SG24-4893-00

DB2 Meets Windows NT

July 1997

International Technical Support Organization
Austin Center

International Technical Support Organization

SG24-4893-00

DB2 Meets Windows NT

July 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 315.

First Edition (July 1997)

This edition applies to Version 2.1.2 of DB2 for Windows NT.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

Contents

Figures

Tables

Preface

DB2 Common Server makes this award-winning database product available on virtually any platform. This means you have more choices and flexibility than ever before to connect to your data wherever it is, from wherever your users and decision makers are. DB2 Common Server gives you the tools and the power to design robust, industrial-strength applications to address all of you database needs, no matter how complex.

Written by DB2 experts from across the globe, this book is unique in its coverage of DB2 Common Server for the Windows NT platform. Everything from detailed security information to using DB2 utilities and and enabling Open Database Connectivity (ODBC) applications is covered in a step-by-step, easy-to-follow manner.

If you are a database administrator, system administrator, security administrator, system operator, or programmer, and need to design, implement and maintain a database system accessed by local or remote clients, or if you need to understand and use DB2 on the Windows NT platform, this book is a must!

How This Redbook Is Organized

- Chapter 1, "Overview" on page 1

  This provides an overview of the DB2 products, in particular DB2 for NT.

- Chapter 2, "DB2 and Windows NT Security" on page 27

  This provides an overall view of security within Windows NT. Also discussed is how security for the DB2 products works in a Windows NT environment.

- Chapter 3, "Using DB2 Utilities in the NT Environment" on page 67

  This chapter looks at some of the utilities that a database administration would use in a DB2 for NT environement. Topics such as the IMPORT/EXPORT utility, LOAD and BACKUP and RESTORE are discussed.

- Chapter 4, "DB2 for NT Server Communication" on page 95

  Everything about configuring a DB2 for NT database server for remote communication is covered in this chapter. Also presented is connection to a DRDA host database.

- Chapter 5, "Enabling Remote Clients" on page 151 discusses the configuration of remote clients connecting to a DB2 for NT database server.

- Chapter 6, "Performance and Event Monitoring in DB2 for NT" on page 225.

  This chapter looks at the performance tools available within NT that you can use to help monitor and tune your DB2 for NT environment. Also discussed is some of the facilities within the DB2 product for performance monitoring.

- Chapter 7, "Problem Determination" on page 267 provides helpful tips and techniques in diagnosing problems that you may encounter in a DB2 for NT environment.

The Team That Wrote This Redbook

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

Calene Janacek is a project leader at the International Technical Support Organization, Austin Center. She writes extensively and teaches IBM classes worldwide on all areas of DB2 Universal Database.

Anthony Cunningham is a DB2 support specialist in the UK. He has three years of IBM experience in DB2, mainly in MVS and recently in the Common Server platforms. His areas of expertise in DB2 include DRDA, LAN, and Common Server. In addition to this book, Anthony contributed to the *DB2 Common Server Connectivity Guide*.

Samit S. Korgaonkar is a Senior IT database specialist in India. He has been with IBM for two years and has four years experience with the Software Solutions division. His areas of expertise include implementing enterprise wide database solutions. He has written extensively on Client/Server implementation, including connectivity, ODBC and JDBC. He holds a Bachelor's degree in computer engineering from Bombay University.

Tomonari Kunimori is a database specialist in Japan.

Rodolphe Michel is a database specialist currently on assignment in the UK. Prior to this assignment he was a member of the DB2 Common Server development team at the IBM Almaden and Santa Teresa laboratories in California. He has 12 years experience in relational and object-oriented technologies and has been with IBM for 10 years. Rodolphe holds a Ph.D. in Computer Science from the University of Paris and has published several papers on database management. Prior to joining IBM he was a research fellow at UCLA (Los Angeles).

Grant Sainsbury has been a database architect for the National Bank of New Zealand for three years. His areas of expertise include DB2, Windows NT and Client/Server.

Dwaine Snow is currently the Certification and Education Coordinator for DB2 Common Server in Toronto, Canada. He has been involved in the development and support of DB2 Common Server since 1992.

Joachim Stumpf has degrees in mechanical engineering and Information technology. During his studies, he learned about database design and technology. His dissertation, ″Design of a Process in a Manufacturing Control System,″ included DB2 for OS/2 as the database. An IBMer since 1990, Joachim began working with DB2 in 1994 in marketing, giving product overviews and attending fairs. Currently he is in second-level product support for DB2 Common Server for NT, which includes education and support for business partners.

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

Dale McInnis
IBM Toronto Lab

Tom Arbuckle
International Technical Support Organization, Austin Center

Eddie Daghelian
IBM Toronto

Comments Welcome

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

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Overview

This chapter gives you an overview of the DB2 family of products for the workstation, in particular DB2 for NT. We briefly discuss the DB2 family of products, focusing on DB2 Common Server, and also look at some of the hardware requirements for DB2 NT. The chapter is organized as follows:

- Introduction to the DB2 family of database servers
- Introduction to DB2 for Windows NT components and products
- Overall information about Windows NT, especially hardware and software requirements for DB2
- Sample configurations
- Summary of advantages and the integration of DB2 in Windows NT

Throughout this chapter, we will illustrate some of the features that are found only in DB2 for Windows NT. These features are related to the operating system and although there are similarities between platforms that use DB2, the implementation on Windows NT may be slightly different or even unique because of the operating system environment.

1.1 Introduction of the DB2 Family of Database Servers

In a client/server environment, systems share resources with other systems, either asking for or providing resources. A DB2 database server can be found in a client/server environment. The resources to be shared are the databases and the information stored in the databases. Also, the ability to manage and administrate the resources can be shared.

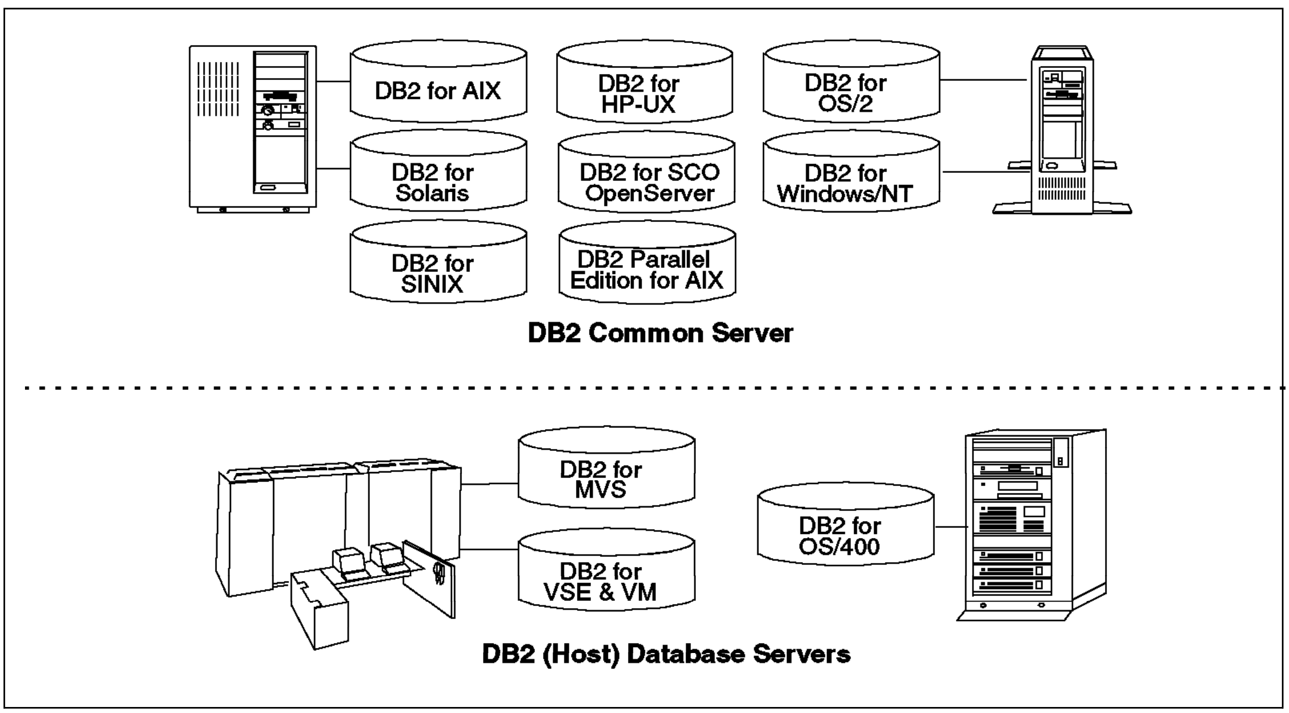


Figure 1. DB2 Family of Database Servers

Figure 1 shows the DB2 family of products. The database servers located above the dotted line are collectively known as DB2 Common Server. DB2 for Windows

NT is part of this group of database servers. The DB2 family of products can be divided into two main groups:

- DB2 for mid range and large systems. This group includes:
  - DB2 for OS/400
  - DB2 for VM
  - DB2 for VSE
  - DB2 for MVS
- DB2 for the workstation environment. This is the DB2 Common Server family of products. This group includes:
  - DB2 for OS/2
  - DB2 for Windows NT
  - DB2 for AIX
  - DB2 for UNIX (various versions)
  - DB2 Parallel Edition for AIX

For simplicity, we will refer to DB2 Common Server as DB2 throughout this document. For more information about the DB2 family, see the *DB2 Planning Guide.*

1.2 DB2 Products and Components for Windows NT

This section discusses the DB2 Common Server products. DB2 products are comprised of a series of components, some of which are automatically installed and some of which can be optionally installed.

The DB2 for NT products described in this chapter are all part of the DB2 Common Server family. Briefly, the DB2 products are:

- DB2 Single-User and Server - provide a relational database engine.
- Distributed Database Connection Services (DDCS) - provides communication facilities to access DB2 databases using the Distributed Relational Database Architecture (DRDA).
- DB2 Software Developer's Kit (SDK) - provides tools for application development.

Before discussing the DB2 for NT products, we will discuss how DB2 components appear in Windows NT.

1.2.1 DB2 Components and Windows NT Features

Each DB2 product includes a number of components. Some of the components can be optionally installed, and some are parts of the base product. They provide a variety of functions such as database administration or application development.

This section looks at the DB2 components as they are found in DB2 for NT products. We will also discuss some of the inherent features or utilities found within the Windows NT environment that have functions similar to the DB2 components. Some of the DB2 components, because they are installed on Windows NT, take advantage of Windows NT features and functions.

The DB2 components are packaged together within the DB2 product family to provide the required functionality. For example, the user-interface tool known as the Command Line Processor (CLP) is available with all DB2 (Common Server) products.

Some of these DB2 components are:

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

The relationship between DB2 products and components is shown in Figure 2.

Product / Component	DB2 Server	DB2 Single-User	DB2 SDK	DDCS Single-User	DDCS Multi-User
Client Application Enabler	✓	✓	✓	✓	✓
Command Line Processor	✓	✓	✓	✓	✓
Database Director	✓	✓	✓	✓	✓
Visual Explain	✓	✓	✓		
Performance Monitor	✓	✓	✓		

Figure 2. Relationship between DB2 Components and Products

For example, the Client Application Enabler, or CAE, is a component that is common to all DB2 for NT products.

A component usually relates directly to an executable application or utility. On the Windows NT platform, one of the ways of invoking certain applications or utilities is the Windows NT desktop. You simply place the mouse on the application icon and click to start the utility. This is the most commonly used method of invoking programs on Windows NT. See Figure 3 on page 4 for a list of the utilities installed on the Windows NT desktop during the installation of DB2 for Windows NT.

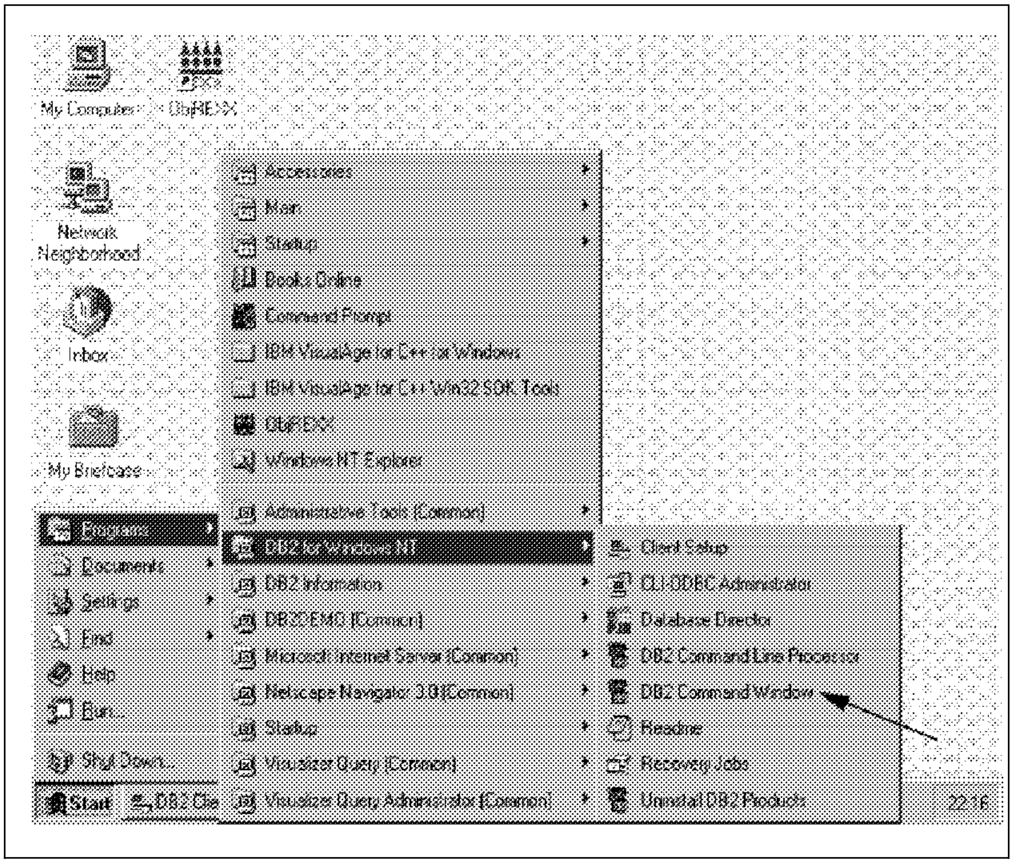


Figure 3. NT Desktop with DB2 Product Installed

The implementation of the DB2 components may be slightly different, depending on the operating system. For example, the DB2 Command Line Processor (CLP) is a multi-purpose tool used to connect to DB2 databases and perform various operations. As shown by the arrow in Figure 3, DB2 for Windows NT contains a ″DB2 Command Window.″ In all other operating systems where DB2 runs, this does not exist. This is required in DB2 for Windows NT due to operating system limitations linking parent/child processes. DB2 commands can only be executed within the CLP or in a DB2 Command Window. They will not execute from a normal Windows NT command prompt.

1.2.1.1 DB2 Client Application Enabler (CAE)

DB2 Client Application Enabler (CAE) is a component common to all DB2 products. It allows access to the database server from local or remote clients. Once an application has been developed, the DB2 CAE component must be installed on each workstation executing the application. Figure 4 on page 5 shows the relationship between the application, CAE and the DB2 for NT database server. If the application and database are installed on the same workstation, the application is known as a *local client*.

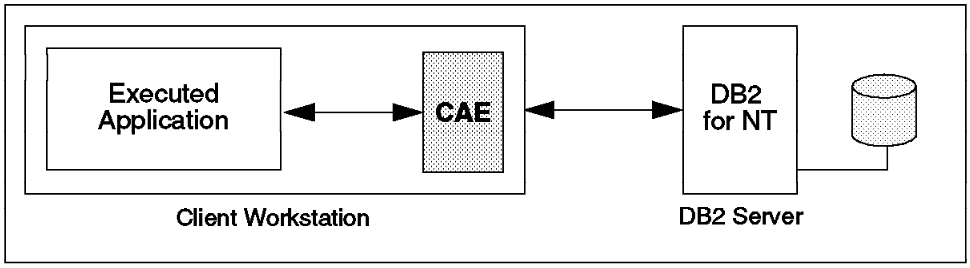


Figure 4. Remote Client Accessing DB2 for NT Database Server Using CAE

The Client Application Enabler comes with an unlimited license when the server product is purchased. The DB2 database server is licensed on a per user basis.

The CAE installation depends on the operating system environment. A complete set of CAEs for all supported client environments is provided with both the DB2 Server and the DDCS Multi-Gateway products. This set is known as the CAE Client Pack.

1.2.1.2 Command Line Processor

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

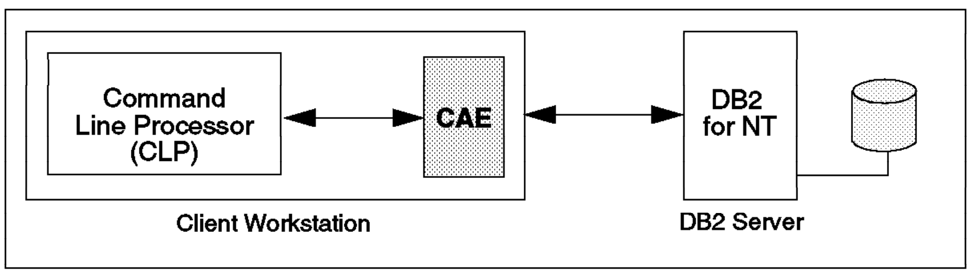


Figure 5. DB2 Command Line Processor

Figure 5 shows the CLP as an application that uses the CAE to communicate with the DB2 database server.

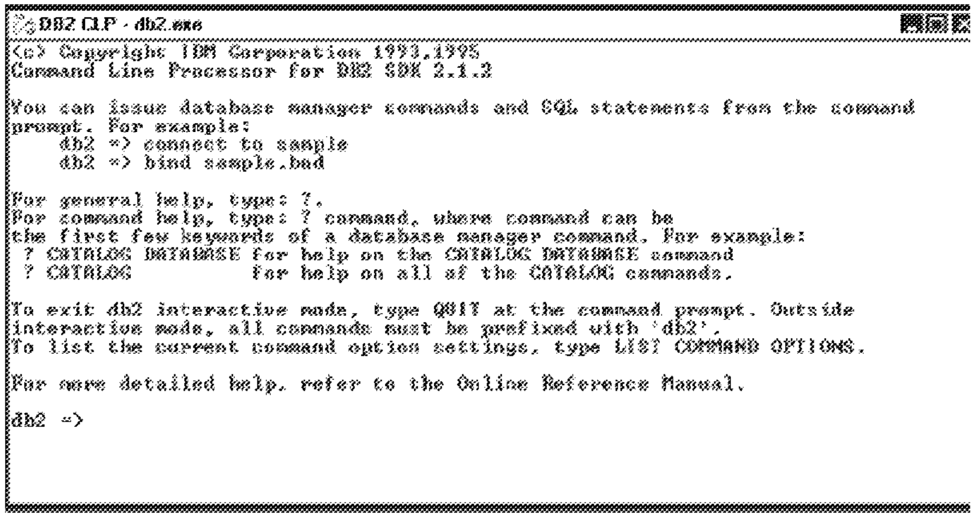


Figure 6. DB2 Command Line Processor

The CLP, as it appears in the Windows NT environment, is shown in Figure 6. From here, you may issue interactive SQL statements and/or DB2 commands. These statements and commands can also be placed in a file and executed in a batch environment, or they may be entered interactively.

An additional environment found only in Windows NT is the DB2 Command Window. You can invoke this through the Windows NT desktop as shown in Figure 3 on page 4. It is similar to an MS-DOS Window, but it also gives you the ability to invoke the Command Line Processor and run DB2 scripts. Samples of such files may be found in the subdirectory SQLLIBSAMPLESCLP.

1.2.1.3 Database Director

The Database Director is a graphical interface used to execute DB2 tasks. In contrast, the Command Line Processor is a text-based interface used to issue DB2 commands and SQL statements.

