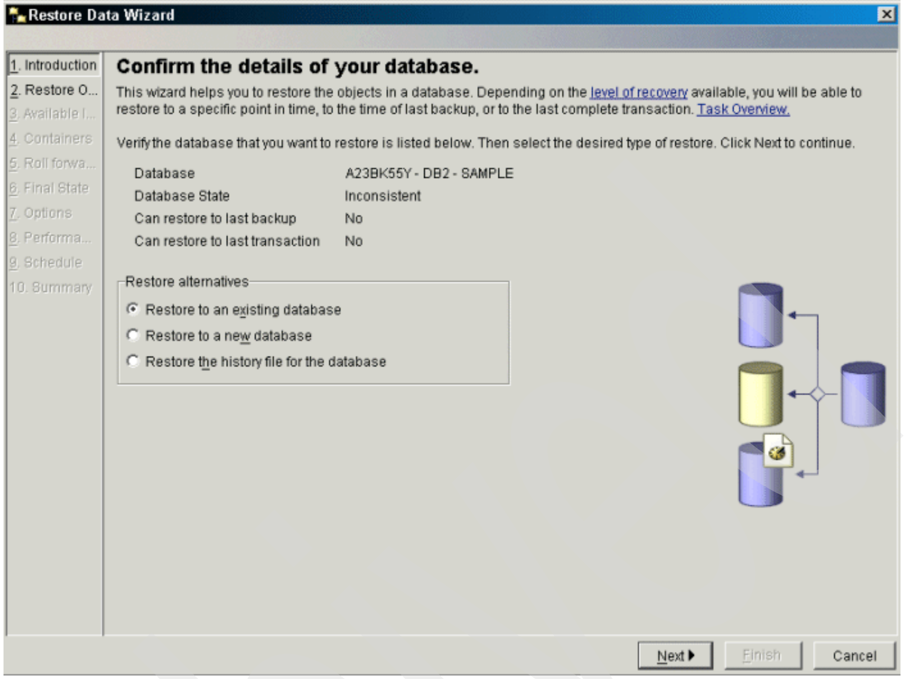


Figure 6-26 Select Restore database option

4. The Confirm the details of your database window appears (Figure 6-27). Leave the *Restore to an existing database* radio button selected and click **Next**.



Confirm the details of your database.

This wizard helps you to restore the objects in a database. Depending on the level of recovery available, you will be able to restore to a specific point in time, to the time of last backup, or to the last complete transaction. Task Overview.

Verify the database that you want to restore is listed below. Then select the desired type of restore. Click Next to continue.

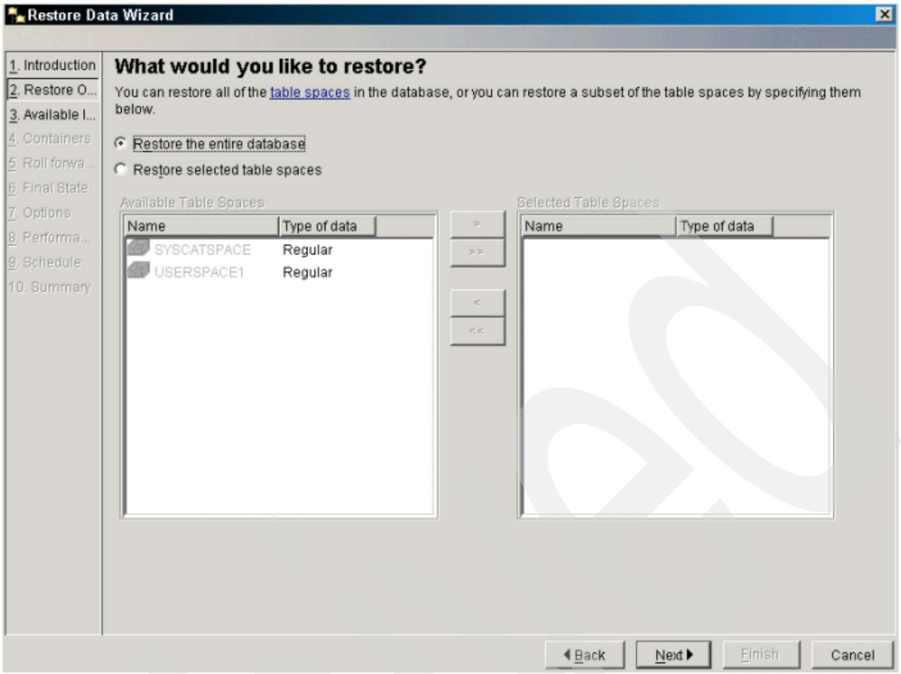
Database	A23BK55Y - DB2 - SAMPLE
Database State	Inconsistent
Can restore to last backup	No
Can restore to last transaction	No

Restore alternatives

- ● Restore to an existing database
- ○ Restore to a new database
- ○ Restore the history file for the database

Figure 6-27 Confirm the details of your database window

5. The following window asks what you would like to restore (Figure 6-28). Leave the *Restore the entire database* radio button selected and click **Next**.



Restore Data Wizard

1. Introduction
2. Restore O...
3. Available I...
4. Containers
5. Roll forwa...
6. Final State
7. Options
8. Performa...
9. Schedule
10. Summary

What would you like to restore?

You can restore all of the table spaces in the database, or you can restore a subset of the table spaces by specifying them below.

● Restore the entire database

○ Restore selected table spaces

Available Table Spaces

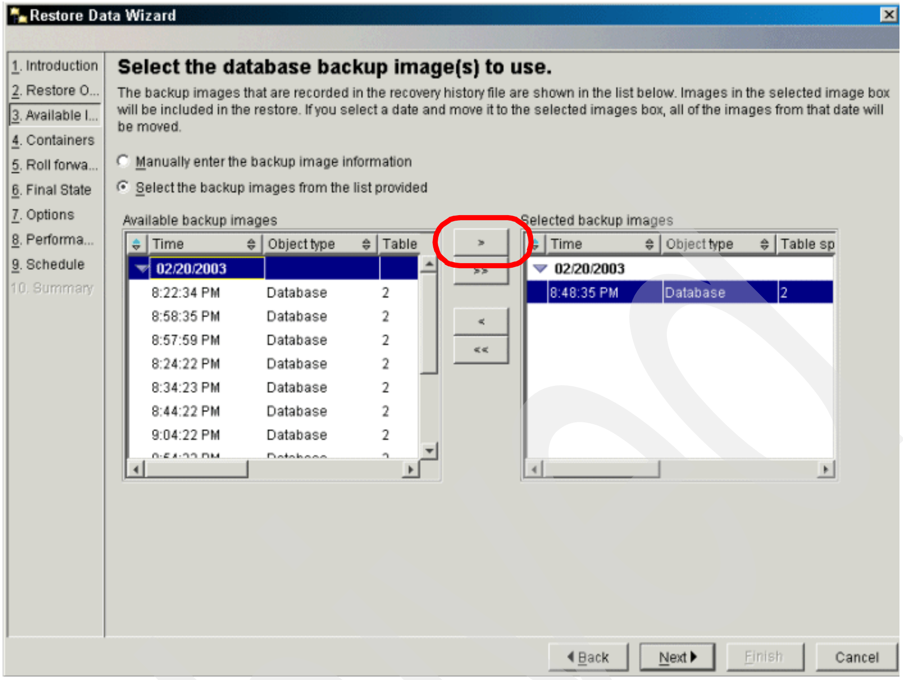
Name	Type of data
SYSCATSPACE	Regular
USERSPACE1	Regular

Selected Table Spaces

Name	Type of data

Figure 6-28 What would you like to restore? window

6. The following window asks for the database image to restore (Figure 6-29). The left pane shows the available database images to restore. Select the online backup image. If you have followed the examples in this chapter, it should be the most recent backup image. Use the arrow button in the center of the window to move the selected image name to the right pane. Click **Next**.



Select the database backup image(s) to use.

The backup images that are recorded in the recovery history file are shown in the list below. Images in the selected image box will be included in the restore. If you select a date and move it to the selected images box, all of the images from that date will be moved.

○ Manually enter the backup image information

● Select the backup images from the list provided

Available backup images

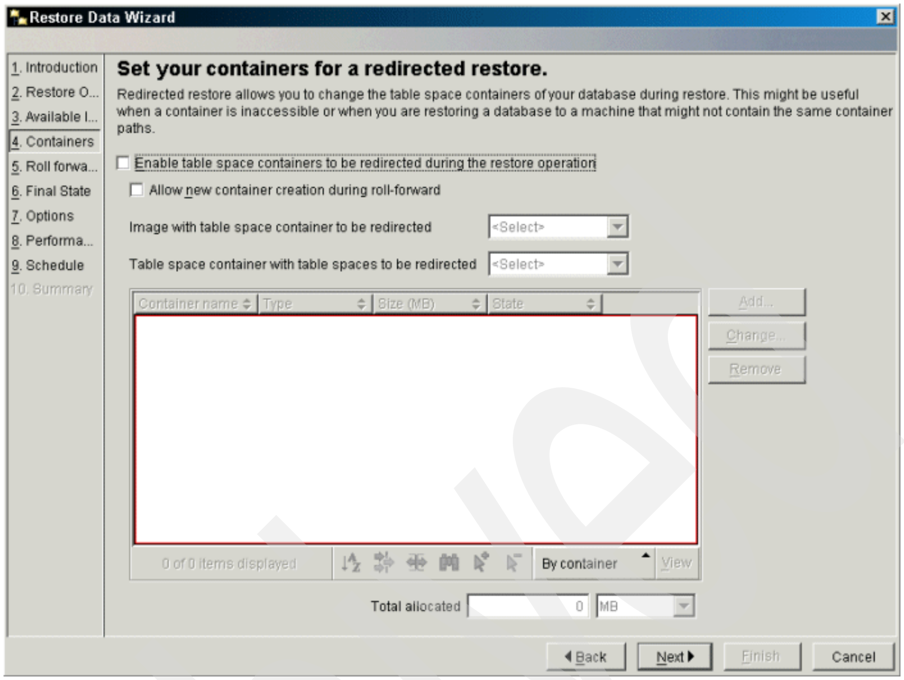
Time	Object type	Table
02/20/2003		
8:22:34 PM	Database	2
8:58:35 PM	Database	2
8:57:59 PM	Database	2
8:24:22 PM	Database	2
8:34:23 PM	Database	2
8:44:22 PM	Database	2
9:04:22 PM	Database	2

Selected backup images

Time	Object type	Table sp
02/20/2003		
8:48:35 PM	Database	2

Figure 6-29 Select the database backup image(s) to use window

7. The following window asks if you would like to set your containers for a redirected restore (Figure 6-30). Since we are not performing a redirected restore, leave all check boxes unchecked and click **Next**.



Set your containers for a redirected restore.

Redirected restore allows you to change the table space containers of your database during restore. This might be useful when a container is inaccessible or when you are restoring a database to a machine that might not contain the same container paths.

□ Enable table space containers to be redirected during the restore operation

□ Allow new container creation during roll-forward

Image with table space container to be redirected <Select>

Table space container with table spaces to be redirected <Select>

Container name	Type	Size (MB)	State

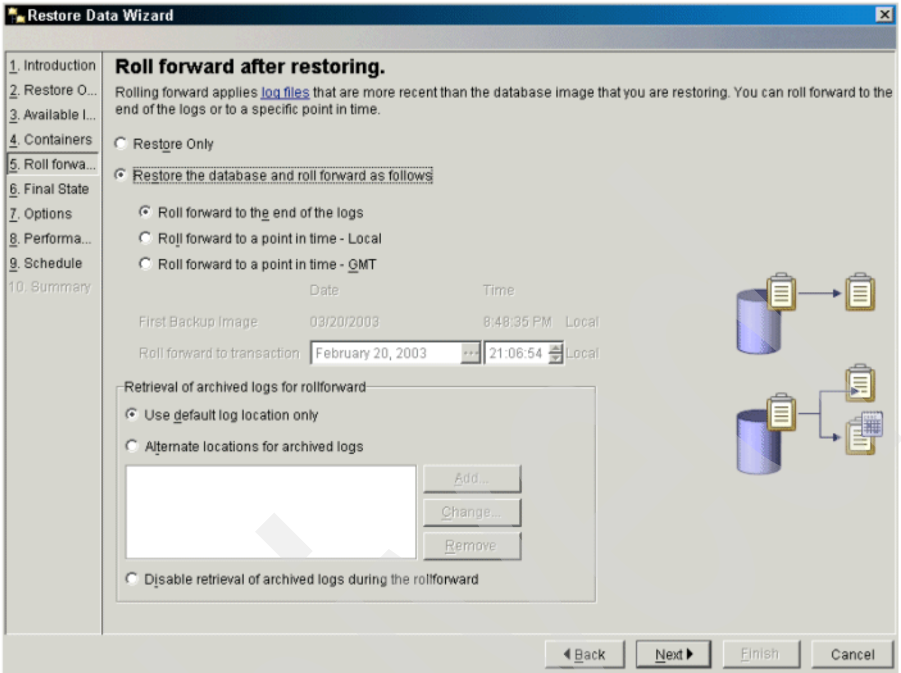
0 of 0 items displayed

By container

Total allocated: 0 MB

Figure 6-30 Set your containers for a redirected restore window

8. The following window allows you to set up a roll forward operation after the restore (Figure 6-31). Select the **Restore the database and roll forward as follows** radio button and leave the Rollforward to the end of logs radio button selected. Since our log files are stored in the default location (that is, we did not specify a different location to store the log files), we will leave the Use default log location only radio button selected. Click **Next**.



Restore Data Wizard

1. Introduction
2. Restore O...
3. Available I...
4. Containers
5. Roll forwa...
6. Final State
7. Options
8. Performa...
9. Schedule
10. Summary

Roll forward after restoring.

Rolling forward applies log files that are more recent than the database image that you are restoring. You can roll forward to the end of the logs or to a specific point in time.

○ Restore Only

● Restore the database and roll forward as follows

● Roll forward to the end of the logs

○ Roll forward to a point in time - Local

○ Roll forward to a point in time - GMT

	Date	Time	
First Backup Image	03/20/2003	8:48:35 PM	Local
Roll forward to transaction	February 20, 2003	21:06:54	Local

Retrieval of archived logs for rollforward

● Use default log location only

○ Alternate locations for archived logs

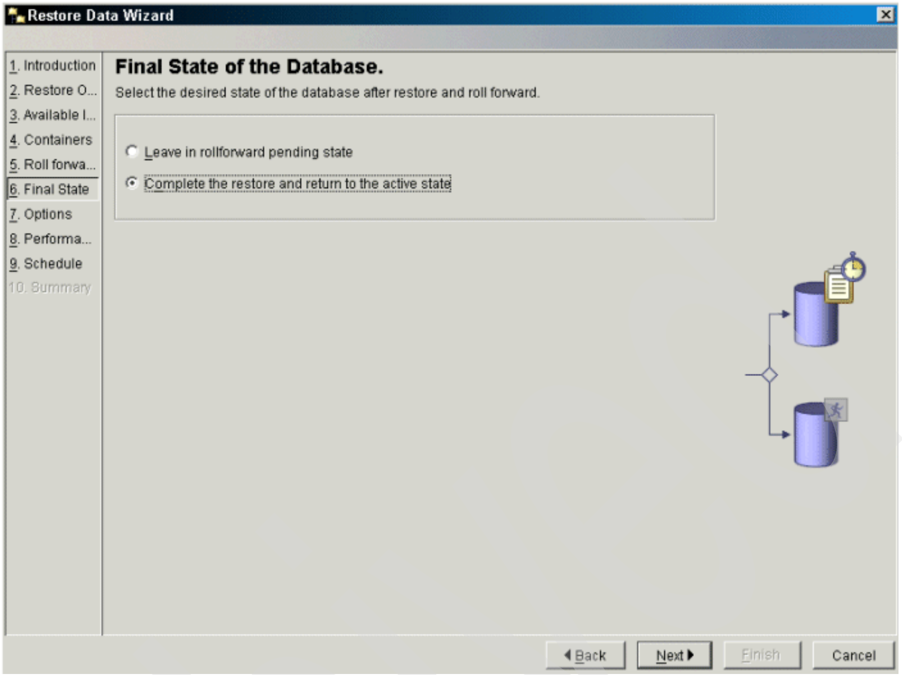
Add... Change... Remove

○ Disable retrieval of archived logs during the rollforward

Back Next Finish Cancel

Figure 6-31 Rollforward after restoring window

9. The following window asks if you would like to leave the database in roll forward pending state after the recovery operation is complete (Figure 6-32). Since all of the log files will be rolled forward at the end of our recovery operation, select the **Complete the restore and return to the active state** radio button. Click **Next**.



Restore Data Wizard

1. Introduction
2. Restore O...
3. Available I...
4. Containers
5. Roll forwa...
6. Final State
7. Options
8. Performa...
9. Schedule
10. Summary

Final State of the Database.

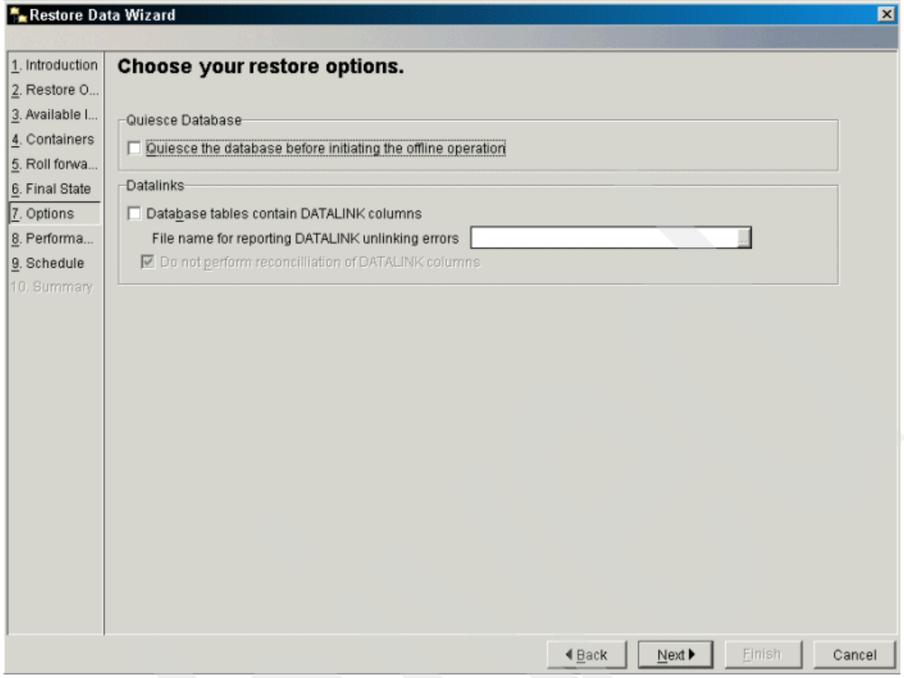
Select the desired state of the database after restore and roll forward.

○ Leave in rollforward pending state

● Complete the restore and return to the active state

Figure 6-32 Final State of the Database window

10. The following window gives you the options of quiescing the database to prevent users from connecting to the database immediately after the restore (Figure 6-33). Leave the *Quiesce the database before initiating the offline operation* check box unchecked for this operation since we do not have any users trying to access the database. Leave the *Database tables contain DATALINK columns* check box unchecked since we do not have datalink columns in our tables. Click **Next**.



Choose your restore options.

Quiesce Database

☐ Quiesce the database before initiating the offline operation

Datalinks

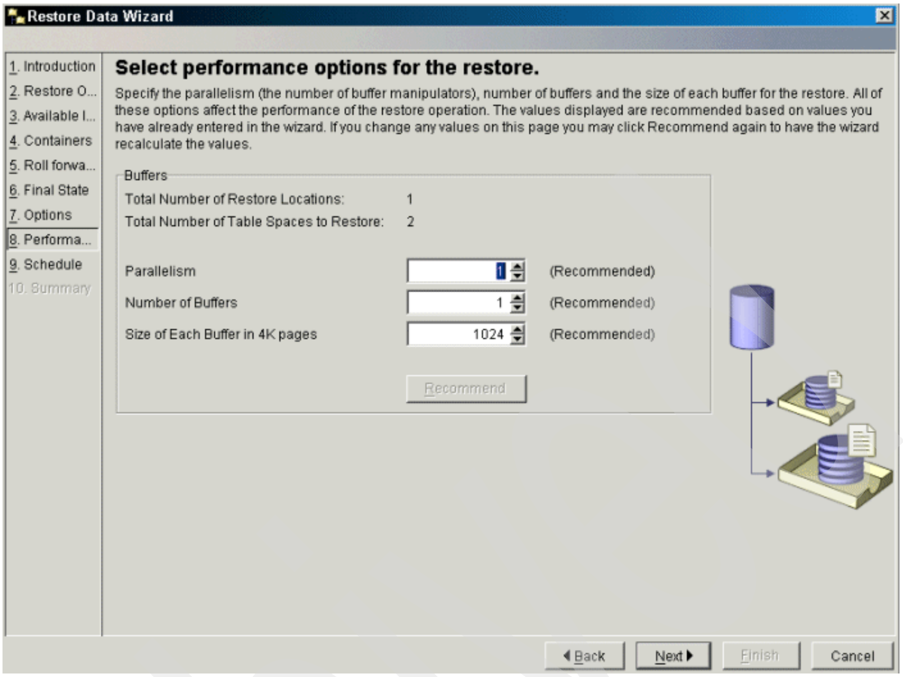
☐ Database tables contain DATALINK columns

File name for reporting DATALINK unlinking errors

☑ Do not perform reconcilliation of DATALINK columns

Figure 6-33 Choose your restore options window

11. The following window asks for performance options related to the restore operation (Figure 6-34). Leave the current recommendations selected and click **Next**.



Select performance options for the restore.

Specify the parallelism (the number of buffer manipulators), number of buffers and the size of each buffer for the restore. All of these options affect the performance of the restore operation. The values displayed are recommended based on values you have already entered in the wizard. If you change any values on this page you may click Recommend again to have the wizard recalculate the values.

Buffers

Total Number of Restore Locations: 1

Total Number of Table Spaces to Restore: 2

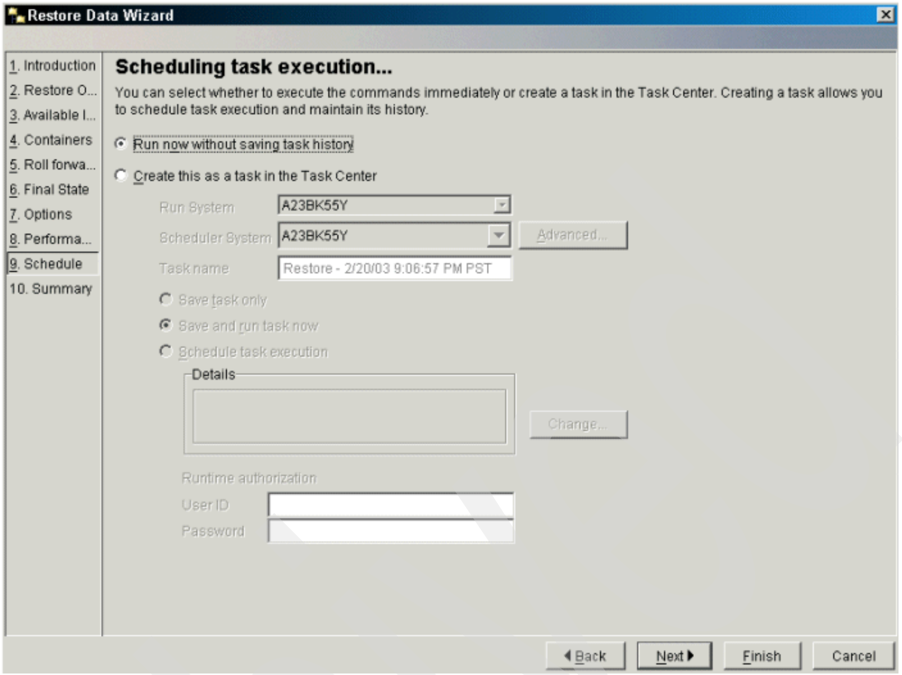
Parallelism: 1 (Recommended)

Number of Buffers: 1 (Recommended)

Size of Each Buffer in 4K pages: 1024 (Recommended)

Figure 6-34 Select performance options for the restore window

12. In the Scheduling the task execution window, select the **Run now without saving task history** radio button and click **Next** (Figure 6-35).



Scheduling task execution...

You can select whether to execute the commands immediately or create a task in the Task Center. Creating a task allows you to schedule task execution and maintain its history.

● Run now without saving task history

○ Create this as a task in the Task Center

Run System: A23BK55Y

Scheduler System: A23BK55Y

Task name: Restore - 2/20/03 9:06:57 PM PST

○ Save task only

● Save and run task now

○ Schedule task execution

Details

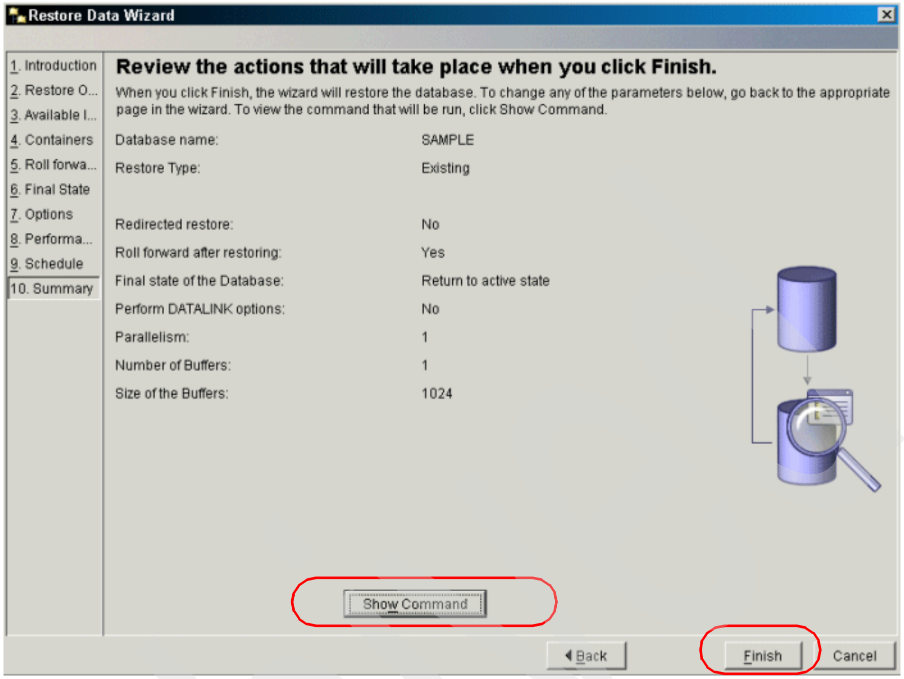
Runtime authorization

User ID:

Password:

Figure 6-35 Scheduling task execution window

13. The following window shows the actions to be performed (Figure 6-36). You may want to click the **Show Command** button to see the syntax that will be submitted for execution of the recovery operation. Click **Finish** to start the recovery operation.



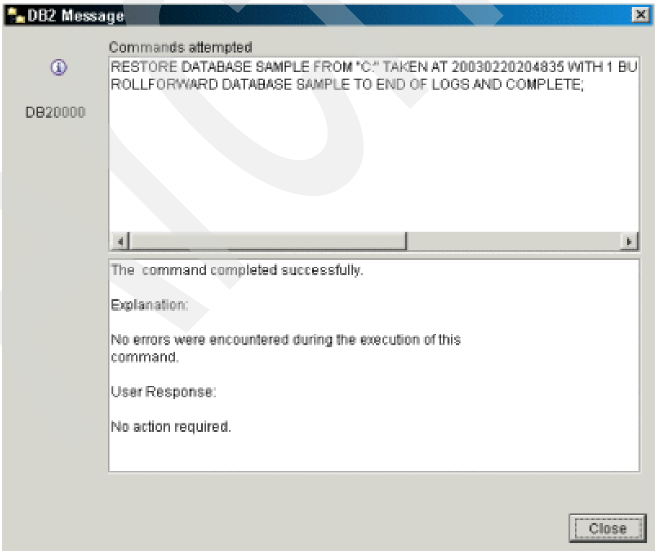
Review the actions that will take place when you click Finish.

When you click Finish, the wizard will restore the database. To change any of the parameters below, go back to the appropriate page in the wizard. To view the command that will be run, click Show Command.

Database name:	SAMPLE
Restore Type:	Existing
Redirected restore:	No
Roll forward after restoring:	Yes
Final state of the Database:	Return to active state
Perform DATALINK options:	No
Parallelism:	1
Number of Buffers:	1
Size of the Buffers:	1024

Figure 6-36 Restore Data Wizard: Review the actions window

14.A window appears showing success of the execution (Figure 6-37).



Commands attempted

```
RESTORE DATABASE SAMPLE FROM "C:" TAKEN AT 20030220204835 WITH 1 BU
ROLLFORWARD DATABASE SAMPLE TO END OF LOGS AND COMPLETE;
```

The command completed successfully.

Explanation:

No errors were encountered during the execution of this command.

User Response:

No action required.

Figure 6-37 Success message window

Other recovery options

In addition to performing a restore on top of an existing database, you also have the option of restoring the database to a new database or perform a redirected restore. Both of these options are available from the Restore Data Wizard in the Control Center. Also mentioned below is the use of a point-in-time recovery with the **`rollforward`** command.

Restore to a new database

Restoring a database image to a new database is typically used when you want more than one copy of the database. This is beneficial when development and/or data warehouse functions are performed in parallel with a production database. By restoring the backup image of a production database to another machine, developers are able to perform tests and data warehouse queries can be generated using a database copy of the production data.

> **Note:** In Linux and Windows environments, when restoring a database image to a different machine, the source and target operating system platforms must be the same.

If you want to restore to a new database on the existing machine and the database code page is the same as the operating system code page, you can type the following:

```
db2 restore database sample into newsamp
```

Where SAMPLE is the original database that has a backup image and NEWSAMP is the database to create using the backup image. If the database backup image has a code page that is different from the operating system, you will need to create the database with the same code page of the database backup image. For instance, if the SAMPLE database has a code page of 1208 (UTF-8), you will need to first create NEWSAMP in the following manner:

```
db2 create database newsamp using codeset UTF-8 territory US
```

Redirected restore

When a redirected restore is performed, you are asking for the tablespace containers to be placed in a directory on the target machine that is different from that specified in the backup image. Once you specify the new location of the tablespace containers on the target system, the restore can completed successfully.

A redirected restore operation is typically used when you need to restore a database to another machine but do not have the same path to the database directory as the source machine. When a database is restored, the restore utility only knows where to restore the database based on the backup image information. It searches on your target machine for the path specified in the backup image. If the path cannot be found on the target machine, a redirected restore is necessary to define a new path for the tablespace containers.

> **Note:** It is possible to restore the database to the target machine if you create the same database path found at the source machine.

Rollforward to point in time

There may be a time when it is necessary to bring your database to a state earlier than the time of a database failure. Using the **`rollforward`** command after a database restore gives you the option to roll forward to a specific point in time. DB2 Version 8.1 is friendlier from previous versions, allowing you to roll forward your log files without having to specify Coordinated Universal Time. Now you have the option of specifying local time thereby eliminating the need to perform time conversions between London, England and your locale. Here is an example of using the local time option:

```
db2 rollforward db sample to 2003-03-07.21.00.40 using local time and stop
```

Crash recovery

If there is a server error or conditions like power or hardware failure, the database or database manager can be ended abnormally and leave the transactions uncompleted and the database in inconsistent and unusable state. Crash recovery is the process by which the database is moved back to a consistent and usable state. By default, this process is enabled as the database configuration parameter *autorestart* is set to *ON*. DB2 will attempt to rollback the incomplete unit of work and complete committed transactions that were still in memory when the crash occurred.

Tablespace level recovery is also available in DB2. For more information about crash and tablespace recovery, refer to the DB2 UDB V8 manual *Data Recovery and High Availability Guide and Reference*, SC09-4831.

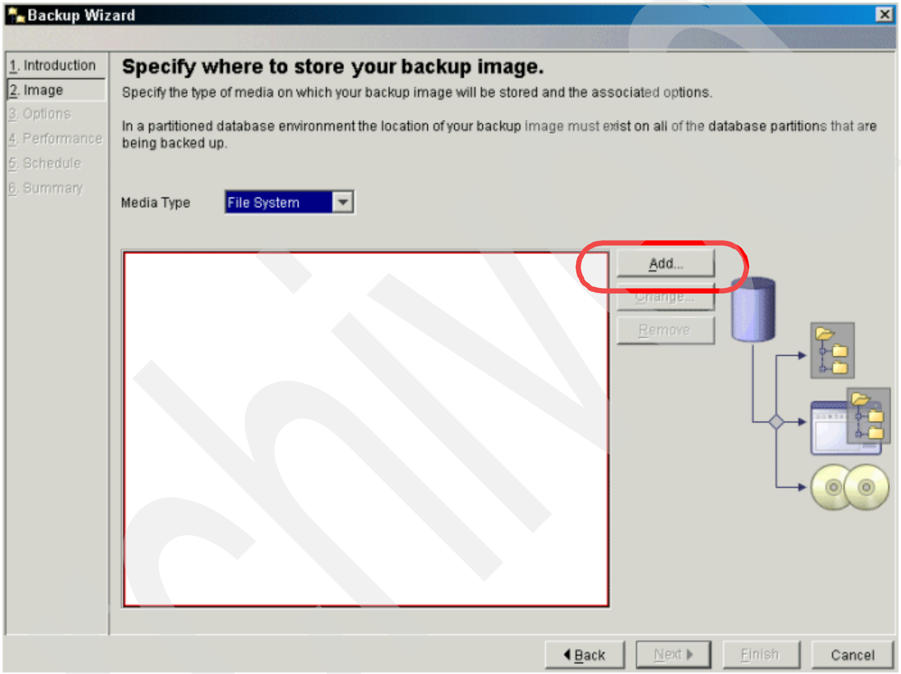
6.3.3 Task scheduling

For those database administrators who are looking for a hands-off approach for routine maintenance tasks, we suggest scheduling your tasks to the Task Center.

The following steps show how to schedule recurring weekly database backups from Control Center so that a task is created in Task Center:

1. From a DB2 CLP or terminal window, type **`db2cc`** to bring up the Control Center.

2. On the left pane of Control Center, expand the directory tree down to the SAMPLE database.
3. Right-click the **SAMPLE database icon** and select **Backup**.
4. The Confirm the details of your database window appears. Select the **Backup entire database** radio button and click **Next**.
5. The Specify where to store your backup image window appears. Leave the Media Type drop-down box as File System and click **Add**.



Specify where to store your backup image.

Specify the type of media on which your backup image will be stored and the associated options.

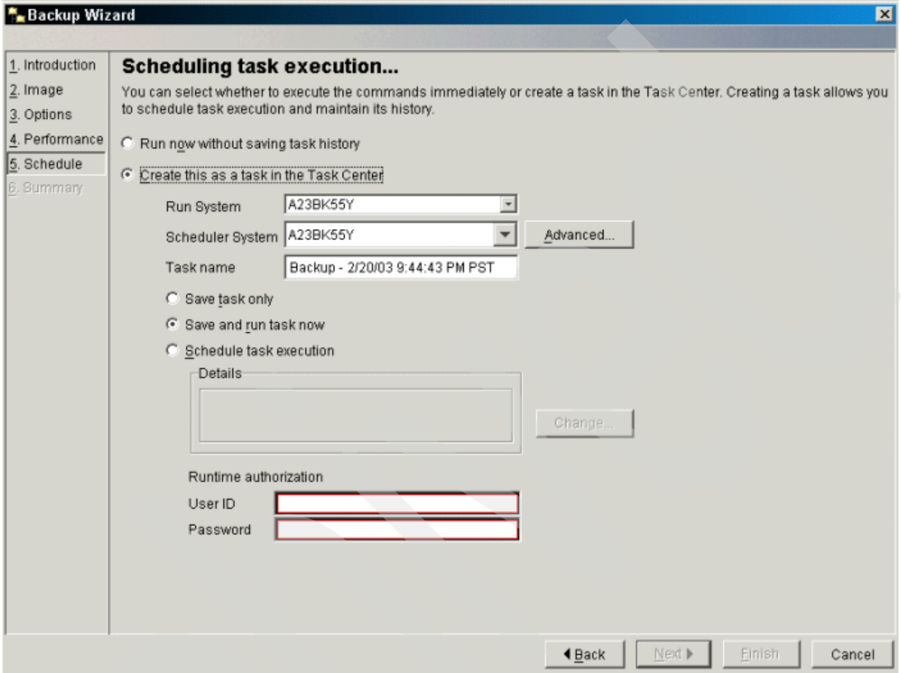
In a partitioned database environment the location of your backup image must exist on all of the database partitions that are being backed up.

Media Type: File System

Figure 6-38 Specify where to store your backup image window

6. A Path Browser window appears. Select a file system and, optionally, a directory to store the database backup image. Click **OK** to go back to the Specify where to store your backup image window. Click **Next** to continue.
7. The Choose your backup options window appears. Select the **Full Backup** radio button. Select the **Online** radio button. Click **Next**.
8. The Specify performance options window appears. Leave this as is. Click **Next.**
9. The Scheduling task execution window appears (Figure 6-39). Select the **Create this as a task in Task Center** radio button.