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

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

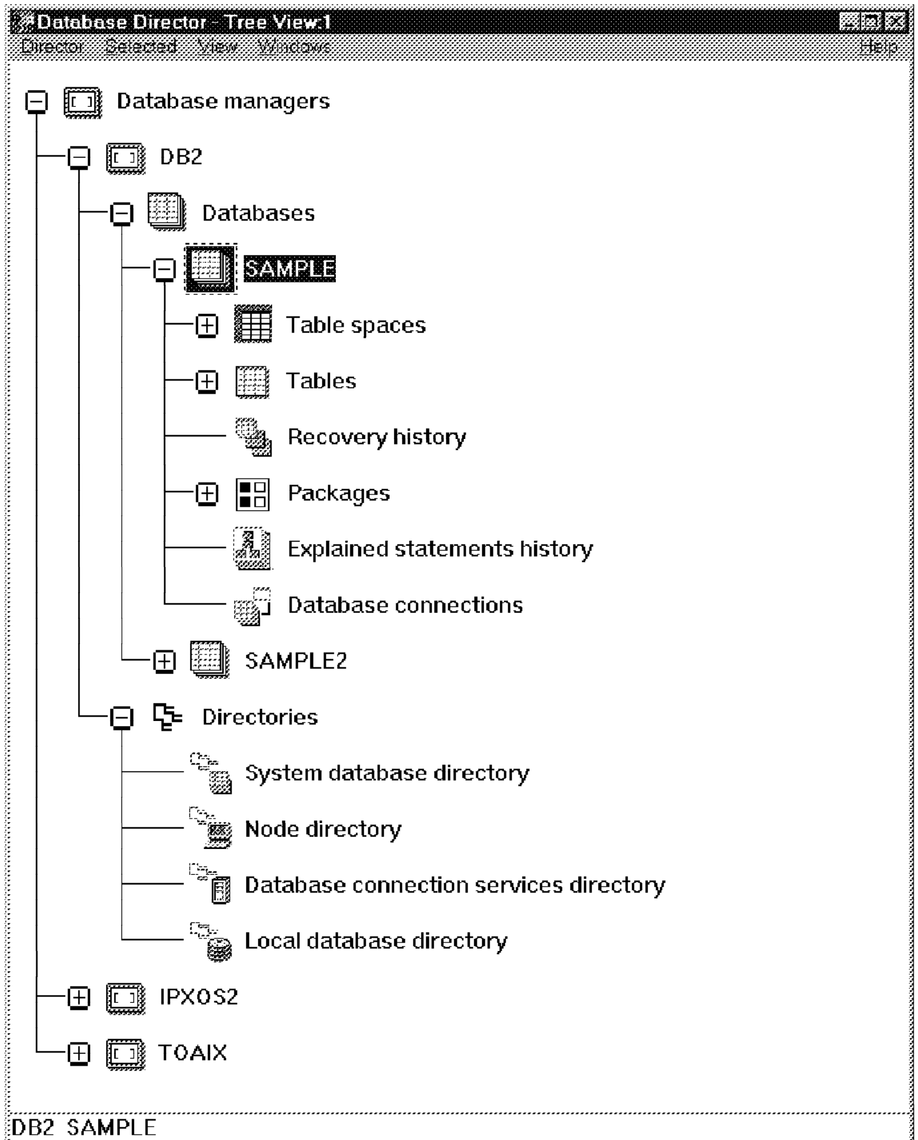


Figure 7. Database Director - Tree View

The Database Director, by default, displays the database objects in a tree view. You can change the view representation to a list view. Using the mouse, you can highlight an object and then open the object to perform various actions. These actions are different for each database object. A pop-up menu will be displayed when an object has been clicked once. From this menu, you can perform an operation on the object. For example, to configure a database, simply click on the database you wish to configure and select **Configure** from the pop-up menu. The Database Director component can be used to manage local and remote database resources.

In Figure 7, the Database Director utility displays three DB2 instances. DB2 is the default instance that is created when the product is installed. The instances IPXOS2 and TOAIX are additional DB2 instances. DB2 databases are created within a DB2 instance. We can see in Figure 7 that there are two databases, SAMPLE and SAMPLE2, created within the instance called DB2.

1.2.1.4 DB2 Performance Monitor

The DB2 Performance Monitor is a graphical utility available with the DB2 Server. There are two basic monitoring facilities:

- Event Monitor
- Snapshot Monitor

The Event Monitor captures database activity defined in the monitor definition. The Event Monitor records are stored on disk and analyzed after the data has been captured.

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

The Windows NT operating system contains a built-in performance monitor. DB2 is enabled to record monitor information here in addition to its own performance monitor.

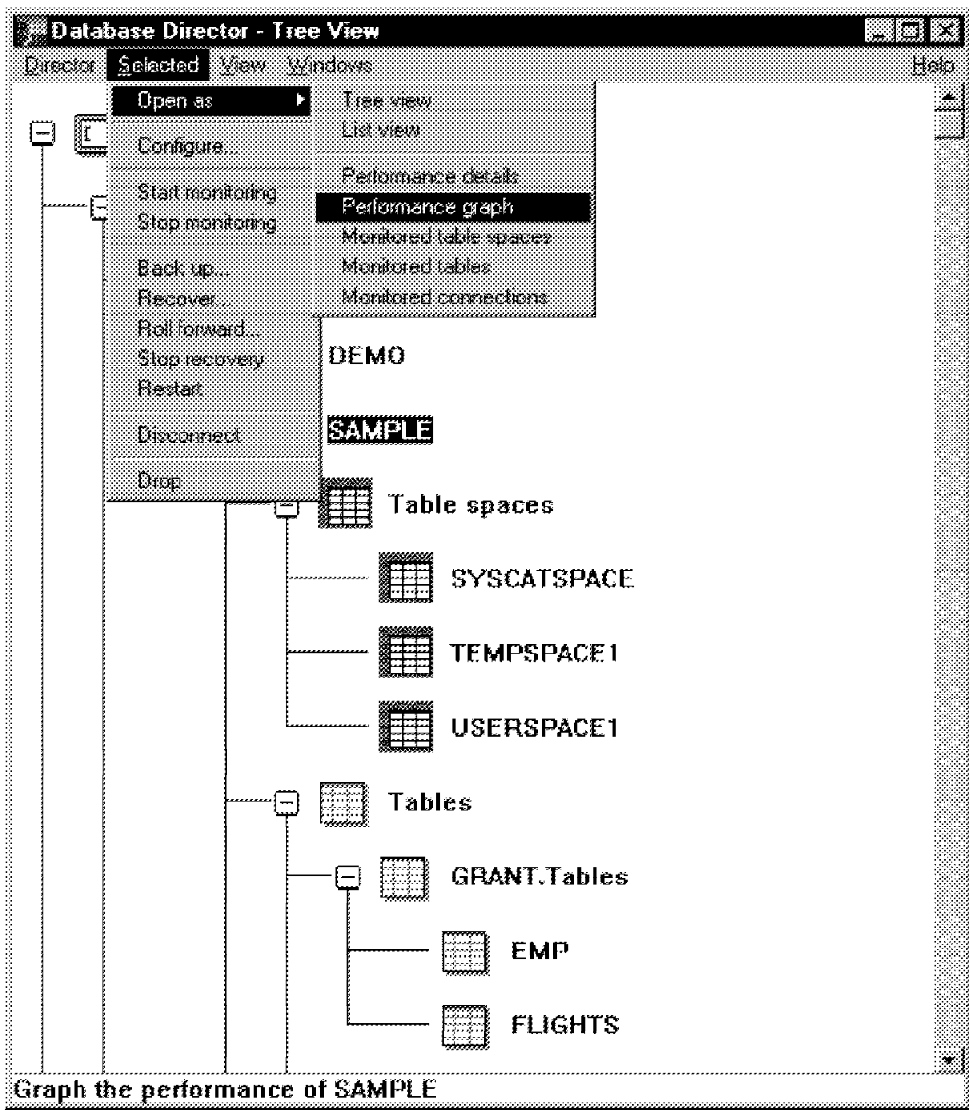


Figure 8. Invoking the Snapshot Monitor from the Database Director

The Snapshot Monitor can be invoked from the Database Director shown in Figure 8. Using the Snapshot Monitor, you can monitor the following objects previously displayed in Figure 7 on page 7:

- Instances
- Databases
- Tablespaces
- Tables

An example of the output from monitoring a database (here, the database is SAMPLE) is shown in Figure 9.

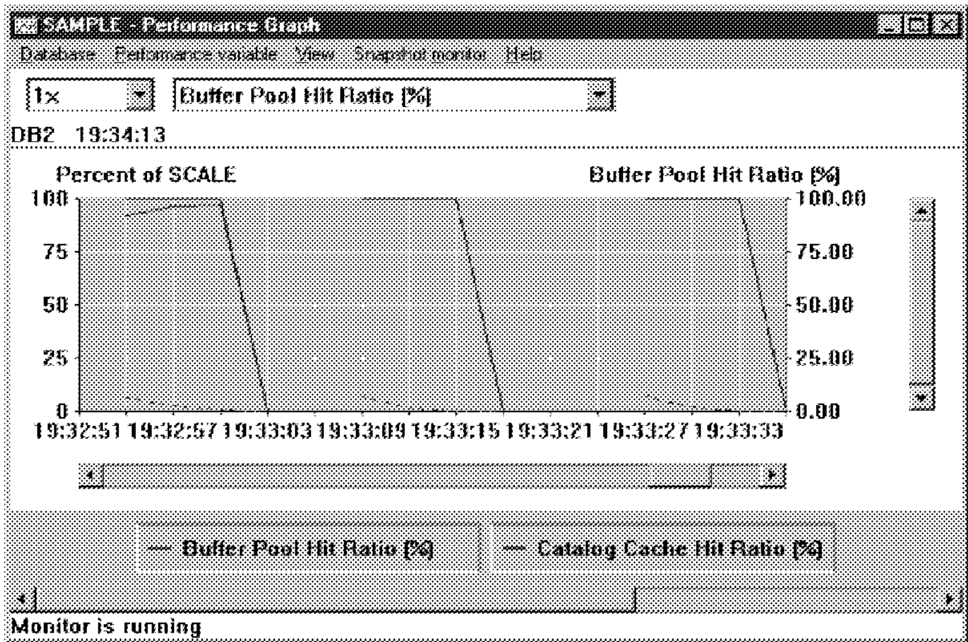


Figure 9. Output from DB2 Performance Graph for the SAMPLE Database

The Snapshot Monitor captures database information for the database SAMPLE at specific intervals. The interval time and the data displayed in the performance graph can be configured. Figure 9 shows a sample output from the Snapshot Monitor displaying the buffer pool hit ratio. A buffer pool hit ratio of 100 percent means that the data for a query was retrieved from memory (buffer pool) and disk I/O was not required. This is a desirable result. The Snapshot Monitor can help to analyze performance problems, tune SQL statements and identify exception conditions based on limits or thresholds.

When you click on **View** in the menu bar of the Performance Graph, you will obtain a sub menu where you can select the item **Include Performance Variables**. Selecting this item will display the window in Figure 10 on page 10. This window shows the performance variables that can be monitored. The left side includes the parameters that have **Collected** Data which is displayed in the graph. The right side includes all the parameters that have been selected for display.

The following output is a list of the parameters that were monitored for the database SAMPLE.

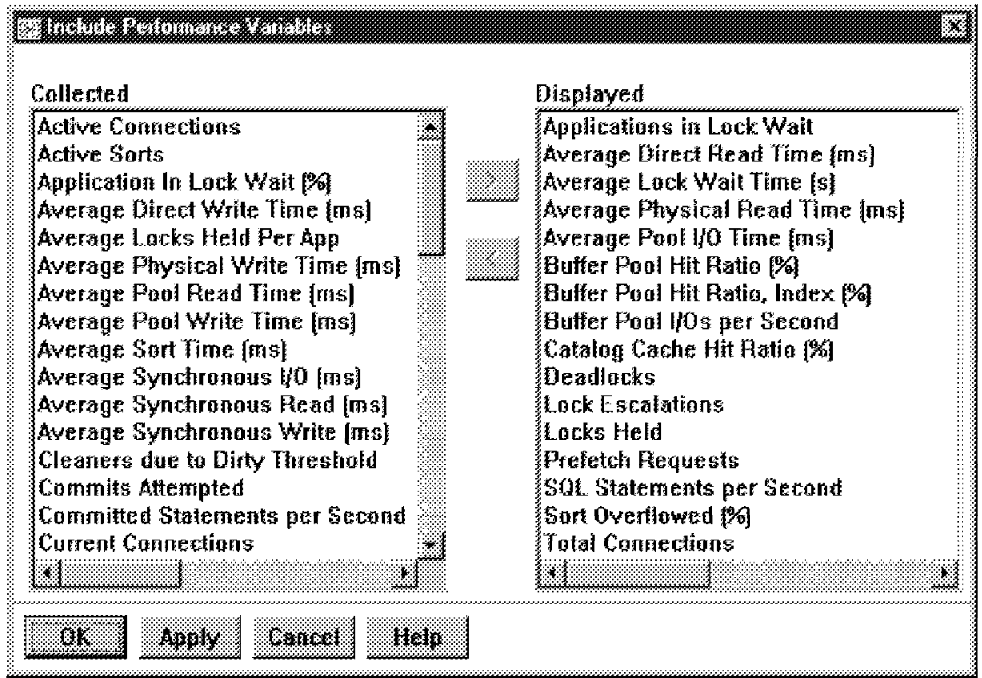


Figure 10. Selecting Variables for Monitoring with DB2 Performance Monitor

1.2.1.5 Windows NT Performance Monitor

Similar monitoring information can be obtained using the Windows NT Performance Monitor. The same information that can be displayed using the DB2 Performance Monitor can also be displayed using the Windows NT Performance Monitor. Figure 11 shows the number of logical reads and writes that occurred in the buffer pool for the SAMPLE database.

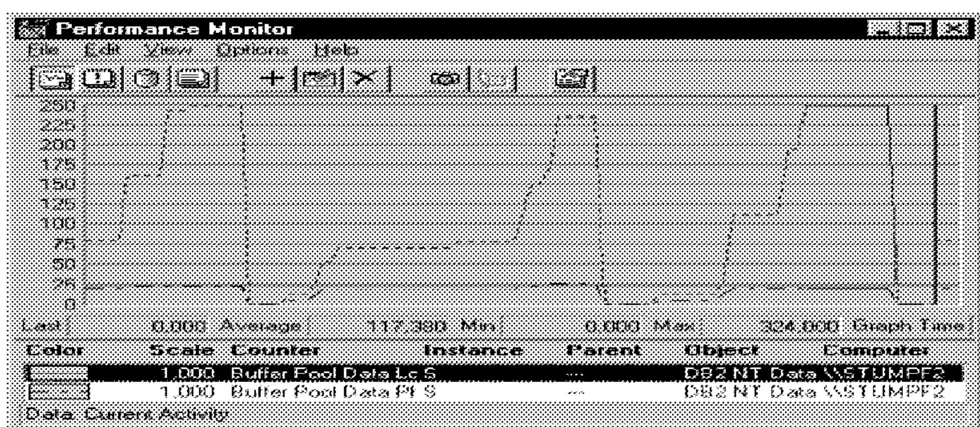


Figure 11. Windows NT Performance Monitor - SAMPLE Database

The DB2 utility DB2PERFI must be executed to integrate the DB2 performance variables in the Windows NT Performance Monitor. If you click on **Edit** in Figure 11, an itemized list will pop up. One of the items is **Add to chart.** If you select this, the Figure 12 on page 11 appears:

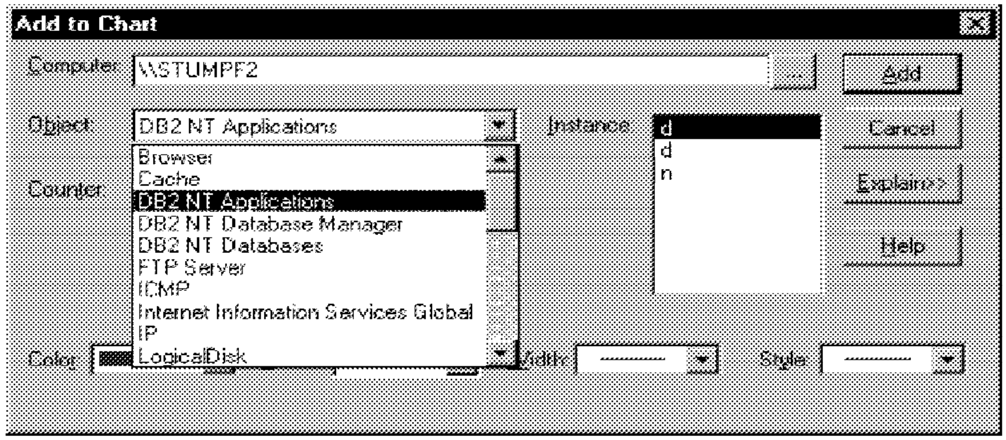


Figure 12. Adding Counters to Chart for DB2 Objects

Using the button beside the object in the window shown in Figure 12, you get an overview of all the objects that can be monitored by the Windows NT Performance Monitor. The DB2 objects start with DB2 NT**.** From this window, we can look at the individual parameters that can be monitored for an object. For example, if we want to monitor a database, the following window appears:

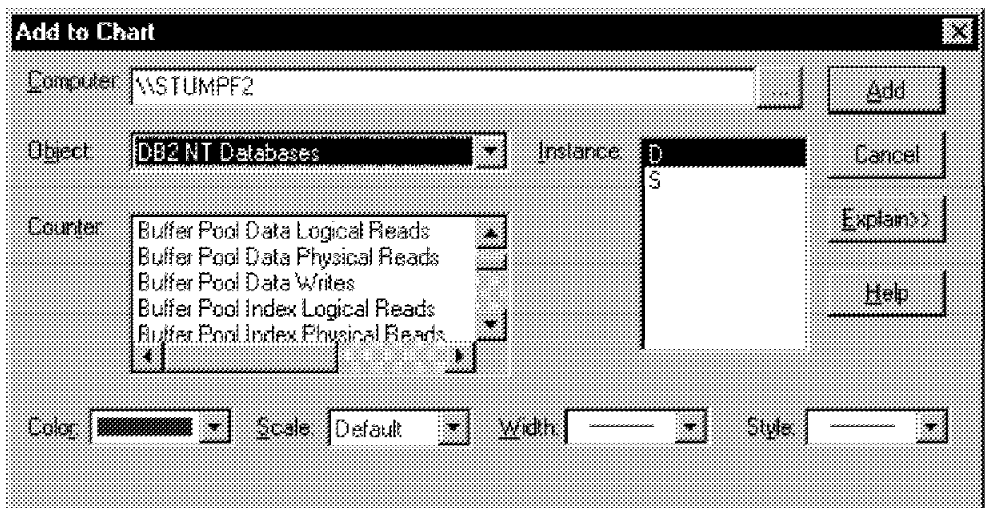


Figure 13. Monitoring Using Windows NT Performance Monitor

When you select an object, all of the performance variables (counters) for that object will be displayed in the Counter window, Figure 14 on page 12.

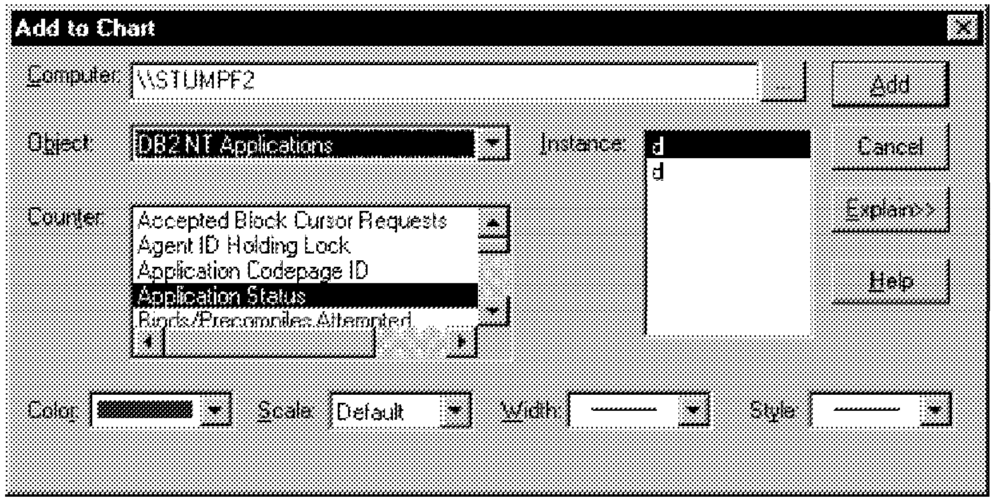


Figure 14. Counters for Object in Windows NT Performance Monitor

1.2.1.6 DB2 Visual Explain

Visual Explain is a DB2 graphical utility that provides a visual representation of the access plans that the DB2 Server used to obtain the data for an SQL statement. It enables you to study queries in more detail, especially those that contain complex sequences of operations. Figure 15 on page 13 shows a sample of the output of the Visual Explain utility.

The optimizer chooses an access plan, and Visual Explain can then display this information as an access plan graph in which tables and indexes, and each operation on them, are represented as nodes. The flow of data is represented by the links between the nodes. From an access plan graph, you can view details for some of the following:

- Objects, such as tables and indexes
- Operators, such as TABLESCAN, SORT and JOIN

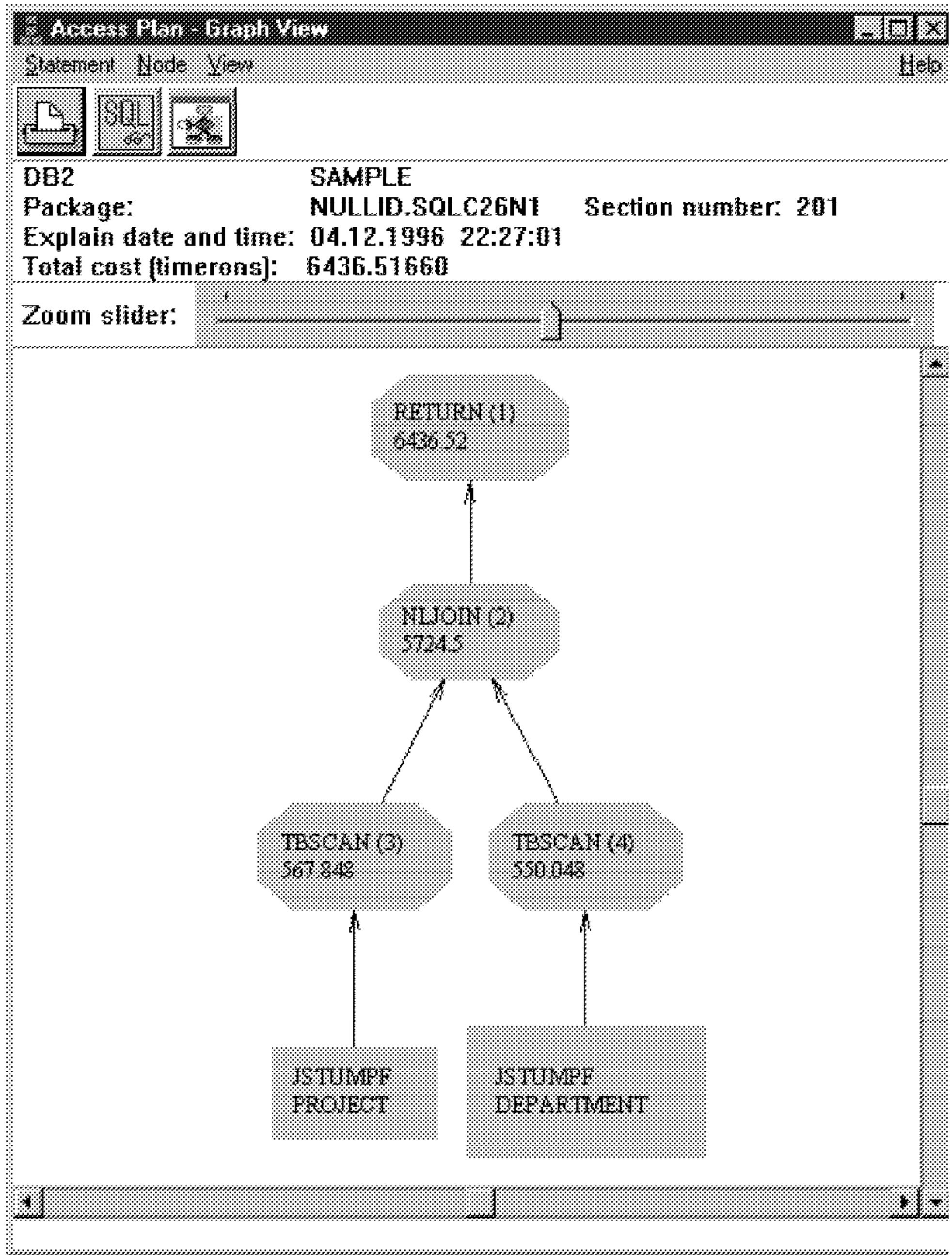


Figure 15. Output from Visual Explain

Figure 15 is an example of the Visual Explain output from an SQL statement that joins data from two tables. The Visual Explain utility uses different symbols for **Nodes** like Objects and Operators. Objects are, for example, the table DEPARTMENT. One of the operators shown in Figure 15 is the NLJOIN (Nested Loop Join) to join the two tables. Underneath each action, you will see numbers which give cumulative information in counting units called timerons. Timerons are calculated with CPU costs, which, in turn, depend on the number of instructions needed and the number of I/O operations. The number in brackets in each node shown in Figure 15 gives you information about the sequence of work.

You can then use the information available from the graph to tune your SQL queries for better performance. The steps you may take are similar to the following:

- View the statistics that were used at the time of optimization. Compare these statistics to the current catalog statistics to help you determine whether rebinding the package might improve performance.
- Determine whether or not an index was used to access a table. If an index was not used, the Visual Explain utility can help you determine which columns might benefit from being indexed.
- View the effects of performing various tuning techniques by comparing the before and after versions of the access plan graph for a query.
- Obtain information about each operation in the access plan, including the total estimated cost and number of rows retrieved (cardinality).

1.2.2 DB2 Products

The DB2 Common Server family consists of a number of DB2 products and components. We have discussed some of the DB2 components as well as some of the features in Windows NT that can be used in conjunction with DB2 products, such as the Windows NT Performance Monitor. There are three main DB2 products, including:

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

This section discusses the DB2 products and describes some main DB2 components found in each product. There are two DB2 products that provide a Relational Database Management System (RDBMS). The database engine is available in a stand-alone version called DB2 Single-User and a multi-user version called DB2 Server. The functionality of the database engine in both products is identical. The only difference is the ability to accept database requests from remote clients.

1.2.2.1 DB2 Single-User

DB2 Single-User provides the same base-level engine functions found in DB2 Server. However, DB2 Single-User also includes the full DB2 application development environment found in the Software Developer's Kit.

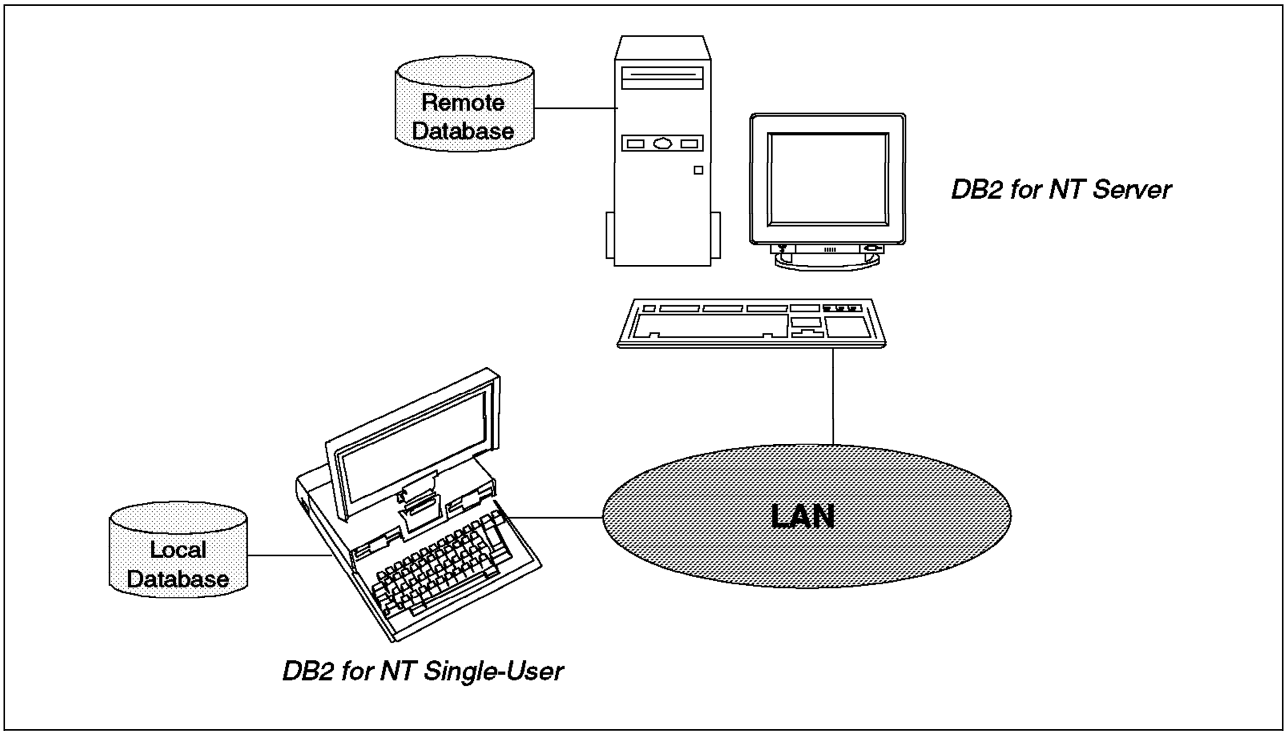


Figure 16. DB2 for NT Single-User

Figure 16 shows an example of using DB2 for NT Single-User. As the product name suggests, DB2 Single-User allows one user to create databases on the workstation where it was installed. DB2 Single-User contains the CAE component. Therefore, once the DB2 Single-User product has been installed, you can use this Windows NT workstation as a remote client to connect to a DB2 Server on a Local Area Network (LAN).

The DB2 Single-User product is appropriate for the following users:

- DB2 developers creating applications that access local databases
- DB2 end users requiring access to local and remote databases

DB2 Single-User contains the Database Director, Visual Explain, Command Line Processor (CLP) and the Software Developer's Kit.

1.2.2.2 DB2 for NT Server

DB2 for Windows NT Server contains all the DB2 for NT Single-User functions, except the application development environment. For this functionality, you need to install the Software Developer's Kit (SDK) on the same Windows NT workstation. In Figure 16, the DB2 Single-User workstation is shown as a mobile user that is occasionally connected to a LAN. This mobile user may access any of the databases on the DB2 Server workstation.

DB2 Server is designed for use in a LAN environment. It provides support for both remote and local clients. A workstation with DB2 Server installed can be connected to a network and participate in a client/server environment as shown in Figure 17 on page 16.

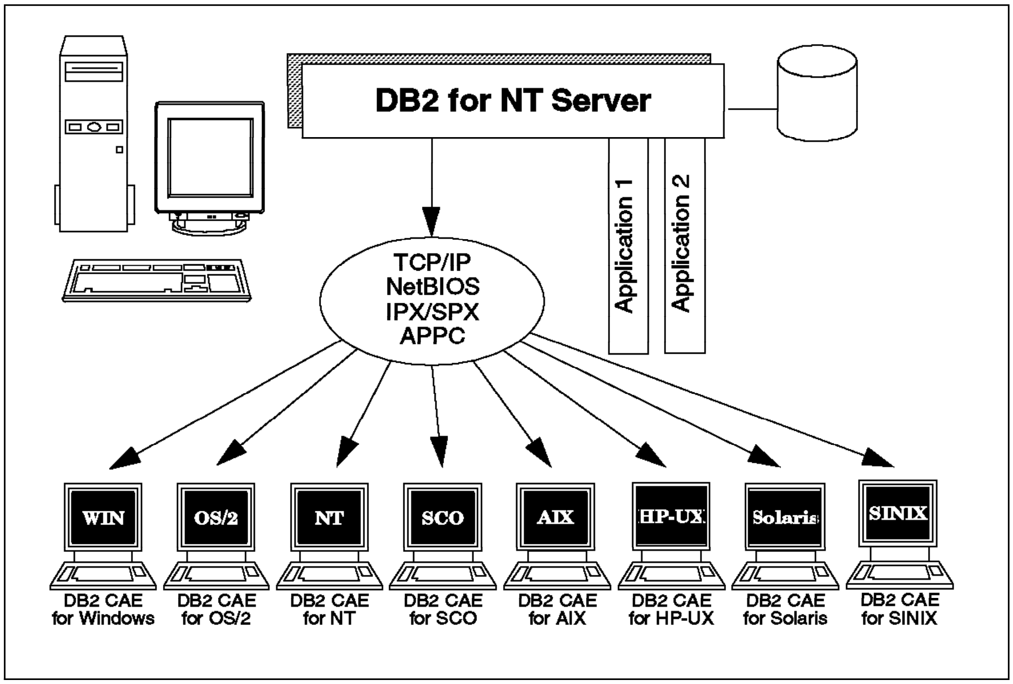


Figure 17. DB2 for NT Server with Remote and Local Clients

In Figure 17, Application 1 and Application 2 are local database applications. Remote clients can also execute Application 1 and Application 2 if the necessary client/server setup has been performed. DB2 remote clients communicate with the DB2 Server using a client/server-supported communication protocol and the CAE component. A DB2 for NT database server supports TCP/IP, NetBIOS, Internetwork Packet eXchange/Sequenced Packet eXchange (IPX/SPX), and Advanced Program-to-Program Communications (APPC).

DB2 for NT Server includes the Database Director, Visual Explain, and Performance Monitor.

1.2.2.3 Distributed Database Connection Services (DDCS)

The Distributed Database Connection Services product allows clients access to data stored on a database server that implements the Distributed Relational Database Architecture (DRDA). The target database server for a DDCS installation is known as a DRDA Application Server. In Figure 18 on page 17, the flow of database requests through DDCS is shown.

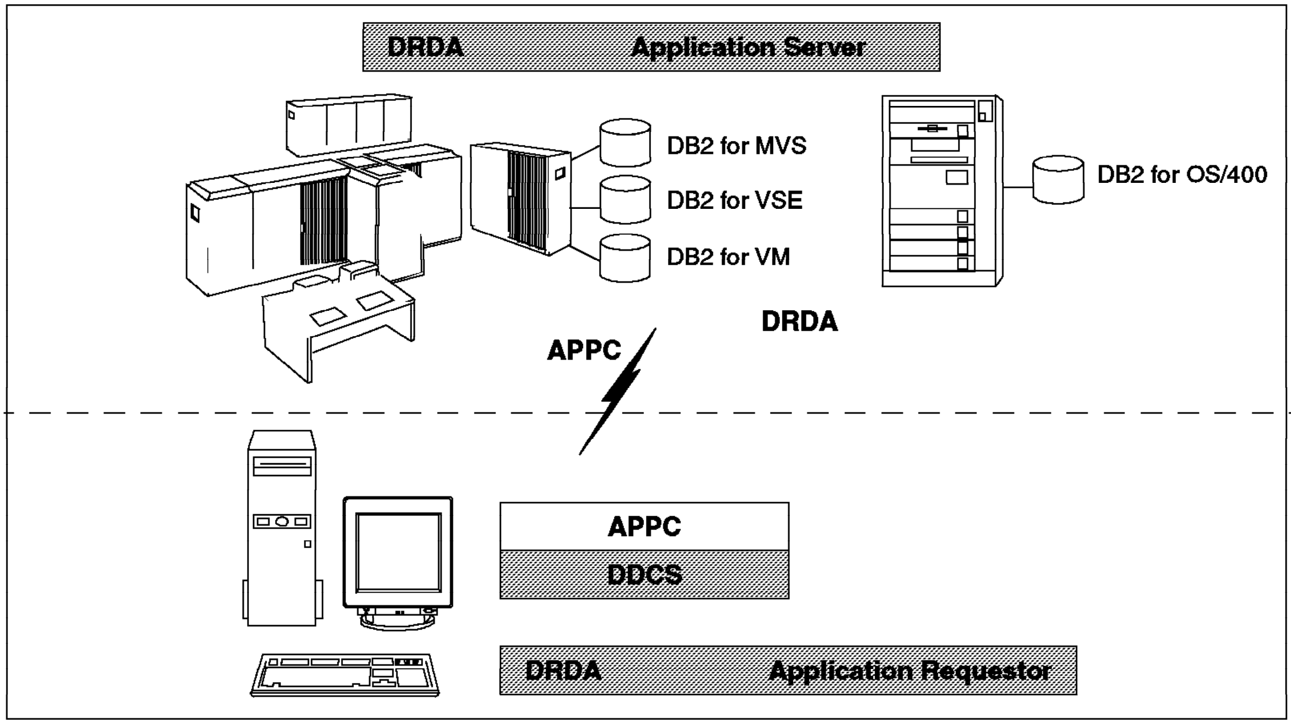


Figure 18. DRDA Application Flow

DDCS currently requires the APPC communication protocol to provide the communications support between Application Servers and Application Requestors.

APPC Support and Windows NT

SNA Server V2.11 provides APPC support for the Windows NT operating system.

The database application must request the data from a DRDA Application Server through a DRDA Application Requestor. The DB2 CAE component is not a DRDA Application Requestor. The DDCS product provides the DRDA Application Requestor functionality.

The DRDA Application Server could be any of the following DB2 servers:

- DB2 for MVS
- DB2 for OS/400
- DB2 for VM
- DB2 for VSE
- DB2 Common Server (DB2 Server, on supported platforms)

In Figure 18, the DB2 CAE component is not shown because the DB2 CAE component can only access DB2 Common Servers directly. The data flow from DB2 CAE and DB2 Server for NT is a private database protocol. This protocol can only be used to communicate with DB2 Common Server.

1.2.2.4 DDCS Single-User

DDCS Single-User is available is only available on the Windows NT and OS/2 platforms. It provides access to host databases from the workstation where it is installed. It includes the Database Director and the Command Line Processor (CLP).

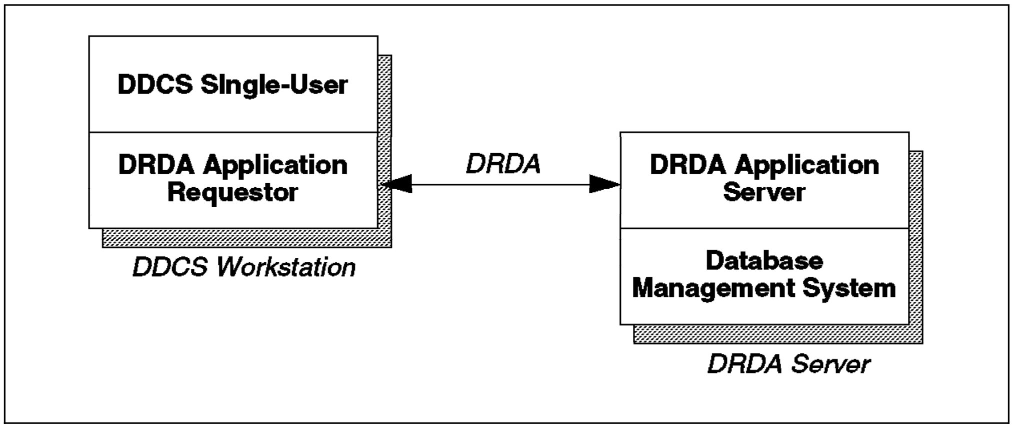


Figure 19. DRDA Flow in DDCS SIngle-User

Figure 19 shows the DRDA flow between the Application Requestor and the Application Server using the DDCS Single-User product.

1.2.2.5 DDCS Multi-User Gateway

The DDCS Multi-User Gateway product provides the ability for multiple clients to access host data. A DDCS gateway routes each database request from the DB2 client to the appropriate DRDA Application Server database. Figure 20 shows the addition of a remote client. The remote client communicates with the DDCS workstation using any of the supported communication protocols.

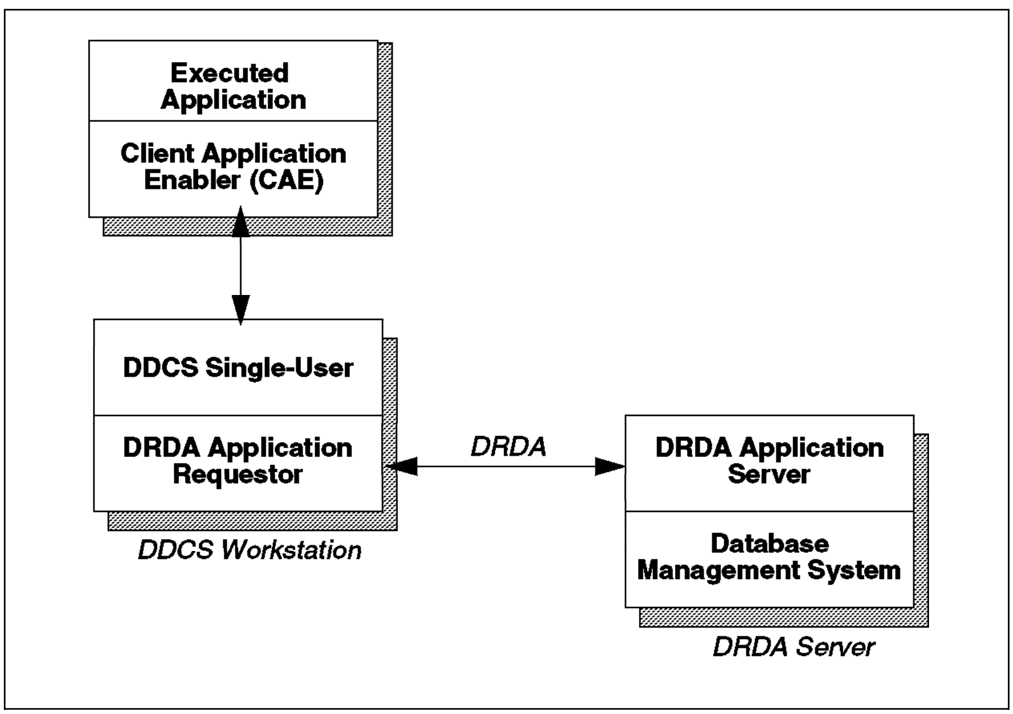


Figure 20. DRDA Flow in DDCS Multi-User

The DDCS Multi-User product is acting as a DRDA Application Requestor, and it is also acting as a DB2 Server (in relation to the remote client connection). The DDCS product includes the Database Director and the Command Line Processor.

1.2.2.6 Software Developer's Kit (SDK)

The Software Developer's Kit (SDK) is a separate product that can be installed on either the DB2 Server or on a DB2 client. It provides all of the necessary data access tools for developing embedded SQL application and callable SQL applications, as shown in Figure 21.

The application development environment provided with the SDK allows application developers to write programs using the following methods:

- Embedded SQL
- Callable SQL interface (compatible with the Microsoft ODBC standard)
- DB2 Application Programming Interfaces (APIs)

The programming environment also includes the necessary programming libraries, header files, code samples, and precompilers for the supported programming languages. Several programming languages, including COBOL, FORTRAN, REXX, C, and C++ are supported by DB2.

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

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

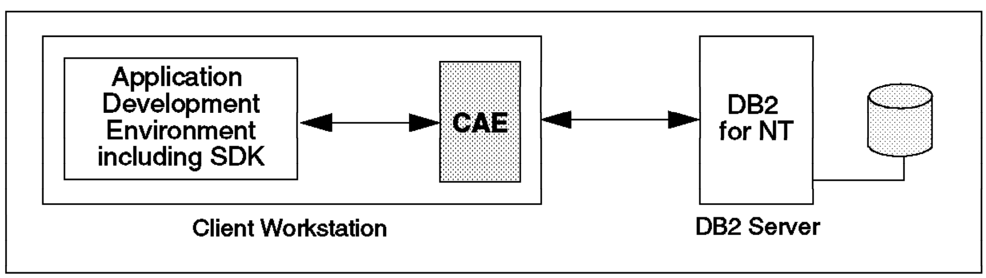


Figure 21. Software Developer's Kit

1.3 Windows NT

This section looks at Windows NT in general, its interaction with DB2 and the supported clients that you may find in a DB2 for NT environment.