In the Run System drop-down list box, select your local system. Since the TOOLSDB database is located on your local system, select your local system from the Scheduler System drop-down list box. Leave the Task Name text box as is and select the **Schedule task execution** radio button. Click **Change**.



Backup Wizard

1. Introduction
2. Image
3. Options
4. Performance
5. Schedule
6. Summary

Scheduling task execution...

You can select whether to execute the commands immediately or create a task in the Task Center. Creating a task allows you to schedule task execution and maintain its history.

○ Run now without saving task history

● Create this as a task in the Task Center

Run System: A23BK55Y

Scheduler System: A23BK55Y

Task name: Backup - 2/20/03 9:44:43 PM PST

○ Save task only

● Save and run task now

○ Schedule task execution

Details

Runtime authorization

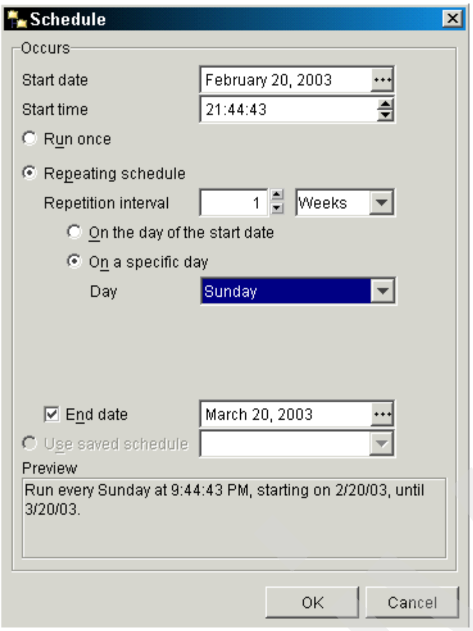
User ID

Password

Figure 6-39 Scheduling task execution window

10.The Schedule window appears (Figure 6-40). Specify a start date and start time, then select the **Repeating schedule** radio button.

For the Repetition interval boxes, type **1** for the first box and **Weeks** for the second box. Select the **On a specific day** radio button and select **Sunday** from the Day drop-down list box. Check the **End date** check box and select one month from your current date. Click **OK**.



Schedule

Occurs

Start date: February 20, 2003

Start time: 21:44:43

○ Run once

● Repeating schedule

Repetition interval: 1 Weeks

○ On the day of the start date

● On a specific day

Day: Sunday

☑ End date: March 20, 2003

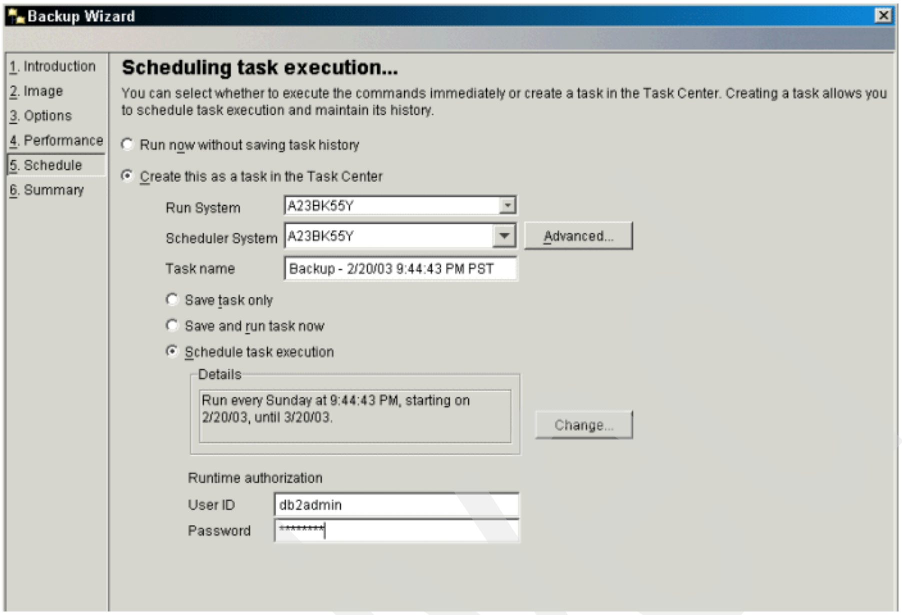
○ Use saved schedule

Preview

Run every Sunday at 9:44:43 PM, starting on 2/20/03, until 3/20/03.

Figure 6-40 Schedule window

11. The Scheduling task execution window appears again (Figure 6-41). For Runtime Authorization, enter your user ID and password. Click **Next**.



Scheduling task execution...

You can select whether to execute the commands immediately or create a task in the Task Center. Creating a task allows you to schedule task execution and maintain its history.

○ Run now without saving task history

● Create this as a task in the Task Center

Run System: A23BK55Y

Scheduler System: A23BK55Y

Task name: Backup - 2/20/03 9:44:43 PM PST

○ Save task only

○ Save and run task now

● Schedule task execution

Details: Run every Sunday at 9:44:43 PM, starting on 2/20/03, until 3/20/03.

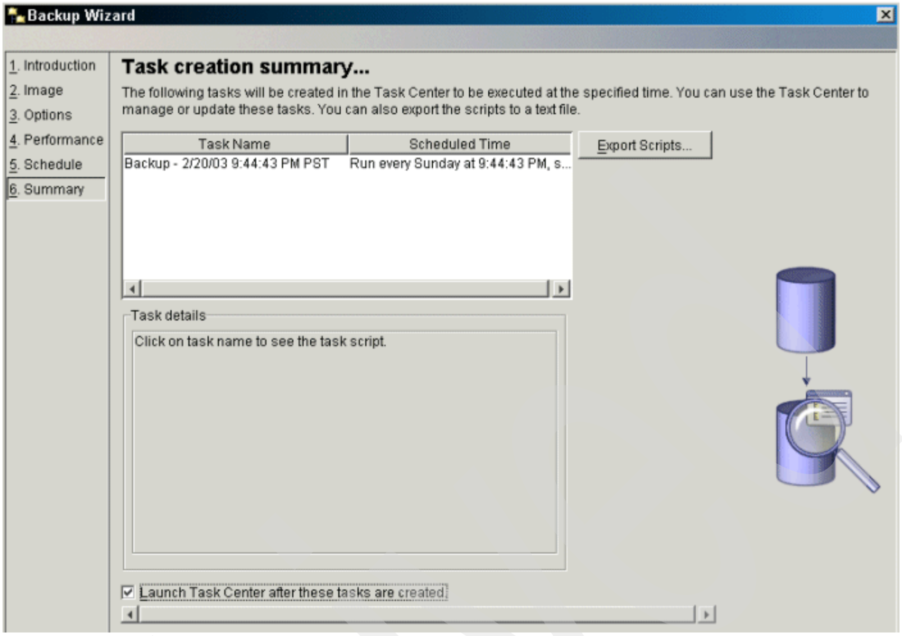
Runtime authorization

User ID: db2admin

Password: ********

Figure 6-41 Scheduling task execution window

12..The Task Creation Summary window appears (Figure 6-42). Verify the summary of your task and check the **Launch Task Center after these tasks are created** check box. Click **Finish**.



Task creation summary...

The following tasks will be created in the Task Center to be executed at the specified time. You can use the Task Center to manage or update these tasks. You can also export the scripts to a text file.

Task Name	Scheduled Time
Backup - 2/20/03 9:44:43 PM PST	Run every Sunday at 9:44:43 PM, s...

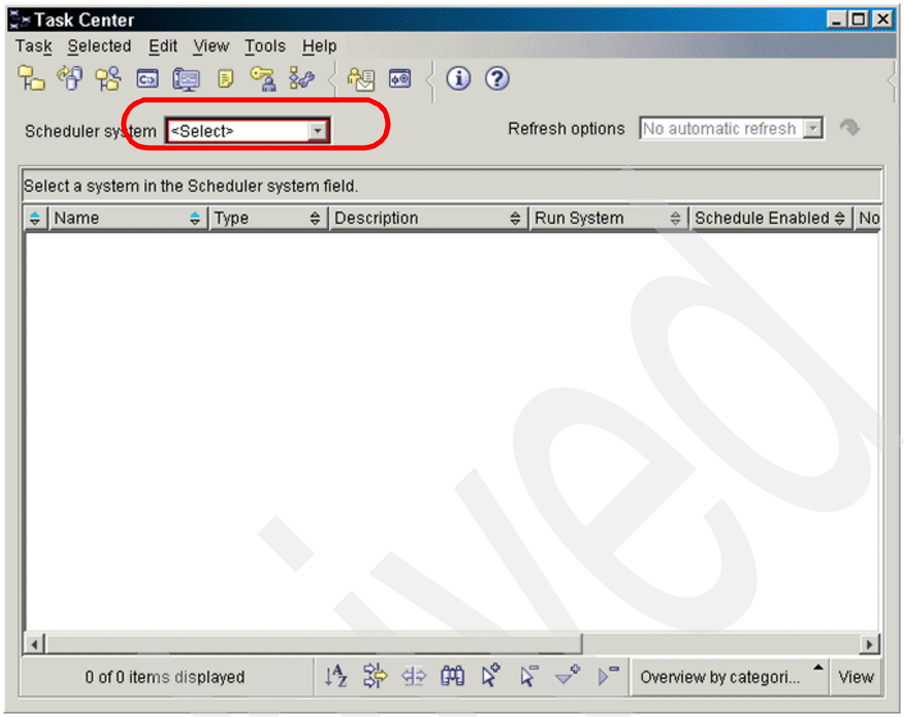
Task details

Click on task name to see the task script.

☑ Launch Task Center after these tasks are created.

Figure 6-42   Task creation summary window

13. The Task Center window appears (Figure 6-43). Select your system from the Scheduler system drop-down list to show the scheduled backup task in the main window. From this point, you can manage the backup task from the Task Center.



Scheduler system <Select>

Refresh options No automatic refresh

Select a system in the Scheduler system field.

Name	Type	Description	Run System	Schedule Enabled	No

0 of 0 items displayed

Overview by categori...

Figure 6-43 Task Center main interface window

6.3.4 Data movement

DB2 Version 8.1 provides a number of commands for moving your data. In this section, we describe how the **`export`**, **`import`**, **`load`**, **`db2move`**, and **`db2look`** utilities are used to help you move data. In addition, an example is provided for each utility.

Export

The **`export`** command is used to move data out of a table and into a file. Example 6-9 is a simple example of the **`export`** command.

Example 6-9 Export employee table data

```
db2 connect to sample
db2 export to data.ixf of ixf select * from employee
```

In the Example 6-9, we are exporting a copy of all the data inside the employee table to the data.ixf file. More complicated select statements can be used to get only data specific to your needs. If you open the data.ixf file, you will not be able to read the file. This is because we used the IXF file format in our export command. The **export** command has three types of file formats available:

- **IXF** (integrated exchange format, PC version), in which most of the table attributes, as well as any existing indexes, are saved in the IXF file, except when columns are specified in the SELECT statement. With this format, the table can be recreated, while with the other file formats, the table must already exist before data can be imported into it.
- **DEL** (delimited ASCII format), which is used by a variety of database manager and file manager programs.
- **WSF** (work sheet format), which is used by programs such as:
  - Lotus® 1-2-3®
  - Lotus Symphony

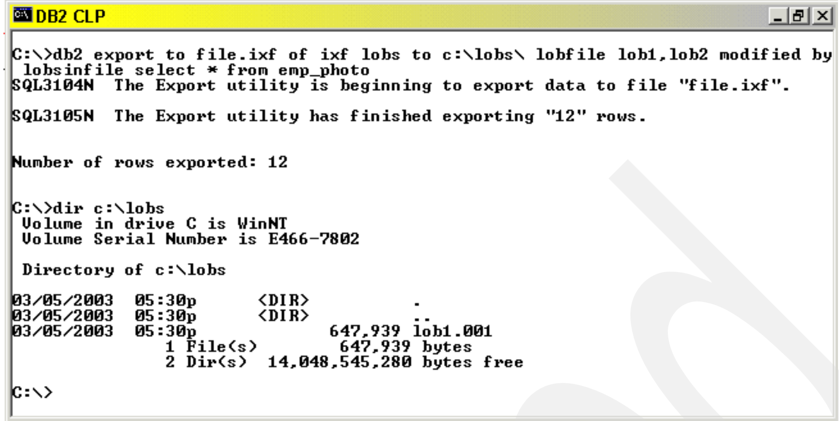
The **export** command also provides modifiers to enable specific functionality during the export operation. For instance, when moving large objects (LOBs), such as character large objects (CLOBS) or binary large objects (BLOBS), you may consider using the LOBSINFILE modifier to create pointers within the exported file to the path where the LOB files are stored. Example 6-10 shows the exporting of data from the emp_photo table that contains BLOB data.

Example 6-10 Export emp_photo table data

```
db2 connect to sample
db2 export to file.ixf of ixf lobs to c:\lobs\ lobfile lob1,lob2 modified by
lobsinfile select * from emp_photo
```

In Example 6-10, the **lobs to** syntax specifies the directory where the LOB columns are stored. The **lobfile lob1,lob2** syntax specifies the LOB file names that are designated to the LOB files. The export utility uses the LOB file names that you provide and creates up to 1000 files for each name by appending the file extension 000 —> 999. For instance, lob1 can have lob1.000 through lob1.999 files. Once lob1.999 is filled, the export utility uses lob2.000 —> lob2.999.

Figure 6-44 shows the execution of the **export** command and the c:\lobs directory contents after the export.



```
C:\>db2 export to file.ixf of ixf lobs to c:\lobs\ lobfile lob1,lob2 modified by
 lobsinfile select * from emp_photo
SQL3104N  The Export utility is beginning to export data to file "file.ixf".

SQL3105N  The Export utility has finished exporting "12" rows.


Number of rows exported: 12


C:\>dir c:\lobs
 Volume in drive C is WinNT
 Volume Serial Number is E466-7802

 Directory of c:\lobs

03/05/2003  05:30p      <DIR>          .
03/05/2003  05:30p      <DIR>          ..
03/05/2003  05:30p             647,939 lob1.001
               1 File(s)        647,939 bytes
               2 Dir(s)  14,048,545,280 bytes free

C:\>
```

Figure 6-44   Export results and c:\lobs directory

Import

The **`import`** command is used to insert data into an existing or new table using SQL inserts. To import from a file, you must have already used the export or another utility to create the data file. Just as with the **`export`** command, the IXF, DEL, and WSF file formats are supported when importing data. Example 6-11 is a simple example of the **`export`** command. Make sure the data.ixf file is located in your current directory.

Example 6-11   Export employee table data

```
db2 connect to sample
db2 import from data.ixf of ixf create into new_employee
```

In Example 6-11, we are importing the table data located in data.ixf and automatically creating a new table before inserting the table data. Because we are using the IXF format, we can create a new table immediately using the table definition that is stored in the data.ixf file.

In Example 6-12, we will use the file.ixf output from Example 6-10 to import into a new table. Make sure the file.ixf data file is located in your current directory:

Example 6-12   Import emp_photo table data into new_emp_photo

```
db2 import from file.ixf of ixf lobs from c:\lobs\ modified by lobsinfile
create into new_emp_photo
```

In Example 6-12, the **`lobs from`** syntax specifies the directory where the LOB columns are stored. A new table called new_emp_photo is created from the table definition information stored in the file.ixf data file.

Load

The load utility is another method for moving data into a table. Unlike the import utility which uses SQL inserts and logging, the load utility writes the data directly to the data pages and reduces logging thereby greatly increasing the speed of the load session for large tables. Since the load does not perform regular logging during the load session, DB2 requires the database to be backed up after the load session completes so that the table can be preserved in case of database failure. This requirement is only applicable to databases that have the database configuration parameter *logretain* set to *recovery*. There is a *nonrecoverable* option syntax that may be used to bypass the back up requirement, but this marks the table to be loaded as non-recoverable.

The load utility supports the following file formats:

- ASC (non-delimited ASCII format)
- DEL (delimited ASCII format)
- IXF (integrated exchange format, PC version), exported from the same or from another DB2 table
- CURSOR (a cursor declared against a SELECT or VALUES statement)

Here is a simple example of the **load** command using a delimited file (Figure 6-13). Since the delimited file format does not have the ability to store the table definition, we need to have an existing table to insert the data into. Using the DEPARTMENT table from the SAMPLE database, we export the data to a file and load the data back into the same table.

Example 6-13 LOAD example

```
db2 export to department.del of del select * from department
db2 load from department.del of del insert into department
```

Backup and restore

The **backup** and **restore** commands may be used to move a database across homogeneous platforms. For instance, you can backup a database on Windows and restore to another Windows environment, but cannot restore that image to a UNIX environment. Since the backup command only saves database files, it will be necessary to manually move stored procedures and user-defined functions to the target machine.

For more information on the **backup** and **restore** commands, refer to sections “Database backup” on page 217 and “Database recovery” on page 236.

db2move

If you are trying to move a database across heterogeneous platforms, you might use the db2move tool to accomplish this. The **db2move** tool enhances the **export**, **import**, and **load** utilities by allowing the movement of many tables within a single command. Some of the DB2 objects like functions and stored procedures are not supported by db2move. You can use db2look to get the DML of these objects and recreate them in the target system. Example 6-14 shows how you can use the db2move tool to export all of the tables in the SAMPLE database in a Windows environment and import the data into the NEWDB database in a Linux environment.

On Windows, run the following command:

```
db2move sample export
```

Example 6-14 EXPORT.out file

```
*****  DB2MOVE  *****
Action:      EXPORT
Start time:  Wed Feb 19 18:26:04 2003

Connecting to database SAMPLE ... successful!  Server: DB2 Common Server V8.1.0
EXPORT:      8 rows from table "UDBRS02 "."ORG"
EXPORT:     35 rows from table "UDBRS02 "."STAFF"
EXPORT:      9 rows from table "UDBRS02 "."DEPARTMENT"
EXPORT:     32 rows from table "UDBRS02 "."EMPLOYEE"
EXPORT:     75 rows from table "UDBRS02 "."EMP_ACT"
EXPORT:     20 rows from table "UDBRS02 "."PROJECT"
EXPORT:     12 rows from table "UDBRS02 "."EMP_PHOTO"
EXPORT:      8 rows from table "UDBRS02 "."EMP_RESUME"
EXPORT:     41 rows from table "UDBRS02 "."SALES"
EXPORT:      0 rows from table "UDBRS02 "."CL_SCHED"
EXPORT:      0 rows from table "UDBRS02 "."IN_TRAY"
EXPORT:     32 rows from table "UDBRS02 "."EMPLOYEE_WORK"
EXPORT:     12 rows from table "UDBRS02 "."NEW_EMP_PHOTO"
EXPORT:     32 rows from table "UDBRS02 "."NEW_EMPLOYEE"
EXPORT:     12 rows from table "UDBRS02 "."NEW_EMP_PHOTO2"

Disconnecting from database ... successful!

End time:  Wed Feb 19 18:26:06 2003
```

After exporting the tables using the db2move tool, an EXPORT.out file is created (Example 6-14) that shows a summary of the export execution and a db2move.lst file is created to be referenced by the db2move when you import or load the tables. In addition, there will be an IXF file and a message file for each table exported into the current directory (Example 6-15). If a table contains large object columns (LOBS), there will also be files to hold the LOBS which have a naming convention similar to **tab15a.001**, where the position of **a** can be a-z and **.001** can range from 001->999.

> **Note:** With the db2move utility, there is a limitation of 26000 rows for a table containing LOB columns. If one of your tables has more than 26000 rows and contains LOB columns, you may consider using the export and import/load utilities to move that specific table.

Example 6-15 Directory listing of the files exported with **`db2move`** *command*

```
02/19/2003  06:30p      <DIR>          .
02/19/2003  06:30p      <DIR>          ..
02/19/2003  06:26p                 668 db2move.lst
02/19/2003  06:26p               1,100 EXPORT.out
02/19/2003  06:26p               6,602 tab1.ixf
02/19/2003  06:26p                 145 tab1.msg
02/19/2003  06:26p               5,213 tab10.ixf
02/19/2003  06:26p                 146 tab10.msg
02/19/2003  06:26p               5,213 tab11.ixf
02/19/2003  06:26p                 146 tab11.msg
02/19/2003  06:26p              17,929 tab12.ixf
02/19/2003  06:26p                 147 tab12.msg
02/19/2003  06:26p               5,024 tab13.ixf
02/19/2003  06:26p                 147 tab13.msg
02/19/2003  06:26p             647,939 tab13a.001
02/19/2003  06:26p              17,929 tab14.ixf
02/19/2003  06:26p                 147 tab14.msg
02/19/2003  06:26p               5,023 tab15.ixf
02/19/2003  06:26p                 147 tab15.msg
02/19/2003  06:26p             385,683 tab15a.001
02/19/2003  06:26p               9,763 tab2.ixf
02/19/2003  06:26p                 146 tab2.msg
02/19/2003  06:26p               6,640 tab3.ixf
02/19/2003  06:26p                 145 tab3.msg
02/19/2003  06:26p              17,929 tab4.ixf
02/19/2003  06:26p                 146 tab4.msg
02/19/2003  06:26p              11,244 tab5.ixf
02/19/2003  06:26p                 146 tab5.msg
02/19/2003  06:26p              10,535 tab6.ixf
02/19/2003  06:26p                 146 tab6.msg
02/19/2003  06:26p               5,012 tab7.ixf
02/19/2003  06:26p                 146 tab7.msg
02/19/2003  06:26p             647,939 tab7a.001
02/19/2003  06:26p               4,765 tab8.ixf
02/19/2003  06:26p                 145 tab8.msg
02/19/2003  06:26p              12,754 tab8a.001
02/19/2003  06:26p               8,079 tab9.ixf
02/19/2003  06:26p                 146 tab9.msg
```

```
              38 File(s)      1,836,259 bytes
```

After exporting the files, you can zip all of the files in the directory and ftp the file in binary mode to the Linux machine. Once the files are unzipped on the Linux machine, create a new database called NEWDB and load the data into the NEWDB database. Make sure you execute the **db2move** command in the same directory where the IXF files are located:

In the target Linux system, login as the instance owner ID and enter the following commands on command prompt. Press **Enter** after each command:

```
db2 create database newdb
db2move newdb import
```

Example 6-16 IMPORT.out file

```
*****  DB2MOVE  *****
Action:      IMPORT
Start time:  Wed Feb 19 11:18:04 2003

Server: DB2 Common Server V8.1.0

*** Table "UDBRS02 "."ORG":  ERROR 3107.  Check message file tab1.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."STAFF":  ERROR 3107.  Check message file tab2.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."DEPARTMENT":  ERROR 3107.  Check message file tab3.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."EMPLOYEE":  ERROR 3107.  Check message file tab4.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."EMP_ACT":  ERROR 3107.  Check message file tab5.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."PROJECT":  ERROR 3107.  Check message file tab6.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."EMP_PHOTO":  ERROR 3107.  Check message file tab7.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.
*** Table "UDBRS02 "."EMP_RESUME":  ERROR 3107.  Check message file tab8.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."SALES":  ERROR 3107.  Check message file tab9.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."CL_SCHED":  ERROR 3107.  Check message file tab10.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."IN_TRAY":  ERROR 3107.  Check message file tab11.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."EMPLOYEE_WORK":  ERROR 3107.  Check message file
tab12.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."NEW_EMP_PHOTO":  ERROR 3107.  Check message file
tab13.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."NEW_EMPLOYEE":  ERROR 3107.  Check message file
tab14.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

*** Table "UDBRS02 "."NEW_EMP_PHOTO2":  ERROR 3107.  Check message file
tab15.msg!
***  SQL Warning!  SQLCODE is  3107
***  SQL3107W  There is at least one warning message in the message file.

End time:  Wed Feb 19 11:18:38 2003
```

Once the import is complete, an IMPORT.out file is created along with a message file for each table. The IMPORT.out file shows a summary of the operation, as shown in Example 6-16. Since the database on Windows has a different code page from the database on Linux, code page conversions will occur and warnings are written to note these conversions. No other action is necessary.

If you want to use the load utility to move the data, it is necessary to create all of the tables ahead of time before inserting the data. Section “db2look” on page 263 shows how the db2look utility can capture the table definitions. For more options specific to the **db2move** command, type db2move at the command line.

db2look

When porting large tables across heterogeneous platforms, it may be necessary to use **db2move** with the **load** option, or use the **load** command rather than **import** so as to speed up the data movement process. Since the load utility does not allow for the automatic creation of tables, it is necessary to create the tables before inserting the data. A helpful tool to aid in this process is the db2look utility. The db2look utility generates a DDL script from your source database and formats it so that you can create the tables on your target database using the script.

The following **db2look** command shows how you can generate a script of all tables in the SAMPLE database:

```
db2look -d SAMPLE -e -o db2look.out
```

Example 6-17   Partial output generated from db2look utility

```
-- This CLP file was created using DB2LOOK Version 8.1
-- Timestamp: 02/19/2003 08:08:02 PM
-- Database Name: SAMPLE
-- Database Manager Version: DB2/NT Version 8.1.0
-- Database Codepage: 1252
-- Database Collating Sequence is: UNIQUE

CONNECT TO SAMPLE;

------------------------------------------------
-- DDL Statements for table "UDBRS02 "."ORG"
------------------------------------------------

CREATE TABLE "UDBRS02 "."ORG"  (
"DEPTNUMB" SMALLINT NOT NULL ,
"DEPTNAME" VARCHAR(14) ,
"MANAGER" SMALLINT ,
"DIVISION" VARCHAR(10) ,
"LOCATION" VARCHAR(13) )
IN "USERSPACE1" ;

------------------------------------------------
-- DDL Statements for table "UDBRS02 "."STAFF"
------------------------------------------------

 CREATE TABLE "UDBRS02 "."STAFF"  (
"ID" SMALLINT NOT NULL ,
"NAME" VARCHAR(9) ,
"DEPT" SMALLINT ,
"JOB" CHAR(5) ,
"YEARS" SMALLINT ,
"SALARY" DECIMAL(7,2) ,
"COMM" DECIMAL(7,2) )
IN "USERSPACE1" ;
```

In Example 6-17, a partial output of the db2look results is displayed. For details regarding specific options with the db2look utility, type **`db2look`** at the command prompt.

6.4 Monitoring

This section discusses the various DB2 and operating specific monitoring and event tools available to measure DB2 performance, utilization, behavior, and event data. Specifically, we describe the uses of the DB2 Health Monitor and Event Monitor tools, the Windows Task Manger and Event Viewer, and the Linux monitoring commands: **`top`**, **`iostat`**, **`sar`**, and **`vmstat`**.

In DB2 Version 8, many tools and functionality have been added or modified to take advantage of features in the Windows environments. DB2 Version 8 enhances its autonomic capability with new self-healing technology. To accomplish that functionality, DB2 is bundled with many tools and special commands that allow a broad analysis of the DB2 environment.

DB2 implements some types of proactive and self-healing techniques with the concept of monitoring by exception. Now on DB2, when some problems appear, DB2 recommends some options for what you can do, and offers tools to do what you choose immediately.

Monitors can be classified as follows:

- **Data monitors:** These are responsible to get the various measurements and information from each module of the system. The data for each is stored on its own repository. For example, Perfmon is a data monitor.
- **Event monitors:** These only generate information when a pre-defined condition or an exception occurs, like Windows memory Page Faults/sec and DB2 Application Sort Overflow. Some event monitors can handle messages from more than one module. For example, DB2 log monitor and Windows EventLog service are event monitors.

Elements of both monitors can be created with a combination of the two types. For example, in DB2 Application sort overflow, the DB2 monitor detects when a sort overflows occurs and automatically increments an internal counter.

6.4.1 DB2 Health Monitor

The Health Monitor is a server-side tool that constantly monitors the health of the instance and all active database objects when DB2 is started. The Health Monitor automatically evaluates a set of health indicators, even without user interaction. If the current value of a health indicator is outside the acceptable operating range or in a non-normal state, the Health Monitor generates a health alert.

The Health Monitor gathers information about the health of the system using new interfaces that do not impose a performance penalty. It does not turn on any snapshot monitor switches to collect information. The Health Monitor is enabled by default when a instance is created; you can deactivate it using the database manager configuration parameter *health_mon*. If the Health Monitor finds that a defined threshold has been exceeded (for example, the available log space is not sufficient), or if it detects an abnormal state for an object (for example, an instance is down), the Health Monitor will raise an alert.

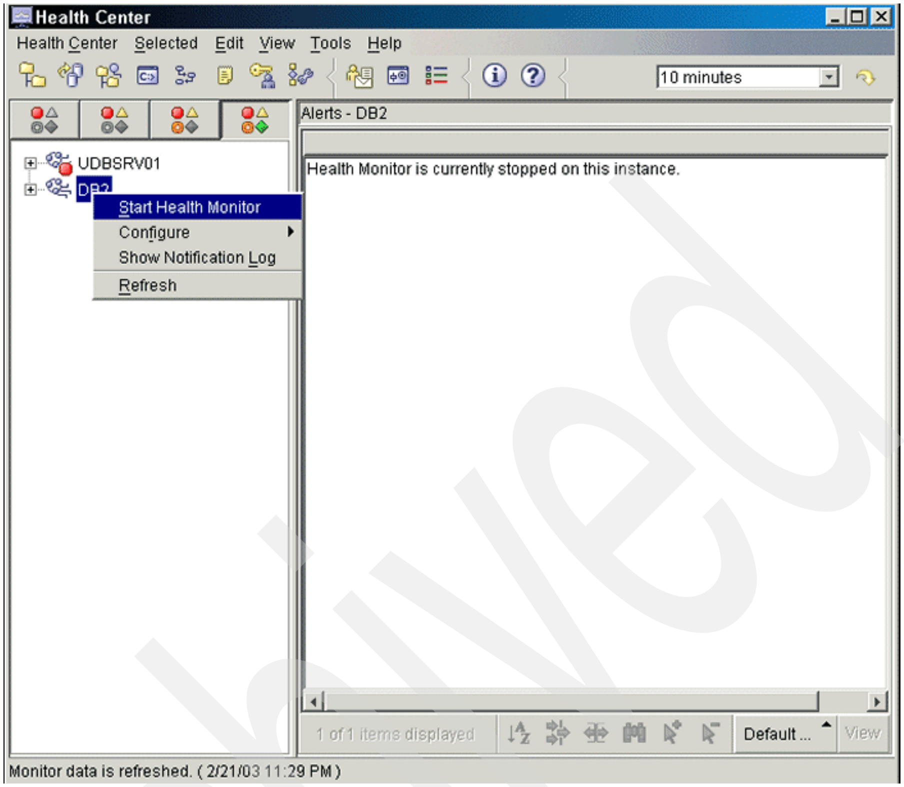
Starting Health Center and Health Monitor

You can start Health Center by selecting Health Center from the Tools menu within any DB2 GUI tools where the Tools menu is available. You can also start Health Center by clicking the Health Center Status beacon when it appears on the status line of the DB2 Control Center window; see Figure 6-45.

The Health Monitor is activated by default, and if it is stopped for any reason, you can reactivate it by using Health Center or the DB2 CLP command **`db2 update dbm cfg using health_mon ON`**.

Depending on the configuration of the DB2 instance, the following actions can occur when the Health Monitor generates an alert:

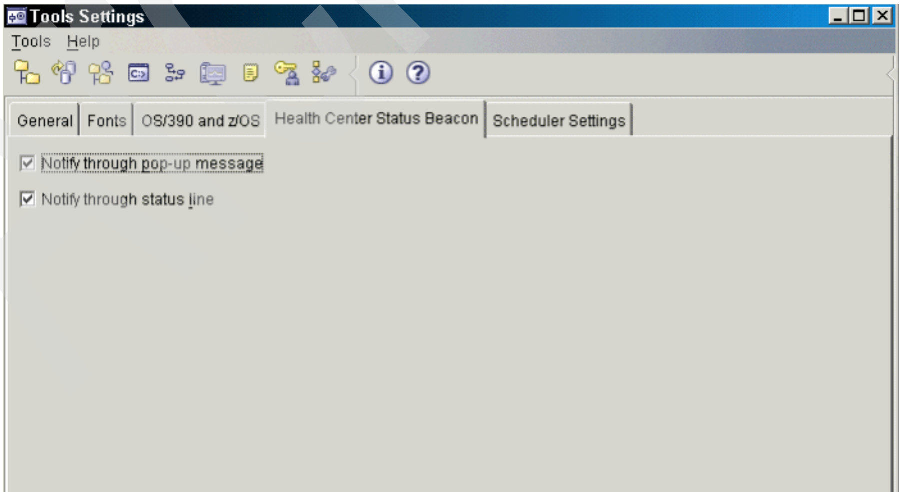
- An entry is written in the administration notification log, which you can read from the Journal.
- The Health Center status beacon appears in the lower right corner of the DB2 GUI Tools window. You can set the status beacon options opening on any DB2 GUI tool by clicking the menu **Tools—>Tools Settings**, and then click the tab Health Center Status Beacon (Figure 6-46).



Health Monitor is currently stopped on this instance.

Monitor data is refreshed. (2/21/03 11:29 PM)

Figure 6-45 Starting Health Monitor from Health Center



☑ Notify through pop-up message

☑ Notify through status line

Figure 6-46 Setting Health Center Status Beacon through Tools Settings

The Status Beacon configuration has two options:

- A script or task is executed.
- An e-mail or pager message is sent to the contacts that you specify for this DB2 instance.

The *health_mon* parameter can be dynamically changed. The update to the *health_mon* takes effect immediately. Once the *health_mon* is turned on (the default), an agent collects information about the health of the objects that are active in your database environment. If an object is considered to be in an unhealthy condition, based on thresholds or object state, notifications are sent and actions can be taken automatically.

For a step-by-step example of the Health Monitor tool, see 3.8.1, “Health Center” on page 93. Additional information on the Health Monitor can be found in Appendix A of the *DB2 UDB Version 8 manual System Monitor Guide and Reference* and the IBM Redbooks:

- *Up and Running with DB2 for Linux*, SG24-6899
- *DB2 UDB Exploitation of the Windows Environment*, SG24-6893

DB2 Event Monitor

In this section we describe the main functionality and the best resources found on DB2 event monitors. You can find an extensive explanation of DB2 monitoring in the IBM DB2 UDB Version 8 manual *System Monitor Guide and Reference*.

In DB2 Version 8, considerable effort was expended to improve event monitor functionality and usability. These are the most interesting capabilities added:

- **Creation of event monitor with SQL commands**

  Now it is possible to create DB2 event monitors using standard SQL. For example:

  ```
  db2 “CREATE EVENT MONITOR CONN_MON FOR CONNECTIONS WRITE TO FILE
  ‘C:\CONN_MON.EMD’”
  ```

  This command will create an event monitor called CONN_MON to monitor connections and save the data in a file called CONN_MON at the c:\ directory. The command line program to create a DB2 event monitor is also included in this version, for example:

  ```
  db2evmon [-db ] <database-alias>[-evm ] <event-monitor-name>
  ```

  or

  ```
  db2evmon -path <event-monitor-target>
  ```

- **Directing the output data to a table or pipe**

  DB2 Version 8 also offers the possibility of directing the output data to tables and pipes. For example:

  ```
  CREATE EVENT MONITOR CONN_MON FOR CONNECTIONS WRITE TO TABLE
  ```

  or

  ```
  CREATE EVENT MONITOR CONN_MON FOR CONNECTIONS WRITE TO PIPE
  '/UDBRED04/EVMON/PIPE'
  ```

Windows specific monitoring tools

A well tuned and performing Windows environment impacts positively on the database overall. In the same way monitoring is done on DB2, you should monitor Windows environments.

You can find more information about Windows 2000 in the Windows 2000 Resource Kit. Also, a nice group of tools to manage and obtain information about Windows 2000 can be found at this Web site:

http://www.sysinternals.com

In this topic, we explain more about each of the Windows tools and how it can be used with DB2 and Windows.

Task Manager

Task Manager is the most basic monitoring tool in Windows. It includes:

- Applications running
- Processes running
- Memory usage
  - Kernel memory counter and usage
  - Physical memory counter and usage
  - Commit charge
  - General counters
  - Handles
  - Threads
  - Processes

To run Windows Task Manager, right-click the taskbar on the desktop and choose Task Manager. There are three tabs: Applications, Processes, and Performance. The Processes and Performance monitoring can be used to get a quick overview of the currently active DB2 processes and overall system performance and CPU utilization. Figure 6-47 shows the Processes tab, while Figure 6-48 displays the Performance tab.