1.3.1 Introduction of Windows NT

Microsoft has different operating systems in the market that target different customer needs. Windows NT is positioned in the enterprise area as a system that fits your needs from a small department server to a centralized enterprise system.

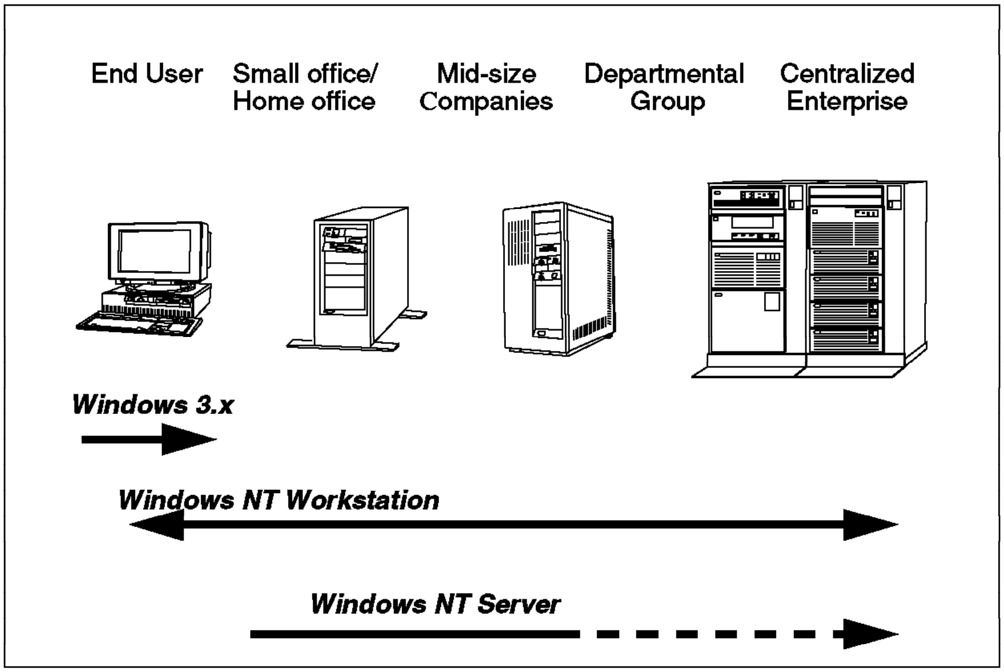


Figure 22. Positioning of Windows NT

Windows NT has two different strategic implementations that differ in some features and in the behavior of the system:

- Windows NT Workstation

  This version is designed as an interactive desktop operating system. It is focused on the interaction of local users sharing their resources, such as the CD-drive, disk and software in a small workgroup. The system resources give priority to the local user and then to remote clients. Each workstation has its own user administration.

- Windows NT Server

  This version is a superset of Windows NT Workstation with several enhancements, like domain-based naming and logon system. This makes it easier for sharing resources in an environment with a large number of PCs.The primary focus of a Windows NT Server is to serve remote clients as a high-performance operating system. The Windows NT Server can be installed in three different configurations:

  – Server

    This system serves applications and resources to users in a network; it might be used as a print server, for instance.

  – Primary Domain Controller (PDC)

    This system is focused on user administration and authorization checking.

  – Backup Domain Controller (BDC)

    This systems focus is to act as backup for the PDC.

For more detail on definitions and functions of the different server versions, please see Chapter 2, "DB2 and Windows NT Security" on page 27.

1.3.2 DB2 Server Requirements for Windows NT

This section provides some of the requirements that you would need for the DB2 Server product on the Windows NT operating system. Discussed are types of processor, memory, disk, communication support, and compilers. For a detailed discussion, please see the *DB2 for NT V2.1 Planning Guide*.

1.3.2.1 Processor

Windows NT as an operating system can be found on different types of hardware, such as Intel or RISC architectures. All of the DB2 for NT Common Server products are supported on the Intel platform completely. However, only the Software Developer's Kit (SDK) for NT is supported on the PowerPC (RISC architecture).

1.3.2.2 Memory

Memory requirements will depend on your specific environment. The exact amount of memory on your DB2 for NT Server will depend on the following:

- Number of instances
- Number of concurrent active users
- Number of active databases

1.3.2.3 Disk Space

Windows NT supports large amounts of disk space, thus making it ideal for large database systems.

1.3.2.4 Communication Support

Windows NT supports the following communication protocols:

- NetBIOS
- TCP/IP
- IPX/SPX
- APPC

All the communication protocols with the exception of APPC are provided in the Windows NT base operating system. APPC support is provided with the SNA Server V2.11 product.

1.3.2.5 Compilers

DB2 NT on Intel platforms supports the following compilers:

- Microsoft Visual C++ Version 2.11 or higher
- Microfocus COBOL Version 3.3.33 or higher

DB2 for NT on the PowerPC supports the following compiler:

- Microsoft Visual C++ Version 4.0

1.4 DB2 for NT Installation Scenarios

This section outlines a few sample installations using the various DB2 for NT products. Also, some of the terminology specific to the Windows NT environment is used.

1.4.1 Example One

A small group of programmers is developing applications to access data from a DB2 database server. They are using Windows NT Workstations running an application development tool, and they want to write database applications. All developers will access the same database server.

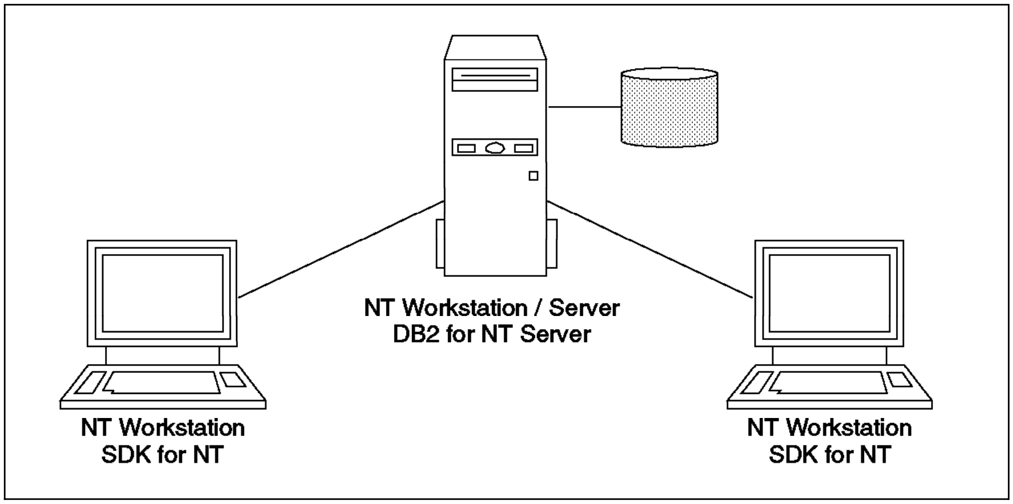


Figure 23. Example One - Small Workgroup

One possible configuration is shown in Figure 23. Each application developer will have SDK for NT installed on their Windows NT Workstation. There will be one DB2 for NT database server on a central Windows NT workstation/server. To improve performance, Windows NT Server can be used on the DB2 database server instead of Windows NT Workstation.

1.4.2 Example Two

This scenario involves a small-to medium-sized enterprise where end-users are using a popular application, such as Lotus Approach, to produce weekly reports. There is a DB2 for Windows NT database server. This group uses the domain concept with a trusted and a trusting domain. There are users in the domain of the database server, but most of them are located in one domain, the trusted domain. This reduces the amount of user administration, since Windows NT cannot share the database over the two domains. The concept of a trusted domain and a trusting domain are explained in Chapter 2, “DB2 and Windows NT Security” on page 27.

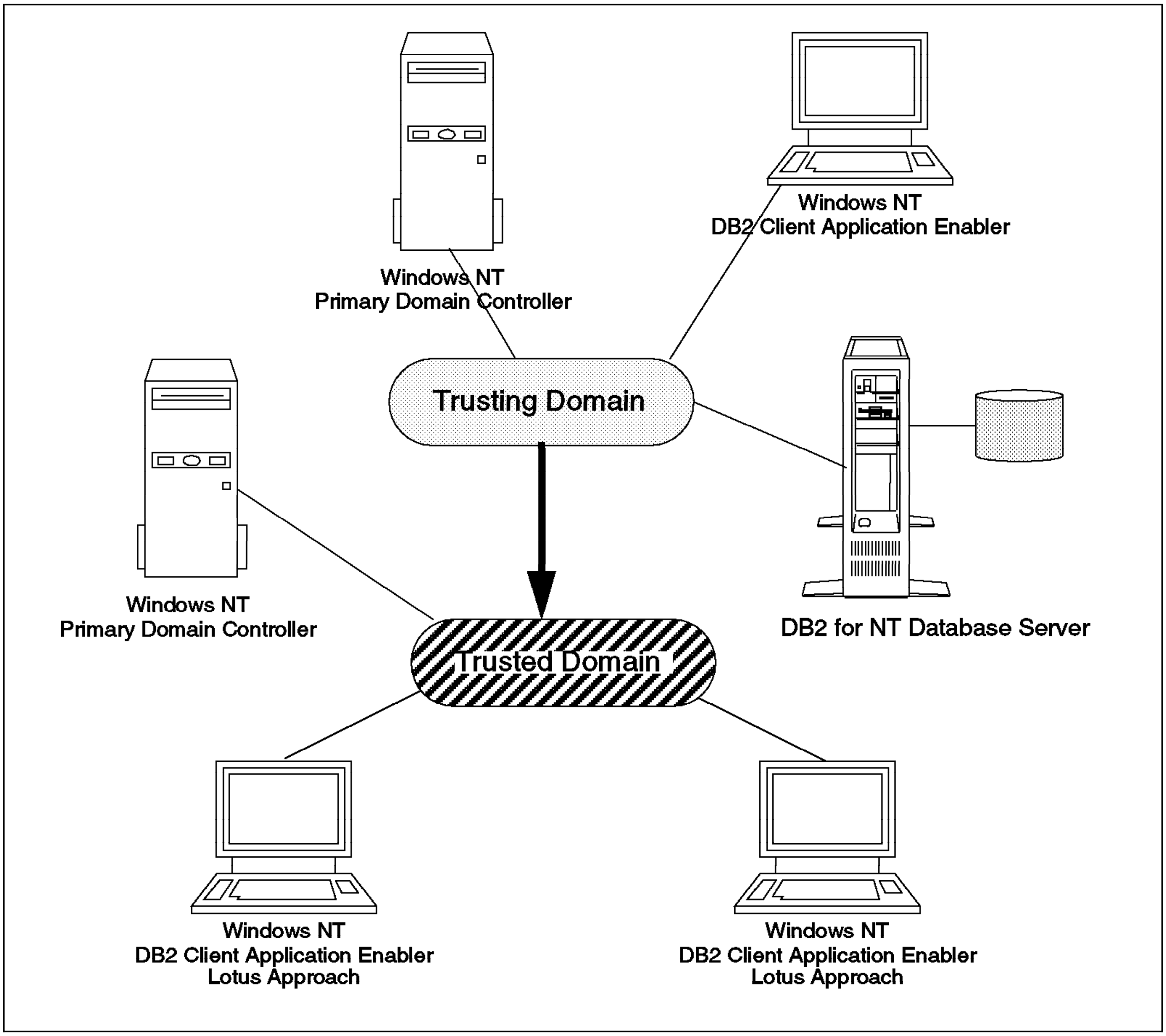


Figure 24. Domain Concept - Trusting and Trusted Domain

The requirements for the scenario just described are the following:

- There will be only one database server; so the database will be shared in the environment.
- There will be minimum effort for user administration.
- Different departments also need resources that will not be shared with other users.
- Users will execute applications, either ODBC-enabled or otherwise, which both access the databases on the database server.
- Databases are only on Windows NT workstations.

Figure 24 shows a possible configuration of the environment. End-users executing the Lotus applications need only the DB2 CAE installed on their Windows NT workstation.

The DB2 for NT Server product can either be installed on Windows NT Workstation or Windows NT Server. The installation of DB2 for NT on a Primary Domain Controller (PDC) is not recommended, in order to avoid resource conflicts. However, installing DB2 for NT on a Backup Domain Controller (BDC) has proven to reduce authentication time when the database is cataloged as an Authentication Server.

Figure 24 on page 23 shows this scenario based on the domain concept of Windows NT. The main user administration will be done in the trusted domain. The arrow in the figure indicates the direction of the trust. If a user logs on in a trusting domain and is not in the local security database, the Primary Domain Controller of the trusting domain will ask the Primary Domain Controller of the trusted domain for authentication of the user. DB2 will also ask the Primary Domain Controller of the trusted domain for group resolution.

1.4.3 Example Three

In this scenario, Windows NT Workstation will be used on end-user and development workstations. There will also be a need to access data from a host database. Some of the data used by the host applications is distributed over the LAN to be used on DB2 for Windows NT database servers.

The end users will be executing ODBC-enabled applications such as Lotus Approach and Microsoft Access, to get data from a DB2 for MVS database as well as data from a DB2 for NT database server. Application developers need to be able to write and test database access modules for the LAN and host database systems. One of the immediate goals is to start developing these host and LAN database applications with a minimum number of workstations. An additional requirement is that host applications may need to access data from a DB2 for NT database server.

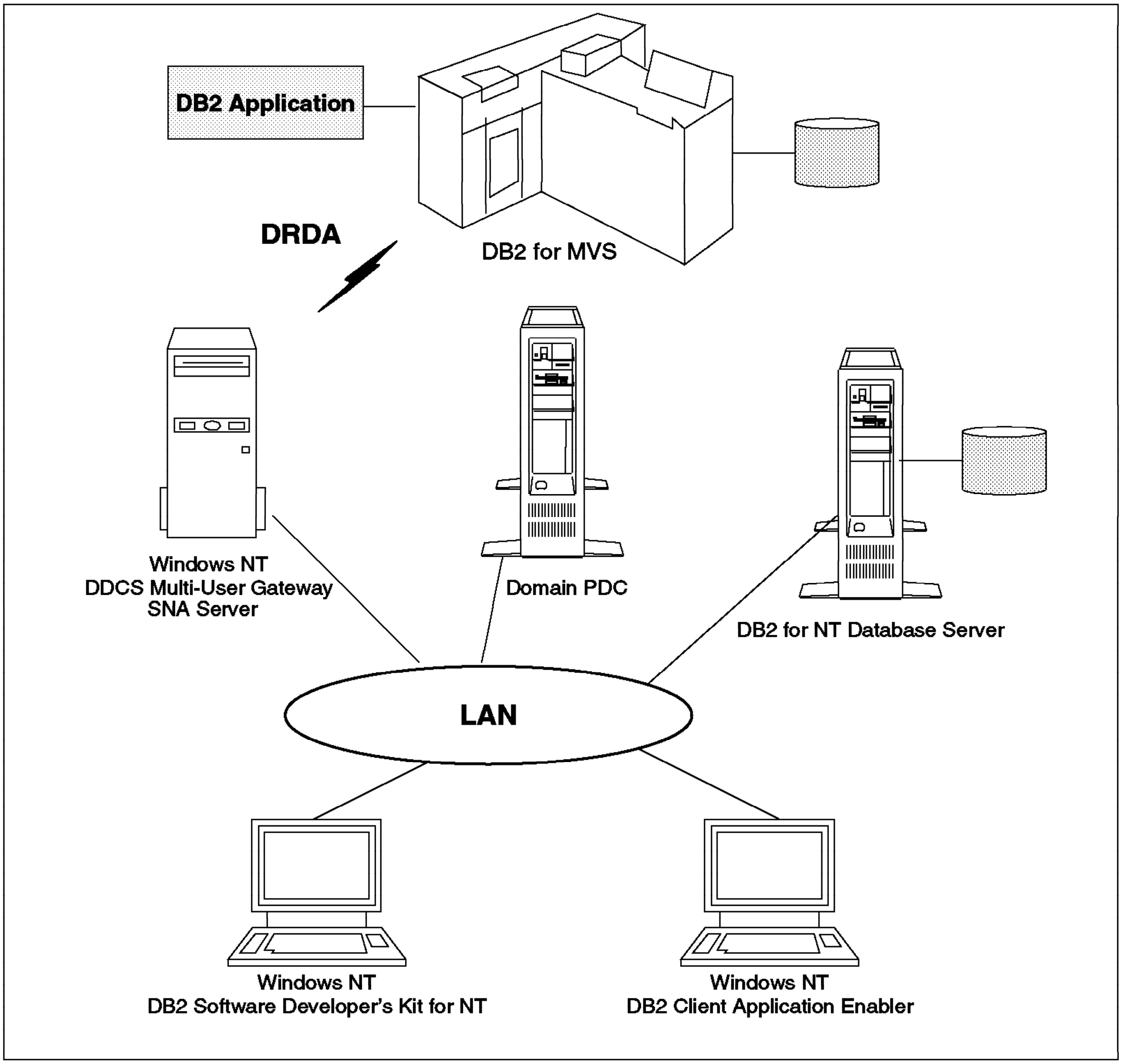


Figure 25. DB2 in an Enterprise Environment

Figure 25 is a possible configuration. Here, we have chosen to place the DDCS product on one Windows NT workstation, while the DB2 for NT product is on another Windows NT workstation. Depending on the system resources, namely memory, disk and CPU on the database server, and the number of clients accessing either the gateway or the database server, this could be achieved by installing both the DDCS and DB2 for NT products on the same workstation.

There are two possible kinds of clients in this enterprise:

- An end user using an ODBC-enabled application, such as Lotus Approach or Microsoft Access, needs the DB2 CAE and required communication protocol.
- An application developer needs DB2 SDK and communication protocol for the operating system platform.

Notice that both types of remote clients need only the LAN communication protocol on their workstation. The DDCS Multi-User gateway will have both the LAN communication protocol and APPC installed. SNA Server V2.11 is the product that provides APPC support on Windows NT. The LAN protocol can be either TCP/IP, NetBIOS, IPX/SPX, or APPC.

1.5 Summary

This chapter provided an introduction to the DB2 Common Server products, especially DB2 for NT, as well as an overview of Windows NT. We looked at DB2 components and how they are integrated on the Windows NT platform. There are many features of Windows NT that make it a desirable platform on which to run DB2. For example:

- DB2 can take advantage of the multithreading processor in Windows NT to allow the execution of multiple jobs in parallel.
- DB2 supports two methods of storage for database objects, the native NT file system and raw I/O.
- DB2 is integrated in the Windows NT service utility so that it can be started automatically when the system is started.
- You have the choice of either using the NT Performance Monitor or the DB2 Performance Monitor to control database parameters.
- DB2 is automatically added in the Windows NT registry.
- You are able to install DB2 on different workstations without an additional local CD-ROM drive from a code server running a script file.
- DB2 uses Windows NT for authorization and group resolution. This allows you to have a central administration control point in Windows NT.
- All system messages generated by DB2 will be logged in the Windows NT event viewer. This assists you in analyzing problems by isolating the source.

Chapter 2. DB2 and Windows NT Security

This chapter discusses one of the main features of Windows NT that separates it from other operating environments and how DB2 utilizes that feature. That feature is security. One of the advanced features of DB2 for Windows NT is the way DB2 takes advantage of the inherent security features of the Windows NT operating system. A greater level of security and control over the DB2 environment is available through Windows NT. This security is achieved through privileges that can be granted to individual users or groups of users and by setting rights and permissions at the file system level.

Users cannot access the Windows NT desktop until they have successfully logged on, either locally or to a domain. This makes NT secure when compared to other windows operating systems, such as Windows 95 or Windows 3.x.

To adequately explain DB2 security and authentication on the Windows NT platform, it is first necessary to understand how security within Windows NT operates. This chapter is therefore structured to introduce some Windows NT concepts. This is followed by a discussion on how security works in terms of user authentication and access to domain objects through user rights and permissions.

Much of the literature on Windows NT and the concepts relating to it is not always clear on the components being discussed. Table 1 on page 28 is a list of definitions that will be followed throughout this document, especially in this chapter.

This chapter is outlined as follows:

- Workgroups in Windows NT
- Domains, including a Primary Domain Controller (PDC) and Backup Domain Controller (BDC)
- Group and User Authentication in Windows NT
- Trust Relationships between Domains
- DB2 for NT Authentication and Security
- The DB2 for NT User Environment

Table 1. List of Definitions

Terms	Definition (for this document)
Windows NT	Either NT Workstation or NT Server. Reference will be made to the common Windows NT features of the two products.
NT Workstation	The NT Workstation product. Cannot be a domain controller.
NT Server	The NT Server product. It is a superset of the NT Workstation product. A machine running NT Server may be a print/file server or a domain controller.
Domain Controller	Either a Primary Domain Controller or Backup Domain Controller.
Primary Domain Controller	By definition, the first domain controller in a domain (for instance, creates the domain). There is only one per domain. Also called a PDC.
Backup Domain Controller	Subsequent NT Servers that join a domain as domain controllers. Also called a BDC.
Server	NT Server that is part of a domain as a file or print server, but is not a domain controller.
Workstation	A machine running NT Workstation or NT Server in a domain that is not a domain controller or a file / print server.
Right	A right is the ability of a user or group of users to perform a Windows NT operation. Examples of rights are logging on to a server and performing backups. Rights are different between users and administrators.
Permission	Authority granted to a user or group of users to perform operations on files, directories, printers, and other resources. Examples of permissions are read, change, full control, and no access. Permissions are applied on a user-by-user basis.

2.1 Workgroups in Windows NT

This section concentrates on Windows NT concepts and discusses security within the Microsoft model. In the section titled, 1.3, "Windows NT" on page 19, some terms were introduced. The concept of the client/server environment was discussed. Also introduced were the NT Workstation and NT Server products as well as the DB2 products that are available for the Windows NT platform.

Two concepts are introduced here: workgroup and domain. Both workgroups and domains are logical organizations of computers that can and do share resources, such as file and printing services. They do have some significant differences, however, the primary one being security.

2.1.1 The Concept of a Workgroup

A workgroup can be described as a collection of Windows NT or Windows workstations. Machines in the workgroup can share their objects with other members (clients) of the workgroup. These objects might include shared directories or printers. Figure 26 on page 29 shows items that might be shared in a workgroup.

Although these resources are shared, user logons are specific to each computer in the workgroup. Therefore, if a user requires access to all file servers in a workgroup, an account must be created for that user on each machine.

A workgroup is identified by a name that can be up to 15 characters long, and users are able to join a workgroup at will. A workgroup can be joined or created when an NT Workstation or NT Server is installed, or at any later stage. An Administrator user ID and password are not required to join a workgroup; only the name of the workgroup has to be known.

A workgroup can contain computers running any number of operating systems, including DOS, Windows 3.x, Windows for Workgroups, Windows 95, NT Workstation and NT Server. An advantage of workgroups is that they are a good way of controlling the size of browse lists on networks, but security can be a problem.

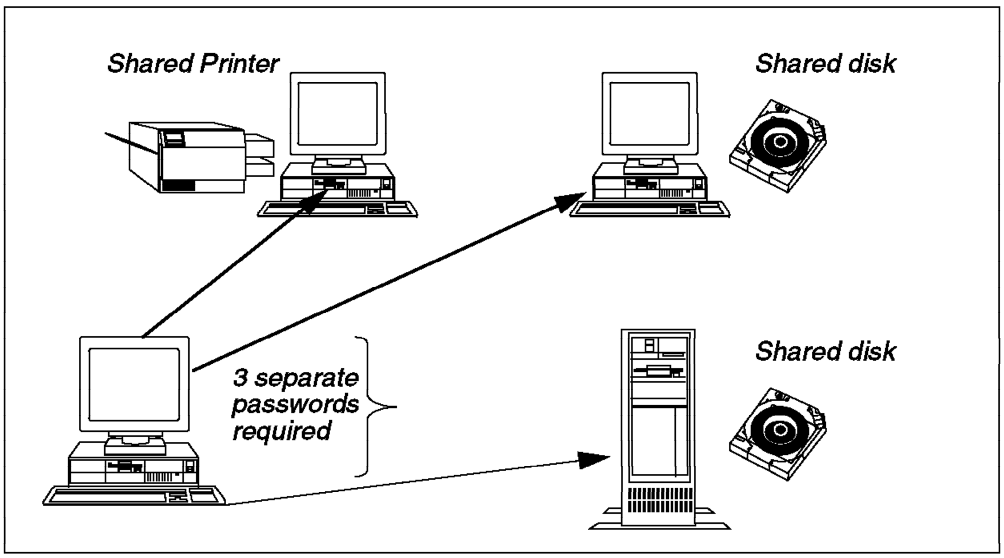


Figure 26. Workgroup Security

In Windows NT, it is possible to share resouces on the computer. The users connected over a network can access the shared resource or share. A share will have a share name associated with it. A share could be a shared directory, a printer queue or a named pipe, for example. Figure 26 is an illustration of the items that can be shared. Once a share has been declared, anyone can connect to it. Shares on a network can be made more secure by either hiding the share or by putting a password on it. Hiding a share means that it will not appear on browse lists on other machines in the workgroup and has to be known by anyone who wants to use it.

Figure 27 on page 30 shows the creation of a share under Windows NT. To obtain the screen shown in Figure 27 on page 30, right-click the mouse on the object (the DB2LOG directory in this example) that you want to share in the NT Explorer. Click on **Shared As,** and then either replace the share name with one you want and/or add a dollar sign ($) to make it hidden. You can also set the number of users who can concurrently connect to this share.

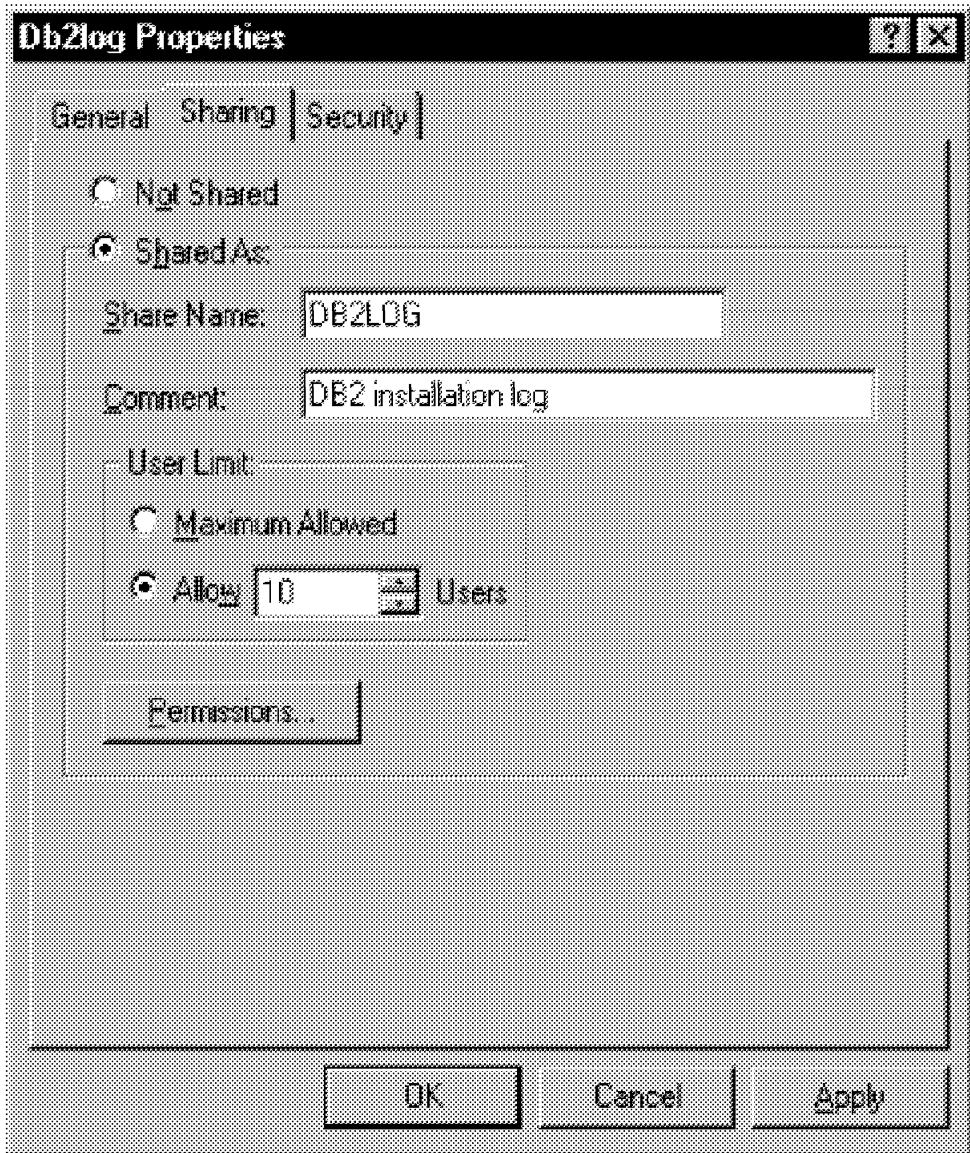


Figure 27. Creating a Windows NT Share

Setting a password is a way to control security. However, a share can only have one password for the network.

2.1.1.1 Windows NT in a Workgroup

Windows NT machines can enhance workgroup security at the user level. A particular user can be granted one of the following permissions to a share:

- No access
- Read
- Change
- Full Control

Figure 28 on page 31 is an example of applying security on a share. The default when a share is created is to grant everyone full control. To apply restrictions, remove this group and add users or groups with the desired type of access.

Priority Levels

> The No Access type of access is the highest priority, followed by Full Control, Change and Read. If a user is granted both Full Control and No Access permissions because of different groups he/she belongs to (groups are discussed later), he/she will be denied access to the object because No Access overrides all other permission levels.

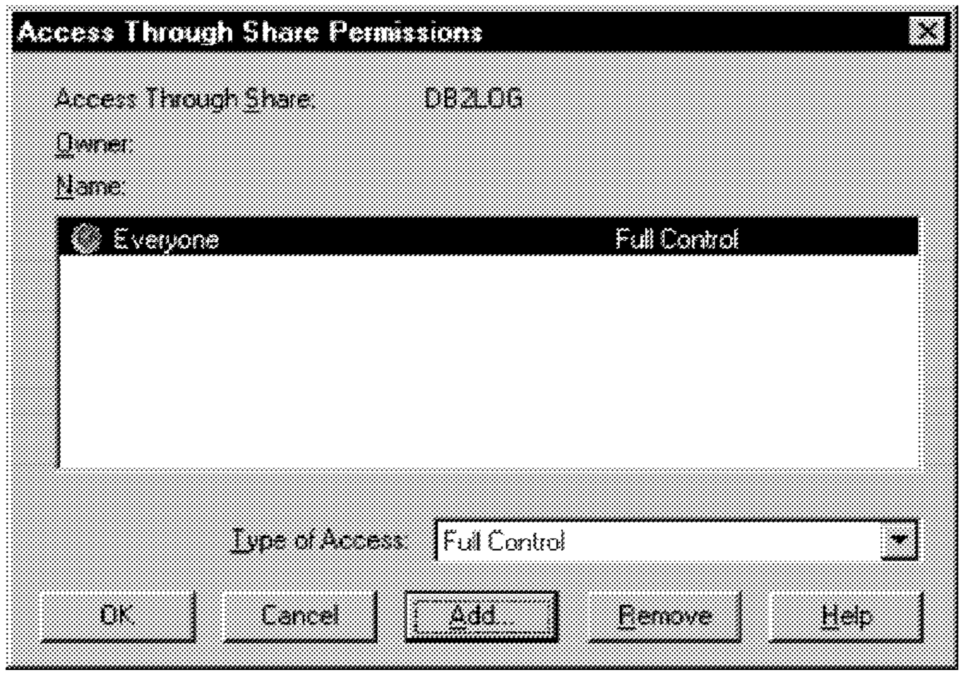


Figure 28. Setting Permissions on a Share

Windows NT comes with a number of default shares, namely:

- Each hard disk partition and CD-ROM drive have a share at their root directory. Those shares are C$, D$, E$, and so on.
- The directory that contains the Windows NT programs (Winnt). That share is ADMIN$.
- In the case of domain controllers, the directory that contains the logon scripts (WinntSystem32ReplImportScripts) is a default share. This share is NETLOGON$.

These shares are hidden and are called *administrative shares*, because only administrators can access them. These shares are automatically created when Windows NT is booted. However, in Windows NT 4.0, these default shares can be unshared by updating the registry.

If the Windows NT workstation is formatted with an NTFS volume, then permissions can also be set on files and directories. An example of this is shown in Figure 29 on page 32, where administrators are updated to have Full Control on files and directories.

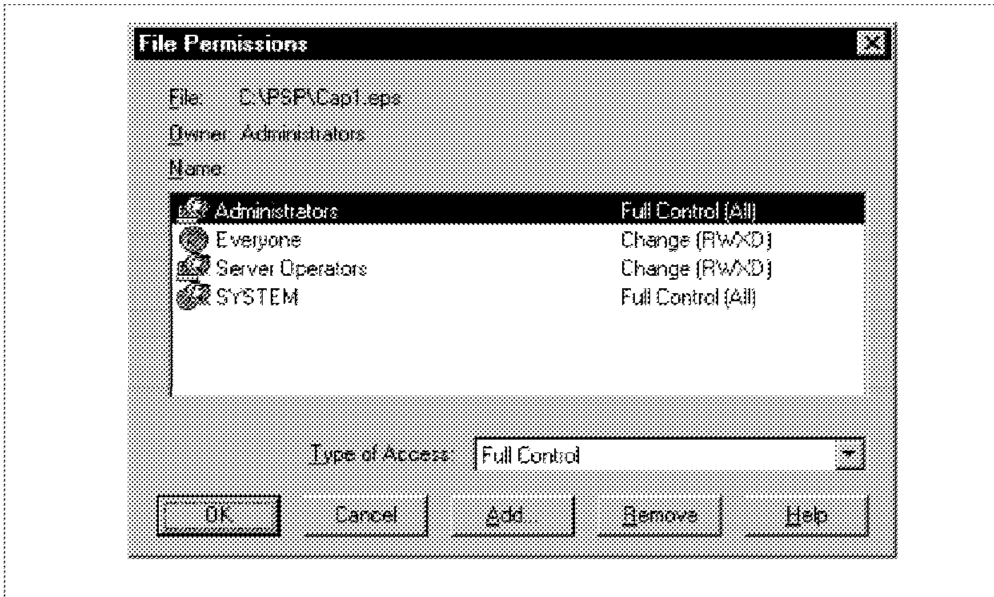


Figure 29. Setting File Permissions

2.2 Domains in Windows NT

A Windows NT domain (hereafter referred to as a domain) is a logical organization of computers. A domain is similar to a workgroup in that they are both a logical organization of computers. The key difference between a workgroup and a domain is that the computers that make up the domain share a common centralized user logon database. This is an important concept because this is one of the differences between DB2 for Windows NT and the rest of the DB2 Common Server family.

A domain can contain Windows NT Servers and LAN Manager 2.x Servers. However, a LAN Manager Server cannot authenticate a Windows NT client.

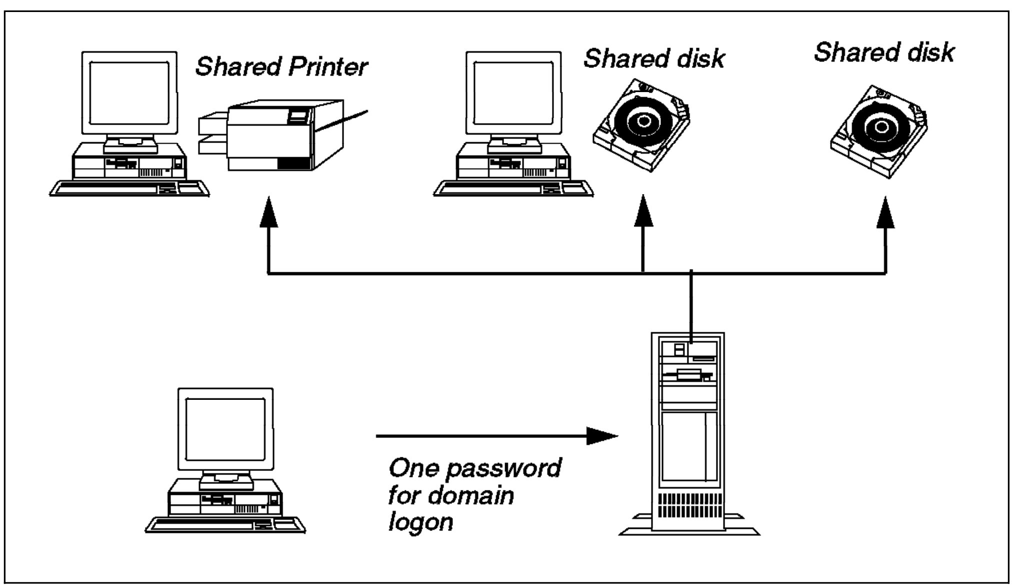


Figure 30. Domain Security

A domain is established when an NT Server is created as a domain controller in a new domain name. This is done when an NT Server is installed on a machine. Two options appear during the setup procedure to either create a controller in a new domain or a server in a domain. Selecting controller in a new domain will make that server the Primary Domain Controller (PDC) by default. A domain name must be supplied. The domain name must not be the same as any other domain on your network (WAN) or any machine name within another domain.

As Primary Domain Controller, that NT Server will have the master copy of the Security Access Manager (SAM) database. This is the place where user IDs, passwords and group information is stored. It is encrypted and resides in the following file:

WINNTSYSTEM32CONFIGSAM

All other servers added to the domain as Backup Domain Controllers will hold a copy of that database, which is regularly synchronized against the master copy on the PDC. When Windows NT servers are booted, a copy of the SAM is copied into memory. The SAM database is further discussed in 2.2.2, “Primary Domain Controller.”

A computer running Windows NT Workstation or NT Server can belong to either a workgroup, a domain or neither (stand-alone), but cannot belong to both at the same time. Computers with either of these operating systems that are not domain controllers will have their own local copy of a SAM database (located in the same local directory).

2.2.1 NT Server and NT Workstation

NT Server can be installed on a machine without that machine becoming a domain controller. If this NT Server workstation joins a domain, but not as a domain controller, it can operate as a file or print server. The workstation with NT Server installed behaves similarly to a system with NT Workstation installed.

A Windows NT machine will have a local copy of a SAM database. It can authenticate users logging in locally and decide what rights users have when logging on from that machine. This Windows NT machine is not capable of becoming a Backup or Primary Domain Controller. To become a domain controller, NT Server would have to be installed.

2.2.2 Primary Domain Controller

The Primary Domain Controller (PDC) is the first domain controller in a domain. It is also, by necessity, the first computer/server in a domain. Effectively a domain is created when an NT Server is installed as a controller in a new domain.

Each domain contains only one PDC, but can have any number of other domain controllers called Backup Domain Controllers (see 2.2.3, “Backup Domain Controller” on page 34).

The PDC holds the master copy of the SAM database. The SAM database contains information about what users can log onto the domain, their passwords and what groups they belong to. It also records what machine names are members of the domains and what other domains the domain knows about.

The PDC is responsible for the authentication of users in the domain. It also is responsible for the updates or maintenance of the domain SAM database. The SAM database is an important object to understand in Windows NT. It is often the feature that will determine if an enterprise will implement a multiple domain network. See 2.2, "Domains in Windows NT" on page 32, for more information.

Another machine within a domain can be prompted to replace a server in its role as PDC. This is discussed in the next section, 2.2.3, "Backup Domain Controller."

To assist in the exercise of planning domain requirements, the size of a given SAM database file can be estimated fairly accurately. Table 2 summarizes the figures to use.

Table 2. Sizing the SAM Database

SAM Component	Size SAM increases by
User account	1024 bytes (1KB)
Computer or machine account	512 bytes (0.5KB)
Global group	12 bytes/user + 512 bytes
Local group	36 bytes/user + 512 bytes

The size of the SAM database is the system architect's choice. There are a few considerations that may help you decide its size:

- The SAM database gets read into the memory of each PDC and BDC. Therefore, there is an overhead of how much RAM a domain controller server will require.
- The larger the SAM database, the longer a domain controller server will take to boot. This is because the SAM database gets read into memory when the system boots up.
- The size of the processor in a domain controller will influence how quickly the SAM database gets read into memory and how quickly users get authenticated.
- Microsoft used to recommend a SAM database no larger than 10 MB in size. However, in 1996, this figure was revised to no larger than 40 MB.
- A reasonable size is 10 MB. However, to confirm the actual size of a SAM database, view the contents of the following file:

  WINNTSYSTEM32CONFIGSAM

2.2.3 Backup Domain Controller

Once a domain has been established (a Primary Domain Controller was created), other NT Servers can be added to the domain as either ordinary file or print servers or as domain controllers. If added as another domain controller, an NT Server is called a Backup Domain Controller (BDC).

The major difference between a PDC and a BDC is the security database (SAM). A BDC holds a copy of the SAM database from the PDC. The BDC can authenticate users to the domain on behalf of the PDC, and any updates the BDC makes are copied back to the SAM database on the PDC.

A disadvantage of domain controller servers (PDCs and BDCs collectively) having the same SAM database is that users or administrators cannot log on locally to the machine in the same way that they log on to a workstation.

A BDC polls the PDC regularly for any changes that have been made to the SAM database. The default is to poll every five minutes, but the frequency of polling can be changed in the following registry:

HKEY_LOCAL_MACHINESystemCurrentControlSetServicesNetlogonPulse

It is of type DWORD, and you can set the value from 60 to 3600 seconds. If there have been any changes since the last poll, only the changes are transmitted, not another copy of the whole database. The BDCs can also be manually synchronized with the PDC through the Synchronize Entire Domain option on the Server Manager (see Figure 31).