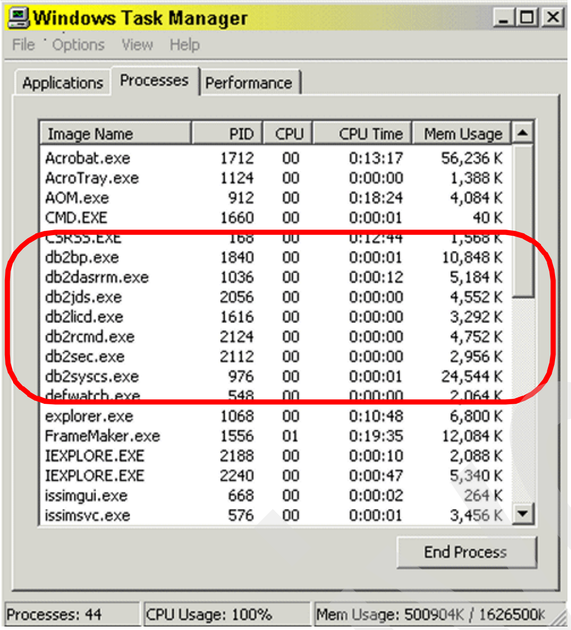
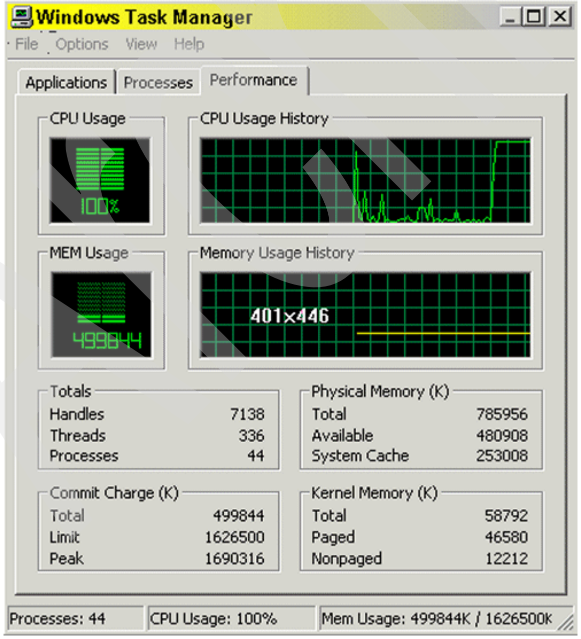


Image Name	PID	CPU	CPU Time	Mem Usage
Acrobat.exe	1712	00	0:13:17	56,236 K
AcroTray.exe	1124	00	0:00:00	1,388 K
AOM.exe	912	00	0:18:24	4,084 K
CMD.EXE	1660	00	0:00:01	40 K
CSRSS.EXE	168	00	0:12:44	1,568 K
db2bp.exe	1840	00	0:00:01	10,848 K
db2dasrrm.exe	1036	00	0:00:12	5,184 K
db2jds.exe	2056	00	0:00:00	4,552 K
db2licd.exe	1616	00	0:00:00	3,292 K
db2rcmd.exe	2124	00	0:00:00	4,752 K
db2sec.exe	2112	00	0:00:00	2,956 K
db2syscs.exe	976	00	0:00:01	24,544 K
defwatch.exe	548	00	0:00:00	2,064 K
explorer.exe	1068	00	0:10:48	6,800 K
FrameMaker.exe	1556	01	0:19:35	12,084 K
IEXPLORE.EXE	2188	00	0:00:10	2,088 K
IEXPLORE.EXE	2240	00	0:00:47	5,340 K
issimgui.exe	668	00	0:00:02	264 K
issimsvc.exe	576	00	0:00:01	3,456 K

Processes: 44 CPU Usage: 100% Mem Usage: 500904K / 1626500K

Figure 6-47 Processes tab in Windows Task Manager



Totals	
Handles	7138
Threads	336
Processes	44

Physical Memory (K)	
Total	785956
Available	480908
System Cache	253008

Commit Charge (K)	
Total	499844
Limit	1626500
Peak	1690316

Kernel Memory (K)	
Total	58792
Paged	46580
Nonpaged	12212

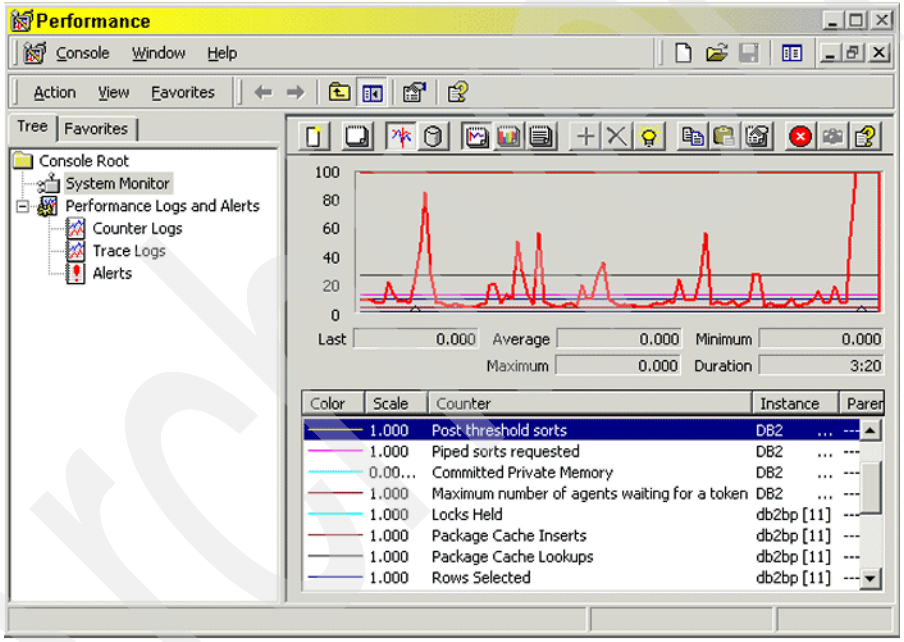
Processes: 44 CPU Usage: 100% Mem Usage: 499844K / 1626500K

Figure 6-48 Performance tab in Windows Task Manager

We have used this tool to quickly monitor the basic performance and utilization data. This is an easy way to diagnose CPU and network utilization. This tool does not have an option to save the monitored data.

Performance Monitor and alert

DB2 is integrated with the underlying operating system where it resides. All DB2 monitoring information is available through the Performance Monitor, which is the standard tool to do graphical analysis in the Windows environment. It contains a comprehensive group of counters for each Windows component, for example; networking counters, physical disk counters, process counters, and so on. It also provides a way to save performance data and afterwards replay it to analyze the environment (Figure 6-49).



Last 0.000 Average 0.000 Minimum 0.000
Maximum 0.000 Duration 3:20

Color	Scale	Counter	Instance	Parer
—	1.000	Post threshold sorts	DB2 ...	---
—	1.000	Piped sorts requested	DB2 ...	---
—	0.00...	Committed Private Memory	DB2 ...	---
—	1.000	Maximum number of agents waiting for a token	DB2 ...	---
—	1.000	Locks Held	db2bp [11]	---
—	1.000	Package Cache Inserts	db2bp [11]	---
—	1.000	Package Cache Lookups	db2bp [11]	---
—	1.000	Rows Selected	db2bp [11]	---

Figure 6-49 Performance Monitor in action

When DB2 is installed, it registers its counters as a Perfmon provider. These counters are grouped as follows:

- DB2 Database Manager elements
- DB2 Database elements
- DB2 Applications

To start the Performance Monitor, go to **Start —> Control Panel —> Administrative Tools —> Performance**. To add DB2 counters, put the cursor in the middle of the chart panel and right-click. A menu will open; choose **Add Counters** (Figure 6-50).

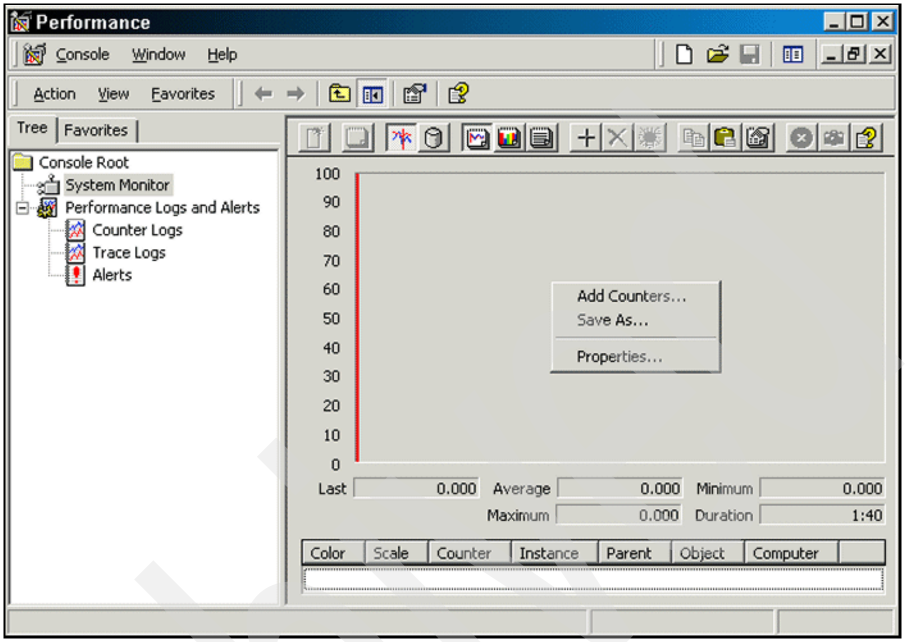
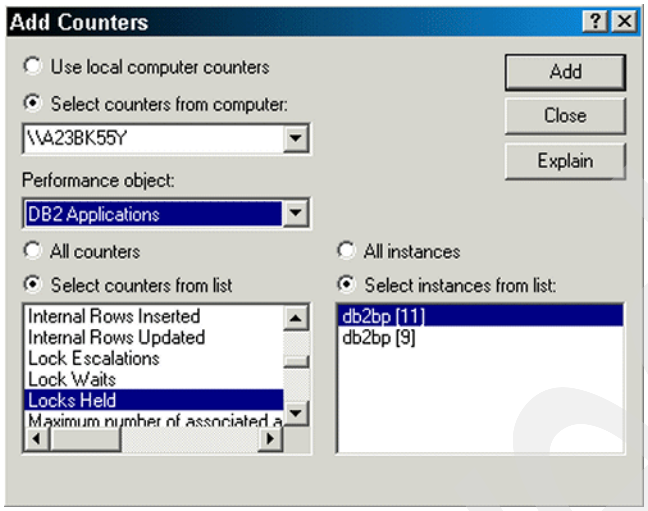


Figure 6-50 Performance Monitor and Alert: Adding DB2 counters

In Figure 6-51, the three groups are shown for the DB2 performance objects. When you select the desired group, their elements are shown on the list. It is possible to select all the counters, but the chart panel will become overwhelmed. You can avoid this problem by selecting only the desired elements. When you select an element, all the objects that are currently instantiated on DB2 that have the selected element are shown on the right list of the panel. Select the desired object and click **Add** (Figure 6-51).


Add Counters

○ Use local computer counters

● Select counters from computer:

\\A23BK55Y

Performance object:

DB2 Applications

○ All counters

● Select counters from list

Internal Rows Inserted
Internal Rows Updated
Lock Escalations
Lock Waits
Locks Held
Maximum number of associated a

○ All instances

● Select instances from list:

db2bp [11]
db2bp [9]

Figure 6-51 Performance Monitor and Alert: Adding DB2 counters

Repeat this procedure on all elements that you want to monitor, and finally, click **Close**. Now the chart panel will display a measure of the elements updated on the default time. To change the update time, right-click the chart panel and select **Properties**. There is a text box on the right side of the Update automatically every: check box. Change the value to the desired interval.

If the performance counters are not shown, it could be that they were not registered. To handle the Performance Monitor functions, there are three tools:

- **db2perfi**: Performance counters registration utility
- **db2perfr**: Performance Monitor registration tool
- **db2perfc**: Reset database performance values

To register DB2 performance counters, open a command line and run the command `db2perfi -i` . After that, you should be able to open the performance counters.

As mentioned before, the Performance Monitor can start background monitoring services to get data elements. With the DB2 db2perfmon provider, you can do background monitoring of the following types:

- **Counter logs**: Counter logs are monitoring tasks that could be programmed to run at a specific hour and for a defined amount of time. They collect the same monitoring data from the performance elements that the real time chart collects. But instead of displaying on the window, the data is directed to text or binary files.

- **Alert logs**: Alert logs are monitoring tasks that could be programmed to run at a determined hour and for a defined amount of time. They collect the same monitoring data from the performance elements that the real time chart collects. But instead of displaying on the window, the data is verified against a condition. If this condition is true, the alert log can do the following tasks:
  - Log an entry in the application event log
  - Send a network message to a user or a computer on the network
  - Start a performance data log or counter log
  - Run a program

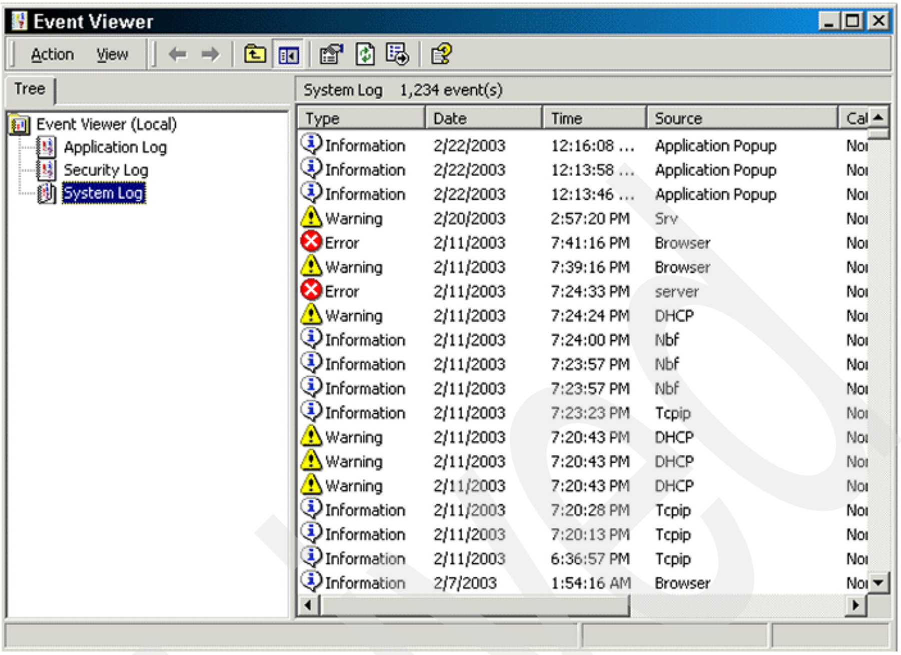
Since the Alert Logs allow you to run a wide variety of tasks, it is possible to do some automated management with them in a similar way as you can with the DB2 Health Monitor.

Event viewer

The Windows environment has standard log mechanisms. System and applications can use these mechanisms to set a notification level. There are three levels of log entry:

- **Information**: Used when the system or one application wants to send a message about successful operation or assorted information.
- **Warning**: Used when the system or one application detects that a problem could happen or is about to happen, or when an expected behavior is not found or met, but the system could continue working without affecting the functioning at all.
- **Error**: Used when the system or one application wants to send or write an error message about an unexpected termination or a failed operation. When a message of this type is generated, there is a potential chance of damage or data loss.

To view the Event Viewer, go to **Start —> Control Panel —> Administrative Tools —> Event Viewer**. Figure 6-52 shows the Windows Event Viewer.



System Log 1,234 event(s)

Type	Date	Time	Source	Ca
Information	2/22/2003	12:16:08 ...	Application Popup	No
Information	2/22/2003	12:13:58 ...	Application Popup	No
Information	2/22/2003	12:13:46 ...	Application Popup	No
Warning	2/20/2003	2:57:20 PM	Srv	No
Error	2/11/2003	7:41:16 PM	Browser	No
Warning	2/11/2003	7:39:16 PM	Browser	No
Error	2/11/2003	7:24:33 PM	server	No
Warning	2/11/2003	7:24:24 PM	DHCP	No
Information	2/11/2003	7:24:00 PM	Nbf	No
Information	2/11/2003	7:23:57 PM	Nbf	No
Information	2/11/2003	7:23:57 PM	Nbf	No
Information	2/11/2003	7:23:23 PM	Tcpip	No
Warning	2/11/2003	7:20:43 PM	DHCP	No
Warning	2/11/2003	7:20:43 PM	DHCP	No
Warning	2/11/2003	7:20:43 PM	DHCP	No
Information	2/11/2003	7:20:28 PM	Tcpip	No
Information	2/11/2003	7:20:13 PM	Tcpip	No
Information	2/11/2003	6:36:57 PM	Tcpip	No
Information	2/7/2003	1:54:16 AM	Browser	No

Figure 6-52 Windows Event Viewer

The Windows Event Viewer has three event logs:

- **Application log**: The application log is the log where DB2 services and modules write their messages. All DB2 messages mentioned before are written inside the application log.
- **System log**: The system log applies to events that are related to the environment and basic structure and functionality of the system.
- **Security log**: Since DB2 relies on the underlying operating system to do user authentication, all the tries that are done to log on are registered on the security log.

To control the DB2 event generation on Windows Event Logs, DB2 Version 8 adds a new parameter *notifylevel* to the database manager.

The event entries are recorded on these event logs on the following way: All common and administrative messages are registered on Windows events logs, and all messages that are for DB2 diagnosis are saved on the db2diag.log files. If DB2 db2diag.log file logs an error about an internal program detection, a log entry is also recorded on the Windows event log advising that an internal message has been generated.

Linux specific monitoring tools

There are a lot of Linux system monitoring tools available to assist you in identifying where the performance or system setting issue occurs. These tools could be used to monitor system resource usage, such as disk I/O, memory consumption, CPU activities and network status. Through the utilization of comprehensive information provided by these operating system tools and combining database system monitoring tools from DB2 UDB, you can understand your system more clearly. For example, which tablespace containers are under high disk I/O pressure, whether or not the excessive sort heap allocation will lead to a mass of paging space activities, if the network bandwidth is the bottleneck for poor response time of client applications, and so on. Then you can make pertinent adjustments to your applications, database system, or operating system to improve the system performance.

In this section, we discuss some of the most commonly used performance monitoring tools for the Linux platform: top, iostat, vmstat, and sar.

Top

Top gives you a lot of information on one window. Top shows you which process is taking the most processor time, the most amount of memory (both in percentages), how long the system has been up, and the amount of free memory. By default, the utility will refresh the display every 5 seconds. You can override the default refresh interval via command option “d”. The users should be aware of that the top command does not report total memory used for DB2 correctly. Top will overstate memory usage by counting shared memory segment multiple times. The top function itself also uses high amount of resource. In general, it is not a recommend monitor tool.

Some frequently used command line options for top are listed in Table 6-1.

Table 6-1 Frequently used command options for top monitoring

Command option	Description
d	Specifies the delay between screen updates
p	Monitor only processes within a given process id. This flag can be given up to twenty times.
i	Start top ignoring any idle or zombie processes

Note: In Linux, DB2 will not free shared memory automatically. To free the shared memory, you need to recycle DB2.

If you are not using batch mode to run **top**, then when the output screen of top displays, you may use the interactive command to control the display. For example, you can toggle off/on the display of CPU states, or sort the processes list by the memory occupancy percentage, or kill a process in the list. Table 6-2 shows some commonly used commands.

Table 6-2 Frequently used interactive commands for top monitoring

Commands	Description
space	Immediately updates the display.
h or ?	Displays a help screen giving a brief summary of commands, and the status of secure and cumulative modes.
n or #	Change the number of processes to display. You will be prompted to enter the number.
q	Quit
f or F	Add fields to display or remove fields from the display.
c	Toggle display of command name or full command line.
A	Sort tasks by age (newest first)
P	Sort tasks by CPU usage (default)
M	Sort tasks by memory usage.
W	Write current setup to top configuration file ~/.toprc.

Example 6-18 shows top output, where some interactive commands are used, such as press “M” to re-order the processes list by resident memory usage (by default, it is ordered by process ID), press “c” to toggle full command line display, and using “n” to control the number of tasks that will be displayed.

Example 6-18 Using top to monitor system resource usage

```
2:21pm up 2:18,3 users,load average:1.75,1.77,1.37
74 processes:71 sleeping,3 running,0 zombie,0 stopped
CPU states:1.1%user,0.9%system,0.0%nice,97.8%idle
Mem:578608K av,448292K used,130316K free,0K shrd,78904K buff
Swap:1753304K av,0K used,1753304K free 290556K cached
PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND
820 root 5 -10 23288 18M 5284 S <0.1 3.2 16:34 X
1538 root 15 0 14136 13M 9328 S 0.0 2.4 0:12 nautilus
1545 root 15 0 12608 12M 8604 S 0.1 2.1 4:36 rhn-applet-gui
```

```
1536 root 15 0 10676 10M 8172 R 0.3 1.8 17:36 gnome-panel
1556 root 15 0 8648 8644 6752 S 0.0 1.4 0:08 gnome-terminal
1551 root 15 0 8468 8464 6828 S 0.0 1.4 0:02 gweather-applet
1418 root 15 0 8212 8208 6324 S 0.0 1.4 0:00 gnome-session
1496 root 15 0 6988 6984 5576 S 0.0 1.2 0:01 gnome-settings-
1494 root 15 0 6360 6360 5204 S 0.0 1.0 0:02 metacity
1540 root 16 0 5892 5888 4952 S 0.7 1.0 35:23 magicdev
1490 root 15 0 5088 5088 1972 S 0.0 0.8 0:01 gconfd-2
1543 root 15 0 4064 4064 3440 S 0.0 0.7 0:03 pam-panel-icon
738 xfs 15 0 3276 3276 884 S 0.0 0.5 0:00 xfs
771 root 15 0 2920 2920 2804 S 0.0 0.5 0:00 gdm-binary
1492 root 15 0 2248 2248 1848 S 0.0 0.3 0:00 bonobo-activati
1557 root 16 0 1520 1520 1152 S 0.0 0.2 0:00 bash
560 root 18 0 1468 1468 1224 S 0.0 0.2 0:00 sshd
3855 root 15 0 1448 1412 848 S 0.0 0.2 0:00 db2fmcd
6561 root 15 0 1412 1412 1116 S 0.0 0.2 0:00 bash
695 postfix 15 0 1324 1324 1060 S 0.0 0.2 0:00 nqmgr
```

Regarding field descriptions for the output screen of top command, refer to man help pages for top or other Linux documentations.

Example 6-19 is another example for using top to monitor designated processes only. You can specify up to 20 processes for top to monitor by using “p” command option. In addition, by taking advantage of the function provided by UNIX/Linux shell, you can use “pgrep db2sampl” as the example shows to get the process ID for process db2sampl and add it into top monitor process list directly. In addition to “p” command option, the “d” command option is also specified in the command. It means the output screen of top will be refreshed every 1 second.

Example 6-19 Using top to monitor specified processes

```
[root@localhost root ]#top d 1 p `pgrep db2sampl`p 3766 p 3767
9:16pm up 1:07,3 users,load average:2.54,0.99,0.38
3 processes:3 sleeping,0 running,0 zombie,0 stopped
CPU states:6.3%user,37.3%system,0.6%nice,55.6%idle
Mem:255448K av,242584K used,12864K free,0K shrd,16156K buff
Swap:265064K av,0K used,265064K free 148216K cached
PID USER PRI NI SIZE RSS SHARE STAT %CPU %MEM TIME COMMAND
4365 db2inst1 15 0 7744 7728 3856 S 0.4 3.0 0:00 db2sampl
3766 db2inst1 15 0 40732 39M 40564 S 0.0 15.9 0:00 db2sysc
3767 db2inst1 15 0 24308 23M 24152 S 0.0 9.5 0:00 db2sysc
```

Vmstat

This tool can be used to report virtual memory statistics as well as information about processes, paging, block IO and CPU activities. It is contained in the procps package for the Linux platform. The first report produced gives averages since the last reboot. Additional reports give information on a sampling period of length delay. The process and memory reports are instantaneous in either case.

The following are field descriptions for the output report generated by vmstat. To acquire more information regarding this command, refer to *man* help pages for vmstat or other Linux documentations.

- **Procs**

  r: The number of processes waiting for run time
  b: The number of processes in uninterrupted sleep
  w: The number of processes swapped out but otherwise can be run. This field is calculated, but Linux never desperation swaps.

- **Memory**

  swpd: The amount of virtual memory used (KB)
  free: The amount of idle memory (KB)
  buff: The amount of memory used as buffers (KB)

- **Swap**

  si: Amount of memory swapped in from disk (KB/s)
  so: Amount of memory swapped to disk (KB/s)

- **IO**

  bi: Blocks sent to a block device (blocks/s)
  bo: Blocks received from a block device (blocks/s)

- **System**

  in: The number of interrupts per second, including the clock
  cs: The number of context switches per second

- **CPU**

  These are percentages of total CPU time.

  us: User time
  sy: System time
  id: Idle time

A sample report generated by vmstat is shown in Example 6-20.

Example 6-20 Using vmstat to observe virtual memory statistics

```
[root@UDBLNX06 instance ]#vmstat -n 1
procs memory swap io system cpu
r b w swpd free buff cache si so bi bo in cs us sy id
0 0 0 47144 8576 16800 466852 0 5 37 121 569 159 3 1 96
4 0 2 47140 8572 16668 466336 4 0 2572 1160 2557 2225 48 20 33
3 1 0 47140 8568 16672 466636 0 0 2560 2012 2190 1754 75 17 9
2 0 0 47140 8568 15644 467908 0 0 3076 1536 2444 2065 62 19 20
```

```
3 0 0 47140 8568 15656 467680 0 0 2564 1728 2535 2018 67 17 16
3 1 0 47136 8572 15212 467792 4 0 3076 1684 2574 2030 68 22 11
5 0 1 47136 8568 15112 467936 0 0 2564 1696 2559 2045 65 20 16
4 1 0 47136 8568 15096 468916 0 0 2564 1412 2190 1678 59 15 26
4 1 0 47136 8568 15092 468340 0 0 3072 1828 2573 2023 68 20 12
1 0 0 47136 8572 14956 468048 0 0 2564 1720 2561 2117 66 22 13
3 1 0 47132 8580 14968 468040 4 0 3080 1684 2543 1975 69 17 15
3 1 0 47132 8572 14476 468560 0 0 3076 1720 2610 2046 66 20 15
2 0 0 47132 8568 14476 469168 0 0 2560 1444 2194 1690 56 15 29
2 1 0 47132 8568 14488 468792 0 0 2564 1540 2487 1965 63 21 17
3 1 0 47128 8572 14504 468532 4 0 2572 1712 2522 2003 66 18 17
```

Iostat

The iostat command is used for reporting Central Processing Unit (CPU) statistics and monitoring system input/output device statistics for devices and partition. The report generated by iostat command could be used to assist in changing system configuration to better balance the input/output load between physical disks.

The first report generated by the iostat command provides statistics concerning the time since the system was booted. Each subsequent report covers the time since the previous report. The iostat command is contained in the sysstat package (the sar command is also included in this package). Before using this command, you need to install this package.

There are some useful command options that can be used for iostat command. For example, using “-k” to display statistics in kilobytes instead of blocks, using “-x” to obtain extended statistics information for devices, and using “-t” to print the time for each report displayed, and so on. For details, refer to man help pages for iostat or other Linux documentations.

Example 6-21 is an example of using iostat command to monitor CPU and device I/O statistics.

Example 6-21 Using iostat to get CPU and I/O statistics

```
[root@UDBLNX06 root ]#iostat 1
Linux 2.4.18-14 (UDBLNX06)11/06/2002
avg-cpu:%user %nice %sys %idle
1.98 0.00 0.66 97.36
Device:tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
dev3-0 5.44 50.72 119.68 1269102 2994466
dev3-1 0.00 0.01 0.00 304 0
avg-cpu:%user %nice %sys %idle
64.00 0.00 19.00 17.00
Device:tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
dev3-0 184.00 5120.00 3216.00 5120 3216
```

```
dev3-1 0.00 0.00 0.00 0 0
avg-cpu:%user %nice %sys %idle
65.00 0.00 20.00 15.00
Device:tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
dev3-0 208.00 6144.00 3328.00 6144 3328
dev3-1 0.00 0.00 0.00 0 0
avg-cpu:%user %nice %sys %idle
69.00 0.00 18.00 13.00
Device:tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
dev3-0 180.00 5136.00 3192.00 5136 3192
dev3-1 0.00 0.00 0.00 0 0
```

Sar

The sar command is an integrated and powerful System Activity Report (sar) utility. You could use it to collect and report CPU utilization, memory and swap space utilization statistics, I/O and transfer rate statistics, as well as network statistics, and so forth. You can save the output of sar command into a designated file and retrieve information from that file at a later time.

The general syntax to run sar command is:

```
sar [ options ] [ interval [ count ] ]
```

Table 6-3 shows some frequently used command options for sar which are extracted from sar's man pages.

Table 6-3 Frequently used command options for sar

Options	Description
-A	This is equivalent to specifying -bBcdqrRuvwWy -I SUM -I PROC -n FULL -U ALL.
-b	Report I/O and transfer rate statistics.
-c	Report process creation activity.
-d	Report activity for each block device.
-e	Set the ending time of the report.
-f	Extract records from filename (created by the -o filename flag). The default value of the filename parameter is the current daily data file, the /var/log/sa/sadd file. The -f option is exclusive of the -o option.
-n	Report network statistics.

Options	Description
-o	Save the readings in the file in binary form.
-q	Report queue length and load averages.
-r	Report memory and swap space utilization statistics.
-s	Set the starting time of the data, causing the sar command to extract records time-tagged at, or following, the time specified.
-u	Report CPU utilization.
-U	Report CPU utilization for a given processor.
-v	Report status of inode, file and other kernel tables.
-w	Report system switching activity.
-x	Report statistics for a given process.

Some samples of using sar command are provided below. In Example 6-12, at first, sar is started with “-A” command option and the output is saved in sartest.bin file. For testing purposes, you can generate some CPU load for sar to report, for example, loading data into DB2 database in a multiple partition environment. Then you can obtain the desired report from the saved file for different system activities. For example:

```
sar -u -s 11:09:00 -f sartest.bin|more
```

The above command can be used to get CPU utilization information, since a specified starting time.

```
sar -w -c -s 11:09:10 -e 11:09:25 -f sartest.bin|more
```

The above command could be used to get process creation/fork and system context switch information. And the following command could be used to obtain network statistics for a specific network interface:

```
sar -n DEV -s 11:09:25 -f sartest.bin|egrep "eth0|IFACE"
```

See Example 6-22 for details.

Example 6-22 Sar output

```
[root@UDBLNX06 root ]#sar -A -o sartest.bin 2 20
[root@UDBLNX06 root ]#sar -u -s 11:09:00 -f sartest.bin|more
Linux 2.4.18-14 (UDBLNX06)11/07/2002
11:09:01 AM CPU %user %nice %system %idle
11:09:03 AM all 0.00 0.00 0.00 100.00
```

```
11:09:05 AM all 0.50 0.00 0.50 99.00
11:09:07 AM all 0.00 0.00 0.50 99.50
11:09:09 AM all 0.00 0.00 0.00 100.00
11:09:11 AM all 1.00 0.00 1.00 98.00
11:09:13 AM all 8.50 0.00 32.50 59.00
11:09:15 AM all 14.50 0.00 48.00 37.50
11:09:17 AM all 5.50 0.00 6.50 88.00
11:09:19 AM all 17.50 0.00 11.00 71.50
11:09:21 AM all 11.00 0.00 12.50 76.50
11:09:23 AM all 23.50 0.00 9.00 67.50
11:09:25 AM all 67.00 0.00 19.50 13.50
11:09:27 AM all 62.00 0.00 18.00 20.00
11:09:29 AM all 65.00 0.00 20.00 15.00
11:09:31 AM all 63.00 0.00 19.00 18.00
11:09:33 AM all 63.00 0.00 18.00 19.00
11:09:35 AM all 66.00 0.00 18.00 16.00
Average:all 27.53 0.00 13.76 58.71
[root@UDBLNX06 root ]#sar -w -c -s 11:09:10 -e 11:09:25 -f sartest.bin|mor
Linux 2.4.18-14 (UDBLNX06)11/07/2002
11:09:11 AM proc/s
11:09:13 AM 3.00
11:09:15 AM 4.50
11:09:17 AM 1.50
11:09:19 AM 0.00
11:09:21 AM 2.00
11:09:23 AM 0.00
11:09:25 AM 0.00
Average:1.83
11:09:11 AM cswch/s
11:09:13 AM 190.00
11:09:15 AM 1013.50
11:09:17 AM 1349.00
11:09:19 AM 2071.00
11:09:21 AM 2132.00
11:09:23 AM 1643.50
11:09:25 AM 2194.00
Average:1399.83
[root@UDBLNX06 root ]#sar -n DEV -s 11:09:25 -f sartest.bin|egrep "eth0|IFACE"
11:09:25 AM IFACE rxpck/s txpck/s rxbyt/s txbyt/s
11:09:27 AM eth0 1187.00 2459.00 1511133.00 3007010.50
11:09:29 AM eth0 1218.00 2528.50 1558313.00 3078301.00
11:09:31 AM eth0 1219.00 2534.00 1558488.00 3085869.00
11:09:33 AM eth0 1209.00 2504.50 1541410.50 3051547.50
11:09:35 AM eth0 1181.00 2502.50 1504659.50 3050603.50
Average:eth0 1202.80 2505.70 1534800.80 3054666.30
```

As mentioned in “Iostat” on page 279, the sar utility is contained in sysstat package. Before using this command, you need to make sure sysstat package is installed in your system. For details regarding the usage of the sar command, refer to man help pages for sar or other Linux documentation.

Other Linux system monitoring tools

There are also a multitude of Linux system monitoring tools available for monitoring Linux system resource usage, for example, the **`ps`** command is used to report processes status, **`pstree`** can be used to display a tree of processes, **`netstat`** for printing network related information, **`nfsstat`** for NFS specific statistics, and so on. In addition, **`pgrep`** and **`pkill`** commands are very helpful when you want to look up or signal processes based on name and other attributes of processes. And if you like monitoring tools with a graphical user interface, utilities like gnome-system-monitor and ksysguard are also usable. For more information, please refer to Linux specific documentation.

6.5 Performance tuning

When you first install and start to use DB2, all of the configuration parameters will be set for you, tables will be created in default tablespaces, and applications will use DB2 in a certain way. Therefore, *out of the box*, DB2 functions without any need for performance tuning. Indeed our PHP application uses a DB2 database with the default settings. Therefore, for a prototype application or an evaluation of feature function within DB2 there may not be a great need to fine tune that database. All this said and done, if you are planning to develop databases for applications that run in a production environment, then you need to be aware of what can be tuned within DB2 and how to go about tuning DB2.

The topic of performance can have whole chapters dedicated to it and even complete books. Therefore within this section we aim to cover:

- DB2 system design
- Using the configuration parameters
- SQL tuning
- Application considerations

What we will not be covering is:

- Theory behind performance tuning
- Tuning the operating system
- Additional system hardware
- Type of disk storage

Even though the above topics are important and the addition of CPUs, memory, and configuring your RAID array will all make a difference to performance, our aim in this section is to concentrate on the configurable options within DB2.

There are some excellent books available that cover the specifics of performance tuning in much greater detail:

- Chapter 5 of *DB2 UDB Exploitation of the Windows Environment*, SG24-6893
- *Database Performance Tuning on AIX*, SG24-5511 (Although this book is based on the AIX platform it contains some very useful general principles for system design)
- *DB2 Administration Guide: Performance*, SC09-4821

We now continue by covering the features available in DB2 UDB Version 8 that can affect performance.

6.5.1 DB2 system design

When we are designing a database system on Windows or Linux, there are a number of design factors we can consider that can affect the performance of the system. These include:

- 64-bit or 32-bit DB2
- Distribution of data
- Tablespace type
- Table type

64-bit or 32-bit DB2

Since version 7.2, DB2 has allowed the creation of 64-bit databases as well as 32-bit databases on platforms with supported hardware. Our application uses the 32-bit trial code. If you have the appropriate 64-bit architecture you may want to consider installing the 64-bit version of DB2. The Windows trial code can be installed on either a 32-bit or a 64-bit server, but the Linux code is for 32-bit architecture only.

64-bit DB2 allows the creation of larger buffer pools, sort heaps, and other memory-intensive resources. Exploiting 64-bit features will also result in performance improvements, as more data can be moved in a single CPU cycle. There are no connectivity issues between 32-bit applications and 64-bit servers, because the new DB2 *common client* provides connectivity throughout the whole DB2 family.

Distribution of data

If you have a system that has multiple physical disk drives available to DB2 you would probably want to split out the DB2 data across these multiple disks to reduce I/O contention as much as possible. In an ideal environment this could mean, separate disks for:

- Database logs

- Mirrored database logs (if this option has been set up in the database configuration file)
- Temporary space (used for sorting data and storing intermediate result sets)
- Table data
- Index data

Due to cost constraints, this ideal scenario may not be completely possible. However, you should try to maximize the utilization of the available disks in your system. In our environment which we tested and created our application, our systems only had two physical drives. In this case we installed DB2 and kept the logs on one drive and created the database and the physical data on the other.

Tablespace type

Once we have created a DB2 system and chosen our disk distribution, then we can decide how we want to store the data. DB2 has two types of tablespaces to store data, System Managed Space (SMS) and Database Managed Space (DMS).

- SMS is the default tablespace type and three tablespaces of this type are created after the install of DB2. These tablespaces allow the operating system to allocate and manage the space where the table data resides.
- DMS tablespaces give DB2 the ability to control storage space. The amount of space allocated to a DMS tablespace must exist upon creation, since these files are allocated upon creation, as opposed to SMS, which grow within the specified file system.

Both types of tablespaces have their own advantages and disadvantages, but in general SMS is easier to manage, whereas DMS can be faster. DMS is also more flexible. The regular data, large data, and indexes can be split between different DMS tablespaces. DMS can be created on raw unformatted disks. Recent enhancements to DMS tablespaces have meant it is now possible to add containers to a tablespace, remove containers and reduce the size of containers

The Create Tablespace Wizard in the Control Center guides you through creating a tablespace.