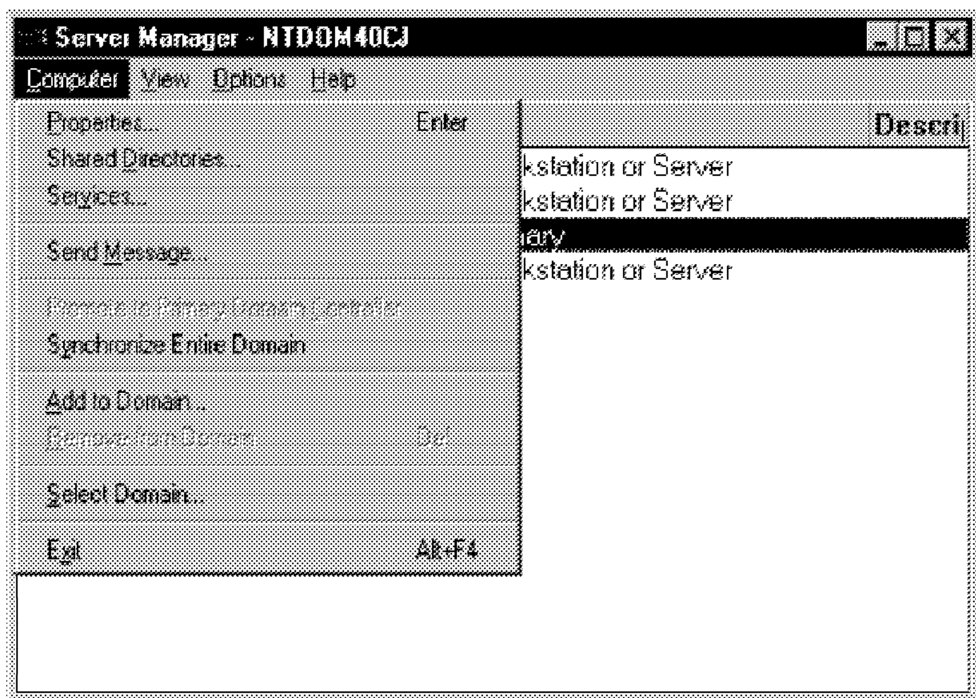


Figure 31. Server Manager

Deciding how many BDCs to have in a domain can provide some challenges. The physical distribution of the network and enterprise architecture are considerations in determining how many BDCs to have in a domain. BDCs can offer a faster logon time for users, as well as redundancy for the system, should the PDC fail. However, backup domain controllers generate network traffic, thereby slowing down a network. You must also consider the cost of multiple BDC servers.

Although a domain can have any number of BDCs, Microsoft recommends that the network traffic issues of a BDC outweigh their benefits as domain controllers. This document does not discuss NT networking. We can say that although some BDCs are good, and even necessary, consider the number of NT Servers in your network. If your network contains a large number of NT Servers, they do not all need to be BDCs. Keep some NT Servers only as file and print servers.

In the previous section (2.2.2, "Primary Domain Controller" on page 33), it was suggested that, depending on the size of a domain, it is desirable to keep the PDC as a dedicated server if the resources are available. BDCs, however, are far more likely to be file, print or application servers first, and domain controllers second. 2.6.4, "DB2 for NT in a Domain" on page 64, discusses the placements of applications (DB2) on PDCs and BDCs.

A BDC can be promoted to PDC, but only a BDC can do this. An NT Server that has not been installed as a BDC cannot become a PDC. This is performed through the Server Manager (Figure 31 on page 35). If the PDC is active when a BDC is promoted, then the PDC is automatically and dynamically demoted to being a BDC.

If a PDC fails and a BDC is promoted to assume the PDC duties, then when the original PDC is started again, the Server Manager will show there are two PDCs in the domain. One of them must be demoted to a BDC. This is done through the Demote to Backup Domain Controller option on the Server Manager. This only occurs when two PDCs exist.

2.3 Groups and User Authentication

Another part of security in Windows NT is the concept of groups. Groups give Windows NT administrators the ability to grant rights and privileges to a number of users at once, without having to maintain each user individually. Groups, like user accounts, are defined and maintained in the SAM database of Windows NT machines.

There are two types of groups in the Windows NT architecture:

- Global groups
- Local groups

Global groups exist only on a domain. However, local groups can exist either on a domain or on a specific machine. The key to understanding whether a group is local to a domain or to a workstation is knowing in which SAM database the group is defined. Remember that a workstation can be a Windows NT Server that is not a domain controller.

The Primary Domain Controller holds the SAM for the domain. This SAM is replicated to any BDCs in the domain. Domain controllers do not have a *local* SAM database. They hold user and group data for the domain. In this sense, any groups created on the PDC, local or global, are domain groups.

Windows NT machines that are not domain controllers (NT workstations and some NT servers) will each have their own SAM databases. User accounts and groups created on those machines are local to that machine. There is no Create Global Group option on machines that are not domain controllers.

User accounts, global and local groups will be discussed in more detail.

2.3.1 User Accounts

Each user in a Windows NT domain will have a user account created for him/her. A user account is essentially a record of a user ID, password and groups of which the user is a member, along with a few other details, such as logon restrictions, logon scripts to be executed, the user's home directory, mandatory profile, and account expiration. Figure 32 shows the screen in the User Manager used to create a new user account. Once a user account has been created, it can be made a member of a group.

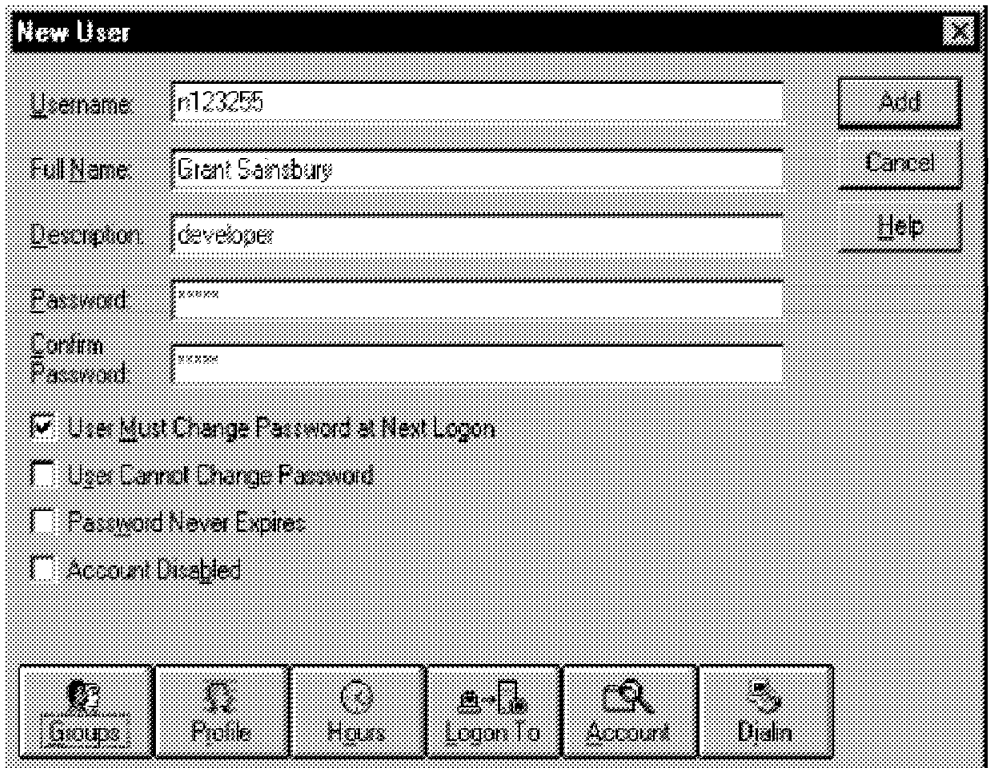


Figure 32. Creating a User Account

2.3.2 Global Groups

Global groups are domain objects. They can only be created on a domain. Figure 33 on page 38 shows the User Manager for Domains screen used to create a global group. They are called global groups because they can be accessed by any machine in a domain and can be seen across domains.

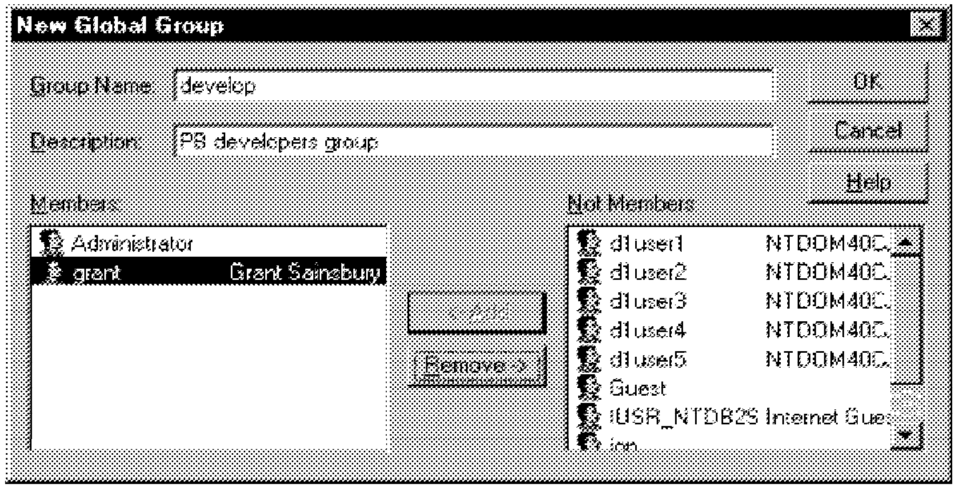


Figure 33. Creating a Global Group

Global groups can only contain user accounts from the domain on which they were created. They cannot contain any other groups as members. However, once created, global groups can be seen and used by any trusting domains. (See 2.4.1, “Trusted Domains” on page 43).

Global groups can have permissions granted to them. Microsoft does not recommend granting permissions to global groups, and rights cannot be granted to a global group.

NT Server comes with a number of default global groups. Those groups are:

- Domain Users
- Domain Admins
- Domain Guests

Domain Users contain all user accounts created on the domain.

Domain Admins contain designated administrator accounts. By default, Domain Admins contain only the Windows NT default administrator account called Administrator.

Domain Guests contain all guest accounts for the domain. By default, Domain Guests contain only the default guest user account called Guest.

Within User Manager, global groups are identified by a picture of a globe being included in the icon, as opposed to a computer for a local group.

2.3.3 Local Groups

Local groups are local to the Windows NT machine on which they are created. Remember that group information will be stored in a machine’s SAM database. A local group created on a workstation will be specific to that workstation. A local group created on the PDC will, however, apply to all domain controllers in that domain because BDCs receive a copy of the PDC SAM.

Rights and Permissions

It is local groups, or individual users, that should be granted rights and permissions, as opposed to global groups.

Local groups can contain individual user accounts that are local to the machine, and found within either the machine's domain or any trusted domains. Local groups can also contain any global groups from the machine's domain or fom any trusted domains. (Trusted domains are explained in detail in 2.4.1, "Trusted Domains" on page 43).

Global groups from trusted domains can be seen in the New Local Group window of the User Manager, under the List Names From drop-down window (see Figure 34).

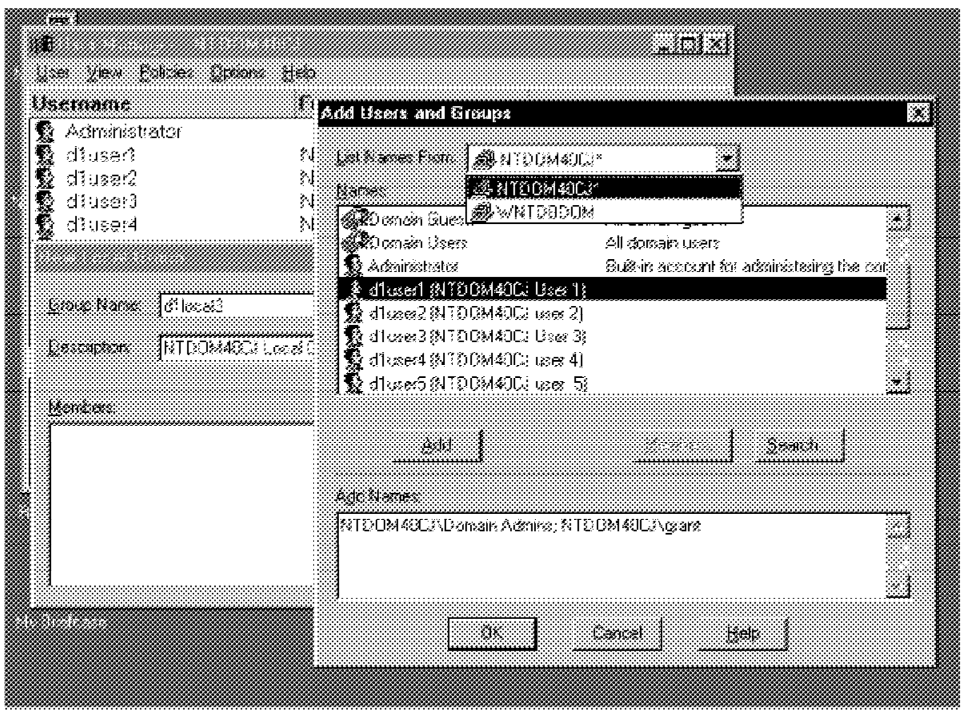


Figure 34. Creating a Local Group

To access an individual user account, click on the **Members** button.

A local group cannot contain other local groups.

Windows NT has a number of default local groups established at installation. The default local groups are:

- Account Operators (NT Server only)
- Administrators
- Backup Operators
- Guests
- Power Users (NT Workstation only)

- Print Operators (NT Server only)
- Replicator
- Server Operators (NT Server only)
- Users

There is one additional group called Everyone. This group will not appear on the list of groups in the User Manager. However, it can be assigned rights and permissions. Anyone who has a user account in the domain, including all local and remote users, is a part of the Everyone local group. The Everyone group also contains all global groups of any trusted domains.

The Administrators group is the most powerful group. By default, it contains only the default administrator account called Administrator. It is important to remember the password selected for this account when Windows NT was installed. If the password is forgotten and no other administrator accounts exist, Windows NT has to be reinstalled.

When an NT Workstation or NT Server joins a domain (but not as a domain controller, in the case of NT Server), the Domain Admins global group is added to the Administrators local group on the machine. This gives any member of the Domain Admins group administrative privileges on that machine.

The Administrators local group on the Primary Domain Controller is the Administrators group for all domain controllers of that domain.

Members of the Users group have minimal rights on machines running NT Server, but they do have rights on NT Workstations. They do have the right to manage and create local groups. When a Windows NT machine joins a domain (not as a domain controller, in the case of NT Server), the Domain Users global group is added to the Users local group on the machine. This allows all domain users to log on at the machine.

A user ID must be known to at least one local group on a Windows NT machine before that user is allowed to log on at that machine.

Within User Manager, local groups are identified by a picture of a computer being included in the icon, as opposed to a globe for a global group.

2.3.3.1 Domain Scenario

To help illustrate the concept of group, Figure 35 on page 41 is a representation of a domain. This domain has the following:

- One PDC (machine A)
- Two BDCs (machines B and C)
- Another NT Server (machine D)
- Two NT Workstations (machines E and F)

A global group was created on machine A, the PDC, and will therefore also exist on B and C, the BDCs. This global group will be global to the domain. This implies that it can be made a part of any local group in this domain (on A, D, E or F) or part of a local group in a trusting domain. Such a global group could only contain user accounts defined on A.

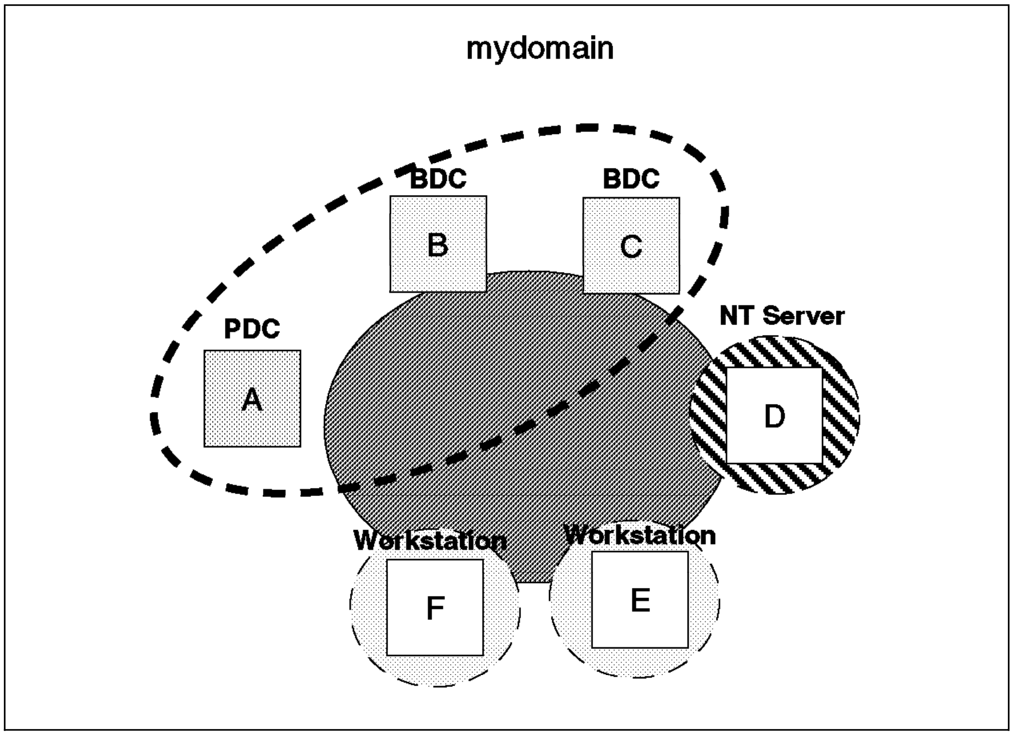


Figure 35. Local Groups in a Domain

A local group created on machine A is local only to the domain controllers. That is, if a user account was created and made only a member of a local group on machine A, that user could only log on to machines A, B and C. A local group on machine A could contain individual user accounts or global groups defined either on machine A or from a trusted domain.

Machines B and C, the Backup Domain Controllers, cannot and do not have their own local groups. This is because their SAM is a copy of that from the PDC, machine A.

A local group defined on D (or E or F) will be local to machine D. Such a local group can contain user accounts that are local to machine D or are domain accounts on machine A. It could also contain global groups from machine A or from a trusted domain.

A user must be a member of a local group on any given Windows NT machine before being able to log on from that machine.

2.3.4 Authentication

Thus far, we have discussed the concepts of Windows NT user accounts, local and global groups, domains and trust relationships. The next logical topic is authentication. The actual process of user authentication is a relatively simple one. (Trust relationships are discussed in 2.4.1, "Trusted Domains" on page 43). Authentication is verifying that a user is who they say they are. This is done by matching a user ID with a password.

Recall that user IDs and passwords are stored in the SAM database on Windows NT machines, but a user's password does not necessarily have to reside on the machine from which he/she logs on, because of the domain concept.

When Windows NT authenticates a user, it follows a simple hierarchy to look for a user ID and password. This hierarchy of authentication is summarized in Figure 36 on page 42.

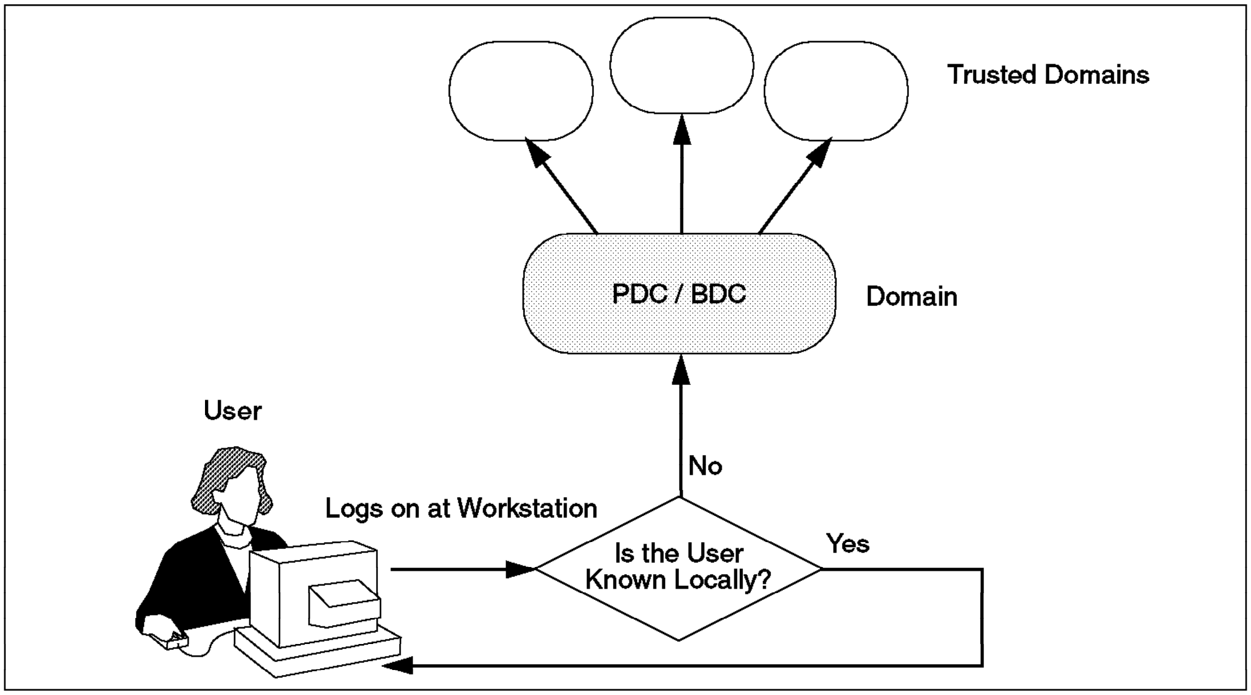


Figure 36. User Authentication in the Windows NT Environment

If the DB2 for NT server is part of a domain, it first checks the SAM database of the domain to which it belongs. If the ID is not found there, it then checks any trusted domains.