Within Control Center:

1. Launch the wizard from **Tools —> Wizards —> Create Tablespace Wizard** then click **OK**.
2. Select your System, Instance and Database and click **OK**.

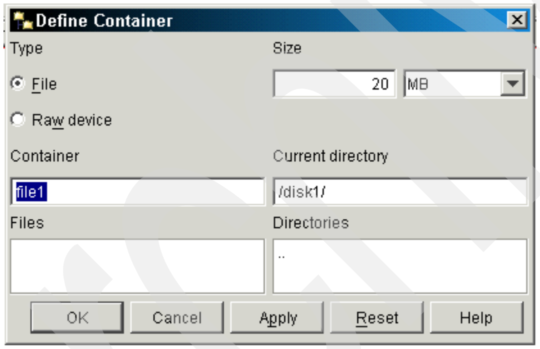
Within the Create Tablespace Wizard:

3. Name your tablespace and click **Next**.

4. Select Regular and click **Next**.
5. You will be using the default bufferpool, so click **Next**.

> **Note:** If you were planning to create tables within the tablespace that are greater than the default 4K pages (such as a table with a single `varchar(4001)` column) you would need to click **Create** here and create an appropriately large bufferpool.

6. For our example, we want to select a DMS tablespace so that we can split our indexes and large objects (LOBs). Then click **Next**. If we had selected to create a Large tablespace in Step 4, then the SMS option would be greyed out.
7. For our example, we are going to create two 20 MB files as our tablespace containers. Click **Add** to add the containers. Click **OK** once you have entered the details for the tablespace, which should be similar to Figure 6-53, and then click **Add** to add another container. You can specify the size of the tablespace in either MB or 4K pages.



● File
○ Raw device

Size: 20 MB

Container: file1

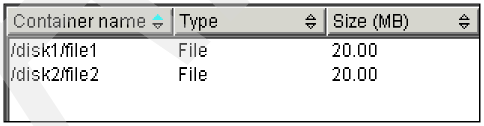
Current directory: /disk1/

Files

Directories: ..

Figure 6-53 Define Container window within the Create Tablespace Wizard

8. Click **Next** once you have added all the containers you want (Figure 6-54).



Container name	Type	Size (MB)
/disk1/file1	File	20.00
/disk2/file2	File	20.00

Figure 6-54 Multiple containers defined within a tablespace

Note: If you specify multiple containers, DB2 will automatically stripe the data across all of the containers. If you are using a RAID disk array then the striping of data across multiple disks will already be striped by the disk array. Therefore, you would only define one physical drive to DB2, this will require the setting of a DB2 registry variable discussed in 6.5.2, “Using the DB2 configuration parameters” on page 291.

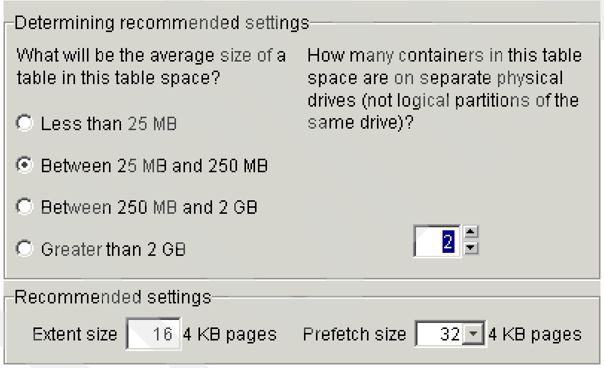
9. This will suggest an extent and prefetch size for the tablespace.

   The extent size recommendation is based on the expected data size in the tables (in general terms, the smaller the data volumes, the smaller the extent size). In our example, we are going to select Between 25 MB and 250 MB, which gives an extent size recommendation of 16 4K pages.

   As our two files sit on two different disks, we will select two physical drives, which will change our recommended prefetch size to be double our extent size. This means that when data is prefetched from disk into the bufferpool, it can be performed in parallel on the two physical disks.

   In our example, the wizard looks like Figure 6-55.

   Click **Next**.



Determining recommended settings

What will be the average size of a table in this table space?

○ Less than 25 MB

● Between 25 MB and 250 MB

○ Between 250 MB and 2 GB

○ Greater than 2 GB

How many containers in this table space are on separate physical drives (not logical partitions of the same drive)? 2

Recommended settings

Extent size 16 4 KB pages Prefetch size 32 4 KB pages

Figure 6-55 Creating tablespace with extent size of 16 and prefetch size of 32

10. If you know the exact characteristics of your disks then enter them here; otherwise, choose the best general match and click **Next**.

11. Click to enable dropped table recovery as you can restore a dropped table more easily. Click **Next**.

12. In the summary window click **Show SQL** to show the statement you have created, it should resemble Example 6-23.

Example 6-23 Create tablespace statement

```
CREATE  REGULAR  TABLESPACE TEST PAGESIZE 4 K  MANAGED BY DATABASE  USING (
FILE '/disk1/file1' 5120,  FILE '/disk2/file2' 5120 ) EXTENTSIZE 16 OVERHEAD
10.5 PREFETCHSIZE 32 TRANSFERRATE 0.33 BUFFERPOOL IBMDEFAULTBP DROPPED TABLE
RECOVERY ON@
COMMENT ON TABLESPACE TEST IS ''@
```

13.Click **Finish** to create the tablespace.

Within Control Center:

14.Click the **Tablespaces** folder to show the new tablespace.

In a production system it is generally advisable to create the following tablespaces:

- At least one for regular data (DMS)
- At least one for long data, if required (DMS)
- At least one for indexes (DMS)
- Drop the existing temporary tablespace (TEMPSPACE1) and create a new SMS temporary tablespace.

If possible the above tablespaces should be striped across multiple disks either using striping defined within the tablespace definition or by using a disk array.

Table Type

As with creating tablespaces tables can now be created in two distinct ways that can significantly effect performance:

- The classic way to create a table is to associate the table with a clustering index (typically the primary key) so that as much as possible the order of entries in the index corresponds with entries in the table. This makes queries on the clustering index perform well, because the data is ordered in the correct way. However, queries using other indexes will not perform as well, because the data is not ordered in the same way as the index.
- The new way to create a DB2 table is using the technique of Multi-Dimensional Clustering (MDC). When you create a table you specify multiple dimensions (which can be single or multiple columns) that you want to cluster by. This means that any rows of data that have the same clustering dimension values will be stored in the same portion of disk (extent). This can enable queries with predicates on all, or some, of the dimensions to perform better. Even though you have clustered the table, you are still able to create regular indexes on the table, which ensures that queries that do not use the clustering dimensions will still perform.

Figure 6-56 illustrates the difference between using a single clustering index and using MDC.

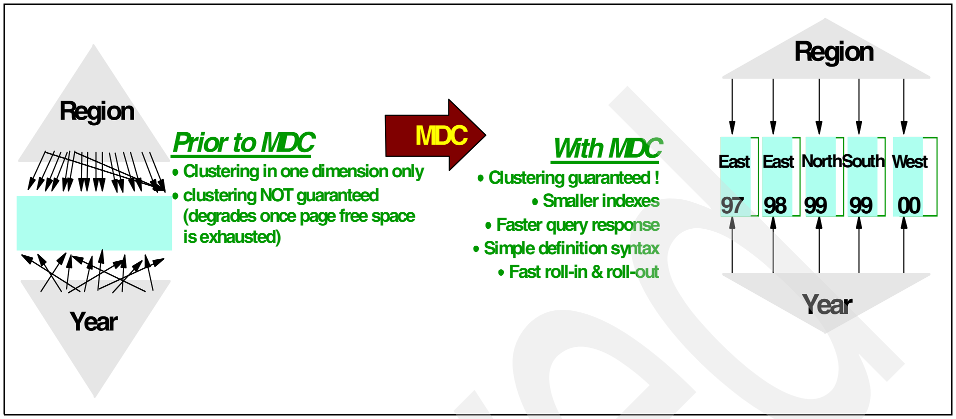

Region
Year
Prior to MDC
• Clustering in one dimension only
• clustering NOT guaranteed (degrades once page free space is exhausted)
MDC
With MDC
• Clustering guaranteed !
• Smaller indexes
• Faster query response
• Simple definition syntax
• Fast roll-in & roll-out
Region
East
East
North
South
West
97
98
99
99
00
Year

Figure 6-56 Comparison of a traditional clustering index and MDC

There are two important considerations regarding the use of MDC in a table design.

- The clustering dimensions have to be chosen carefully. You need to ensure that the columns that you are clustering by have a low cardinality (there aren't many distinct values). A good example is `region` (north, south, east, west), a bad example is `customer_number` (potentially thousands+).
- As each unique combination of clustering dimensions creates at least one extent, the extent size should be kept small (8 rather than 64).

If we consider the `CUSTOMER` table from our application. This table is a good candidate for MDC as it contains two columns that have a low cardinality, `c_nationkey` and `c_mktsegment`.

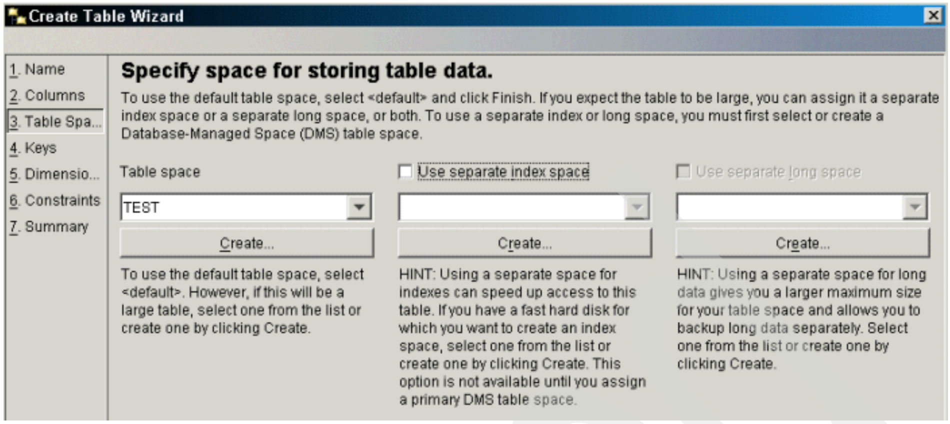
> **Note:** We use the following SQL statement on the customer table to calculate how many extents would be created by clustering on these dimensions:
>
> ```
> select count(*) from (select distinct c_nationkey,c_mktsegment from
> customer) TabA
> ```

Example 6-24 shows how the two DDL statements for the customer table would compare.

Example 6-24 Comparison of DDL statements

```
--Traditional DDL

CREATE TABLE DB2INST1.CUSTOMER  (
        C_CUSTKEY INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY ( START
WITH +1 , INCREMENT BY +1 , CACHE 20 ) ,
        C_NAME VARCHAR(25) NOT NULL ,
        C_ADDRESS VARCHAR(40) NOT NULL ,
        C_NATIONKEY INTEGER NOT NULL ,
        C_PHONE CHAR(15) NOT NULL ,
        C_ACCTBAL DECIMAL(15,2) NOT NULL ,
        C_MKTSEGMENT CHAR(10) NOT NULL ,
        C_COMMENT VARCHAR(117) NOT NULL );

--MDC DDL

CREATE TABLE DB2INST1.CUSTOMER_MDC  (
        C_CUSTKEY INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY ( START
WITH +1 , INCREMENT BY +1 , CACHE 20 ) ,
        C_NAME VARCHAR(25) NOT NULL ,
        C_ADDRESS VARCHAR(40) NOT NULL ,
        C_NATIONKEY INTEGER NOT NULL ,
        C_PHONE CHAR(15) NOT NULL ,
        C_ACCTBAL DECIMAL(15,2) NOT NULL ,
        C_MKTSEGMENT CHAR(10) NOT NULL ,
        C_COMMENT VARCHAR(117) NOT NULL )
ORGANIZE BY (C_NATIONKEY, C_MKTSEGMENT);
```

The Create Table Wizard (**Tools —> Wizards —> Create Table Wizard)** available from within the Control Center includes support for creating MDC tables. The wizard also includes the facility to specify where to place the regular data, long data, and index data, as Figure 6-57 shows.


Create Table Wizard

1. Name
2. Columns
3. Table Spa...
4. Keys
5. Dimensio...
6. Constraints
7. Summary

Specify space for storing table data.

To use the default table space, select <default> and click Finish. If you expect the table to be large, you can assign it a separate index space or a separate long space, or both. To use a separate index or long space, you must first select or create a Database-Managed Space (DMS) table space.

Table space: TEST

□ Use separate index space

□ Use separate long space

To use the default table space, select <default>. However, if this will be a large table, select one from the list or create one by clicking Create.

HINT: Using a separate space for indexes can speed up access to this table. If you have a fast hard disk for which you want to create an index space, select one from the list or create one by clicking Create. This option is not available until you assign a primary DMS table space.

HINT: Using a separate space for long data gives you a larger maximum size for your table space and allows you to backup long data separately. Select one from the list or create one by clicking Create.

Figure 6-57 Options for storing different object types in different tablespaces

This step, whether specified in the wizard or as part of a `create table` statement, is important, because if you want to use this performance feature, you have to specify these additional tablespaces when you create the table, not when you create the index.

6.5.2 Using the DB2 configuration parameters

Once you have designed and implemented your DB2 system, you can then apply some configurations to customize the environment. There are three main levels for configuring a DB2 system, all of which have been discussed earlier in this chapter:

- The db2set profile registry — discussed in 6.1.4, “The DB2 profile registry” on page 206
- The database manager configuration file — discussed in 6.1.1, “Database manager configuration” on page 197
- The database configuration file — discussed in 6.1.2, “Database configuration file” on page 200

db2set

As discussed earlier, the `db2set` parameters are changed from a command line/shell, and requires that the db2 instance is restarted for the change to be effective. Table 6-4 provides some examples of these parameters that can make a performance difference.

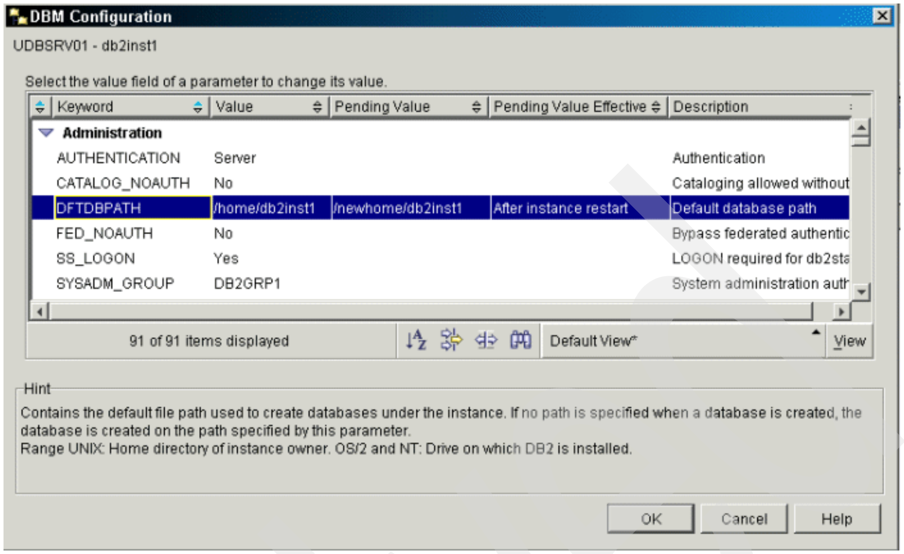
Table 6-4 Sample db2set parameters

db2set parameter	Why
DB2_PARALLEL_IO	While reading or writing data from and to tablespace containers, DB2 may use parallel I/O for each tablespace value that you specify. The degree of parallelism is determined by the prefetch size and extent size for the containers in the tablespace. When this variable is not enabled, the number of prefetcher requests created is based on the number of containers in the tablespace. Therefore without this parameter being set, a tablespace with one RAID array would only use 1 prefetch even though the array contains multiple disks. The command is: **db2set DB2_PARALLEL_IO=*** (for all tablespaces)
DB2NTNOCACHE	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. This variable is particular to Windows NT only

Database manager configuration file (dbm cfg)

As discussed earlier in 6.1.1, “Database manager configuration” on page 197, the dbm cfg file controls the allocation of resources across the entire DB2 instance, which can include multiple databases. The visualization of this file has been improved in DB2 Version 8 so that when a change is made you are informed when this change will be implemented.

From within Control Center: If you right-click your database instance (Linux: **db2inst1**; Windows: **DB2**) and select **Configure Parameters**, you will see a window similar to Figure 6-58.



UDBSRV01 - db2inst1

Select the value field of a parameter to change its value.

Keyword	Value	Pending Value	Pending Value Effective	Description
Administration				
AUTHENTICATION	Server			Authentication
CATALOG_NOAUTH	No			Cataloging allowed without
DFTDBPATH	/home/db2inst1	/newhome/db2inst1	After instance restart	Default database path
FED_NOAUTH	No			Bypass federated authentic
SS_LOGON	Yes			LOGON required for db2sta
SYSADM_GROUP	DB2GRP1			System administration auth

91 of 91 items displayed

Default View*

Hint

Contains the default file path used to create databases under the instance. If no path is specified when a database is created, the database is created on the path specified by this parameter.
Range UNIX: Home directory of instance owner. OS/2 and NT: Drive on which DB2 is installed.

Figure 6-58 Updating the DBM Configuration file

In this example, we clicked on the value for DFTDBPATH to change it from `/home/db2inst1` to `/newhome/db2inst1`

DB2 provides a hint of what the parameter does and also in some cases what the allowable range of values are.

The Pending Value Effective column informs us when the change we made will be performed to the system. (For DBM CFG parameters this is generally after the instance is restarted.)

> **Tip:** If you make a change that you wish to undo, then simply click the **Value** field and click the **Reset** button to reset the value back to the system default.

Table 6-5 below provides some examples of dbm parameters that can make a performance difference.

Table 6-5 Sample dbm cfg parameters

dbm cfg parameter	**Why**
KEEPFENCED	This parameter indicates whether or not a fenced mode process is kept after a fenced mode routine call is complete. Fenced mode processes are created as separate system entities in order to isolate user-written fenced mode code from the database manager agent process. If we had a stored procedure that executed in a fenced mode process. If this parameter is set to **no**, each transaction incurs the overhead of creating a new fenced mode process, resulting in some performance reduction. If, however, this parameter is set to yes, each transaction would try to use an existing fenced mode process, which would avoid this overhead.
MAXAGENTS	This parameter indicates the maximum number of database manager agents, whether coordinator agents or subagents, available at any given time to accept application requests.This parameter can be useful in memory constrained environments to limit the total memory usage of the database manager, because each additional agent requires additional memory.

Database configuration file (db cfg)

As introduced earlier in 6.1.2, "Database configuration file" on page 200, and just as the `dbm cfg` file configures each DB2 instance, each database configuration file configures a DB2 database. When you create a database, the `db cfg` parameters are set to defaults. If you want to change these parameters to better suit your application requirements then the best initial step is to use the Configuration Advisor.

We have already introduced the mechanics of using the Configuration Advisor in 3.4, "Configuration Advisor" on page 64, so here we provide details about using it to run against our sample application. We also discuss some of the performance changes recommended by the advisor.

From within Control Center:

- Launch the advisor from **Tools —> Wizards —> Configuration Advisor** then click **OK**.

- Choose the database you want to work with and click **OK**.

From within the Configuration Advisor, we have 10 steps to configure our database:

1. Introduction: This page lists the database for which you want to configure performance parameters. Verify that the correct database is shown. Click **OK.**
2. Server: Use this page to specify what percentage of the server's memory is to be used by the database manager. If other applications are running on the server, set the value to less than 100%.

   In our example we are also running Apache on the same server, so we’ll accept the default of 80%. Click **Next.**
3. Workload: On this page, indicate the type of workload for which the database is used. You can indicate if your database is used mainly for queries (such as in a data warehousing environment), for transactions (such as order entry), or mixed (for a combination of queries and transactions).

   In our application we will be inserting data and querying that data so keep the default and click **Next**.
4. Transactions: Use this page to describe a typical SQL transaction for the database. Indicate whether the average number of SQL statements per unit of work is less than 10, or more than 10. Also indicate approximately how many transactions per minute run on the database. This information can be garnered from using the event monitoring discussed in the previous section.

   In our application our statements are short, basically consisting of either an insert, update or delete followed by a commit, so we should click the **Fewer than 10** button. As it is a Web based application the default of 60 transactions a second is possible. We will accept this now, but if this usage were to change then we can easily rerun the Advisor and change this value. Click **Next**.
5. Priority: On this page you specify the administration priority for the database. If you optimize for faster transaction performance, the database will take longer to recover. If you optimize for faster data recovery time, transaction performance will be slower.

   In our example, it is important to optimize both, so accept the default of **Both** and click **Next**.
6. Populated: Use this page to indicate whether the database is already populated with data. Data volumes can affect the recommendations made by the advisor.

   As we already have data accept the default of **Yes** and click **OK**.

7. Connections: On this page, indicate the average number local applications and the average number of remote applications that connect to the database.

   In our example the Apache Web server and DB2 are on the same physical machine, so any connections will be local (despite the fact that the requests coming into the Web server are remote http requests). We then change **local** to **10** and **remote** to **3** (allow some remote DBAs to administer the system). Click **Next**.

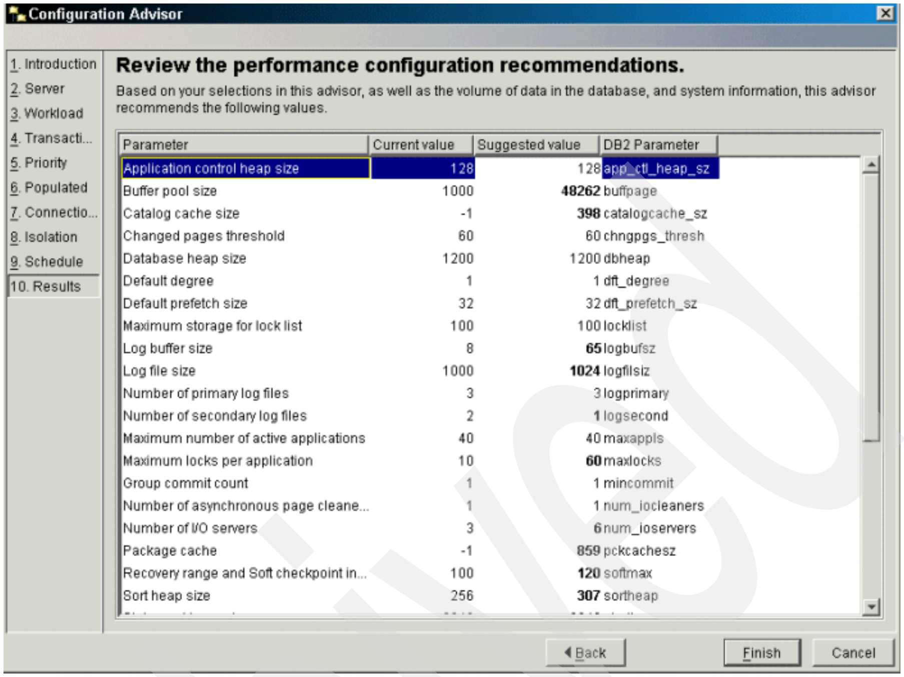
8. Isolation: Use this page to specify the isolation level the database will use when a user either reads or changes data. The isolation level determines the number and duration of the locks on the data. We discuss using isolation levels for applications in 6.5.4, “Application considerations” on page 306.

   Click **Cursor Stability** (which is the DB2 default) and click **OK**.

9. Schedule: Here you can specify how and when you want the generated command to be run. If you want to schedule the task, it is saved within the Task Center.

   In our example we will just run the task, so select the **Run now without saving task history button** and the click **OK**

10. Results: On this page the wizard lists the recommended configuration parameter settings, based on the information you have provided. Your window should resemble Figure 6-59.



Review the performance configuration recommendations.

Based on your selections in this advisor, as well as the volume of data in the database, and system information, this advisor recommends the following values.

Parameter	Current value	Suggested value	DB2 Parameter
Application control heap size	128	128	app_ctl_heap_sz
Buffer pool size	1000	**48262**	buffpage
Catalog cache size	-1	**398**	catalogcache_sz
Changed pages threshold	60	60	chngpgs_thresh
Database heap size	1200	1200	dbheap
Default degree	1	1	dft_degree
Default prefetch size	32	32	dft_prefetch_sz
Maximum storage for lock list	100	100	locklist
Log buffer size	8	**65**	logbufsz
Log file size	1000	**1024**	logfilsiz
Number of primary log files	3	3	logprimary
Number of secondary log files	2	**1**	logsecond
Maximum number of active applications	40	40	maxappls
Maximum locks per application	10	**60**	maxlocks
Group commit count	1	1	mincommit
Number of asynchronous page cleane...	1	1	num_iocleaners
Number of I/O servers	3	**6**	num_ioservers
Package cache	-1	**859**	pckcachesz
Recovery range and Soft checkpoint in...	100	**120**	softmax
Sort heap size	256	**307**	sortheap

Figure 6-59 Output from the Configuration Advisor

Table 6-6 lists some of the recommended changes made by the Configuration Advisor for our environment and the reason for the changes.

Table 6-6 Sample Configuration Advisor recommendation

Parameter	Old Value	New Value	Reason
Buffer pool size	1000	48262	More data pages in memory allows for quicker access. The system has much more memory than 1000 4K pages (4MB) so the bufferpool is increased.
Catalog cache size	-1 (indicates it is a derived value)	398	This means that 398 data pages will be kept in the catalog cache. This will help the performance of queries against the same table as the table information has been cached. (This has been changed from its derived value as the derived value is based on the maximum number of applications, which has been kept at 40.)

Parameter	Old Value	New Value	Reason
Log buffer size	8	65	Increases to the size of the log buffer will result in more efficient logging file I/O, because the log records will be written to disk less frequently and more log records will be written at each time.
Maximum locks per application	10	60	With this increase the application can hold more row locks before the lock is escalated so that the applications locks the whole table. This has probably been changed due to the fact that we chose an isolation level that requires many short locks.
Package Cache	-1	859	DB2 will now have a specific amount of pages that will cache SQL statements which will speed up execution if the same query is required more than once. (This has been changed from its derived value as the derived value is based on the maximum number of applications, which has been kept at 40).

These are just some of the recommended changes, for a greater level of understanding, we recommend:

- Run the Configuration Advisor for yourself.
- Consult Chapter 13 of *Administration Guide: Performance*, SC09-4821, which contains detailed information on the configuration parameters and their dependencies.

As with updates to the `dbm cfg` file, you can make your own changes to the db cfg file by right-clicking on the database name in Control Center and selecting **Configure Parameters**.

Here you can make changes, view the update, see when the update will be implemented (in many cases immediately) and then if necessary reset the update.

6.5.3 SQL tuning

Once you have designed your system and configured your database, you can then start to run some SQL statements. In our application, we have either coded the SQL as ODBC calls or created stored procedures with the statements. Either way we have had control over what statements we created which might not be the case with a third-party application that runs on DB2.

Two of the most common ways that DB2 can be configured to optimize the performance of SQL queries are through:

- Design Advisor (to help with the creation of indexes)
- Changing the dbm cfg and db cfg parameters

Index Creation

We have already introduced the use of indexes in 6.5.1, “DB2 system design” on page 284, where we compared the use of traditional clustered indexes and MDC tables. As well as indexes that physically order our data, there are other indexes that provide pointers to the physical location of the data. There is usually less data to search in these indexes, so it is quicker to search them for the location of the data in the table than to scan through the table itself.

To assist us in the creation of indexes, DB2 provides the Design Advisor which helps us to create the appropriate indexes for our specific workload.

> **Note:** The design advisor uses a command line tool called `db2advis`, which can be used by entering **`db2advis`** at a command prompt/shell.

Our example of using the Design Advisor consists of four steps:

- Running an existing query against the database
- Using the Design Advisor to check if indexes will help the query execution
- Checking that the indexes have been created
- Running the query again with the indexes

From within Command Center we can run a query against the database:

1. Connect to the RBORDERS database.
2. Copy the query in Example 6-25 into the Interactive screen.

Example 6-25 Query to be optimized by Design Advisor

```
select c_name, c_custkey, o_orderkey, o_orderdate, o_totalprice,
sum(l_quantity) from customer, orders, lineitem where o_orderkey in ( select
l_orderkey from lineitem group by l_orderkey having sum(l_quantity) > 252) and
c_custkey = o_custkey and o_orderkey = l_orderkey group by c_name, c_custkey,
o_orderkey, o_orderdate, o_totalprice order by o_totalprice desc, o_orderdate
```

3. Ensure that the Command Center is configured to display an access plan:

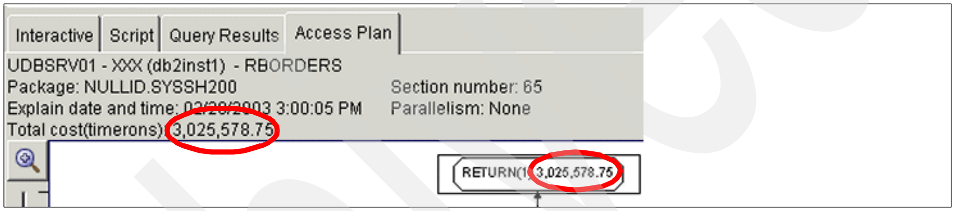
   Command Center —> Options

In the **Access Plan** tab ensure that the **Automatically generate access plan** box is selected. Click **OK**.

4. Execute the query by clicking on the gears icon.

> **Tip:** As well as clicking the gears icon, you can simultaneously press `ctrl-enter` to run the command.

5. The results of the query are generated in the **Query Results** tab and an **Overview** window is generated with the access plan that DB2 used to return the data. The Access Plan tab also contains the access plan.
6. Click the **Access Plan** tab and note the total time taken to run the query and also that all the tables were read directly as the database didn't have any appropriate indexes.



UDBSRV01 - XXX (db2inst1) - RBORDERS
Package: NULLID.SYSSH200 Section number: 65
Explain date and time: 02/28/2003 3:00:05 PM Parallelism: None
Total cost(timerons): 3,025,578.75

Figure 6-60 Query execution time from the Command Center

> **Note:** The Total cost (timerons) is relative measurement to allow the comparison of queries it is not a measurement of real time.

7. Do not close the **Overview** window, because this will allow you to compare the access plan before and after adding the indexes.

From within Control Center, we can launch the Design Advisor:

- Launch the advisor from **Tools —> Wizards —> Design Advisor**.
- Choose to work with the **RBORDERS** database and click **OK**.

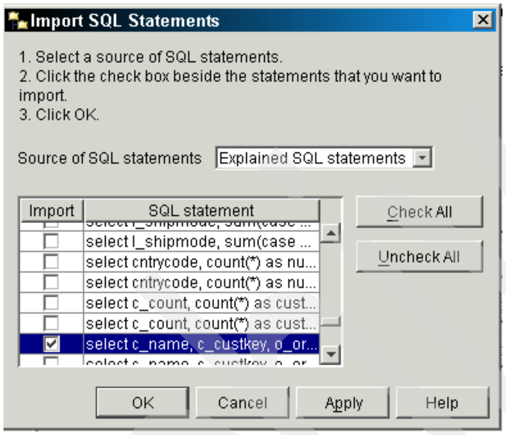
From within the Design Advisor, we have nine steps to create our indexes:

1. Introduction: This page lists the database for which you want to configure performance parameters. Verify that the correct database is shown. Click **Next.**
2. Workload: On this page, you define the workload. You can create a new workload or specify an existing workload. Once you create or specify the workload, you can work with its SQL statements. You can add or import more statements, change the existing statements, or remove statements.

- Create a workload name of **Workload1**.
- Click **Import** to import statements that have been previously run.
- Select **Explained SQL Statements** as the source of the SQL statement. This will query the DB2 explain tables and return the queries, like ours, that have been explained.
- Select your query, as in our query that we explained earlier, it will be displayed as:

  ```
  select c_name, c_custkey, o_or...
  ```

- Figure 6-61 illustrates the **Import SQL Statement** window, click **OK**.



Import SQL Statements

1. Select a source of SQL statements.
2. Click the check box beside the statements that you want to import.
3. Click OK.

Source of SQL statements: Explained SQL statements

Import	SQL statement
☐	select l_shipmode, sum(case ...
☐	select cntrycode, count(*) as nu...
☐	select cntrycode, count(*) as nu...
☐	select c_count, count(*) as cust...
☐	select c_count, count(*) as cust...
☑	select c_name, c_custkey, o_or...

Figure 6-61 Using the Design advisor with a previously explained statement

- A default name of **Explained SQL statements** is given to the statement.
- Click **OK** to close the **Create Workload** window, then click **Next** to move to the next window.

3. Collect Statistics: On this page, you can specify to update the table statistics on selected tables. To get the best recommendations from the wizard, your table statistics should be current.

   Select the tables whose statistics you want to update.

   In our query the tables are:

   - CUSTOMER
   - ORDERS
   - LINEITEM

   Select each of these tables and click the **>** button to move them to the right of the window. Click **Next.**

4. Disk Usage: Use this page to set a maximum disk space usage for the creation of the recommended optimization objects. If disk space is not an issue, and you want the best set of recommendations, do not set a disk space limit.

   Accept the defaults of DEFAULT and no maximum index size and click **Next**.

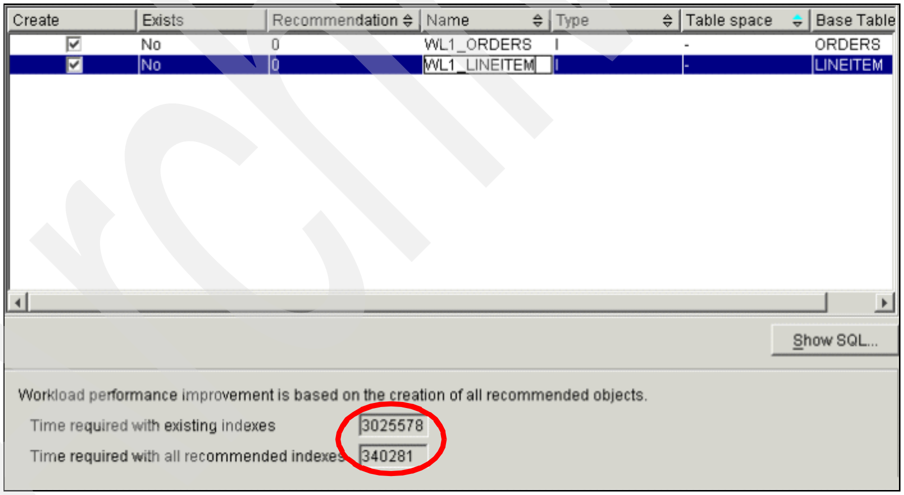
5. Calculate: On this page, you specify when you want DB2 to calculate the optimization recommendations. You can select to calculate immediately or you can schedule a later time. You can also decide how long you want the wizard to calculate the recommendations by setting a maximum calculation time.

   Keep the default start time of **Now** and the 10 minute maximum time default.

   Click **Next** to start the calculation to determine the indexes.

6. Recommendations: On this page, you review the materialized query tables and indexes that DB2 recommends for performance improvement, and select the candidates that you want created. You can also select a tablespace for the materialized query tables.

   Figure 6-62 shows the output from the Design Advisor with the system generated index names changed.



Create	Exists	Recommendation	Name	Type	Table space	Base Table
☑	No	0	WL1_ORDERS	I	-	ORDERS
☑	No	0	WL1_LINEITEM	I	-	LINEITEM

Workload performance improvement is based on the creation of all recommended objects.

Time required with existing indexes 3025578

Time required with all recommended indexes 340281

Figure 6-62 Output from the Design Advisor

You can click the **Name** column and change the system generated index names.

Note the two times, with existing indexes and with all the recommended indexes. The existing time will correspond to the time it took to run the query, while the second time should correspond to the time it takes the query to run after the indexes are applied. Click **Next** to continue.

7. Unused Objects: On this page, you review the materialized query tables and indexes in the database that were not used during the execution of the workload. You can then select to drop these objects. However, this could cause a degradation in the performance of the database for other workloads.

   We will leave all of these current indexes. Click **Next**.

8. *Enabling Scheduling (*Should not appear)

   This page only appears if you have not already created a scheduling database. On this page, you enable the scheduling function by creating a database for the DB2 tools catalog. The scheduling function allows you to run a task immediately, save the task to the Task Center, or schedule a task to be run at a later time.

9. Schedule: On this page, you specify when you want DB2 to perform the workload optimization task. You can run the task immediately, save the task to the Task Center, or schedule the task to be run at a later time.

   We will run the task now, so select **Run now without saving task history** and click **Next**

10. Summary: The Summary page that appears at the end of the wizard is dependent on your selections on the Schedule page. If you selected to run the task immediately, the Summary page shows a list of objects (materialized query tables and indexes) that will be created, and a list of objects that will be dropped. You can review the objects, go back in the wizard and make any necessary changes, and then click Finish to execute the commands.

    If you selected to save the task to the Task Center and run the task later, the Summary page shows the task name, task script, and other information about the task. You can also select to launch the Task Center from this page.

    We selected to run the task immediately, so click **Finish** to create the indexes.

From Control Center we can check that the indexes have been created:

- Navigate to the **Indexes** folder within the **RBORDERS** database.
- Right-click and choose **Refresh**.
- The two indexes should be displayed.

From within Command Center we can rerun the query against the database

1. Our query should still be in the Interactive tab.
2. Execute the query again.
3. The same results will be displayed as well and a new **Overview** window will be created.