If the user ID is not locally known, then an authentication request is sent to a domain controller for the domain in which the user is requesting a logon. The domain controller that does the authentication can be either the PDC or a BDC. BDCs have a copy of the PDC's SAM database. To determine which domain controller will perform the authentication, a broadcast message is sent out from the user's machine and the first domain controller to respond to the message will perform the authentication.

If the user is not known to the domain, (for example, his or her user ID is not in the SAM database of the PDC), then domain controllers of any trusted domains are queried. It can be either the PDC or a BDC that responds to an authentication request from a trusting domain.

Once the user account has been found and the password authenticated, any account or policy restrictions are determined, as well as a list of groups of which the user is a member.

DB2 authentication is discussed in 2.5, "DB2 for NT Authentication and Security" on page 50.

2.4 Trust Relationships between Domains

Thus far, only the concept of a single domain has been discussed. However, an enterprise may wish to establish more than one domain. These domains do not have to exist independently, nor do separate user accounts have to exist for each domain a given user wishes to log into. Multiple domains can be achieved through relationships between domains called trusts.

2.4.1 Trusted Domains

The main reason trust relationships are established is that users from one domain have the ability to access resources in another domain without having to be reauthenticated.

There are two characteristics in a trust relationship:

1. One domain trusts another to authenticate users on its behalf and therefore grant access to resources in its domain without reauthenticating users.
2. An administrator from one domain trusts an administrator from another domain to administer resources in their respective domains.

It is recommended that the two domains in a trust relationship first be defined. They are called the trusting and the trusted domain. A scenario representing this is shown in Figure 24 on page 23. A trust relationship lets an administrator of one domain (the trusting domain) grant rights and permissions to global groups and users of another domain (the trusted domain). The administrator of the trusted domain must be, in turn, trusted since this administrator can control which users are members of global groups. For example, if a DB2 server was part of a trusting domain, the administrator of that domain could issue an SQL statement (GRANT SELECT) to give permission on a database to users from a trusted domain.

Trust is one-directional. Figure 37 represents a trust relationship created between two domains, A and B.

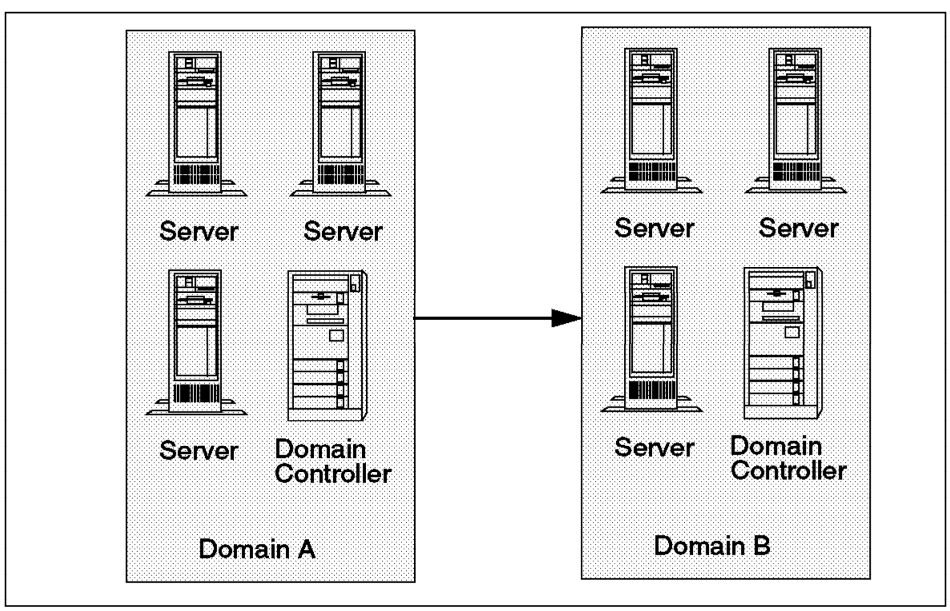


Figure 37. Domain A Trusts Domain B

In Figure 37, domain A trusts domain B to authenticate users on its behalf and therefore will grant resource access to users from domain B. In this example, B is the trusted domain, and A is the trusting domain. In practice, users logging onto domain B would see both domains in the domain drop-down menu on the Windows NT log on box, while users in domain A would see only domain A.

Trust relationships are not transitive. This means that explicit trust relationships need to be established in each direction between domains. There is no concept of an implicit or piggybacked trust relationship.

Figure 38 represents three domains where domain A trusts domain B and domain B, in turn, trusts domain C. With just these two relationships, domain C cannot authenticate users on behalf of domain A. Domain A does not trust domain C.

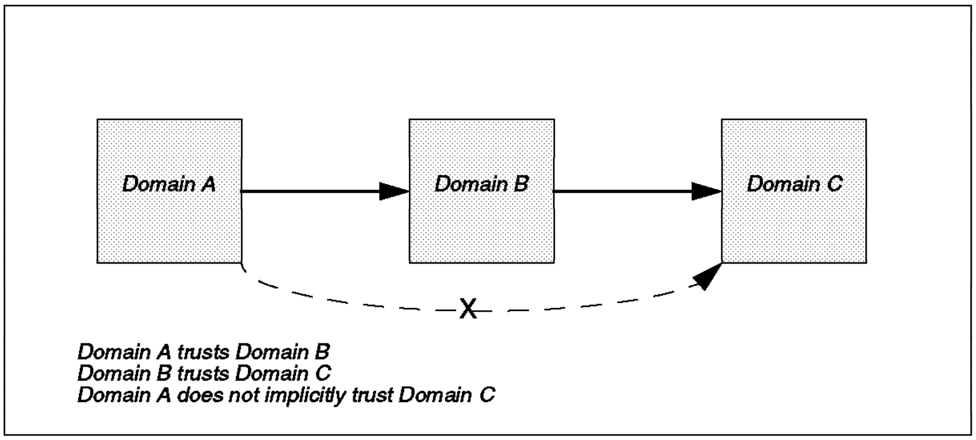


Figure 38. Trust Relationships Between Domains A, B and C

2.4.2 Creating a Trust Relationship

A trust relationship is relatively simple to maintain. The maintenance of a trust relationship is done by administrators in the User Manager on both domain controllers. The relationship first needs to be set up on the PDC in what is to be the trusted domain.The following are the steps to create the trusted domain on the PDC:

1. In the User Manager for Domains, select **Policies | Trust Relationships**. A window will be displayed.

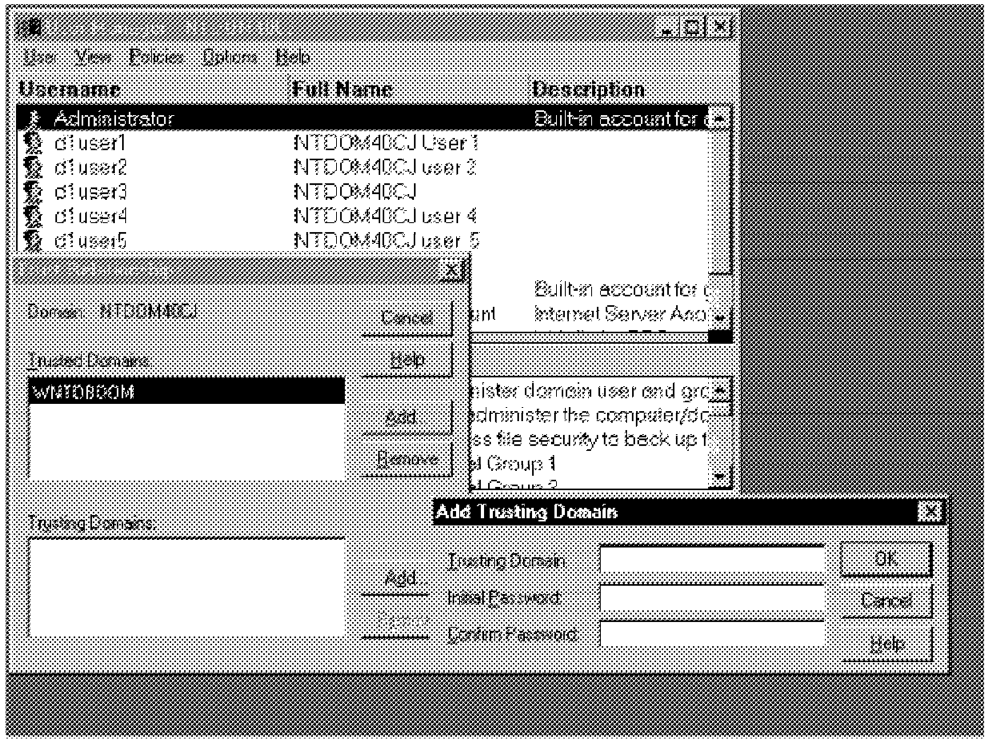


Figure 39. Creating a Trust Relationship between Domains

2. Next, select **Add next to Trusting Domains**.
3. Enter the name of what will be the trusting domain.
4. Enter and confirm a password that the administrator of the trusting domain will use to complete the relationship. (See Figure 39.)
5. Next, on the PDC for the trusting domain select **Policies | Trust Relationships** in User Manager for Domains.
6. Select **Add next to Trusted Domains** and enter the name of the trusted domain and the password selected to be the administrator of the trusted domain. The trust relationship is now complete.

The two domains will agree on a new password that will be unknown to either administrator. If the relationship needs to be broken, then it must be removed from both domains. This is performed in the same way as adding a trust relationship, except select **Remove**. A broken trust relationship can be reestablished by repeating the create trust relationship procedure described.

2.4.3 Models of Domain Trust

Choosing the right architecture for an enterprise can be an involved and complex task, with any number of considerations to be taken into account. To assist in this process, Microsoft suggests four models of domain organization. They are:

1. The Single Domain Model
2. The Master Domain Model
3. The Multiple Master Domain Model
4. The Complete Trust Model

These should be treated just as models. Organizations should configure their domain(s) to best suit their individual needs. However, each of these models will now be briefly explained, along with how the model might be used with DB2. There will be other important factors that influence the model an organization will choose. A Windows NT planning reference should be consulted for more information.

2.4.3.1 The Single Domain Model

The single domain model is represented in Figure 40.

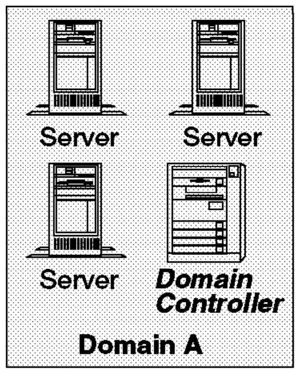


Figure 40. The Single Domain Model

All servers and workstations belong to one domain. There are no trust relationships to any other domain.

Advantages of this model include:

- Easy implementation
- Suitable design for a small to medium-sized network (See 2.2.2, "Primary Domain Controller" on page 33, for the size of SAM database.)
- No trust relationships to establish or maintain
- One set of administrators

Disadvantages of this model include:

- List of users and machines can grow to an undesirable size
- Network and server performance problems may arise, keeping domain servers in sync and browsing
- No groupings of users of resources

An example of a single domain model might be a small network with an independent domain. This could be a production environment, where it is desirable to keep the production data separate from the development environment. You might also have a number of small domains for an organization where either the sharing or separation of resources such as databases is not required. The ability to administer each domain separately also is not an issue.

The most compelling reason not to implement with this model, especially in a production environment, is generally the size of the domain, specifically the number of users and machines. These factors impact the size of the SAM database on the domain controllers. The system architects for an enterprise need to determine the optimal size for a SAM database before they consider moving to a multi-domain architecture.

2.4.3.2 The Master Domain Model

Figure 41 is an example of the master domain model.

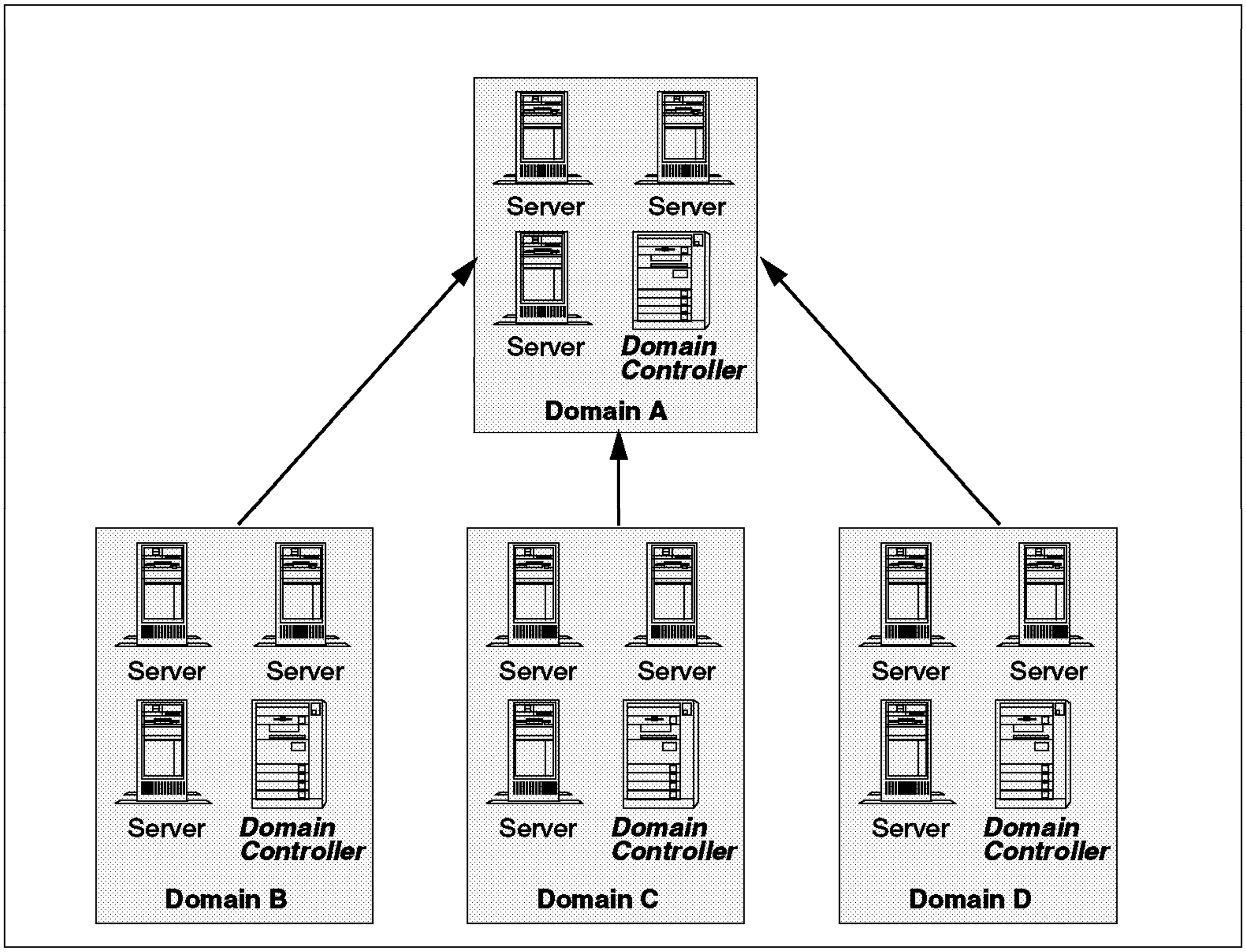


Figure 41. The Master Domain Model

The master domain is one domain in which all users are defined. All other domains trust the master domain. Domains other than the master domain have no users defined. The other domains are called resource or slave domains.

Advantages of this model include:

- Administration for the enterprise is centralized.
- Supports logical grouping of resources (such as divisions or departments).
- Supports geographical division of an enterprise.
- Ability to protect data from Master Domain administrators.
- Global groups are defined only once.

Disadvantages with this model include:

- Performance degradation on WAN or with large number of users.
- Local groups have to be defined on each domain.
- Global administration (this is the same administrator operating from all domains) can be cumbersome to establish.

- Master domain becomes a point of failure.

You might find the master domain model implemented for each department in an organization. However, all administration and authentication occurs in the one master domain. The enterprise is split geographically, and resources are grouped accordingly. However, users are all defined and administered centrally. This model easily supports movement of personnel across domains.

2.4.3.3 The Multiple Master Domain Model

Figure 42 shows the multiple master domain model.

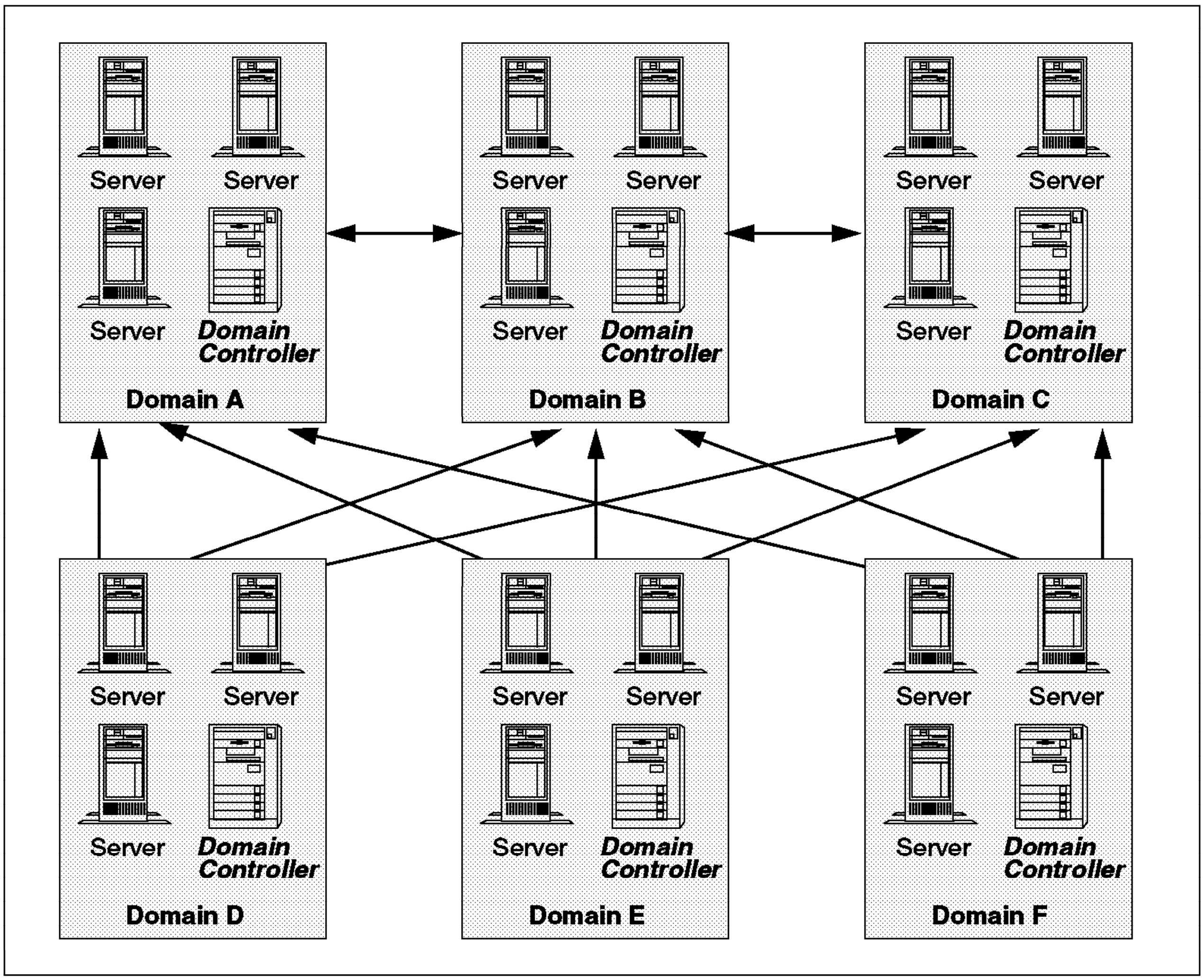


Figure 42. The Multiple Master Domain Model

More than one domain, in which user accounts are held, have trust relationships between each other. Any of the master domains can authenticate each other for a number of resource domains.

Advantages of this model include:

- Supports a large number of users with acceptable performance
- Resources are grouped logically
- Resource domains can be managed independently for security

Disadvantages of this model include:

- Groups may need to be defined more than once for different domains
- Many trust relationships to manage
- Maintenance of user accounts is more difficult because they are in multiple domains

A multiple master domain may be established for the same reasons as a master domain. However, there are too many users for one domain to handle all the authentication requests. Therefore, to ease network traffic and speed user authentication requests, multiple master domains are created to service the local resource domains.

2.4.3.4 The Complete Trust Model

Figure 43 shows the complete trust model.