4. Click the **Access Plan** tab and note the new time taken to run the query (which corresponds to the estimate from the Design Advisor) and also that the access to the LINEITEM and orders tables was done through indexes.
5. Figure 6-63 shows the two access plans, before and after indexes. Note the presence of the yellow hexagons on the right which indicate that indexes are being used.

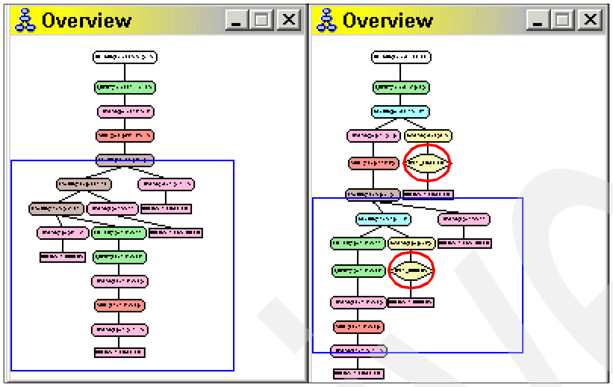


Figure 6-63 Comparison of access plans before and after the addition of indexes

Note: There are some further performance features that can affect the SQL performance with indexes:

- **RUNSTATS**: This command updates statistics about the physical characteristics of a table and the associated indexes. The optimizer uses these statistics when determining access paths to the data. This utility should be called when a table has had many updates, or after reorganizing a table.
- **REORGCHK**: This command line utility calculates statistics on the database to determine if tables or indexes, or both, need to be reorganized or cleaned up. It can either use existing statistics or run runstats to create up to date statistics.
- **REORG**: This command reorganizes an index or a table.
  - The index option reorganizes all indexes defined on a table by rebuilding the index data into unfragmented, physically contiguous pages.
  - The table option reorganizes a table by reconstructing the rows to eliminate fragmented data, and by compacting information. If you specify an index as part of the table reorg then the table will be physically ordered by that index.

With Version 8 of DB2, both table and index reorg can take place online.

Changing the dbm cfg and db cfg parameters

As well as changing the access path through the use of indexes we can make changes to the configuration parameters that, depending on the workload, can make a significant difference.

These are the Database Manager configuration parameters:

- **Maximum query degree of parallelism (MAX_QUERYDEGREE)**

  This parameter specifies the maximum degree of intra-partition parallelism that is used for any SQL statement executing on this instance of the database manager. An SQL statement will not use more than this number of parallel operations within a partition when the statement is executed. The intra_parallel configuration parameter must be set to "YES" to enable the database partition to use intra-partition parallelism.

  In our example, we have kept the system default of ANY (which means system determined).

- **Enable intra-partition parallelism (INTRA_PARALLEL)**

  A value of -1 causes the parameter value to be set to "YES" or "NO" based on the hardware on which the database manager is running. This parameter specifies whether the database manager can use intra-partition parallelism.

  For a multi-CPU system that will be running many large complex queries, setting this parameter to -1 or YES will allow DB2 to split the work between multiple DB2 agents. In a system which is just performing simple inserts or updates, it is generally considered better to have this parameter set to NO.

  In our example we are using the system default of NO.

These are the Database configuration parameters:

- **Default query optimization class (DFT_QUERYOPT)**

  The query optimization class is used to direct the optimizer to use different degrees of optimization when compiling SQL queries. This parameter provides additional flexibility by setting the default query optimization class used when the optimization has not been set when you run the application or when it was bound to the database.

  The default is 5, and the smaller the number, then the less time DB2 spends working out an access plan.

  As our example has a mix of OLTP and DSS activity, we have kept the default of 5.

- **Degree of parallelism (DFT_DEGREE)**

  A value of 1 means no intra-partition parallelism for this database. A value of -1 means the optimizer determines the degree of intra-partition parallelism based on the number of processors and the type of query. This parameter relates to *max_querydegree*, above as the degree of parallelism used is the lowest number of the following:

- max_querydegree configuration parameter
- Application runtime degree (you can set the degree of optimization when you run an application)
- SQL statement compilation degree (when the statement is bound to the database you can specify the optimization degree then)

In our example we have accepted the system default of 1.

6.5.4 Application considerations

Now that we have performed some initial performance tuning on the database system we will now briefly discuss some application considerations. Users of the application want the data to be available, but also want to be reassured that the data they are working with is correct. Therefore, one of the most important considerations in designing a database application are the potential implications of database locking.

Different types of locking

To provide concurrency control and prevent uncontrolled data access, DB2 places locks on tables or table rows. A lock associates a DB2 resource with an application, called the lock owner, to control how other applications can access the same resource. Most database applications access data through cursors (which allow you to navigate through data one row at a time) so the crucial performance and consistency decision is how to lock the rows of data you have returned.

Isolation levels are used to determine how data is locked or isolated from other processes while the data is being accessed. The general tip is to use the minimum isolation level that satisfies your application needs. The DB2 isolation levels are as follows:

- **Repeatable Read**: Repeatable Read (RR) locks all the rows an application references within a unit of work. The Repeatable Read application can retrieve and operate on the rows as many times as needed until the unit of work completes. However, no other applications can update, delete, or insert a row that would affect the result table, until the unit of work completes. Repeatable Read applications cannot see uncommitted changes of other applications.

  This is the highest level of locking and the one that will require the most locks

- **Read Stability**: Read Stability (RS) locks only those rows that an application retrieves within a unit of work. It ensures that any qualifying row (that is, one that satisfies the predicates) read during a unit of work is not changed by other application processes until the unit of work completes, and that any row changed by another application process is not read until the change is committed by that process.

The objective of this level is to provide both a high degree of concurrency as well as a stable view of the data.

- **Cursor Stability**: Cursor Stability (CS) locks any row accessed by a transaction of an application while the cursor is positioned on the row. This lock remains in effect until the next row is fetched or the transaction is terminated. However, if any data on a row is changed, the lock must be held until the change is committed to the database.

  This is the default isolation level and should be used when you want the maximum concurrency while seeing only committed rows from other applications. This is the isolation level that we are using in our sample application.

- **Uncommitted Read**: Uncommitted Read (UR) allows an application to access uncommitted changes of other transactions. The application also does not lock other applications out of the row it is reading, unless the other application attempts to drop or alter the table.

  The Uncommitted Read isolation level is most commonly used for queries on read-only tables, or if you are executing only select statements and you do not care whether you see uncommitted data from other applications.

When to define the isolation level

Although the application interfaces that interact with DB2 all use the same isolation level, the definition of this isolation level depends on the interface:

- **Embedded SQL**: For an application written in a supported compiled language there is an ISOLATION option to change the isolation level. If you create a bind file at precompile time, the isolation level is stored in the bind file. If you do not specify an isolation level at bind time, the default is the isolation level used during precompilation. If you do not specify an isolation level, the default of cursor stability is used.

- **REXX**: When a database is created, multiple bind files that support the different isolation levels for SQL in REXX are bound to the database. Other command-line processor packages are also bound to the database when a database is created. REXX and the command line processor connect to a database using a default isolation level of cursor stability.

  This default can be changed by going to a DB2 Command Window/shell and typing:

  ```
  db2 change sqlisl to ur
  ```

This has to be changed before you connect to a database

- **CLI/ODBC**: The isolation level can be changed at runtime using the `CHANGE ISOLATION LEVEL` command
- **JDBC/SQLj**: You can change the value using the `setTransactionIsolation` method in the java.sql interface connection.
- **Statement**: Use the WITH clause. The statement-level isolation level overrides the isolation level specified for the package in which the statement appears.

  So in our application even though the isolation level is Cursor Stability we can still run queries that use a different isolation level as in Example 6-26.

Example 6-26 Changing the isolation level of a query

```
connect to rborders user db2inst1 using db24me;

select * from nation fetch first row only;
select * from nation fetch first row only with UR;
connect reset;

change sqlisl to ur;
connect to rborders user db2inst1 using db24me;

select * from nation fetch first row only;
select * from nation fetch first row only with CS;

connect reset;
```

7

Additional DB2 features

This chapter provides information on some additional features included within DB2. These features have been placed together in this chapter as they are all extensions to the general usage of DB2. Most of these topics are sufficiently detailed to have warranted redbooks in their own right. This chapter will then seek to introduce these features, discuss how they can be used, and provide references to additional material. The other facet shared by these topics is that they are included as part of the base DB2 product and also the trial code version.

In this chapter, these topics are discussed:

- Replication
- Data Warehouse Center
- DB2 and XML
- Accessing WebSphere MQ

7.1 Replication

This section introduces the scenarios where replication may be appropriate and also the architecture of the DB2 replication solution.

7.1.1 What is replication?

Replication is the copying of data from one place to another. Data can be extracted by programs, transported to some other location, and then loaded at the receiving location. A more efficient alternative is to extract only the changes since the last processing cycle and transport/apply those to the receiving location. During this replication process data may be filtered and transformed.

There may be other requirements for replication, such as time constraints. In most cases, replication must not interfere with existing applications and have minimal impact on production systems. These replication processes need to be managed and monitored.

7.1.2 Why use replication

There are different business justifications for using replication, these can be categorized as:

- Distribution of data to other locations
- Consolidation of data from other locations
- Bidirectional exchange of data with other locations

Distributing data to other locations

Distribution of data involves moving all or a subset of the data to one or more locations. Data can be copied to a central data warehouse or decision support system. This allows you to leverage your enterprise data with business intelligence tools, while maintaining the security and performance of your production applications.

Distribution of data can also be used to provide data to applications in the same or different environments. This can be as simple as maintaining a copy of the production data on another similar system or as complex as providing filtered and transformed data to a new application.

Consolidation of data from other locations

You may have your data stored on many different distributed systems. Replication offers you the capability to copy changes from each of the distributed sites to a central site for analysis, reporting, and for enterprise application processing.

Consolidation of data can be very useful for business intelligence applications such as OLAP or Data Mining.

Bidirectional exchange of data (Update Anywhere)

If the data can be updated at multiple locations, then replication must process changes made at any of the sites in a coordinated fashion. One location serves as the master location and distributes changes to the target locations. Changes made at the targets flow to other target sites through the master. Bidirectional replication can be used for mobile applications where the target may be a computer in a branch office or in a delivery truck. Often, there are many targets and they are only occasionally connected to the source system. This is sometimes called *master-slave* replication.

Another type of bidirectional replication does not have a designated master location. Each location copies changes from all other locations directly. This is often called *multi-master* or *peer-to-peer* replication. Peer-to-peer replication can be used to maintain disaster recovery sites, to provide fail-over systems for high availability, and to balance query workload across multiple locations.

7.1.3 What are the components of DB2 replication?

The IBM replication solution has four components:

- Administration
- Capture
- Apply
- Alert Monitor

The four components communicate via relational tables, called control tables. These control tables are created and populated using the Replication Center. The Capture, Apply, and Alert Monitor programs read and update information in the control tables.

Administration

The Replication Center is a graphical user interface used to define replication sources and map sources to targets. It is also used to manage and monitor the Capture and Apply processes on local and remote systems. The Replication Center runs on both Windows and Linux systems and must have connectivity to both the source and target servers. The Replication center can be invoked from a DB2 Server or the DB2 Administration client by either:

- Clicking **Start —> Programs —> IBM DB2 —> General Administration Tools —> Replication Center**
- Running the `db2cc -rc` command

Capture

Changes to DB2 source tables are captured by a Capture program running at the source server. In our environment the DB2 source server can either be Windows or Linux. The captured changes are stored in a table local to the source table and are automatically deleted after they have been applied.

When changes are made to the source table, DB2 writes log (journal) records. These log records are used for database recovery and for replication. Each source table has a corresponding Change Data (CD) table (automatically created by the Replication Center) where the captured changes are stored.

DB2 Capture holds the changes in memory until a COMMIT is issued for those changes. When a COMMIT is issued for a transaction that involves replication source tables, Capture inserts the captured changes into the appropriate CD tables and stores the COMMIT information in the Unit of Work (UOW) control table. When Capture detects a ROLLBACK, it removes the associated changes from memory.

Apply

Captured changes are applied to target tables by Apply programs. The Apply program can run on any server and must have connectivity to both the source and the target servers. Data can be filtered by column, filtered by row, joined with other data (using views), and transformed with SQL expressions during the Apply process.

Example 7-1 below illustrates the Capture and Apply processes.

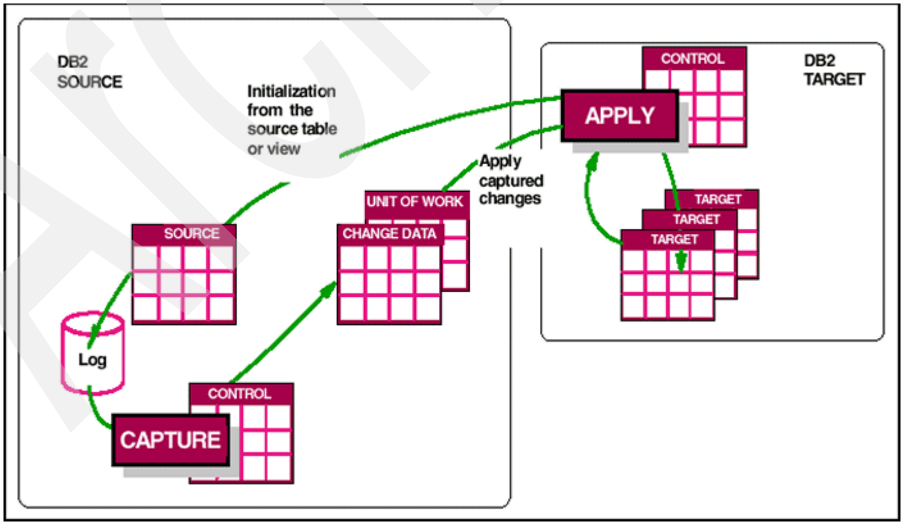


Figure 7-1 Capture and Apply

The Replication Center is used to map a source table or view to a target table. You define a subscription set which is a group of one or more target tables (called subscription members) that will be processed as a unit by Apply. The changes from the CD tables are applied for each table separately, in the same order they occurred at the source. A single COMMIT is issued after the last member in the set is processed.

Apply selects from the source tables for the first initialization of the target tables, using any transformations or filtering you defined. This is called full refresh. After the full refresh, Apply selects changes from the CD tables and applies those changes to the target tables.

Apply can be run as a batch process or as a task that runs all the time. You specify the schedule for replication when you define the subscription set. This schedule can either be time-based or event based.

Alert Monitor

The Replication Alert Monitor is included with DB2 V8 for Windows and Linux. It can be used to monitor replication on those platforms as well as other DB2 platforms. The Alert Monitor has its own set of control tables, defined with the Replication Center.

Replication administrators define thresholds and events through the Replication Center for Capture and Apply servers. You can also define users or groups of users to receive e-mail notification when an alert occurs. The server where the monitor runs is called the Monitor Server and can monitor one or more local and/or remote servers. The Alert Monitor program collects information from the capture and apply control tables. It also uses the Database Administration Server (DAS) installed on the Capture and Apply servers to receive remote commands and supply system information.

7.1.4 What can you do next?

Replication is a detailed subject. For more information about DB2 replication, replicating to and from Infomix, or working through an example scenario, we recommend these books:

- *A Practical Guide to DB2 UDB Data Replication V8*, SG24-6828
- *DB2 UDB Replication Guide and Reference*, SC27-1121 (Chapter 15 includes a step-by-step replication scenario)
- *DB2 UDB Guide to GUI Tools or Administration and Development*, SC09-4851 (Chapter 4)

7.2 Data Warehouse Center

This section introduces the concepts and architecture that comprise this part of the data warehousing capabilities of DB2.

Data Warehousing allows you to create stores of informational data, which is data that is extracted from the operational data and then transformed for decision making.

The typical operation of a data warehousing tool can involve:

- Extracting data from the operational database (such as our sample application)
- Cleaning the data
- Performing calculations to summarize the data
- Writing the summarized data to a target in a separate database

The outcome is that users can then query a separate database (the warehouse) which is more specialized to their requirements and won't impact the performance of the operational database.

7.2.1 What is the Data Warehouse Center?

The Data Warehouse Center is a component of DB2 that automates data warehouse processing. You can use the Data Warehouse Center to define the processes that move and transform data for the warehouse and then to schedule, maintain, and monitor these processes. It can support sources of various types of relational databases, ODBC sources, replication tables, OLE-DB sources, application sources like WebSphere MQ queues, SAP R/3 and Web site information.

7.2.2 What are the tasks required to create a warehouse?

Creating a DB2 data warehouse using the Data Warehouse Center involves the following tasks:

- Identifying the source data (or operational data) and defining it for use as warehouse sources.
- Creating a database to use as the warehouse and defining warehouse targets.
- Defining a subject area for groups of processes that you will define in your warehouse.

- Specifying how to move and transform the source data into its format for the warehouse database by defining steps in the processes.
- Testing the steps that you define and scheduling them to run automatically.
- If you have DB2 Warehouse Manager, you can create an information catalog of the data in the warehouse. An information catalog is a database that contains business metadata. Business metadata helps users identify and locate data and information available to them in the organization. Data Warehouse Metadata can be published to the information catalog. The information catalog can be searched to determine what data is available in the warehouse.
- Defining a star schema model for the data in the warehouse. A star schema is a specialized design that consists of multiple dimension tables, which describe aspects of a business, and one fact table, which contains the facts or measurements about the business.

7.2.3 What is the architecture of the Data Warehouse Center?

Figure 7-2 illustrates the functionality provided by the Data Warehouse Center.

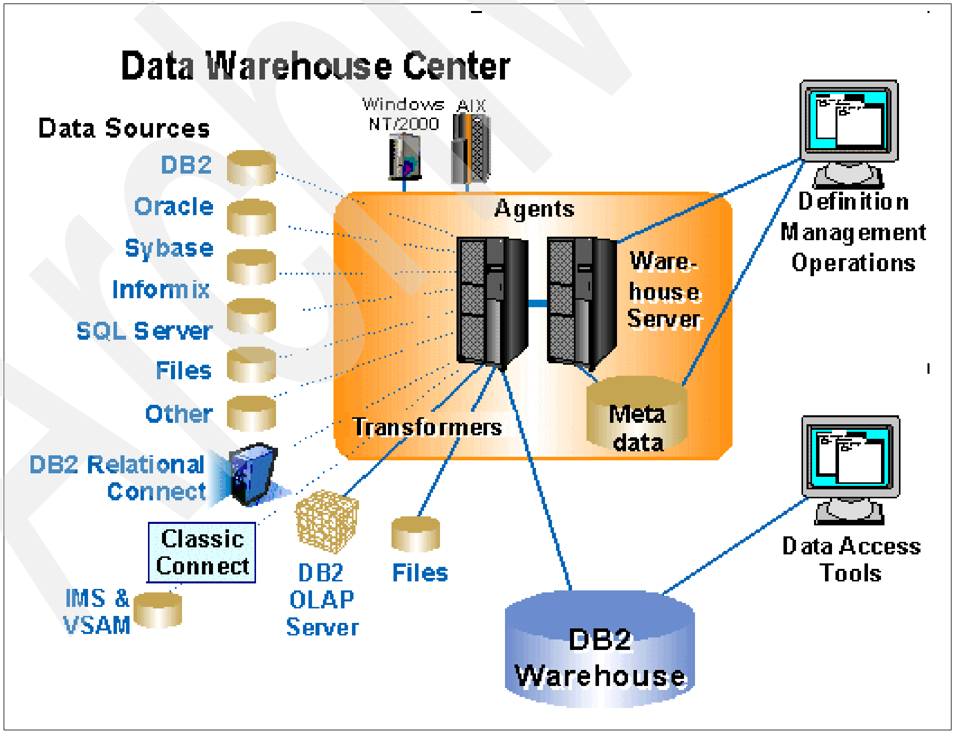


Figure 7-2 Elements of the Data Warehouse Center

The key components are:

Warehouse sources

Warehouse sources identify the tables and files that will provide data to your warehouse. The Data Warehouse Center uses the specifications in the warehouse sources to access the data. The sources can be almost any relational or non-relational source (table, view, file, or predefined nickname).

Warehouse targets

Warehouse targets are database tables or files that contain data that has been transformed. Like a warehouse source, users can use warehouse targets to provide data to other warehouse targets. A central warehouse can provide data to departmental servers, or a main fact table in the warehouse can provide data to summary tables.

Warehouse agents and agent sites

Warehouse agents manage the flow of data between the data sources and the target warehouses. Warehouse agents are available on the AIX, Linux, iSeries, z/OS, Windows NT, Windows 2000, and Windows XP operating systems, and for the Solaris Operating Environment. The agents use Open Database Connectivity (ODBC) drivers or DB2 CLI to communicate with different databases.

Warehouse server (currently Windows NT/2000 and AIX)

The warehouse server interacts with Data Warehouse Center and warehouse agents. It manages the metadata definitions that describe the warehouse processes, automates the workflow for warehouse processes, monitors warehouse process activity, and gathers statistics about the completion of warehouse processes.

As part of this management of the warehouse environment a warehouse control database is defined. This is a DB2 UDB database that hosts the metadata used to define and execute the warehouse processes. It must be stored in a DB2 UDB database on a UNIX or Intel server. It most often resides on the same platform as the warehouse server to provide local access performance. although it can be configured remotely.

Processes and steps

A process contains a series of steps that perform a transformation and movement of data for a specific warehouse use. In general, a process moves source data into the warehouse. Then, the data is aggregated and summarized for warehouse use. A process can produce a single flat table or a set of summary tables. A process might also perform some specific type of data transformation.

A step is the definition of a single operation within the warehouse. By using SQL statements or calling programs, steps define how you move data and transform data. When you run a step, a transfer of data between the warehouse source and the warehouse target, or any transformation of that data, can take place.

These processes and steps are defined within the GUI of the Data Warehouse Center

7.2.4 What can you do next?

The functionality to build and administer warehouses using DB2 Data Warehousing is provided in two ways:

Included with DB2

When you install DB2 UDB on Windows NT or 2000 platforms it provides you with the:

- Data Warehouse Center
- Warehouse server (Windows NT/2000 only)
- Default Windows NT/2000 warehouse agent, which can only be installed on the warehouse server machine

> **Tip:** As part of our installation process in chapter 2 we chose not to install the Data Warehousing components. To add this functionality:
>
> - Run the setup wizard again
> - Choose to modify the install
> - Select to install all the Business Intelligence features

If you require further information then we recommend consulting the following books:

- *DB2 Data Warehouse Center Administration Guide*, SC27-1123
- *DB2 Warehouse Management: High Availability and Problem Determination Guide*, SG24-6828

Purchasing the Warehouse Manager product

The functionality contained within the Data Warehouse Center can be extended by purchasing DB2 Warehouse Manager. This extension to DB2 adds additional capability, including:

- Support for remote native platform execution. Agents that perform the data movement can be installed on any supported platform, you are not restricted to the server where you installed DB2.

- Data Warehouse Center extensions that include extended transformer functions for more sophisticated data transformations.
- Additional support for WebSphere MQ and ODBC targets.
- The Information Catalog that contains business metadata that helps users identify and locate data and information available to them in the organization. When you publish metadata to the information catalog, you provide end users and system analysts with a fast path to seeing what is in the data warehouse.

7.3 DB2 and XML

This section introduces how XML data can be stored in DB2 and also how XML data can be accessed from DB2. We discuss:

- What is XML?
- DB2 XML Extender
- Using SQL/XML to access data
- Using the stored XML
- Further information

7.3.1 What is XML?

XML stands for eXtensible Markup Language. XML is a meta-markup language and is used for creating your own markup languages. Using XML, you can define the tags for your markup language. XML tags are used to describe the contents of the document. This means that any type of data can be defined easily using XML.

XML is universal not only by its range of applications but also by its ease of use: Its text-based nature makes it easy to create tools. It is also an open, license-free, cross-platform standard, which means anyone can create, develop, and use tools for XML. XML is a structured data format, which allows it to store complex data, whether it is originally textual, binary, or object-oriented.

7.3.2 DB2 XML Extender

DB2 XML Extender provides the ability to store and access XML documents, and also to compose XML documents from the existing relational data or to decompose XML documents into relational data. It does so by providing a set of new data types, functions, and stored procedures.

With DB2 XML Extender, you can use SQL as the main access method to your XML documents, which makes your task much easier. It also facilitates your

Administration tasks by providing you with a GUI-based wizard, a set of administrative stored procedures and an Administration command.

Even though XML solves many of the problems by providing a standard format for data interchange, there are other problems, such as storing the XML documents in a centralized repository, as well as the ability to quickly search for information or to trigger automatic data interchange when a particular action occurs. These kinds of issues can be addressed only by a database management system.

Storing the XML document

With DB2 XML Extender you can either store the entire XML document in DB2 as a user-defined data type or map the XML content into columns in DB2 tables. DB2 XML Extender can also store the XML document as an external file in the file system with a pointer to that file stored in the DB2 database so that you can keep working with your legacy flat files but benefit from the advantages offered by DB2 at the same time. DB2 XML Extender provides powerful capabilities to search the XML elements and attributes.

DB2 XML Extender provides two options you can use for integrating XML documents into DB2:

- **XML Columns**: This method allows you to store XML documents in DB2. The documents are inserted into columns which are enabled for XML, and can be updated, retrieved, and searched. Element and attribute data can be mapped to DB2 tables called side tables, which can then be indexed for fast searches.

  With this method once the document is stored, you have a whole range of possibilities for working with the XML document:

  – You can extract and search XML elements and attributes in the XML document directly or for improved performances you can store in the side tables the elements and attributes you are likely to access more often.

  – You have the option to update the content of an XML element or the value of an XML attribute.

  – The XML document can also by retrieved and used by your XML application.

- **XML Collections**: This method allows you to map XML document structures to DB2 tables, so that you can compose XML documents from existing DB2 data, or decompose XML documents into DB2 data.

  With the XML Collections method, you can do the following:

  – You can populate the columns of one or more DB2 tables starting from an XML document. The tags in the XML document and the XML document itself is not stored in DB2; only the values of the elements and attributes you are interested in are stored in DB2. The mapping between elements and attributes in the XML document and the DB2 tables is given by the DAD file.

- You can create an XML document using the data in columns of one or more DB2 tables. DB2 XML Extender composes an XML document using the data stored in columns of DB2 tables. Here again, the mapping between data in the DB2 tables and the composed XML document is done using the DAD file.
- You can work with the data shredded (or decomposed) from the XML document as you would work with any relational data in DB2.

Document validation and relational mapping

The XML extender uses the following two methods to validate the XML document and to map the XML document to relational data, respectively:

- **Document Type Definition (DTD) repository**: The Document Type Definition (DTD) files are used to validate the XML documents you are storing within DB2. The DB2 XML Extender provides an XML DTD repository to store these DTDs. When a database is enabled for XML, a DTD reference table called DTD_REF is created. Each row in this table contains a DTD with additional metadata information about it. You can insert your own DTDs into this table.
- **Document Access Definition (DAD)**: DB2 XML Extender also provides you with a mapping scheme called Document Access Definition (DAD), which is a file used to map the XML document to relational data. The DAD is an XML formatted document which allows you to associate XML document structure to a DB2 database when using either XML Columns or XML Collections. The structure of DAD files is different when using an XML Column or an XML Collection.

7.3.3 Using SQL/XML to access data

DB2 UDB V8 has introduced new built-in functions to allow for relational data to be queried and published as to XML. These functions take a typical DB2 select statement as an input and output a DB2 character large object (CLOB) containing a tagged XML document.

Functions to return SQL data as XML

The three functions are:

- **XMLELEMENT**: This will construct an XML element from the arguments provided. This function takes an element name, an optional collection of attributes, and zero or more arguments that will make up the element’s content. The output result data type is XML.

- **XMLAGG**: XMLAGG is an aggregate function that is used to produce a concatenation of the XML elements from a collection of individual elements.
- **XMLATTRIBUTES**: XMLATTRIBUTES is used to construct an attribute for an element. This function, if specified, must appear as the second argument of an XMLELEMENT call. These attributes are constructed from an expression and an optional alias

These three functions require to be cast by the function

- **XML2CLOB**: This returns the argument of the function as a CLOB data type. The argument must be an expression of data type XML.

Example of using the SQL/XML functions

From our sample applications, we produce a country element (for countries or regions with a code of 24 or 25) with a list of customers, sorted by their name. The syntax and formatted output are in Example 7-1.

Example 7-1   Using SQL/XML to return XML data from relational tables

```
SELECT XML2CLOB(XMLELEMENT(NAME "Country",
XMLATTRIBUTES(c.c_nationkey AS "name"),
XMLAGG(XMLELEMENT(NAME "NAME",c.c_name )
ORDER BY c.c_name
)
))AS "country_list"
FROM db2inst1.customer c
WHERE c.c_nationkey IN (23,24)
GROUP BY c_nationkey;

<Country name="23">
<NAME>Customer#000000011</NAME>
...
<NAME>Customer#000000748</NAME>
</Country>
<Country name="24">
<NAME>Customer#000000117</NAME>
...
<NAME>Customer#000000749</NAME>
</Country>
```

> **Note:** There is another function called REC2XML that returns a string formatted with XML tags and contains row and column information from a table.

7.3.4 Using the stored XML

Once you have the XML stored and managed within DB2, you can then use this XML data to provide content to other sources. In this section we introduce two examples of ways that DB2 and XML can provide functionality with the traditional database. The XML Extender can also be used to access data held in WebSphere MQ and this will be discussed in 7.4, "Accessing WebSphere MQ" on page 324.

Transforming XML content to HTML

The configuring of the XML Extender makes various functions and procedures available for use. Two of these SQL functions, `XSLTransformToClob` and `XSLTransformToFile` allow you to transform XML to HTML.

These functions use Extensible Stylesheet Language Transformation (XSLT) stylesheets to provide the formatting for the translated HTML document. XSLT stylesheets consists of a series of markups that can be used to apply formatting rules to each of the elements inside an XML document. XSL works by applying various style rules to the contents of an XML document based on the elements that it encounters. By design, XSLT stylesheets are regular XML documents.

XSLT is a critical bridge between XML processing and more familiar languages such as HTML. XSLT allows you to transform an XML structure into other data types by removing or replacing the XML tags. It also allows you to change the order of the information, extract some special information or sort the information.

Web services

Web services are XML-based application functions that can be invoked over the Internet. You can define a basic Web service by using standard SQL statements and DB2 XML Extender stored procedures. For Web Services that involve advanced transformations between XML and relational data, you can use the DB2 XML Extender.

The Web services object runtime framework (WORF) provides an environment to easily create simple XML based Web services that access DB2 data and stored procedures. WORF uses Apache Simple Object Access Protocol (SOAP) 2.2 or later and the Document Access Definition Extension (DADX). A DADX document specifies a Web Service using a set of operations that are defined by SQL statements or XML Extender Document Access Definition (DAD) documents. Web services, or functions invoked over the Internet, specified in a DADX file are called DADX or DB2 Web services.

Figure 7-3 illustrates the WORF architecture

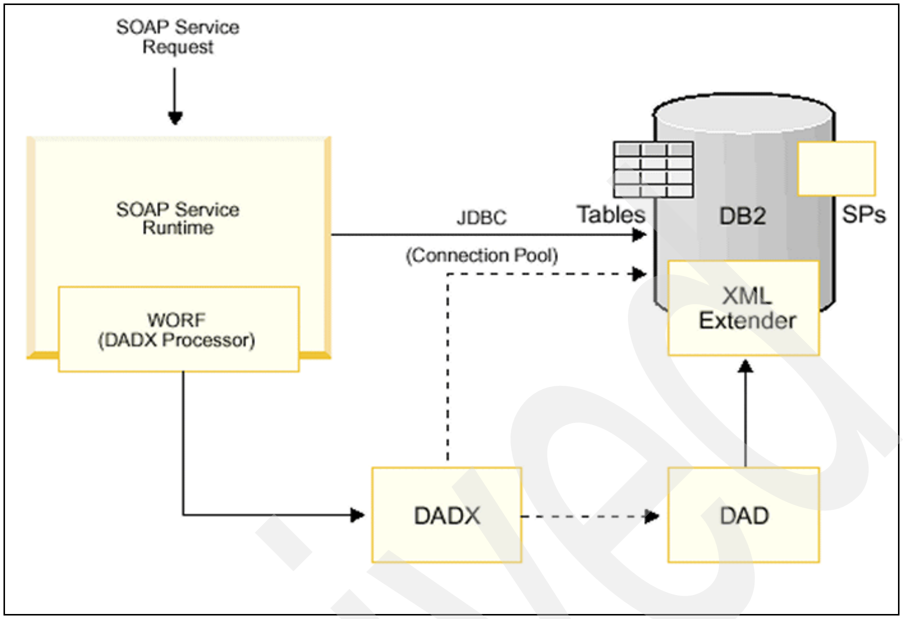


Figure 7-3 WORF architecture

Version 8 of DB2 has extended the Web services capability of DB2 by adding the functionality to invoke other Web services as well as providing Web services to consumers. This allows you to extend the SQL language with Web services operations, enabling you to combine set-oriented data access with the simplicity and power of Web services. Web services may be used to transform existing data or to retrieve real-time data.

7.3.5 Further information

When you install DB2 Version 8 the SQL/XML functions described in Section 7.3.3" on page 320 are part of the default install. The XML Extender is also installed by default however you then have to enable a database to be able to use the XML data types, stored procedures and functions. The command for this is:

```
dxxadm enable_db rborders
```

After this step you then decide which columns you want to enable to be either used for XML Collections or XML Columns. An excellent tutorial on using the XML Extender is provided in Chapter 1 of the *XML Extender Administration and Programming Guide*.

To create Web services the WORF framework is included as part of DB2 Version 8. IBM WebSphere studio further provides a set of tools that make it easier to build DADX Web Services. For more information on WebSphere studio, check DB2 Web services Web site:

`http://www-3.ibm.com/software/solutions/webservices/leader/index.jsp`

For more information, tutorials and examples, we recommend using the following information as a good starting point.

IBM Redbooks:

- *Integrating XML with DB2 XML Extender and DB2 Text Extender*, SG24-6130

Web sites:

- Meet the Experts: Susan Malaika on XML Capabilities in DB2

  `http://www7b.boulder.ibm.com/dmdd/library/techarticle/0212malaika/0212malaika.html`

- Efficient ways to publish DB2 data as XML, Using the new SQL/XML features in DB2 V8

  `https://www6.software.ibm.com/reg/devworks/dw-db2sqlx-i?S_TACT=102B7W84&S_CMP=DB2DD`

- The DB2 Web Services home page can be found at:

  `http://www7b.software.ibm.com/dmdd/zones/webservices/worf/`

DB2 manuals:

- *SQL Reference Volume 1*, SC09-4844
- *XML Extender Administration and Programming*, SC27-1234
- *Web Services Object Runtime Framework: Implementing DB2 Web Services*, at: `http://www7b.boulder.ibm.com/dmdd/zones/webservices/worf/DXXSERVL.PDF`

7.4 Accessing WebSphere MQ

This section briefly discusses how DB2 can be used to access and publish data in the MQ product family. WebSphere MQ is a separate product from DB2, so this section may be of particular interest to those who currently use WebSphere MQ, are planning to use WebSphere MQ or are considering downloading the trial code.

You can download the trial code from:

`http://www7b.boulder.ibm.com/wsdd/downloads/evaluation.html`

This section covers:

- What is WebSphere MQ?
- How does DB2 interact with MQ?
- Further information

7.4.1 What is WebSphere MQ?

The WebSphere MQ range of products provides application programming services that enable application programs to communicate with each other using messages and queues. It provides assured, once-only delivery of messages. If the receiver, or the communication channel to it, is temporarily unavailable, the message can be forwarded at a later time. MQSeries also provides mechanisms for providing acknowledgements of messages received.

Product Family

The WebSphere MQ family of products incorporates:

- WebSphere MQ and the core of the WebSphere MQ family, integrates more than 35 platforms, provides the base messaging functions for servers and clients, and assures once only message delivery. It can be used alone or with other members of the family.
- WebSphere MQ Everyplace™ enables access to enterprise data for mobile workers and remote devices with assured message delivery and dependable security anytime, anyplace, anywhere.
- WebSphere Adapters provide solutions for connecting legacy systems and packaged applications to WebSphere MQ.
- WebSphere MQ Integrator Broker, a powerful information broker, includes a one-to-many connectivity model plus transformation, intelligent routing and information flow modelling across multiple, disparate business systems.
- WebSphere MQ Event Broker enables you to integrate people, applications and telemetry devices in real-time based on an event model.
- MQSeries Workflow, a business process management system, enables the definition, execution and swift change of complete business processes that span systems, applications and people.

Messaging models

MQ supports three messaging models:

- Datagrams: Messages are sent to a single destination with no reply expected.
- Publish/subscribe: One or more publishers send a message to a publication service which distributes the message to interested subscribers.
- Request/reply: Messages are sent to a single destination and the sender expects to receive a response

7.4.2 How does DB2 interact with MQ?

DB2 offers the flexibility to interact with WebSphere MQ queues in two different ways:

- Using built-in DB2 functions
- Using the DB2 XML Extender

WebSphere MQ interaction using built-in DB2 functions

DB2 provides built in functions that allow you to read messages from MQ queues as well as place messages onto these queues. These functions are easily recognizable as they are prefixed by “MQ”.

- MQREAD, MQREADCLOB, MQRECEIVE, MQRECEIVECLOB

  These functions allow DB2 to read messages from MQ queues and store the intact contents of the message using traditional SQL INSERT statements.

- MQPUBLISH and MQSEND

  These functions allow you to place SQL data onto MQ queues. When combined with the capability of SQL/XML this can allow you to publish SQL data as XML onto a WebSphere MQ queue.

WebSphere MQ interaction using DB2 XML Extender

As we discussed in Section 7.3.2 the DB2 XML Extender provides UDFs and stored procedures for storing and accessing XML data.

- **UDFs**: These functions allow the use of SQL statements to publish, read, or receive or send MQ messages as XML:
  - The publish function, MQPublishXML, publishes XMLVARCHAR and XMLCLOB data to MQ
  - The read functions, MQREADXML, MQReadAllXML, MQReadXMLCLOB, MQReadAllXMLCLOB return XMLVARCHAR or XMLCLOB data from MQ either as a scalar value or a table.
  - The receive functions, MQReceiveXML, MQReceiveAllXML, MQRcvAllXMLCLOB, MQReceiveXMLCLOB remove messages from an MQ queue either as XMLVARCHARs or XMLCLOBs.
  - The send functions MQSENDXML, MQSENDXMLFILE, MQSendXMLFILECLOB send XMLVARCAHARs or XMLCLOBs to an MQ queue.
- **Stored procedures**: Using the MQ XML stored procedures, you can retrieve an XML document from a message queue, decompose it into untagged data, and store the data in DB2 tables. You can also compose an XML document from DB2 data and send the document to an MQSeries message queue.

- The composition stored procedures, dxxmqGen(), dxxmqGenCLOB(), dxxmqRetrieve(), and dxxmqRetrieveCLOB() generate XML documents using data in existing database tables, and to send the generated XML documents to a message queue.
- The decomposition stored procedures, dxxmqInsert(), dxxmqInsertAll(), dxxInsertCLOB(), dxxmqShred(), dxxmqShredCLOB, and dxxmqShredAll() are used to break down or shred incoming XML documents from a message queue, and to store the data in new or existing database tables.

What do you need to have installed?

To take advantage of the MQ interaction detailed above you will need the following configuration on your system:

- DB2 Version 8.
- MQSeries Version 5.2 with a default queue manager.
- To make use of the publish/subscribe capabilities of the DB2 MQ functions, you must also install either MQSeries Integrator or the MQSeries Publish/Subscribe Functions.
- Install the MQ Application Messaging Interface (AMI). The DB2 MQFunctions require the installation of this interface.
- Configure and enable a database for the DB2 MQ functions. The **`enable_MQFunctions`** utility is a flexible command that first checks that the proper MQ environment has lbeen set up and then installs and creates a default configuration for the DB2 MQfunctions, enables the specified database with these functions, and lconfirms that the configuration works.
- Ensure that **`dxxadm enable_db`** has been run on the same database so the XML Extender functions can be used also.

Figure 7-4 illustrates this simple DB2/MQ configuration.

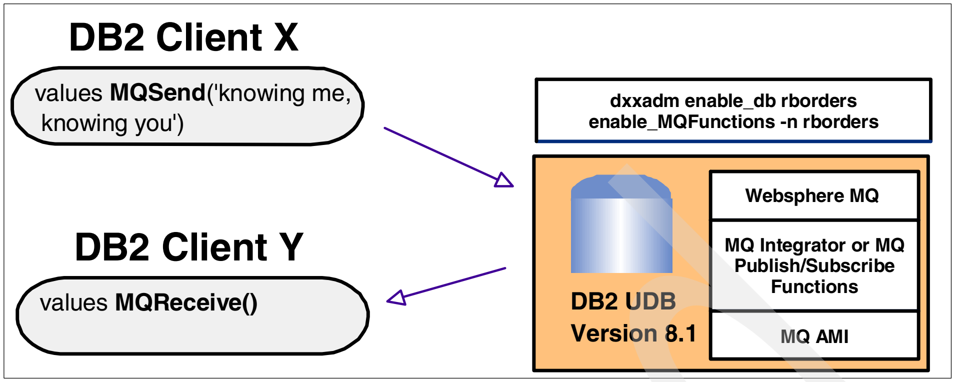

DB2 Client X
values MQSend('knowing me, knowing you')
DB2 Client Y
values MQReceive()
dxxadm enable_db rborders enable_MQFunctions -n rborders
DB2 UDB Version 8.1
Websphere MQ
MQ Integrator or MQ Publish/Subscribe Functions
MQ AMI

Figure 7-4 DB2 interacting with WebSphere MQ

7.4.3 Further information

For more information and examples, we recommend the following information.

IBM Redbook:

- *Getting Started on Integrating Your Information*, SG24-6892-00

Web sites:

- The WebSphere MQ home page can be found at:

  `http://www-3.ibm.com/software/ts/mqseries/`

- DB2 MQ XML Functions: Using MQSeries and XML Extender from DB2 Applications:

  `http://www7b.software.ibm.com/dmdd/library/techarticle/wolfson/0201wolfson.html`

- Using MQSeries from DB2 Applications:

  `http://www7b.boulder.ibm.com/dmdd/library/techarticle/wolfson/0108wolfson.html`

DB2 manuals:

- *SQL Reference Volume 1*, SC09-4844
- *XML Extender Administration and Programming*, SC27-1234

8

High availability, scalability, and parallelism

This chapter discusses three concepts; high availability, scalability, and parallelism. The concept of high availability (HA) translates to having a system available to users on a constant basis with only very brief, planned outages. Unfortunately, unplanned outages seem to find their way into the scheme of things, causing mayhem for those without a tested contingency plan to act upon. We discuss ways to protect your database and log files using DB2 and IBM products. We then discuss various third-party failover products to allow for automatic recovery of a database or system failure. In addition, we describe some high availability maintenance features in DB2 to reduce the need for database or instance downtime.

In the section about scalability, we introduce DB2's partitioning feature to show how a database can run seamlessly across multiple machines while benefitting from each machine's hardware resources.

In the section about parallelism, we show how DB2's partitioning feature can take a database task, such as a query, and break it up into smaller pieces to enable workload sharing across all partitions.

8.1 High availability

In this section we describe the high availability (HA) features of DB2 Universal Database V8.1 for Linux and Windows operating systems. We begin by describing data redundancy strategies available in DB2. Following Data redundancy strategies, we introduce third-party failover products that can be implemented in a DB2 failover plan to essentially eliminate downtime. The last topic for high availability shows how you can maintain your database product through the new DB2 maintenance features to enable 99.999% uptime.

8.1.1 Data redundancy strategies

Many data redundancy strategies are available to suit your database needs. In this section we discuss some of the more common strategies that can be implemented with DB2 V8.1. In the chart shown in Figure 8-1, we describe the *Capabilities* of DB2 on the Y-axis and the *Cost* on the X-axis. Lower cost and lower capability strategies are found closer to the lower left corner of the chart while higher cost and higher capability strategies are found closer to upper right corner of the chart.

Low-cost data redundancy scenarios are available and should be used when relative time to recover and data redundancy is not a high issue. As recovery time and data becomes more important, higher-end strategies should be considered.

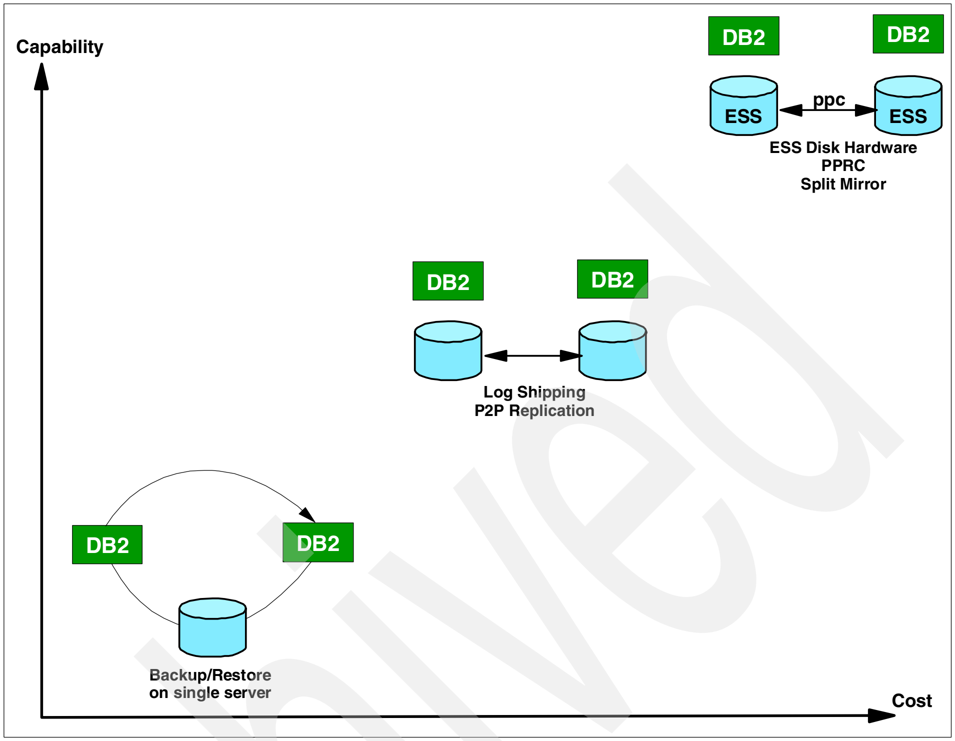


Figure 8-1 High availability solutions

Single server

The single server data redundancy strategy is a low-cost solution for systems that do not need instant recovery. The owner should also be able to accept total data loss if the disks or machine is destroyed. Small businesses find this to be an affordable way to preserve the data in the database by backing up the database and log files to different disks on the same machine or to tapes. If the disk holding the database fails or the database becomes corrupt, the database can be restored from the backup disk or tape and the log files rolled forward.

Since the database needs to be restored, time needs to be factored in depending on the processing and memory hardware available, the size of the database to be restored, and the size and number of log files to be rolled forward.

Log Shipping

Unlike the single server strategy above, log shipping and the rest of the strategies described below implement a two server plan, where the backup server can take over the primary servers workload. The technique for log shipping involves a primary and backup server where the database on the primary server is backed up and restored to the backup server.

Once the database is available on the backup server, the database logs can be continually copied (shipped) over to the backup server either via a user exit program, a script, or by setting the database configuration parameter *logmirrorpath*. The **db2inidb** command may then be used to set the database to a perpetual roll forward pending state whereby each database log passed from the primary server is immediately applied to the database on the backup server. More about the **db2inidb** command is discussed in the section, Initializing the database.

Replication

Another data redundancy strategy may be to use replication through DB2 via a primary and backup server, or two primary servers using peer to peer configuration. In the first scenario, the DB2 Replication Center tool can be used to set-up and administer the replication environment thereby allowing database changes to be replicated to the backup server database. If the primary server goes down, the backup server can be initialized to take over the primary server's workload. Figure 8-2 is an example of a primary/backup scenario.

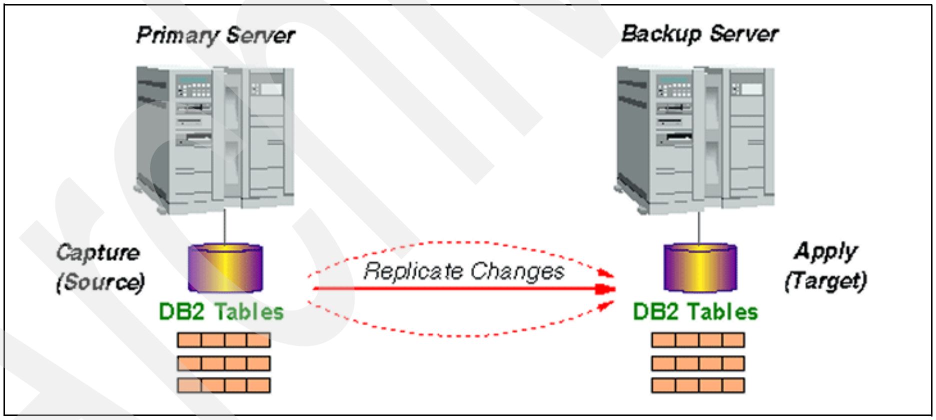


Figure 8-2 Primary/backup replication scenario

In the second scenario, two primary servers acting as peers may be implemented, each serving as a backup for the other when a database failure occurs. The backup server then takes on the failed server's workload in addition to its primary workload. In order to have the backup server automatically take over, it is necessary to have a failover plan. Otherwise, it will be necessary to manually switch over to the backup server. We will discuss failover products available for the Linux and Windows platform in Failover products. Figure 8-3 is an example of a peer-to-peer scenario:

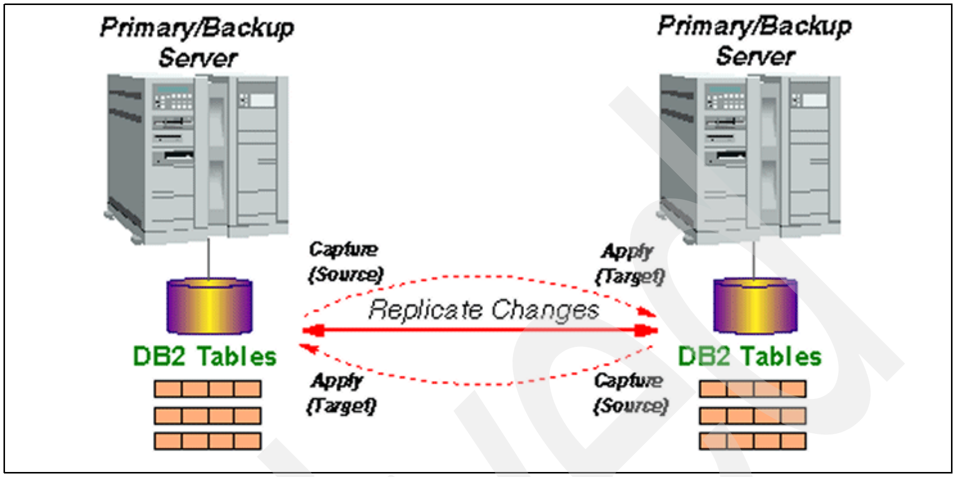


Figure 8-3 Peer-to-peer high availability replication scenario

Peer-to-peer replication is bidirectional. It differs from the primary/backup scenario in the following ways:

- There is no MASTER server.
- Capture and Apply run on both peer servers.
- Administration cannot be done totally through the Replication Center.

The advantages of replication over a hardware solution such as log shipping are:

- Log shipping may not be available on the desired platform.
- Replication may be relatively inexpensive compared to the other solutions, both in the initial cost and ongoing administration.
- Replication can operate across distances that are not easily supported, if supported at all, by log shipping.
- Applications can run on any peer system, so query workload can be balanced across the peer configuration. Note that this is not workload balancing for change processing. Each peer server must be able to handle local changes as well as the replicated changes from all other peers.

The disadvantages of replication are:

- Replication is asynchronous. There is always some delay in copying changes from one system to another. At any given point in time, both peer systems may not exactly match. If a failure occurs on one system and application activity is transferred to the second system, there may be changes that have not yet been replicated to the second system. The replication delay is called latency.

- Subscription sets in each direction are processed independently, so Apply does not have information for conflict detection.
- Most of the peer to peer setup is done through the Replication Center, but there are some updates to control tables that must be done manually.

ESS disk hardware and online split mirror

In this data redundancy strategy, we combine the high level of data redundancy and performance of a RAID system, high bandwidth capabilities, and an online split mirror to create a copy of the database to the backup server.

ESS disk hardware

The Enterprise Storage Server™ (ESS) solution provides a high bandwidth, storage, and data redundancy solution to allow large transactional databases to ship large quantities of database logs to the standby server. The ESS lets you simplify and centralize the management of your storage resources across your data processing enterprise, while providing the highest levels of data redundancy and performance.

ESS is a redundant disk array solution that is packaged with a pair of eight disks, for a total of 16 disks. The package is called an *eight pack* while eight disks is referred to as a *rank*. Many eight packs may be used within a this strategy. Six disks for each rank are striped while the remaining two are reserved, one for parity and one backup. In the event that a disk in one of the ranks fails, a parity disk will reconstitute the data lost from the failed disk and move the data to the backup disk. Figure 8-4 shows an ESS disk array configuration.

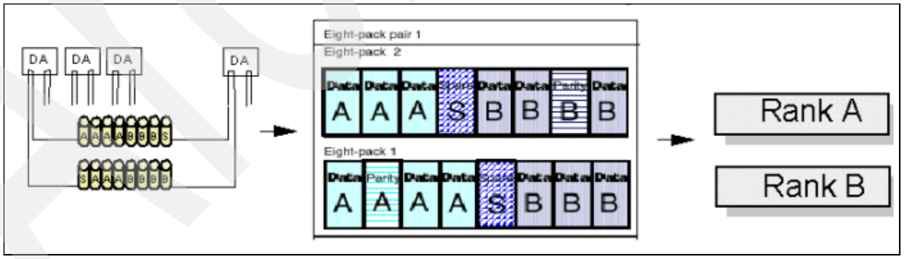


Figure 8-4 ESS disk array configuration

DB2 also supports the following storage server vendors:

- Sun Cluster 2.2
- Veritas Cluster Server
- HACMP

- SteelEye

Since large quantities of data are moved from the primary server to the ESS storage device, it is necessary to have the primary and backup servers in relative proximity to each other for optimal performance. As discussed in Online split mirror, An online split mirror may be used to mirror the database to the backup server.

Online split mirror

Introduced in DB2 UDB v7.2, split mirror images provide high availability support by combining features built into DB2 UDB with advances in high end storage subsystems. The concept of mirrored images is nothing new. We have been using RAID technology for over a decade to protect data in the event of disk failures by mirror disks with RAID controllers. Today, with advances in disk system technology we are able to break mirrored disks and move images stored on these disks to other system. DB2 UDB supports this technique while keeping the database system online.

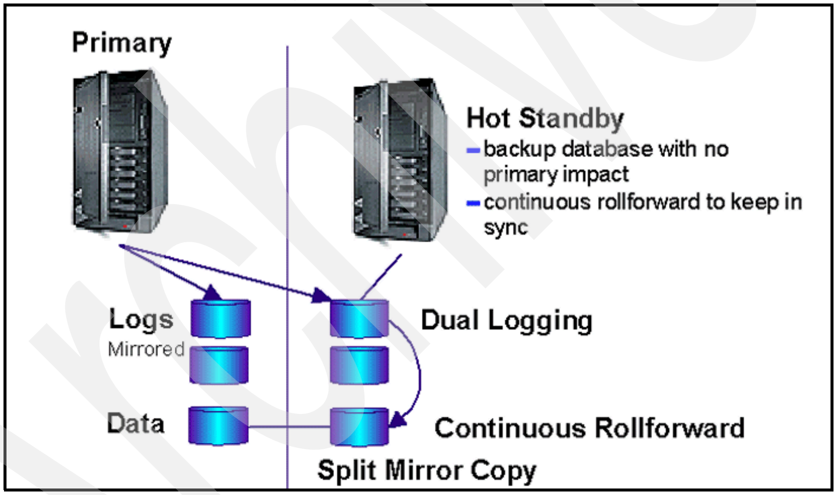


Figure 8-5 Online split-mirror image overview

In Figure 8-5, dual logging can be accomplished by updating the database configuration parameter *logmirrorpath* to the database log directory on the backup server. Note, it is necessary to mount the backup server's log directory to the primary server so that it can be seen by DB2.

There are several methods for splitting mirrored images of databases and they all provide for high availability. First, you can remove the overhead associated with a database backup by splitting the mirror image and backing up the database on a separate system. Once the backup is complete, you simply re-mirror the disk.

Next, you can quickly create a idle standby server by splitting the mirror image and moving the image to the standby server. Finally, you can efficiently create a near real time copy of a database on another system for reporting or OLAP.

Suspending database I/O

DB2 UDB provides support for online split mirror images by temporarily suspending database I/O while the disk subsystem splits mirrored disks. After the mirrored disks images are split, DB2 UDB resumes I/O activity on the primary image of the database. Once the secondary image has been moved to another system running DB2 UDB the secondary database image is initialized. The secondary database image can be initialized in one of three ways, standby, backup, or mirror.

In order to prepare the database for breaking of the mirrored disks the database must suspend I/O to the disk subsystem. The database will remain online and continue processing transactions (that is, select, insert, update, delete) during the suspended I/O state, however no writes to the database data (tablespace containers) or transaction log files will be performed. Database I/O is suspended by first connecting to the database and invoking the DB2 set write suspend command as follows:

```
db2 set write suspend for database
```

Resuming database I/O

After the mirror image has been broken by the disk subsystem, database I/O can be resumed. Database I/O is resumed by first connecting to the database and invoking the DB2 set write resume command as follows:

```
db2 set write resume for database
```

It is important to note that the transaction database will remain online and continue processing transactions (select,insert,update,delete) during the suspended I/O state, however no writes to the database data (tablespace containers) or transaction log files will be performed. This leaves the database in a very delicate state that could result in the complete loss of transaction if the database were to crash during the suspended I/O state.

In the event of a database crash while the database is in the suspended I/O state, the database will need to be restarted with the resume write state so that crash recovery can be initiated and completed. This can be accomplished by invoking the DB2 restart database command with the write resume option as follows:

```
db2 restart database <database_alias>write resume
```

Initializing the database

Once the secondary database image has been moved to another system running DB2 UDB, the image must be initialized in order to reconcile discrepancies in the database transaction log files with data in the database files. This is done using the DB2 Productivity Tool Initialize Database (db2inidb.exe).

The syntax is as follows:

```
db2inidb database_alias as <SNAPSHOT |STANDBY |MIRROR >
[RELOCATE USING config_file ]
```

- **Snapshot database image**: The SNAPSHOT option allows the secondary database image to be initialized as a duplicate copy of the original database. The database will be initialized with a new set of transaction log file independent of the primary database image. Use this option when you want to create a secondary copy of the database to off load reporting or OLAP queries.
- **Standby database image**: The STANDBY option allows the secondary database image to be initialized as a standby copy of the original database. The database will be initialized in roll forward pending state allowing additional database transaction log files from the primary database image to be applied to this image. The database remains in a continuous roll forward pending state until otherwise required at which point the roll forward state is completed and the database is brought online.
- **Mirror database image**: The MIRROR option allows the secondary database image to be initialized as a mirror image of the original database. The database will initialized in a backup pending state so that a backup image can be taken and if required, can be used to restore the primary database image as if the backup image had been taken from the original database.
- **Relocating database image**: The RELOCATE USING option is a new option with DB2 UDB V8.1 and provides the functionality of the Relocate Database (db2relocatedb.exe) Productivity Tool to update internal database structures to the new location of database files. The information is provided using a plain text configuration file. More information about the Relocate Database configuration file can be found in the Online DB2 Command Reference.

8.1.2 Failover products

Clustered computing has been wide used in today's e-business world. One of the objective of clustering architecture is high availability achieving by hardware redundancy and software failover capability. Failover is the automated process of switching databases, metadata, and user connections from one machine to another when there is a hardware failure so that users can still access data.

In Linux, the high availability solutions for Linux clusters are available from several ISVs and span a range of capability and complexity. For Windows environment, Microsoft Cluster Server (MSCS) and Legato Automated Availability Manager facilitates the failover capability. In this section, we discuss third party products for Linux and Windows that allow automatic failover recovery of your database on a clustered environment.

Linux failover products

Linux itself does not have the failover component. However, several failover products are supported on the Linux platform to automatically allow your standby server to take over in the event of database failure. Here are some of the following failover products:

LifeKeeper for Linux (SteelEye, Inc.)

SteelEye's LifeKeeper for Linux is a software application that maintains high availability of clustered Linux systems. LifeKeeper is fault tolerant by empowering other production servers within the cluster to take over failed servers and applications. Cost of ownership is reduced since the need for a hot standby is eliminated. A single application failover, such as DB2 service failure, can be performed without having to perform a full failover on the system thereby reducing the load work for the take over servers. LifeKeeper provides protection for Linux environments to support disaster tolerance, multiple system failures or faster recovery, including:

- **Multi-directional recovery**
  - LifeKeeper for Linux supports a four-node cluster in a multi-directional failover configuration within the context of two nodes connected to the same shared disk.Shared Data Support.
  - LifeKeeper for Linux provides for an N+1 configuration. LifeKeeper supports up to two nodes per shared SCSI bus. This allows shared SCSI disk-based application recovery between two nodes within the cluster that are connected to the same shared disk. In this configuration, one server, in an active-active role, provides backup for failovers from any of the other nodes in the cluster.
- **Cascading failover**
  - LifeKeeper for Linux allows cascading failover for as many as 32 active nodes to ensure continuous client access in the event of system or application failure.

For more information regarding the SteelEye product, visit:

http://www.steeleye.com/products/linux/#1

Legato Automated Availability Manager (Legato Systems)

Legato Automated Availability Manager (LAAM) delivers high availability clustering of DB2 applications. LAAM supports Linux, Solaris, HP-UX, AIX, and Windows NT servers in a heterogeneous clustering solution. LAAM is also uniquely suited to managing application and service-level availability within SAN deployments.

If an application or server fails, then LAAM keeps the application available by restarting it on another server thereby allowing business users to be reconnected and continue to be productive.

Here are some of the highlights for LAAM:

- Non-intrusive
- Ease-of-use
- Heterogeneous UNIX and Windows NT Clusters, sharing configuration and event data
- SAN-ready
- SSI Configuration Database, for practical and easily managed large-scale cluster deployments
- Availability Tracking
- Open, flexible architecture, with APIs for custom deployment
- Value-added modules for enhanced applications sensing
- Supports Shared Disk, NAS, NFS, NT Share, and replicated data
- Integrated with Legato Replication, to automate replication capabilities

For more information regarding the LAAM product, visit:

http://portal2.legato.com/products/manager/

Convolo Cluster Dataguard (Mission Critical Linux, Inc.)

Convolo Cluster Dataguard is another clustering product which provides a failover solution for DB2. Here are some of the highlights for Convolo Cluster Dataguard:

- DataGuard is unique in the industry as the only Linux high availability cluster that preserves locks throughout failover and also maintains full NFS access permissions.
- Supports usage of a wide range of remote power switch types, including both network and serial connected, to ensure that data integrity is not compromised during system failure.

- Maximizes cluster availability with support for the ReiserFS journalled file system.
- Assures transparent operations via network interface failover with optional MAC address failover.
- Runs on a 2.2.19 or 2.4.6 Linux kernel.
- Supports active-active cluster configurations.
- Provides an extensive framework for making off-the-shelf commercial applications highly-available without requiring application modification.
- Does not depend on low-level SCSI reservations to maintain data integrity.
- Includes both a graphical user interface (GUI) and a command line interface (CLI) that facilitate smooth installation and configuration in addition to service and cluster administration.

For more information regarding the Convolo Cluster Dataguard product, visit:

http://www.missioncriticallinux.com/products/convolo-overview.php

Windows failover products

The following products support failover strategies on Windows operating systems:

Microsoft Server Clusters (MSCS)

The Microsoft Server Clusters (MSCS) product is available in Windows Server 2003, Enterprise Edition and Datacenter Edition. MSCS provide high availability and scalability for DB2 databases. If one of the nodes in a cluster becomes unavailable either due to planned downtime for maintenance or unplanned downtime due to failure, another node takes over to provide the service to the end-user. If a failover occurs, users accessing the cluster service are able continue to access the service, and are unaware of the seamless transition of the service from one node to another node in the cluster. Windows Server 2003 supports up to 8-node server clusters in Enterprise Edition and Datacenter Edition.

For more information regarding setting up DB2 on MSCS, refer to:

- *Implementing IBM DB2 Universal Database V8.1 Enterprise Server Edition with Microsoft Cluster Server* at this Web site:

  http://www-3.ibm.com/software/data/pubs/papers/#esemcs
- *DB2 UDB Exploitation of the Windows Environment*, SG24-6893

Legato Automated Availability Manager (Legato Systems)

Legato Automated Availability Manager (LAAM) delivers high availability clustering of DB2 applications. LAAM supports Linux, Solaris, HP-UX, AIX, and Windows NT servers in a heterogeneous clustering solution. LAAM is also uniquely suited to managing application and service-level availability within SAN deployments.

If an application or server fails, then LAAM keeps the application available by restarting it on another server thereby allowing business users to be reconnected and continue to be productive.

Here are some of the highlights for LAAM:

- Non-intrusive
- Ease-of-use
- Heterogeneous UNIX and Windows NT Clusters, sharing configuration and event data
- SAN-ready
- SSI Configuration Database, for practical and easily managed large-scale cluster deployments
- Availability Tracking
- Open, flexible architecture, with APIs for custom deployment
- Value-add modules for enhanced applications sensing
- Supports Shared Disk, NAS, NFS, NT Share, and replicated data
- Integrated with Legato Replication, to automate replication capabilities

For more information regarding the LAAM product, visit:

http://portal2.legato.com/products/manager/

LifeKeeper for Windows (SteelEye, Inc.)

SteelEye's™ LifeKeeper® for Windows 2000 product ensures the continuous availability of applications and data that reside on a Windows 2000 server. The LifeKeeper framework runs on top of the Windows 2000 operating system, allowing LifeKeeper to fail over applications in the event of a failure.

For more information regarding the LifeKeeper for Windows, visit:

http://www.steeleye.com/products/nt/windows_2000.html

8.1.3 HA administration features

In addition to having failover functionality, DB2 Version 8 has improved upon previous versions by allowing utilities to run while databases are online and commands to become effective immediately.

Buffer pool management

Most database professionals will probably tell you that buffer pools are the single most important database resource that can be tuned to improve performance.

Given the importance of memory (specifically, buffer pool tuning) to database performance, it is no wonder that IBM has focused more attention to this area. In previous versions of DB2 UDB, buffer pools could be created, changed, and dropped while the database was online. In fact, the database had to be online; however, these changes did not take effect until the database was recycled, which resulted in downtime.

Online buffer pools

DB2 UDB V8.1 introduces online buffer pool management. This new feature allows you not only to create, change, and drop database buffer pools without having to recycle the database; it allows you to put these changes into effect immediately, providing for high availability during these changes.

New options have been added to SQL statements for managing buffer pools. The default option is now IMMEDIATE, which indicates that changes to buffer pools will take effect without recycling DB2 databases. For backward compatibility, the DEFERRED option allows you to implement changes after the database has been recycled.

For example, let us say you are running on a small 4-way SMP system with 4 GB of memory on Windows 2000 Server. You start out with a single large 1-GB buffer pool, but later decide to create a second buffer pool for a select few tablespaces. Now you can the reduce the size of the first buffer pool and create the second buffer pool on-the-fly, without recycling the database:

```
db2 alter bufferpool mybp1 size 250000 immediate
db2 create bufferpool mybp2 size 250000 immediate
db2 alter tablespace mytbspc2 bufferpool mybp2
```

Tablespace management

Although most database professionals will agree that memory management, specifically buffer pool tuning, is important for performance, most are unlikely to site any specific instances where failure to perform this important task has resulted in system downtime. Tablespace management, on the other hand, is not only vital to database performance, but if neglected could result in system downtime.

In previous versions of DB2 UDB, tablespace management consisted of two types: autopilot and manual. The first type, autopilot or System Managed Storage (SMS), has been part of DB2 on Intel platforms for over a decade. Database Managed Storage, the manual type, was first introduced in DB2 UDB v5, and resulted in faster I/O operation. In this version of DB2 UDB, DMS tablespaces could only be extended by adding additional containers which would immediately invoke tablespace re-balancing. Later, in DB2 UDB v7.2, some relief was provided by allowing DMS tablespace containers to be extended in size.

DB2 UDB V8.1 introduces new management functionality for tablespaces, defined as Database Managed Storage (DMS). Now you can add a DMS tablespace container without having to re-balance the containers, a task that can sometimes overwhelm systems. You can also shrink tablespaces that might have been accidently over-allocated or have had a significant amount of data removed. All of these new options can be performed without impacting performance and without bringing the database or instance offline, providing for high availability.

Configuration parameters

Once common system maintenance task involves making configuration changes to DB2 instances and databases. In the past, very few of these changes could be performed without having to recycle either the database or the instance to get the new configuration parameter value to take effect. DB2 UDB V8.1 provides for high availability by allowing buffer pools to be managed while the database is still online, as well as allowing over 50 configuration parameters at the instance and database levels to be changed without having to bring systems offline.

Online configuration parameters

DB2 UDB V8.1 introduces online configuration parameters that provide for high availability, as changes to these specific configuration parameters take effect immediately without having to bring down DB2 instances or databases. This means that users will not have to be forced off of databases so that database configuration parameters can take effect. It also means that DB2 instances will not have to be stopped and started to have instance (dbm) configuration parameters take effect.

IBM has concentrated on specific instance (dbm) and database configuration parameters in the area of memory utilization, such as catalog and package cache, statement, sort, and utility heaps. In the event that you prefer the old school, IBM provides for backward compatibility, thus allowing you to defer these configuration changes until the next instance or database recycle. This is a key requirement for performance tuning during a series of benchmark runs:

```
db2 get dbm cfg show detail
db2 get db cfg for <database alias> show detail
```

Automatic configuration parameters

DB2 UDB V8.1 also introduces the beginning of self tuning configuration parameters. Only a few key parameters can be set to AUTOMATIC. DB2 UDB V8.1 currently supports automatic settings for instance_memory, database_memory and maxappls.

You can determine the current value of configuration parameters that support the automatic setting by using the SHOW DETAIL option on the DB2 following DB2 commands:

```
db2 get dbm cfg show detail
db2 get db cfg for <database alias> show detail
```

Reorganizing data

After many changes to table data, logically sequential data may be on non-sequential physical data pages so that the database manager must perform additional read operations to access data. Additional read operations are also required if a significant number of rows have been deleted.

As tables are updated with deletes and inserts, index performance can also degrade in the following ways:

- Fragmentation of leaf pages may occur. When leaf pages are fragmented, I/O costs increase because more leaf pages must be read to fetch table pages.
- The physical index page order no longer matches the sequence of keys on those pages, which is referred to as a badly clustered index. When leaf pages are badly clustered, sequential prefetching is inefficient and results in more I/O waits.
- The index develops more than its maximally efficient number of levels. In this case, the index should be reorganized.

For both tables and indexes that have become fragmented, running the REORG utility can improve performance. Prior to DB2 UDB V8, the REORG utility execution resulted in the table and indexes being unavailable for user and application access. We discuss below how to run the REORG utility online for both tables and indexes that permit access while executing, providing you with higher availability.

For detailed information on the REORG utility see the *DB2 UDB V8 Command Reference*.

In-place index reorganization

With DB2 Version 8, you can read and update a table and its existing indexes during an index reorganization using the new REORG INDEXES command. During in-place index reorganization, the entire index object (that is, all indexes on the table) is rebuilt. A shadow copy of the index object is made, leaving the original indexes and the table available for read and write access. Any concurrent transactions that update the table are logged. Once the logged table changes have been forward-fitted and the new index (the shadow copy) is ready, the new index is made available. While the new index is being made available all access to the table is prohibited.

The default behavior of the REORG INDEXES command is ALLOW NO ACCESS, which places an exclusive lock on the table during the reorganization process, but you can also specify ALLOW READ ACCESS or ALLOW WRITE ACCESS to permit other transactions to read from or update the table.

In-place Reorganization

In-place reorganization allows applications to access the table during the reorganization. In addition, online table reorganization can be paused and resumed later by anyone with the appropriate authority by using the schema and table name.

In-place reorganization is allowed only on tables with type-2 indexes and without extended indexes. To utilize online table reorganization, the REORG command syntax has added the following options:

- **INPLACE**: Reorganizes the table while permitting user access.
- **ALLOW READ ACCESS**: Allows only read access to the table during reorganization.
- **ALLOW WRITE ACCESS**: Allows both read and write access to the table during reorganization. This is the default behavior.

Loading data

As data sizes increase in database systems and user demands for that data approach 24x7x365, one of the challenges we face is loading large volumes of data without impacting the availability of that data for users and applications. Prior to V8 of DB2 UDB, the only method to add data to a table and still have the table available for user and application access was to issue SQL INSERT statements and issue frequent COMMITs. This process can be very slow and places large demands on the logging facility for both disk space and CPU. We discuss the new online loading capabilities introduced in DB2 UDB V8.

Online loading

When loading data into a table in Version 8, the tablespace in which the table resides will no longer be locked. Users have full read and write access to all the tables in the tablespace, except for the table being loaded. For the table being loaded, the existing data in the table will be available for read access if the load is appending data to the table. To utilize the online load feature a new option ALLOW READ ACCESS has been added to the syntax of the LOAD command. When using the ALLOW READ ACCESS option load will lock the target table in a share mode. The table state will be set to both LOAD IN PROGRESS and READ ACCESS. Readers may access the non-delta portion of the data while the table is being load. In other words, data that existed before the start of the load will be accessible by readers to the table, data that is being loaded is not available until the load is complete. This option is not valid if the indexes on the target table are marked as requiring a rebuild.

When there are constraints on the table, the table state will be set to CHECK PENDING as well as LOAD IN PROGRESS, and READ ACCESS. At the end of the load the table state LOAD IN PROGRESS state will be removed but the table states CHECK PENDING and READ ACCESS will remain. The SET INTEGRITY command must be used to take the table out of CHECK PENDING. While the table is in CHECK PENDING and READ ACCESS, the non-delta portion of the data is still accessible to readers, the new (delta) portion of the data will remain inaccessible until the SET INTEGRITY command has completed. A user may perform multiple loads on the same table without issuing a SET INTEGRITY command. Only the original (checked) data will remain visible, however, until the SET INTEGRITY command is issued.

For additional options available for use with the LOAD command see the *DB2 UDB V8 Command Reference*, SC09-4828. These new load features significantly improve the availability of the data and help customers deal with the maintenance of large data volumes and shrinking maintenance windows.

8.2 Scalability

DB2 is a very scalable database product allowing you to create multi-terabyte databases through its partitioning feature by spreading a database across many machines (called a cluster). Even if a multi-terabyte database is not in your plans, partitioning a database allows for a *shared nothing* architecture in which each machine in the cluster performs it's own workload independent from all other machines. This means independent processing, independent memory, independent disks. There are many different partitioning scenarios each of which will be discussed:

- Single partition on a single processor (uniprocessor)

- Single partition with multiple processors (SMP)
- Multiple partition configurations
- Partitions with one processor (MPP)
- Partitions with multiple processors (cluster of SMPs)
- Logical database partitions

Single partition on a single processor

This configuration involves one database partition on a single machine with a single processor (Figure 8-6). This type of configuration may be useful for a small business or department where the number of concurrent users accessing a database and workloads are relatively low.

The addition of disks allows for concurrent I/O to take place thereby increasing workload performance and disk capacity. There is a point where disk capacity will be overweighted against a single processor's capabilities. When this point is near, you may consider adding more processors to increase performance.

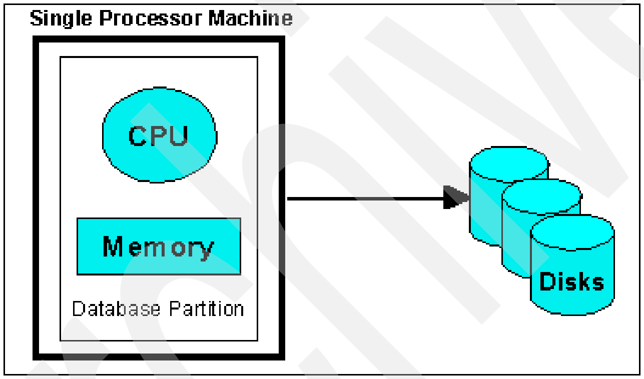


Figure 8-6 Single partition on a single processor

Single partition with multiple processors

This configuration combines several processors to increase workload performance (Figure 8-7). The name for this configuration is called a symmetric multi-processor (SMP) system.

By increasing the number of processors, database operations can occur in parallel thereby increasing processing speed. Workloads such as queries, loading and backing up data, or restoring a database can benefit from multiple processors.

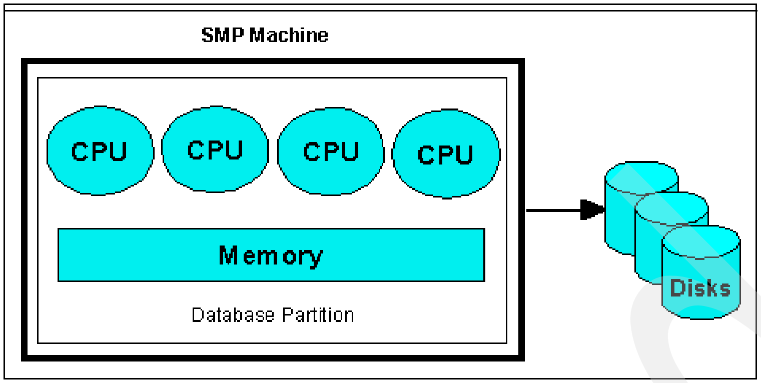


Figure 8-7 Symmetric multiprocessor system

In this configuration, it is helpful to spread your tablespace containers across multiple disks so that the multiple processors are not limited by a single I/O bound disk. If disks are added, you should update the database configuration file parameter num_ioservers to the number of disks available plus an extra two for standby. Having one or more I/O servers for each disk allows for more than one I/O operation to take place at the same time. You can update the database configuration file at the DB2 CLP by typing:

```
db2 update db cfg for <database alias> using num_ioservers <number of
io_servers>
```

If maximum capacity or scalability is reached, you can consider moving to a system with multiple partitions.

Multiple partition on systems with one processor

In this configuration, many database partitions can be added to the cluster of machines (Figure 8-8). Each machine has a physical database partition and its own resources, such as memory, disks, and processors to perform its workload independent of the other machines in the cluster. This type of configuration is called a massively parallel processor (MPP) system.

In a partitioned database environment, database tables are spread across the partitions in approximate equal amounts. When a database operation affects a table, such as a query, each partition performs the query based on the table data it holds. It then sends the data to the main database partition (called a catalog partition) to finish processing the query based on the results returned from each partition.

Although there are separate physical machines involved in the query, the cluster acts a single logical unit and the operation is seamless to the user. The communication that occurs between machines in a cluster is handled by the DB2 communications facility which is configured through the db2nodes.cfg and is located in the SQLLIB directory on Linux and sqllib/DB2 on Windows.

Since each machine handles its own workload independently, we are able to increase memory, disks, and processors multiplied by the number of machines in the cluster. This is a classic example of the *shared nothing* architecture that is emphasized in the database partitioning product.

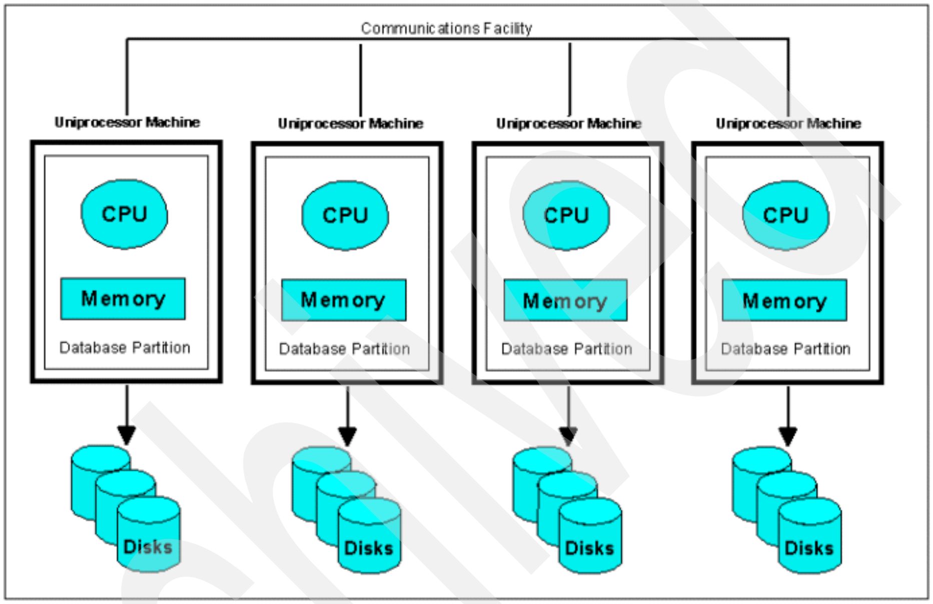


Figure 8-8 Massive parallel processor system

In this environment you can add more database partitions (nodes) to your configuration. If you have reached maximum capacity or scalability, you can consider moving to a system where each partition has multiple processors.

Partitions with multiple processors

As a database grows larger and more concurrent users submit more and larger workloads, processors may need to be added to increase processing speed on each partition. This configuration is called an SMP cluster and has the advantages of both the SMP and MPP system by combining increased speed with the shared nothing physical partition architecture (Figure 8-9).

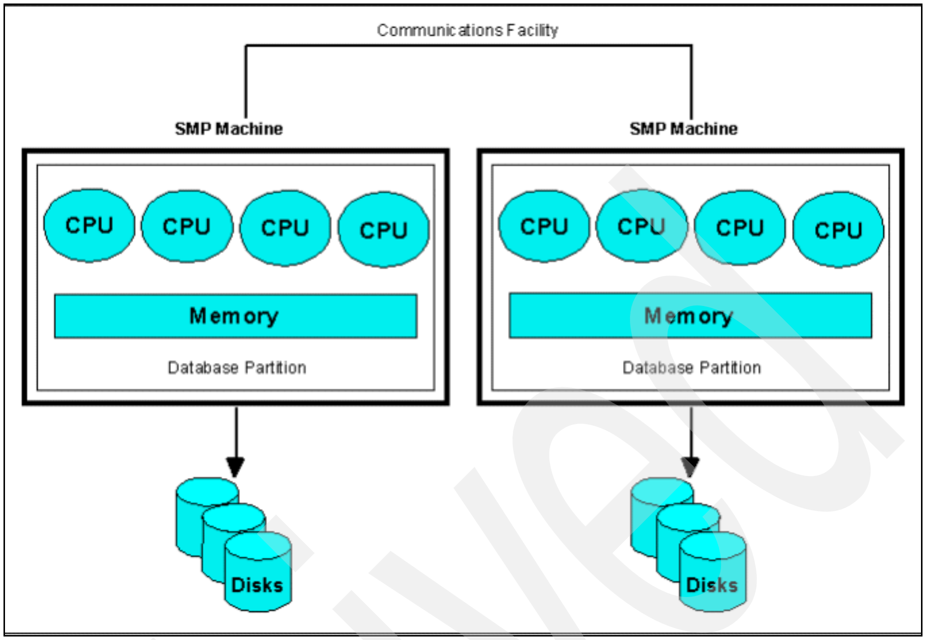


Figure 8-9 SMP cluster system

In this configuration you can add more database partitions and add more processors to existing database partitions.

Logical database partitions

In a logical database partition configuration, a machine has more than one partition. Since two or more partitions reside on the same machine and not physically separated between machines, we call the partitioning logical.

With this configuration, each logical partition allocates a portion of the machine's memory and disk resources and does not share them with the other logical partitions (Figure 8-10). This is a benefit since each partition becomes independent thereby allowing separate administration and recovery of each partition. While disk and memory resources are not shared in a logical partitioned environment, the processors are shared to allow parallel processing to occur within each partition.

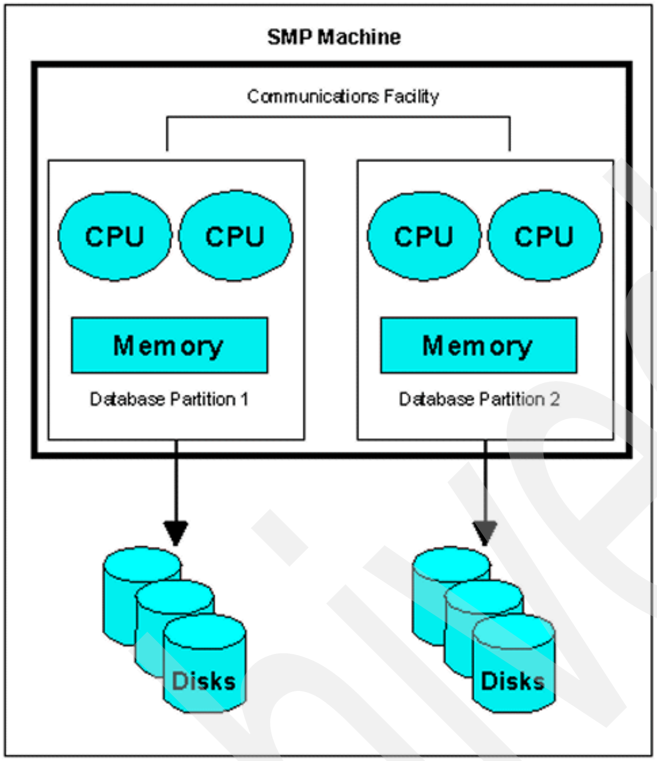


Figure 8-10 Logical partitioned SMP

In Figure 8-10, two logical partitions reside on a single machine. Both disk resources and memory are not shared, but all processors are shared.

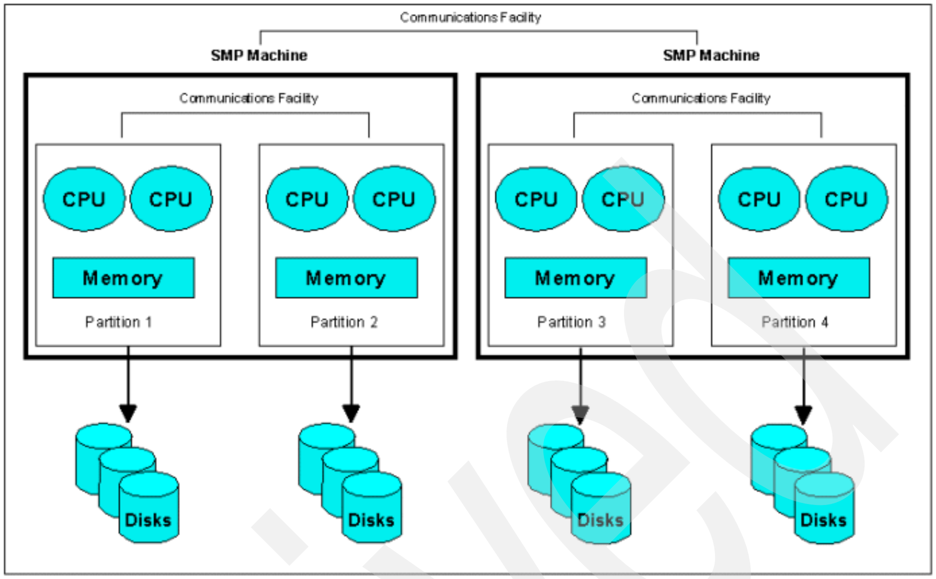


Figure 8-11 Logical and physical partitioning in a clustered SMP system

In Figure 8-11, both logical and physical partitioning are implemented in this two machine cluster. There are two logical partitions for each machine while each set of partitions within a machine are physically partitioned against the other machine.

8.3 Parallelism

To enhance the performance of an operation, such as a database query, it is important to consider such components as the database configuration, the hardware environment, and the type of operation performed. Each of these components should be considered when designing the physical hardware layout and logical database architecture. In this section we discuss the following types of parallelism supported by DB2:

- Input/Output
- Query
- Utility

Input/output parallelism

By spreading one or more tablespace containers across multiple disks, the database manager can exploit parallel I/O. Parallel I/O refers to the process of writing to, or reading from, two or more I/O devices simultaneously. This can result in significant improvements in throughput.

In order to exploit I/O parallelism, there is a registry variable called db2_parallel_io. You can set the db2_parallel_io registry variable in the following manner:

```
db2set DB2_PARALLEL_IO=*
```

The above command will set all tablespaces for I/O parallelism. To determine the degree of parallelism (that is, how many prefetch requests to concurrently issue), the database manager divides the database configuration parameters dft_prefetch_sz by dft_extent_sz. For example, if the prefetch size is three times the extent size, then there are three extent-sized prefetch requests issued concurrently. With a tablespace container on each of three disks, each container simultaneously receives a prefetch request.

Query parallelism

There are two types of query parallelism: inter-query parallelism and intra-query parallelism.

Inter-query parallelism refers to the ability of multiple applications to query a database at the same time. Each query executes independently of the others, but DB2 executes all of them at the same time. DB2 has always supported this type of parallelism.

Intra-query parallelism refers to the simultaneous processing of parts of a single query, using either intra-partition parallelism, inter-partition parallelism, or both.

Intra-partition parallelism

Intra-partition parallelism refers to the ability to break up a query into multiple parts. Some DB2 utilities also perform this type of parallelism.

Intra-partition parallelism subdivides what is usually considered a single database operation such as index creation, database loading, or SQL queries into multiple parts, many or all of which can be run in parallel within a single database partition.

Figure 8-12 shows a query that is broken into four pieces that can be run in parallel, with the results returned more quickly than if the query were run in serial fashion. The pieces are copies of each other. To utilize intra-partition parallelism, you must configure the database appropriately. You can choose the degree of parallelism or let the system do it for you. The degree of parallelism represents the number of pieces of a query running in parallel.

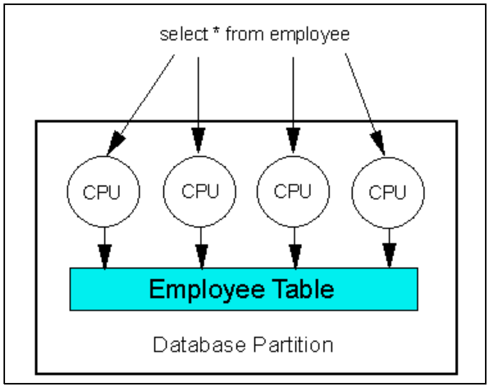


Figure 8-12 Intra partition parallelism

Inter-partition parallelism

Inter-partition parallelism refers to the ability to break up a query into multiple parts across multiple partitions of a partitioned database, on one machine or multiple machines. The query is run in parallel. Some DB2 utilities also perform this type of parallelism.

Inter-partition parallelism subdivides what is usually considered a single database operation such as index creation, database loading, or SQL queries into multiple parts, many or all of which can be run in parallel across multiple partitions of a partitioned database on one machine or on multiple machines.

Figure 8-13 shows a query that is broken into four pieces that can be run in parallel, with the results returned more quickly than if the query were run in serial fashion on a single partition.

The degree of parallelism is largely determined by the number of partitions you create and how you define your database partition groups.

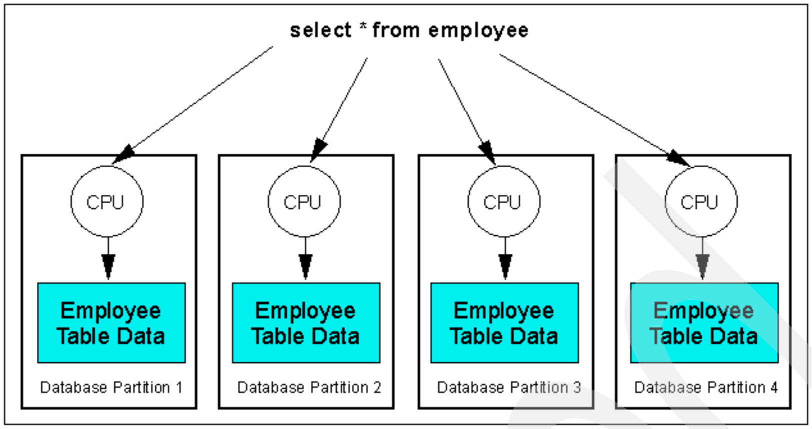


Figure 8-13 Inter partition parallelism

Simultaneous intra-partition and inter-partition parallelism

You can use intra-partition parallelism and inter-partition parallelism at the same time (Figure 8-14). This combination provides two dimensions of parallelism, resulting in an even more dramatic increase in the speed at which queries are processed:

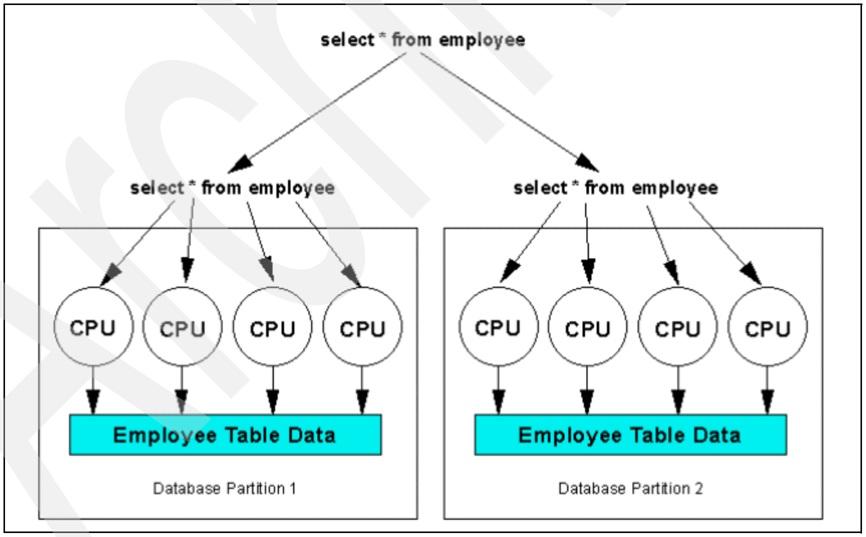


Figure 8-14 Inter and intra-partition parallelism used together

Utility parallelism

DB2 utilities can take advantage of intra-partition parallelism. They can also take advantage of inter-partition parallelism; where multiple database partitions exist, the utilities execute in each of the partitions in parallel.

The load utility can take advantage of intra-partition parallelism and I/O parallelism. Loading data is a CPU-intensive task. The load utility takes advantage of multiple processors for tasks such as parsing and formatting data. It can also use parallel I/O servers to write the data to containers in parallel.

In a partitioned database environment, the LOAD command takes advantage of intra-partition, inter-partition, and I/O parallelism by parallel invocations at each database partition where the table resides.

During index creation, the scanning and subsequent sorting of the data occurs in parallel. DB2 exploits both I/O parallelism and intra-partition parallelism when creating an index. This helps to speed up index creation when a CREATE INDEX statement is issued, during restart (if an index is marked invalid), and during the reorganization of data.

Backing up and restoring data are heavily I/O-bound tasks. DB2 exploits both I/O parallelism and intra-partition parallelism when performing backup and restore operations. Backup exploits I/O parallelism by reading from multiple tablespace containers in parallel, and asynchronously writing to multiple backup media in parallel.

9

Selection guide

Now that you have used DB2 and been introduced to the features on the Linux and Windows platforms, this chapter is answering the question *what's next?* The trial code software you have been using is fully functional but time restricted so in this chapter we will discuss how DB2 is licensed for different requirements. We will also cover the different versions of DB2 that are available covering the products on Linux and Windows in some detail. IBM's new Data Management Tools for multiplatforms are also discussed as they act as an extension to the feature-rich capabilities supplied as standard. Finally we cover how a new DB2 UDB user can obtain education, certification and access a wealth of resources on the Internet.

This chapter contains the following topics:

- DB2 product family
- DB2 UDB product offerings for Linux and Windows
- Licensing
- Data management tools
- Other products
- Going further with DB2 UDB

9.1 DB2 product family

This section introduces the members of the DB2 family. The complete DB2 family comprises the following products:

- **DB2 Universal Database V8.1**

  IBM's relational database management system for AIX, Linux, HP-UX, Sun, and Windows. All of these distributed platforms share 90% of the same code for DB2 with the other 10% used to optimize for the underlying operating system.

- **DB2 Everyplace**

  DB2 relational database and enterprise synchronization architecture for mobile and embedded devices.

- **DB2 Universal Database for iSeries**

  This advanced, 64-bit relational database system provides leading-edge performance in e-business and data warehousing environments.

- **DB2 Universal Database for z/OS and OS/390**

  Use DB2 on the mainframe to run powerful enterprise applications, and make e-commerce a reality.

- **DB2 Server for VSE & VM**

  A full-function RDBMS supporting production and interactive VM and VSE environments for your company

Figure 9-1 illustrates the DB2 product family from mobile devices right through to host systems.

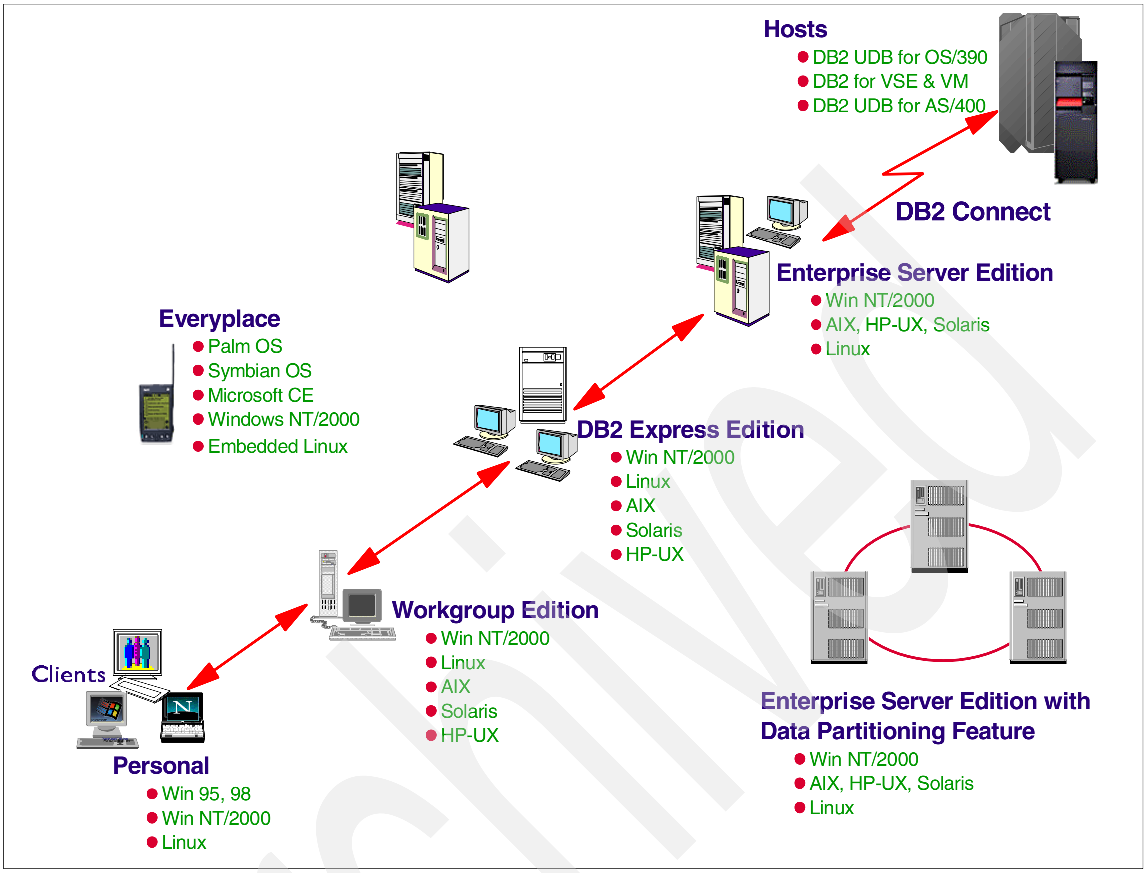


Figure 9-1 The DB2 product family

9.2 DB2 UDB product offerings for Linux and Windows

This section discusses the DB2 UDB product offerings for Linux and Windows. We cover the standard features of each product and any additional features that are available. These products are:

- DB2 Universal Database Enterprise Server Edition
- DB2 Universal Database Workgroup Server Unlimited Edition
- DB2 Universal Database Workgroup Server Edition
- DB2 Universal Database Express Edition
- DB2 Universal Database Personal Edition
- DB2 Universal Developer's Edition
- DB2 Personal Developer's Edition
- DB2 Express

After the products have been discussed, we provide examples of where this product version is most appropriate.

9.2.1 DB2 Enterprise Server Edition (ESE)

DB2 Enterprise Server Edition (ESE) is designed to meet the database server needs of midsize to large businesses. ESE can be deployed in Linux, UNIX, and Windows environments on any size server. The ESE feature set, scalability, reliability, and availability provides the ideal foundation for building data warehouses, transaction processing, or Web based solutions as well as a backend for packaged solutions like ERP, CRM, or SCM. Additionally, ESE offers connectivity and integration for other enterprise DB2 and Informix data sources.

What's included?

- **DB2 Enterprise Server Edition**
- **Administration Client**
- **Application Development Client**
- **Run Time Client**
- **IBM Developer Kit, Java Technology**
- **Distributed Debugger for Java Stored Procedures**
- **Complimentary Products**:
  - **Audio, Image, and Video Extenders**: DB2 Audio Extender adds the power of audio retrieval to SQL queries. You can integrate audio data and traditional business data in a single query. For example, query a music database for audio recordings by a particular artist, and list the date of each recording. Then use the Audio Extender to play the audios.

    DB2 Image Extender adds the power of image retrieval to SQL queries. You can use visual examples of color and texture patterns as search criteria. For example, you can query a photographic database for thumbnail images of all pictures stored in GIF format, and list the name of each picture's photographer. Then use Image Extender to invoke a browser.

    DB2 Video Extender adds the power of video retrieval to SQL queries. You can integrate video data and traditional business data in a single query. For example, you can query a news database for video news clips about a specific subject, and list the playing time of each video clip. Then use the Video Extender to play the video clips
  - **WebSphere Studio Site Developer Advanced (Trial version)**: WebSphere Studio Site Developer is an introduction to the WebSphere Studio experience providing an easy to use, productive and

  comprehensive development environment for building Java, Web and Web services applications.
  - **WebSphere MQ**: WebSphere MQ enables application integration by helping business applications to exchange information across different platforms, sending and receiving data as messages. They take care of network interfaces, assure *once and once only* delivery of messages, deal with communications protocols, dynamically distribute workload across available resources, handle recovery after system problems, and help make programs portable.

    This product was introduced as part of 7.4, “Accessing WebSphere MQ” on page 324.
  - **QMF™ for windows**: QMF is a tightly integrated, powerful, and reliable query and reporting tool set for IBM's DB2 relational database management system, and provides an environment that is easy for a novice to use but powerful enough for an application programmer
  - **Data Management Tools (Try & Buy)**: These tools will be discussed in 9.4, “Data management tools” on page 370.

What is licensed separately?

As well as the products that are included with DB2 UDB ESE there are additional products that can be licensed separately to provide functionality in a specific area. These products are:

- **Database Partitioning Feature (DPF)**: The DB2 UDB Database Partitioning Feature (DPF) is required in order to partition your DB2 UDB ESE database, either within a single server, or across multiple servers. The DPF is a license-only and does not require any additional products on top of DB2 UDB ESE to be installed on your database server to support database partitioning.
- **DB2 Net Search Extender**: This feature combines the in-memory search capability with powerful text search capabilities The high-performance of in-memory search is indispensable for e-commerce applications or any other application with high performance and scalability text-search demands. For example, you can now use in-memory search with your XML documents. Integration with the DB2 optimizer's intelligent strategies ensure high-performance, full-text search working seamlessly within an SQL full SELECT.

  Integrated solutions means ensuring all parts of the solution work together so now you can manage the DB2 Net Search Extender right from the DB2 UDB Control Center significantly simplifying basic tasks like creation of a text index.
- **DB2 Data Links Manager**: This separately ordered DB2 feature that enables DB2 to manage and manipulate data that resides in both unstructured files and in the relational database management system (RDBMS). The files

stored outside of the database reside in file systems accessible over a network, and are managed as if they were stored inside the RDBMS.

DB2 Data Links Manager enables DB2 to provide comprehensive control over files that exist outside your database, ensuring referential integrity, providing enhanced access control, and supporting automatic and coordinated backup and restore capabilities, which are crucial for data management in transactional environments.

- **DB2 Spatial Extender**: DB2 Spatial Extender allows you to store, manage, and analyze spatial data (information about the location of geographic features) in DB2 UDB along with traditional data for text and numbers. This functionality is provided by a set of advanced spatial data types that represent geometries such as points, lines, and polygons and many functions and features that inter-operate with those new data types. These capabilities allow you to integrate spatial information with your business data, adding another element of intelligence to your database.
- **DB2 Warehouse Manager**: DB2 Warehouse Manager adds to the ETL capabilities of Data Warehouse Center including an expanded set of operating systems and a metadata management solution, the Information Catalog Center.

  This feature was discussed in 7.2, “Data Warehouse Center” on page 314.
- **DB2 Intelligent Miner™ Scoring**: DB2 Intelligent Miner Scoring provides scoring technology as a DB2 extender. It extends database capabilities and enables users to deploy data mining in real-time applications with a simple SQL call. DB2 Intelligent Miner Scoring enables your applications to better serve business and consumer users alike - by providing more informed recommendations, personalizing customer and supplier treatment, and building more efficiencies into the online experience to be more responsive to the specific situation at hand
- **DB2 Intelligent Miner Modeling**: DB2 Intelligent Miner Modeling delivers DB2 stored procedures for the following modeling operations: associations discovery, demographic clustering, and tree classification.

  DB2 Intelligent Miner Modeling supports the development of data mining models in a format which conforms with Predictive Model Markup Language PMML) V2.0, the new industry standard for analytic models
- **DB2 Intelligent Miner Visualization**: DB2 Intelligent Miner Visualization provides Java visualizers, to interact and graphically present the results of associations, demographic clustering, and tree classification modeling operations. Input to the visualizers are PMML 2.0 conforming models.

  The visualizers can be included as part of an application or as an applet in a Web browser. The visualizers enable customers to analyze model results in order to gain new business insights

When would you buy DB2 ESE?

DB2 UDB Enterprise Server Edition (ESE) should be considered when either DB2 UDB Workgroup Server Edition or DB2 UDB Workgroup Server Unlimited Edition cannot be used. ESE offers a per processor licensing model. If host system connectivity requirements exceed five concurrent users, additional host user access licenses will be required.

When would you buy the DB2 Database Partitioning Feature?

DPF should be considered for business intelligence (data warehouse) workloads when one or more of the following exist:

- Requirement to spread data across clustered systems
- More than 8-12 CPUs within the SMP system
- Speed of backup/recovery utility operations is critical (as your data volume approaches 100GB)
- Any single table exceeds 64GB using 4K pages, 128GB using 8K pages, 256GB using 16K pages, or 512GB using 32K pages

9.2.2 DB2 Workgroup Server Unlimited Edition (WSUE)

DB2 Workgroup Server Unlimited Edition (WSUE) offers a simplified per processor licensing model for deployment in a departmental or small business environment that has Internet users or number of users that makes per processor licensing more attractive than the WSE licensing model. The WSUE can be deployed in Linux, UNIX, and Windows environments on systems with up to 4 CPU’s.

What’s included?

- DB2 Workgroup Server Unlimited Edition
- Administration Client
- Run Time Client
- IBM Developer Kit, Java Technology
- Distributed Debugger of Java Stored Procedures
- Complimentary Products:
  - Audio, Image, and Video Extenders
  - WebSphere Studio Site Developer Advance (trial version)
  - WebSphere MQ
  - QMF for Windows trial

What is licensed separately?

- DB2 Net Search Extender
- DB2 Spatial Extender

When would you buy DB2 WSUE?

DB2 UDB Workgroup Server Unlimited Edition (WSUE) offers the same capabilities provided with DB2 UDB Workgroup Server Edition, but offers a simplified per processor licensing model.

WSUE should be considered when DB2 UDB Workgroup Server Edition can't be used and you don't require one of the following that require the use of DB2 UDB Enterprise Server Edition:

- Support for more than four CPUs
- Satellite Administration of DB2 UDB Personal Edition installations
- Federated database access
- Database Partitioning Feature
- Host system connectivity required and with the addition of DB2 Connect Enterprise Edition cost exceeds that of DB2 Enterprise Server Edition

9.2.3 DB2 Workgroup Server Edition (WSE)

DB2 Workgroup Server Edition (WSE) is the database server designed for deployment in a departmental or small business environment that involves a small number of internal users. The WSE uses a licensing model designed to provide an attractive price point for smaller installations while still providing a full function database server. WSE licensing includes a base server charge and a charge for each concurrent or registered user. The WSE can be deployed in Linux, UNIX, and Windows environments on systems with up to 4 CPU's.

What's included?

- DB2 Workgroup Server Edition
- Administration Client
- Run Time Client
- IBM Developer Kit, Java Technology
- Distributed Debugger for Java Stored Procedures
- Complementary Products:
  - Net Search Extender (5 user limit)
  - DB2 Spatial Extender (5 user limit)
  - Audio, Image, and Video Extenders
  - WebSphere Studio Site Developer Advanced (trial)
  - WebSphere MQ
  - QMF for Windows (trial)

When would you buy DB2 WSE?

WSE should be considered when the number of concurrent or registered users is small and you don't require one of the following that require the use of either DB2 UDB Enterprise Server Edition or DB2 UDB Workgroup Server Unlimited Edition:

- Support for more than four CPUs
- Number of concurrent/registered users exceed 25-30
- Satellite Administration of distributed PE installations
- Federated database access
- Warehouse Manager feature
- Database Partitioning Feature
- Spatial Extender support of more than 5 users
- Net Search Extender support of more than 5 users
- Host system connectivity required and with the addition of DB2 Connect Enterprise Edition cost exceeds that of DB2 Enterprise Server Edition

9.2.4 DB2 Universal Database Personal Edition (PE)

DB2 Universal Database Personal Edition (PE) provides a single user database engine ideal for deployment to PC based users. The PE includes the ability to be remotely managed, which makes it the perfect choice for deployment in occasionally connected or remote office implementations that don't require multi-user capability. PE is licensed by system, one license for every system it is installed on.

What's included?

- DB2 Personal Edition
- IBM Developer Kit, Java Technology
- Distributed Debugger for Java Stored Procedures
- Complimentary Products:
  - Net Search Extender (singer user)
  - DB2 Spatial Extender (singer user)
  - Audio, Image, and Video Extenders
  - WebSphere Studio Site Developer Advanced (Trial)

When would you buy DB2 PE?

This is the perfect choice for deployment in occasionally connected systems like Laptop or remote office implementations that don't require multi-user capability. PE installed systems can also be a client to a central server with one of the server versions of DB2 UDB installed.

9.2.5 DB2 Universal Developer's Edition

DB2 Universal Developer's Edition offers a low cost package for a single application developer to design, build, and prototype applications for deployment on any of the DB2 client or server platforms. This comprehensive developer's offering includes all client and server DB2 editions, DB2 Connect, the DB2 Extenders™, Warehouse Manager, and Intelligent Miner allowing customers to build solutions that utilize the latest database technologies. The software in this package cannot be used for production systems.

Restriction: Database servers and gateways can be set up for development purposes only and you must acquire one license for each developer.

The package includes:

- DB2 Personal Edition
- DB2 Workgroup Server Edition
- DB2 Enterprise Server Edition
- DB2 Connect Personal Edition
- DB2 Connect Enterprise Edition
- DB2 Data Links Manager
- DB2 Net Search Extender
- DB2 Spatial Extender
- DB2 Warehouse Manager
- DB2 Intelligent Miner Scoring
- DB2 Intelligent Miner Modeling
- DB2 Intelligent Miner Visualization
- Audio, Image, and Video Extenders
- Administration Client
- Application Development Client
- Run Time Client
- IBM Developer Kit, Java Technology
- Distributed Debugger for Java Stored Procedures
- Complementary Products:
  - DB2 Everyplace Software Development Kit: The DB2 Everyplace Software Development Kit consolidates and packages together toolkits used for developing mobile device applications using the DB2 Everyplace Database and the DB2 Everyplace Sync Server
  - WebSphere Application Server, Advanced Developer Edition
  - WebSphere Studio Site Developer Advanced (trial)
  - WebSphere MQ
  - QMF for Windows
  - Data Management Tools: DB2 Web Query Tool, DB2 Table Editor, DB2 High Performance Unload (Try & Buy), Recovery Expert (Try & Buy), Performance Expert (Try & Buy)
  - Borland Products (30-day trial): Borland Delphi Enterprise, Borland C++ Builder Enterprise, Borland Kylix Enterprise

When would you buy DB2 UDE?

UDE should be considered for use on development/test systems where the cost of licensing the developers using the system doesn't exceed that of the licensing of one of the server versions of DB2 UDB.

9.2.6 DB2 Personal Developer's Edition (PDE)

DB2 Personal Developer's Edition (PDE) enables a developer to design and build single user desktop applications. This offering includes Windows and Linux versions of DB2 Personal Edition as well as the DB2 Extenders.

This package includes:

- DB2 Personal Edition
- DB2 Connect Personal Edition
- Application Development Client
- IBM Developer Kit, Java Technology
- Distributed Debugger for Java Stored Procedures
- Complementary Products:
  - DB2 Net Search Extender
  - DB2 Spatial Extender
  - Audio, Image, and Video Extenders
  - WebSphere Studio Site Developer Advanced (trial)
  - Borland Products (30-day trial): Borland Delphi Enterprise, Borland C++ Builder Enterprise, Borland Kylix Enterprise

When would you buy DB2 PDE?

PDE should only be considered for developers that are developing applications to be implemented on DB2 UDB Personal Edition

9.2.7 DB2 Express

DB2 Express is a newly announced full function DB2 database, which is easy to transparently install as part of an application. It has all the autonomic manageability features of the rest of the DB2 family. It comes in minimal number of CDs, and is available in all major geographic languages.

DB2 Express is aimed at the Independent Software Vendors (ISVs) who want to include DB2 as part of an integrated solution within their application, its availability on both SMB pervasive platforms, that is, Linux and Windows, will lower the cost of total IT solution to the end customer.

The DB2 Express most likely will be available in the mid-year of 2003.

When would you buy DB2 Express?

If you are an application vendor wanting to embed a high function, low cost, transparent database into your application

9.3 Licensing

On Windows and Linux platforms DB2 is almost exclusively licensed either by user or by processor. Table 9-1 summarizes the licensing policy for each version of DB2.

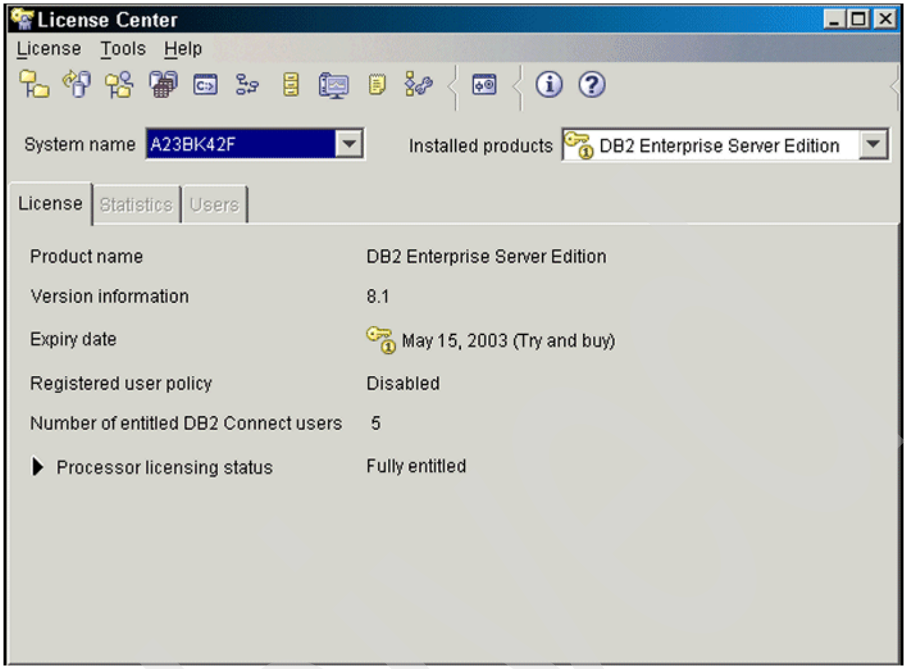
Table 9-1 DB2 licensing

Product	**Per Processor**	**Per User**	**Note**
DB2 ESE	X		
DB2 UDB WSUE	X		
DB2 UDB WSE		X	WSE licensing includes a base server charge and a charge for each concurrent or registered user.
DB2 UDB PE		X	
DB2 UDB UDE		X	
DB2 UDB PDE		X	

9.3.1 Adding a license to DB2

To license your DB2 product the License Center GUI, which can be accessed from within the DB2 Control Center (**Tools —> License Center)**, is provided. Use the License Center to display license status and usage information for DB2 products installed on your system. You can also use the License Center to configure your system for license monitoring. The License Center also allows you to add a new license to change your DB2 product from trial code version or Expired to Licensed status.

Figure 9-2 illustrates the License Center.



System name: A23BK42F

Installed products: DB2 Enterprise Server Edition

Product name	DB2 Enterprise Server Edition
Version information	8.1
Expiry date	May 15, 2003 (Try and buy)
Registered user policy	Disabled
Number of entitled DB2 Connect users	5
▶ Processor licensing status	Fully entitled

Figure 9-2 DB2 Licence Center

There are two ways to add a new license. You can add a license from a file, or by entering the information manually from a license certificate.

To add a license from a file:

1. Open the License Center.
2. In the **System name** field, select the system for which you want to add a new license.
3. Select **License -> Add**. The Add License window opens.
4. Click the **From a file** radio button.
5. Select the license file (*.lic) that you want to add.
6. Click **Apply** to add the license and keep the window open, or click **OK** to close the window.

To add a license manually:

1. Open the License Center.
2. In the **System name** field, select the system for which you want to add a new license.

3. Select **License -> Add**. The Add License window opens.
4. Click the **Manually** radio button. This displays the license information fields for you to fill in.
5. In the **Product name** field, type the product name.
6. In the **Product password** field, type the license password that appears on your license certificate.
7. Optional: In the **Annotation** field, type the annotation data if it appears on your license certificate.
8. Optional: In the **Comment** field, type comments about the license.
9. Optional: In the **Vendor name** field, type the name of the vendor.
10. Optional: In the **Vendor ID** field, type the ID of the vendor.
11. Click **Apply** to add the license and keep the window open, or click **OK** to close the window.

Note: Licensing can also be performed using the `db2licm` command.

9.3.2 Upgrade DB2 trial code

The trial code that you have downloaded can be upgraded to be a fully licensed DB2 product. This upgrade will require a DB2 license. The information of obtaining the DB2 license is discussed in 9.6, “Going further with DB2 UDB” on page 376.

Note: All versions of DB2 Clients (Administration Client, Application Client, and Run-time Client are freely available from IBM.

9.4 Data management tools

DB2 Multiplatform Tools from IBM are key enabling technologies for maintaining high database performance and availability and data integrity. Designed from the ground up to extend and enhance IBM DB2 Universal Database, the tools help database administrators keep their databases continuously operational and optimized for all of the constituencies that need to access them: employees, partners, and customers.

The following are tools that complement and enhance the capability of DB2 UDB:

- DB2 Performance Expert for Multiplatforms
- DB2 Recovery Expert for Multiplatforms
- DB2 High Performance Unload for Multiplatforms

- DB2 Table Editor for Multiplatforms
- DB2 Web Query tool for Multiplatforms

The Information Management tools Web site provides additional information on these products:

http://www.ibm.com/software/data/tools/mptools.html

9.4.1 DB2 Performance Expert

The DB2 Performance Expert for Multiplatforms integrates performance monitoring, reporting, buffer pool analysis, and a Performance Warehouse function into one tool. It provides a single system overview that monitors all subsystems and instances across many different platforms in a consistent way.

Performance Expert features include:

- Analyzes, controls, and tunes the performance of DB2 and DB2 applications
- Provides expert analysis, a real-time online monitor, and a wide range of reports for analyzing and optimizing DB2 application and SQL Statements.
- Includes a Performance Warehouse for storing performance data and analysis tools
- Defines and applies analysis functions (rules of thumb, queries) to identify performance bottlenecks
- Performance Expert on multiplatforms includes the DB2 Buffer Pool Analyzer which collects data and provides reports on related event activity and obtains information on current buffer pool behavior. It can provide these reports in the form of tables, pie charts, and diagrams.
- Includes a starter set of autonomic features that provide recommendations for system tuning to gain optimum throughput
- Monitors DB2 Connect Gateways, including application and system-related information.

9.4.2 DB2 Recovery Expert

IBM DB2 Recovery Expert for Multiplatforms provides simplified, comprehensive, and automated recovery with extensive diagnostics and autonomic capabilities to minimize outage duration.

Recovery Expert features include:

- Providing targeted, flexible and automated recovery of database assets, even as systems remain online.

- Allows expert or novice DBAs to recover database objects safely, precisely and quickly without having to resort to full disaster recovery processes.
- Offers precision recovery options to support safe database development and maintenance.
- Has built-in self-managing and resource tuning features that provide intelligent analysis of altered, incorrect or missing database assets including tables, indexes, or data and automates the process of rebuilding those assets to a correct point in time, all without disruption to normal database or business operations.

9.4.3 DB2 High Performance Unload

IBM DB2 High Performance Unload for Multiplatforms quickly and efficiently unloads and extracts data from DB2 Universal Database Enterprise Edition for movement across enterprise systems or for reorganization in-place.

High Performance Unload features include:

- Performs sequential table reading, providing fast access to IBM DB2 data.
- Creates multiple output files during a single unload at minimal cost.
- Makes the output file-set a tape, disk, or named pipe that is formatted suitably for optional reorganization or reload.
- Performs unloads against backups to eliminate interference with DB2 production databases. The backups can be the last full copy or any full copy.

9.4.4 DB2 Table Editor

IBM DB2 Table Editor for Multiplatforms quickly and easily accesses, updates, and deletes data across multiple DB2 database platforms

Table Editor features include:

- Easily navigate IBM DB2 databases, tables and views; find related data; and quickly updates, deletes, or creates data with full support for your existing DB2 security and logon IDs.
- Edits DB2 tables everywhere with your choice of end-user entry points: Java-enabled Web browsers; Java-based interfaces launched from the IBM DB2 Control Center; Microsoft Windows or an ISPF interface.
- Provides drag-and-drop and wizards to rapidly create versatile, task-specific Java- or Windows-based table editing forms containing built-in data validation and business rules.

9.4.5 DB2 Web Query tool

DB2 Web Query Tool for Multiplatforms enables rapid access to DB2 data from any Web browser. It has the ability to connect users directly to multiple enterprise DB2 and Informix database servers securely, simultaneous, and seamlessly, regardless of hardware or software platforms. The Web Query Tool can also format data into XML and can be interfaced via WAP and e-mail.

Web Query tool features include:

- Enabling complex querying, data comparisons, and customized presentations.
- Providing rapid global access to business information over e-mail clients, including WAP-enabled devices such as PDAs, wireless phones, and text pagers.
- Supporting standard browsers, giving administrators, developers, and end users the ability to build queries that support multiple DB2 platforms, share and run the queries, and convert the results to XML and other highly transportable file formats.
- Is a J2EE-compliant Web application so it can be deployed on WebSphere Application Server.

9.5 Other products

This section introduces additional products in the IBM Data Management portfolio that complement the base DB2 product. These are:

- DB2 Connect
- DB2 Information Integrator

9.5.1 DB2 Connect

DB2 Connect makes your company's host data directly available to your Personal Computer and LAN-based workstations. It connects desktop and palm-top applications to your company's mainframe and minicomputer host databases, leveraging your enterprise information no matter where it is.

DB2 Connects provides the application enablement and robust, highly scalable communication infrastructure for connecting Web, Windows, UNIX, Linux, OS/2® and mobile applications to S/390 and AS/400® data.

DB2 Connect features include:

- Seamlessly connects popular desktop applications to S/390 and AS/400 data

- Enables fast, secure access to legacy data through intranets, extranets or the public Internet
- Integrates new Web-based applications with existing core business applications
- Provides the performance, scalability, reliability and availability needed for the most demanding e-commerce, CRM, BI, and ERP applications
- Provides extensive application programming tools for developing client-server and Web applications using industry standard APIs such as ODBC, ADO, OLE DB, JDBC, SQLJ, DB2 CLI and Embedded SQL.
- Allows you to build new Internet applications and extend existing applications such as data warehousing, data mining, OLTP and OLAP to the Web
- Integrates with both Java and Microsoft models for developing new Web-based applications
- Energizes mobile PC users and users of the new pervasive computing devices with reliable, up-to-date data from S/390 and AS/400 database servers

DB2 Connect is packaged in the following four ways:

- **DB2 Connect Personal Edition**: DB2 Connect Personal Edition provides the application programming interface (API) drivers and connectivity infrastructure to enable direct connectivity from desktop applications to mainframe (zSeries, S/390, VM/VSE) and iSeries (AS/400) database servers. This product is specifically designed and is licensed for enabling two-tier client-server applications running on individual workstations and as such is not appropriate for use on servers.
- **DB2 Connect Enterprise Edition**: DB2 Connect Enterprise Edition addresses the needs of organizations that require robust connectivity from a variety of desktop systems to mainframe and iSeries database servers. DB2 client software is deployed on desktop systems and provides API drivers that connect client-server applications running on these desktop systems to a DB2 Connect server (gateway) that accesses host data. The licensing model for this product is user based.
- **DB2 Connect Application Server Edition**: DB2 Connect Application Server Edition product is identical to the DB2 Connect Enterprise Server in its technology. However, its licensing terms and conditions are meant to address specific needs of multi tier client-server applications as well as applications that utilize Web technologies. DB2 Connect Application Server Edition license charges are based on the size of the number of processors available to the application servers where the application is running.
- **DB2 Connect Unlimited Edition**: DB2 Connect Unlimited Edition product is ideal for organizations with extensive usage of DB2 Connect, especially

where multiple applications are involved. This product provides program code of the DB2 Connect Personal Edition as well as program code identical to the DB2 Connect Application Server Edition for unlimited deployment throughout an organization.

A five registered user license of DB2 Connect is included as part of the DB2 UDB ESE.

DB2 Personal Developer's Edition provides all the tools for one software developer to develop desktop business tools and applications including DB2 Connect

For more information on DB2 Connect, refer to the DB2 Connect home page:

http://www.ibm.com/software/data/db2/db2connect/

9.5.2 DB2 Information Integrator

DB2 Information Integrator V8.1 is a new product to access structured and unstructured information across and beyond the enterprise. Components include a federated data server and a replication server, to integrate diverse data types on demand in real time. Applications that use SQL or tools that generate SQL (e.g. integrated development environments or reporting and analytical tools) can access, integrate, and manipulate distributed and diverse data through a federated system server as if it were a single data source.

At the time of publishing this product is currently in beta.

Federated Data Server

The federated data server lets administrators:

- Configure data source access
- Define integrated views across diverse and distributed data -- including relational databases, content repositories, XML documents, message queues, Web services, spreadsheets, flat files, E-mail databases, Internet content, and more.
- Use integrated graphical tools to configure access to source data, representing that data as logical tables in the federated data server.
- Compose integrated views across these sources using standard SQL view definitions and expressions.
- Define caching strategies for federated data

Applications can:

- Access the server by either traditional database or Web service clients.

- Query or search across the aggregated data sources as if they were in a single database.
  - The query is expressed using standard SQL.
  - SQL expressions may be used to transform the data for business analysis or data exchange.
  - Results can be presented as standard SQL results sets or as XML documents

A developer toolkit is provided to add access to other sources.

Replication Server for Mixed Relational Databases

Administrators can:

- Replicate data between mixed relational data sources including DB2, Informix, Oracle, Microsoft, Sybase, and Teradata.
- Configure a variety of topologies, latency, and consistency characteristics.

The replication server supports distribution (moving data from one database to many) and consolidation (moving data from many databases to one) scenarios. Transformation can be applied in-line with the data movement via standard SQL expressions or stored procedure execution.

Data movement can be:

- Automated to occur on a specific schedule, at designated intervals, continuously, or as triggered by events.
- Managed table-at-a-time (such as for warehouse loading during batch windows) or with transaction consistency (for data that is never off-line).

For more information on IBM Information Integrator, the home page can found at:

`http://www.ibm.com/software/data/integration/iipreview.html`

9.6 Going further with DB2 UDB

Throughout this book we introduce DB2, discuss its main characteristics and provide examples of its use. To conclude we want to provide information about:

- How to buy DB2
- Where you can find out about education on DB2
- How you can get certified in DB2
- Where you can find additional information about DB2

The DB2 Web site provides detailed information about these topics and acts as a pointer to further information.

`http://www.ibm.com/software/data/db2/`

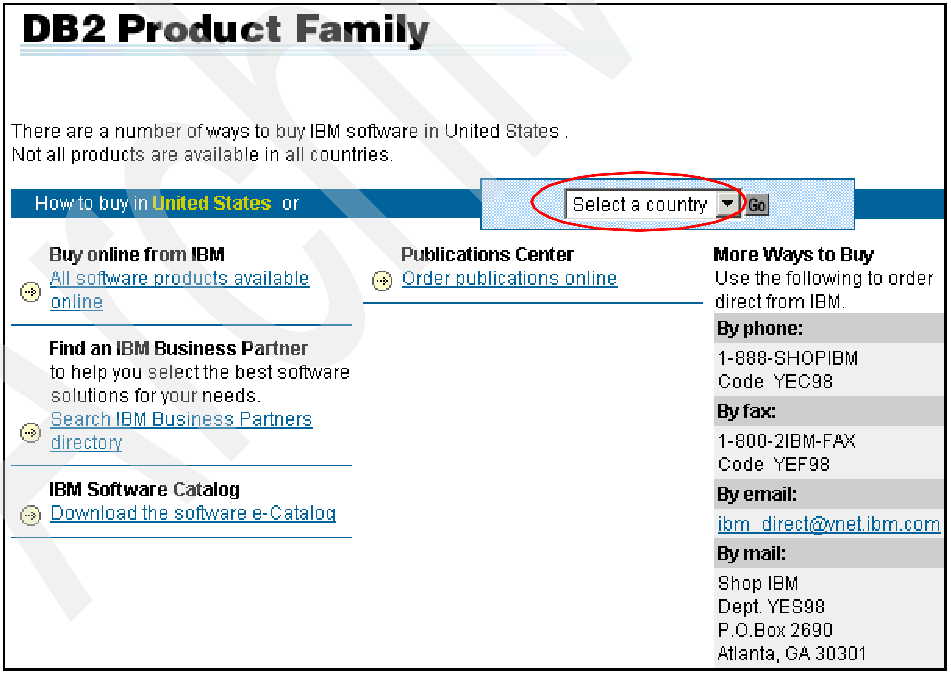
Our aim with this redbook has been to leave the reader proficient with DB2 UDB. By following the links specified in this section we hope the reader will become an expert in the use of DB2 UDB.

9.6.1 How to buy DB2

If you are interested in buying licensed copies of DB2, or upgrading your trial code license, then the available methods of purchase can depend on the country (or region) you are buying the software in.

For country (or region) specific details on how to buy DB2:

- Go to `http://www.ibm.com/software/data/db2/`
- Click **How to Buy**.
- You will then see a window similar to Figure 9-3.



DB2 Product Family

There are a number of ways to buy IBM software in United States .
Not all products are available in all countries.

How to buy in **United States** or

Buy online from IBM
All software products available online

Find an IBM Business Partner
to help you select the best software solutions for your needs.
Search IBM Business Partners directory

IBM Software Catalog
Download the software e-Catalog

Publications Center
Order publications online

More Ways to Buy
Use the following to order direct from IBM.

By phone:
1-888-SHOPIBM
Code YEC98

By fax:
1-800-2IBM-FAX
Code YEF98

By email:
ibm_direct@vnet.ibm.com

By mail:
Shop IBM
Dept. YES98
P.O.Box 2690
Atlanta, GA 30301

Figure 9-3 Buying DB2

- The default country (or region) that the page opens on is the US, but you can select your country (or region) by using the drop down list.
- You will then be offered the options on how you can contact IBM about purchasing DB2.
- As well as the options offered on this page you can also contact your local IBM representative should you require additional information.

9.6.2 Where you can find out about education on DB2

To discover about what DB2 courses are available there is a dedicated Web site:

http://www.ibm.com/services/learning/us/catalog/db2/index.html

This Web site provides a view of the full curriculum for DB2 education and is shown in Figure 9-4.

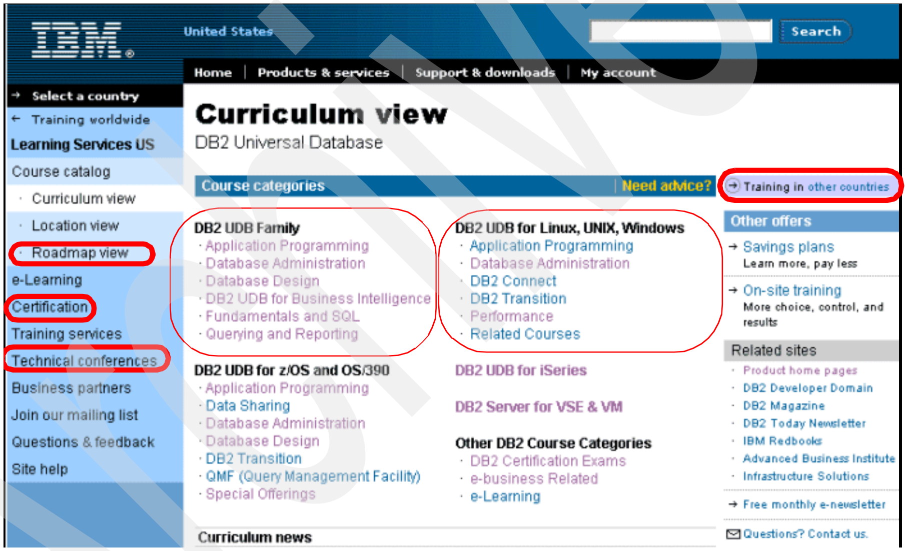


Figure 9-4 DB2 training Web site

Although this initial Web site is again for the US, training in other countries (or regions) can be accessed by selecting the highlighted link.

The initial training Web sites provide links to:

- **e-Learning**: e-learning is the perfect complement to your classroom training. IBM Learning Services offers you the ability to learn in new ways including:

- Digital Video Library: A CD-ROM library of some of our most popular classroom courses packaged in a rich video format and combined with an interactive Web component.
- e-Classes: Scheduled instructor-led classroom experience delivered over the Web.
- e-Tutorials: Online, self-paced courses with access to instructor via instant messaging.

- **Classroom based courses**: As you will see in Figure 9-4 above the training for the Unix and Windows platforms is split into specialized training for these platforms and as part of a more general DB2 curriculum. If you click any of the course categories they will take you to the courses that are part of this category. The course schedules can be examined online and the course booked.
- **Course roadmaps**: Course roadmaps guide you on your path to greater knowledge about DB2. Roadmaps identify courses in their logical sequence to complete a specific curriculum or certification program.

  If you click the **Roadmap view** on the Web site and then **DB2** and then **DB2 UDB for Linux, UNIX, and Windows** you can then choose your area of interest (e.g. DB2 UDB V8 Certification) and be presented with a roadmap that charts what course should be taken in what order. This is illustrated in Figure 9-5.

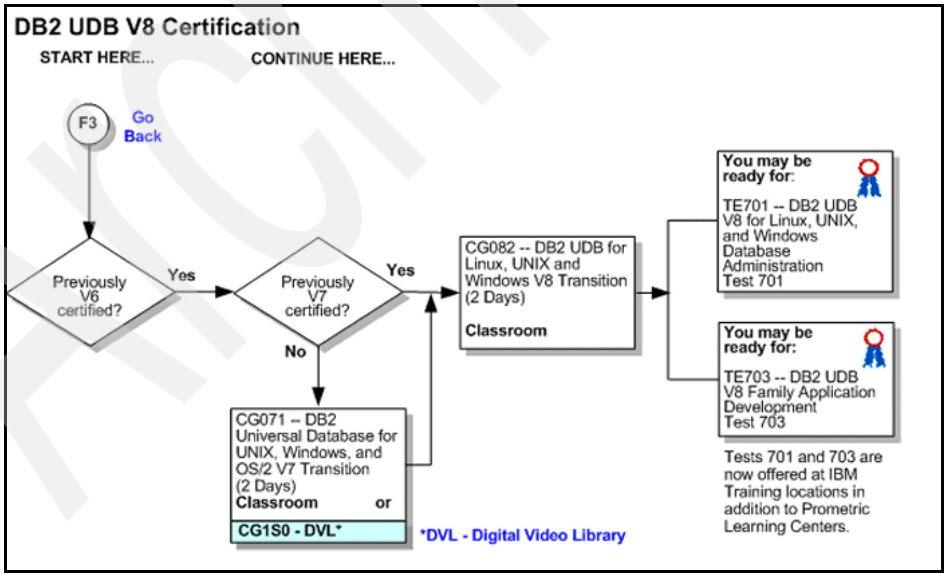

DB2 UDB V8 Certification
START HERE...
CONTINUE HERE...
F3
Go Back
Previously V6 certified?
Yes
Previously V7 certified?
Yes
No
CG082 -- DB2 UDB for Linux, UNIX and Windows V8 Transition (2 Days)
Classroom
CG071 -- DB2 Universal Database for UNIX, Windows, and OS/2 V7 Transition (2 Days)
Classroom or
CG1S0 - DVL*
*DVL - Digital Video Library
You may be ready for:
TE701 -- DB2 UDB V8 for Linux, UNIX, and Windows Database Administration Test 701
You may be ready for:
TE703 -- DB2 UDB V8 Family Application Development Test 703
Tests 701 and 703 are now offered at IBM Training locations in addition to Prometric Learning Centers.

Figure 9-5 Roadmap to DB2 V8 Certification

- **Technical conference information**: The training Web site also provides you with a link to the various technical conferences (click **Technical Conferences**) that are run by IBM.

  There is also a highly respected and well attended DB2 conference run by the International DB2 Users Group (IDUG); details about IDUG can be found at this Web site:

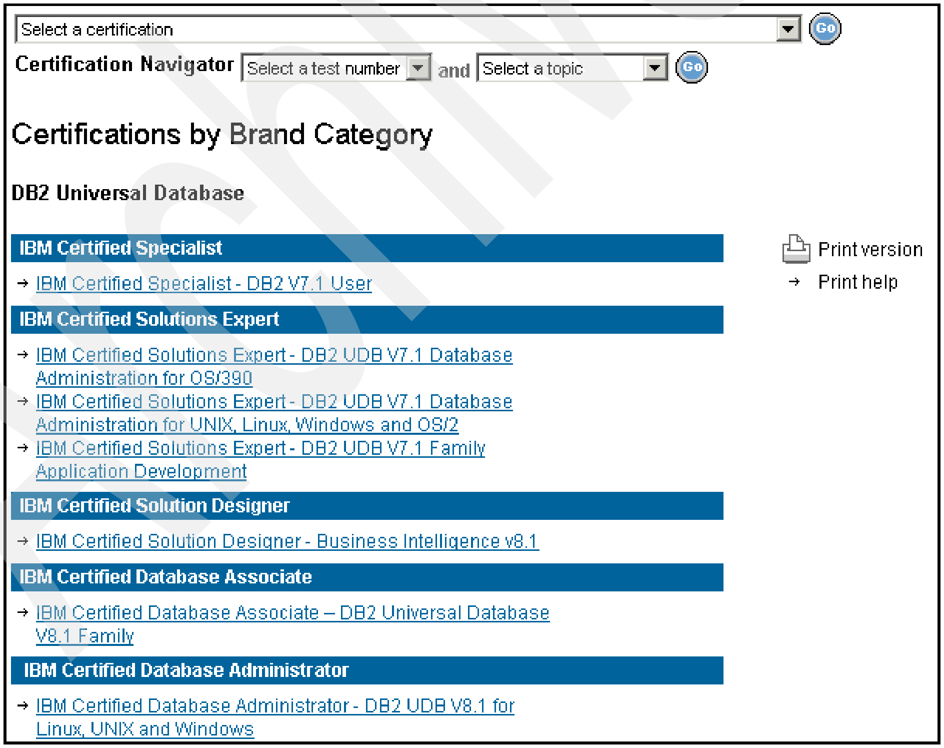
  `http://www.idug.org/`

9.6.3 How you can get certified in DB2

Once you have gained experience with DB2 and received formal classroom experience it is rewarding to have your level of expertise recognized by formal professional certification.

The DB2 certification information can be accessed by selecting **Certification** from the training Web site and selecting **DB2**.

The IBM certification Web site for DB2 is then launched as in Figure 9-6.



Certifications by Brand Category

DB2 Universal Database

IBM Certified Specialist

- IBM Certified Specialist - DB2 V7.1 User

IBM Certified Solutions Expert

- IBM Certified Solutions Expert - DB2 UDB V7.1 Database Administration for OS/390
- IBM Certified Solutions Expert - DB2 UDB V7.1 Database Administration for UNIX, Linux, Windows and OS/2
- IBM Certified Solutions Expert - DB2 UDB V7.1 Family Application Development

IBM Certified Solution Designer

- IBM Certified Solution Designer - Business Intelligence v8.1

IBM Certified Database Associate

- IBM Certified Database Associate – DB2 Universal Database V8.1 Family

IBM Certified Database Administrator

- IBM Certified Database Administrator - DB2 UDB V8.1 for Linux, UNIX and Windows

Figure 9-6 DB2 Certifications Web site

This Web site can also be directly accessed from:

`http://www.ibm.com/certify/certs/db_index.shtml`

The Web site allows you to click on the test and the provides you with:

- Test objectives
- Sample tests, which can be taken online
- Recommended educational resources

Once you have decided you want to sit an official test you can book the test via one of our testing partners:

- Testing in EMEA/AP at VUE:

  `http://www.vue.com/`

- Testing worldwide courteously of Prometric, Inc. (formerly Sylvan Prometric):

  `http://www.2test.com`

9.6.4 Where you can find additional information about DB2

As might be expected there are a large number of places that contain information about DB2. Indeed throughout this book we have provided references to many articles and books. Our aim in this section is to provide a pointers to the most common Web sites.

General DB2 information

- The IBM DB2 UDB Web site comprehensively covers DB2 UDB and acts as a portal to other sources of information

  `http://www.ibm.com/software/data/db2/`

Application Development with DB2

- This page contains information related to application development using DB2 Universal Database Version 8

  `http://www.ibm.com/software/data/db2/udb/ad/`

- The DB2 Developer Domain provides articles, papers, tutorials and general information on technical aspects of DB2 UDB

  `http://www7b.software.ibm.com/dmdd`/

Support

- The DB2 Support Web site provides support, hints and tips, product information and the ability to search for particular issues.

  `http://www.ibm.com/software/data/db2/udb/support.html`

Migrating to DB2

- The DB2 migration Web site has information to aid the migration from other relational databases to DB2 UDB. There is also a freely downloadable tool to automate the process of converting from Oracle, Sybase and SQL Server

  http://www.ibm.com/software/data/db2/migration/

Publications

- DB2 magazine is a solutions oriented magazine intended to provide IT professionals with the strategic and technical information they need to work successfully in the IBM Data Management environment

  http://www.db2mag.com/

- Many IBM Redbooks contain invaluable technical information and experience about DB2. Go to the Web site and search for DB2.

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

User Groups

- The International DB2 Users Group (IDUG) is an independent, not-for-profit, user-run organization whose mission is to support and strengthen the information services community by providing the highest quality education and services designed to promote the effective utilization of the DB2 family of products.

  http://www.idug.org/

News groups

- News groups are an invaluable way to communicate with other DB2 professionals. The main DB2 UDB news group (`ibm.software.db2.udb`) is on the `news.software.ibm.com` server. The comp.databases.ibm-db2 of Internet newsgroup is a forum/discussion group available for communicating with other Db2 professionals.

Search Engine

- It may sound obvious but in some cases putting your DB2 question into a search engine can be very rewarding.

A

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

```
ftp://www.redbooks.ibm.com/redbooks/SG246934
```

Alternatively, you can go to the IBM Redbooks Web site at:

```
ibm.com/redbooks
```

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246934.

Using the Web material

The additional Web material that accompanies this redbook includes the following files:

apacheDB2PHP.sh	Script to download and install Apache and PHP
rborders.ddl	DDL to create RBORDERS database

rborders.load	DDL to load data to tables
insert_part.ddl.trg	DDL to ccreate triggle
update_order.ddl.trg	DDL to create trigger
create_order.pro	DDL to create store procedure
insert_part.pro	DDL to create store procedure
customer.tbl	CUSTOMER table data
forms.tbl	FORMS table data
lineitem.tbl	LINEITEM table data
nation.tbl	NATION table data
orders.tbl	ORDERS table data
part.tbl	PART table data
partsupp.tbl	PARTSUPP table data
region.tbl	REGION table data
supplier.tbl	SUPPLIER table data

Each of the following files contains all the above files:

sg246934_sample.zip	WinZipped Code Samples of Redbooks ordering shopping cart application for Windows platform
sg246934_sample.tar.gz	TarZipped Code Samples of Redbooks ordering shopping cart application for Linux platform

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space:	8MB minimum
Operating System:	Windows 2000 or Linux Red Hat or Linux SuSE
Processor:	Intel 386 or higher
Memory:	16MB

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

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 387. Note that some of the documents referenced here may be available in softcopy only.

- *Up and Running with DB2 for Linux,* SG24-6899
- *DB2 UDB Exploitation of the Windows Environment*, SG24-6893

Other publications

These publications are also relevant as further information sources:

- *IBM DB2 Universal Database Command Reference*, 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 Universal Database Quick Beginnings for DB2 Servers*, GC09-4836
- *IBM DB2 Universal Database Installation and Configuration Supplement*, GC09-4837

Online resources

These Web sites and URLs are also relevant as further information sources:

DB2

- Database and Data Management home page

  http://www.ibm.com/software/data/
- DB2 Developer Domain

  http://www7b.software.ibm.com/dmdd/
- DB2 Universal Database home page

  http://www.ibm.com/software/data/db2/udb/
- DB2 Universal Database for Linux home page

  http://www.ibm.com/software/data/db2/linux/

- DB2 Technical Support

  http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/index.d2w/report
- DB2 Extenders home page

  http://www.ibm.com/software/data/db2/extenders/
- IBM Manuals for Data Management Products

  http://www.ibm.com/software/data/db2/library/

Linux

- IBM Software for Linux

  http://www.ibm.com/software/is/mp/linux/software/
- SuSE home page

  http://www.suse.com/index_us.html
- Red Hat home page

  http://www.redhat.com/

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

Index

IBM

Redbooks

DB2 UDB Evaluation Guide for Linux and Windows

(0.5" spine)
0.475"<->0.875"
250 <-> 459 pages

IBM®

DB2 UDB Evaluation Guide for Linux and Windows

Redbooks

Install and configure DB2 UDB V8.1 for Linux and Windows

Build a Web-based application using DB2 UDB and PHP

Explore the new features of DB2 UDB V8.1

You ask yourself, "So I have downloaded DB2 UDB Version 8 for Linux or Windows. What's next?" This IBM Redbook helps you journey through the experiences of discovering the new DB2 UDB V8.1. This journey introduces you to existing features and enhancements that make the DB2 family of products a superb choice for a relational database.

We start out by showing you how to install DB2 in both the Linux and Windows environments. Next, we introduce you to the graphical DB2 tools, enabling you to become productive more quickly. Once familiarized with DB2, we gear you up for the Internet by providing a dynamic PHP Web application to manage a book store through a DB2 database. This provides a foundation for developing applications with DB2. Building on the Web application, we describe, in detail, all of the DB2 application development features and the programming languages supported.

For the database administrator, the autonomic capabilities of DB2 UDB Version 8.1 act as your 24x7 assistant. We fully cover the database administration functions of DB2 including database configuration, support and monitoring tools, and wizards. Finally, we provide DB2 product information, a selection guide, and a roadmap to help you become a DB2 certified professional.

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

SG24-6934-00 ISBN 0738429260