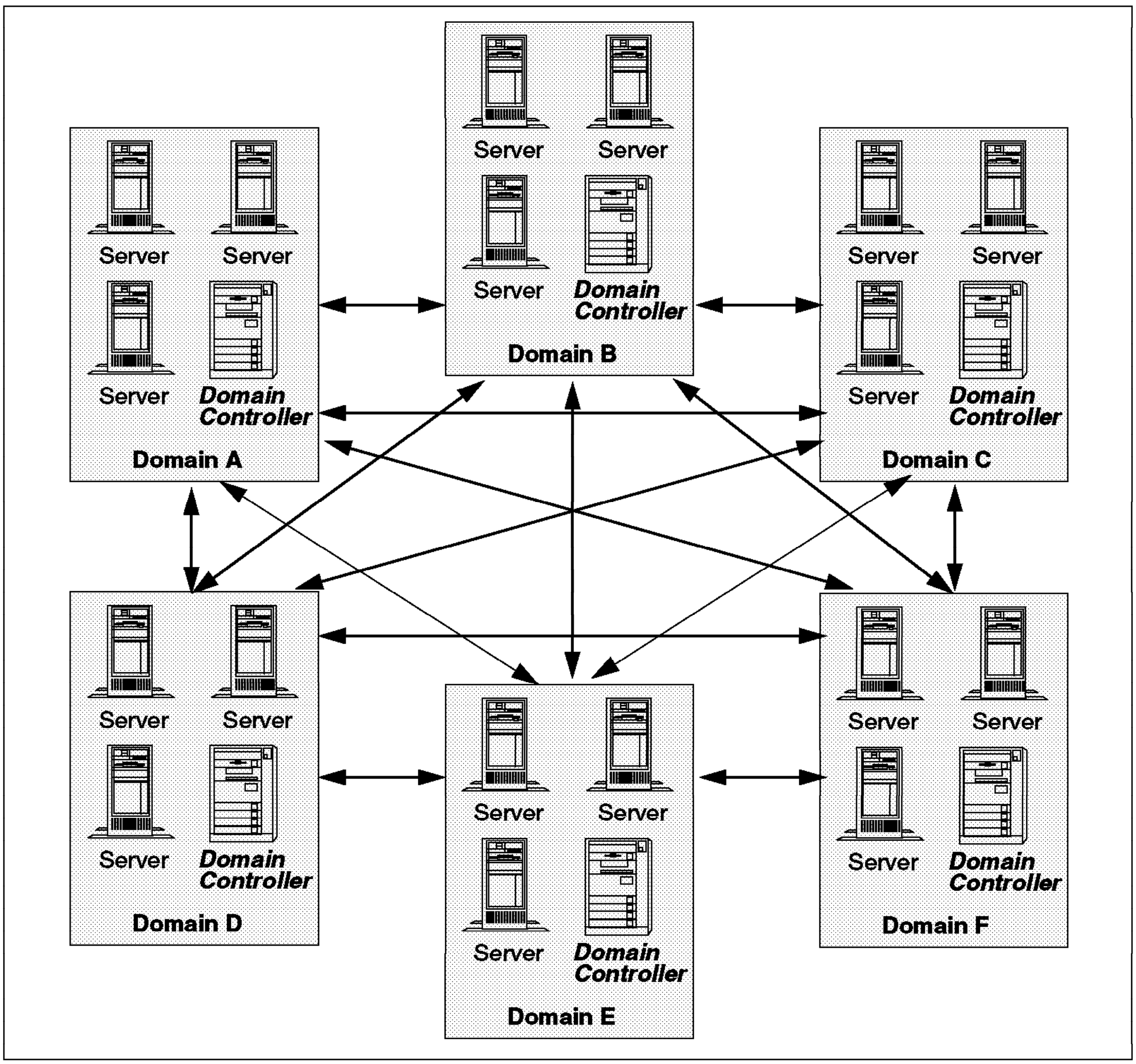


Figure 43. The Complete Trust Model

In a complete trust model, domains exist with trust relationships to and from all other domains on the network.

Advantages of this model include:

- Supports a large number of users
- Does not require central administration
- Resources and users are grouped logically into domains (from browser perspective)
- Independent management of resources for each domain

Disadvantages of this model include:

- Lack of central administration can cause potentially severe network problems
- Large number of trust relationships to manage

An example of where the complete trust model might be implemented is in a development environment.

2.5 DB2 for NT Authentication and Security

Thus far, this chapter has focused on Windows NT security and authentication. This section looks at those concepts from a DB2 for NT perspective.

DB2 for NT Version 2.1.2

This and prior versions of DB2 for NT do not strictly adhere to the Microsoft models of trust. DB2 for NT only supports local groups for granting authorities. Also, user ID, local and global groups must all be defined at the same SAM database in order for DB2 to resolve the group to which a user belongs.

2.5.1 DB2 for NT Group Resolution

There are two rules that must be observed to make DB2 for NT work successfully over a network. They are:

1. Privileges can only be granted to individual users or local groups. Privileges cannot be granted to global groups.

2. Users will only be recognized as being part of a local group if their user account resides on the same Windows NT machine as the local group. That is, the user account must be in the same SAM database as the local group.

DB2 for NT does support trusted domains, but must comply with the two rules stated above. Authorities and privileges can be granted to a local group of a trusted domain, but the users on the DB2 server must have user accounts created or defined on the same Windows NT machine. This machine is the PDC of the trusted domain. This is necessary in order to get a successful connection to DB2.

Figure 44 on page 51 and Figure 45 on page 52 help illustrate how DB2 handles privileges for users and groups.

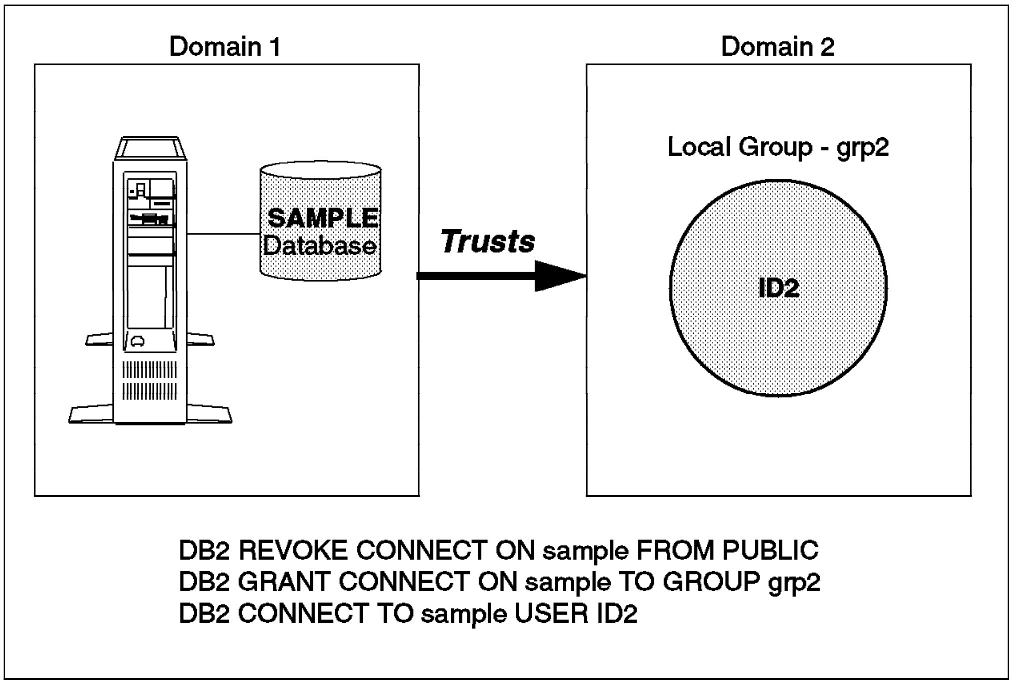


Figure 44. Successful DB2 for NT Database Connection Across Trusted Domains

Figure 44 demonstrates an example that results in a successful connection to the SAMPLE database. There is a trust relationship between Domain 1 and Domain 2. Domain 1 trusts Domain 2 to authenticate users on its behalf and therefore will grant resource access to users from Domain 2. Privileges are revoked from the group public to illustrate this scenario. The DB2 database server on Domain 1 grants access to the SAMPLE database to the local group, grp2, on Domain 2. Because the user whose ID is ID2 is a member of the local group, grp2, the connection to the SAMPLE database is successful.

Defining the User ID and Local Group

The user ID and local group do not need to be defined on the same domain or server where the database manager is running, but they must be defined on the same domain or server as each other.

In Figure 44, the user ID, ID2, and the local group, grp2, are defined in Domain 2. However, the database is defined on Domain 1.

The next example (Figure 45 on page 52) shows an unsuccessful database connection among trusted domains.

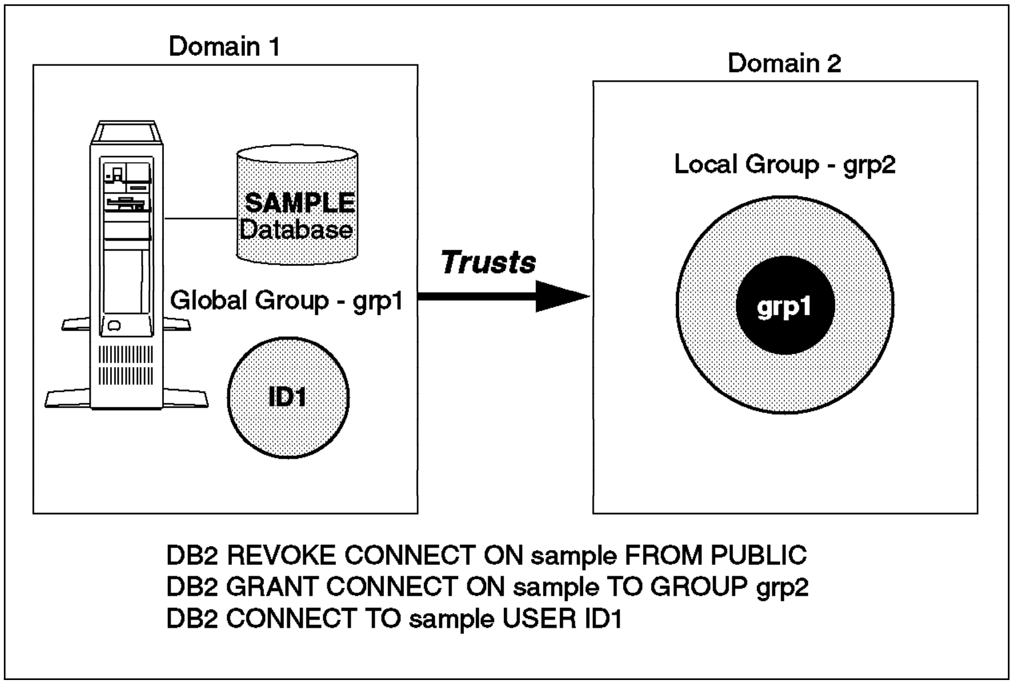


Figure 45. Failed DB2 for NT Database Connection Across Trusted Domains

The scenario depicted in Figure 45 will not work because the user ID and local group are in different domains.

2.5.2 Authority Levels

Authorization levels in DB2 provide a hierarchy for administration capabilities. This can be seen in Figure 46.

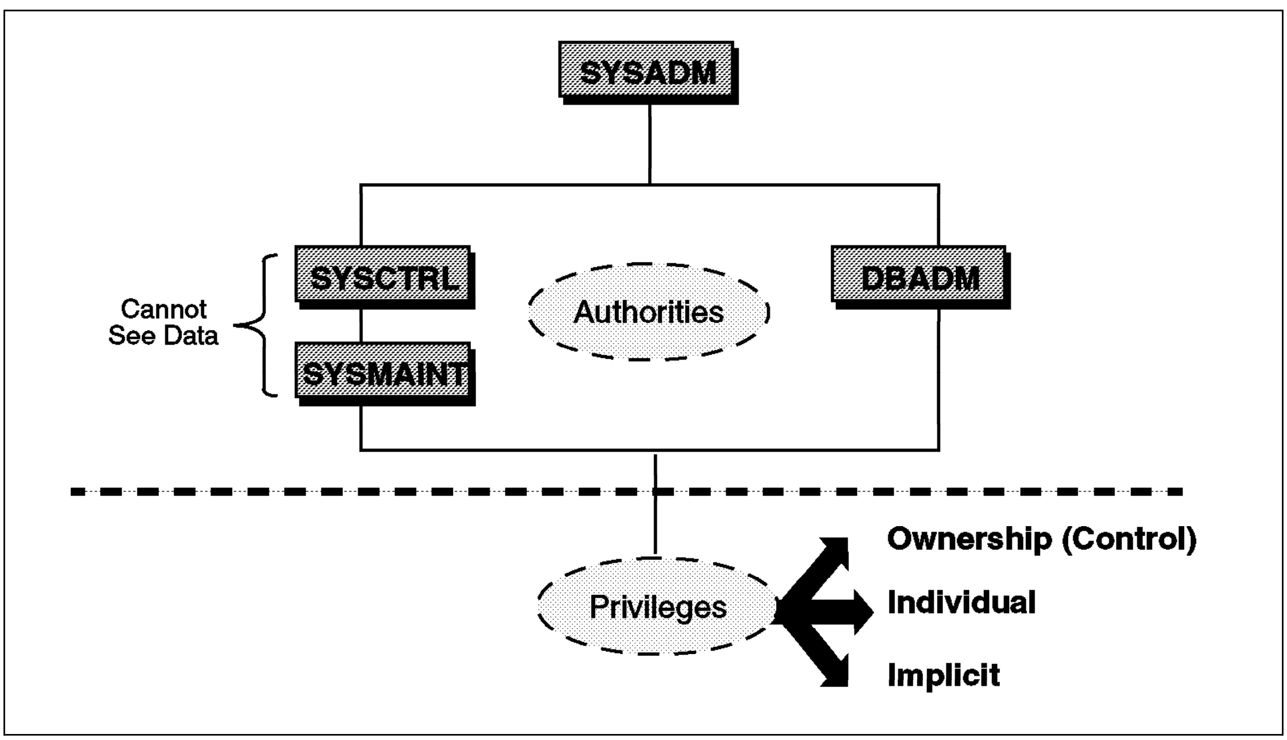


Figure 46. DB2 Access Control Hierarchy

An authority is a set of privileges covering a set of databases or database objects. These authorities are assigned to a group of users. Each member of the group has the same DB2 authority, unless he or she is explicitly removed.

At the top of this hierarchy is the DB2 System Administrator or SYSADM. Any member of the SYSADM user group is able to perform any of the DB2 administration operations as well as access all database objects. SYSADM members are the only users allowed to configure the DB2 instance.

User names that are defined with a user type of Administrator in the Windows NT User Manager have system administrator (SYSADM) privileges by default on a system wide basis.

System administration authority for a particular instance is also given to any user belonging to the group specified by the `SYSADM_GROUP` parameter in the DB2 database manager configuration file.

By default, the system administrator of an instance is anyone who is a Windows NT administrator. You can change this for each instance by changing the `SYSADM_GROUP` parameter. But before you do, ensure that:

1. The group exists. Use the Windows NT User Manager administrative tool to create groups.
2. The group is a local group.
3. To obtain SYSADM authority for an instance, users must be members of that local group. Their user accounts are on the same machine (that is, in the same SAM database) as the local group. This has the benefit of allowing a user to be a DB2 administrator without being a Windows NT administrator.

To specify group names, update the database manager configuration file using one of the following:

- The Command Line Processor documented in the *DB2 Command Reference Manual*

  For example:

  ```
  db2 update database manager configuration using sysadm_group dbadmin
  ```

- The configuration API documented in the *DB2 API Reference*

A user with SYSADM privileges may give authority for both system and maintenance operations, as well as database operations. Two additional levels of system control authority (SYSCTRL and SYSMAINT) are provided. Users receive these authority levels through membership in groups.

These group names must be entered in the database manager configuration file as values for the `SYSCTRL_GROUP` and `SYSMAINT_GROUP` parameters. Initially these parameters are set to null, meaning that no user has these authority levels.

System control (SYSCTRL) provides the ability to perform almost any administration command. A member of the SYSCTRL user group does not have authority to access database objects or modify the instance configuration file (DBM configuration). SYSCTRL offers almost complete control of database objects defined in a DB2 instance, but cannot access user data directly, unless explicitly granted the privilege to do so. A user with this authority, or higher, can perform the following functions:

- Update the database, node and DCS directory entries
- Update database configuration parameters
- Create or drop a database
- Force applications
- Quiesce the DB2 instance or database
- Execute the RESTORE/BACKUP/ROLLFORWARD commands
- Create or drop a tablespace

SYSMAINT, or System Maintenance authority, allows the execution of maintenance activities but not access to user data. Only users with this level of authority (SYSADM or SYSCTRL) can do the following tasks:

- Update database configuration files
- Back up databases and tablespaces
- Restore an existing database
- Restore tablespaces
- Start and stop the DB2 instance
- Run the Database Monitor
- Start and stop traces

The authority levels of SYSCTRL and SYSMAINT provide access to instance-level commands and to a limited number of database-level commands for the purpose of system maintenance. No direct access to data within the database is permitted for users that have only these authorities.

At the database administration level, there is the DBADM authority. The creator of a database will automatically have DBADM authority for the new database. Other users can be granted DBADM authority by using the SQL GRANT statement. It is possible to hold DBADM authority for multiple databases. DBADM provides some common administration tasks, such as loading data, creating database objects and monitoring database activity.

A privilege is the right of a particular user or group to create or access a resource. There are three types of privileges: ownership, individual, and implicit.

1. Ownership or control privileges - For most objects, the user or group who creates the object has full access to that object. Control privilege is automatically granted to the creator of an object. There are some database objects, such as views, that are exceptions to this rule. Having control privilege is like having ownership of the object. You have the right to access the object and give access to others. Privileges are controlled by users with ownership or administrative authority. They provide other users with access using the SQL GRANT statement.
2. Individual privileges - These are privileges that allow you to perform a specific function, sometimes on a specific object. These privileges include select, delete and insert.
3. Implicit privileges - As an example, a user who can execute a package that involves other privileges obtains those privileges while executing the package. This is known as the execute privilege.

This section describes the possible classes of users and their related authority levels. Note that the first three (SYSADM, SYSCTRL and SYSMAINT) are controlled outside of DB2 and recorded in the database manager configuration file, while DBADM is controlled within DB2 through the SQL GRANT statement.

Table 3 shows the valid authorities for the various DB2 levels.

Table 3. Database Authorities

Function	SYSADM	SYSCTRL	SYSMAIN	DBADM
CATALOG/UNCATALOG DATABASE	YES			
CATALOG/UNCATALOG NODE	YES			
CATALOG/UNCATALOG DCS	YES			
UPDATE DATABASE MANAGER CONFIGURATION FILE	YES			
GRANT/REVOKE DBADM	YES			
GRANT/REVOKE SYSCTRL	YES			
GRANT/REVOKE SYSMAINT	YES			
FORCE USERS	YES	YES		
CREATE/DROP DATABASE	YES	YES		
QUIESCE DATABASE	YES	YES		
CREATE/DROP/ALTER TABLESPACE	YES	YES		
RESTORE TO NEW DATABASE	YES	YES		
UPDATE DATABASE CONFIGURATION FILE	YES	YES	YES	
BACK UP DATABASE/TABLESPACE	YES	YES	YES	
RESTORE TO EXISTING DATABASE	YES	YES	YES	
PERFORM ROLL FORWARD RECOVERY	YES	YES	YES	
START/STOP INSTANCE	YES	YES	YES	
RESTORE TABLESPACE	YES	YES	YES	
RUN TRACE	YES	YES	YES	
OBTAIN MONITOR SNAPSHOTS	YES	YES	YES	
QUERY TABLESPACE STATE	YES	YES	YES	YES
UPDATE LOG HISTORY FILES	YES	YES	YES	YES
QUIESCE TABLESPACE	YES	YES	YES	YES
LOAD TABLES	YES			YES
SET/UNSET CHECK PENDING STATUS	YES	YES		YES
READ LOG FILES	YES	YES		YES
CREATE/ACTIVATE/DROP EVENT MONITORS	YES	YES		YES
USE IMPORT/EXPORT UTILITY	YES			YES

Assigning Authorities

Access to SYSADM, SYSCTRL, SYSMAINT and DBADM privileges should be restricted to avoid the risk of compromising system integrity. These authority levels are not required for general use. For a more detailed discussion of privileges and authorizations, see the *DB2 Administration Guide for Common Server*. For a complete list of the minimum authority levels needed to execute DB2 commands and SQL statements, see the *DB2 Command Reference* or the *DB2 SQL Reference*.

2.5.3 Controlling Client Access to DB2 Databases

Controlling database access involves the control of access both to DB2 resources and to your data. A plan for controlling database access should be developed by defining your objectives for a database access control scheme and specifying who shall have access to what and under what circumstances. Such a plan should also describe how to meet these objectives by using database functions, functions of other programs, and administrative procedures. For example, if a database contains sensitive data, database access must be planned carefully to allow access to items only when necessary. A plan for controlling database access must include the necessary actions to protect databases containing sensitive data and to ensure that the databases are physically secure.

This section describes how to control access to the database manager and how the database manager controls access within itself. It also describes how you can customize access to the databases in your instance. This section provides additional information about the needs and functions of different users and using the system catalog tables to monitor access.

2.5.3.1 Importance of Security

One of the most important responsibilities of the database administrator is database security. Securing your database involves several activities:

- Preventing accidental loss of data or data integrity from equipment or system malfunction.
- Preventing unauthorized access to valuable data. The database administrator must ensure that sensitive information is not accessed by those without a need to know.

2.5.4 Authentication

Access to an instance or to a database is first validated outside the database manager. This process, known as authentication, verifies that users are really who they claim to be.

The authentication type for each instance determines how and where a user will be verified. The authentication type is stored in the database manager configuration file at the server. It is initially set when the instance is created.

All databases will have the same authentication type as the instance in which the database was created. For remote databases that reside on a down-level server or DRDA host, the authentication type can, optionally, be specified when a database is cataloged.

The following authentication types are provided:

- **SERVER** Specifies that authentication occurs on the server. The user name and password specified during the connection or attachment attempt are compared to the valid user name and password combinations on the server to determine if the user is permitted to access the instance. This is the default.
- **CLIENT** Specifies that authentication occurs on the node where the application is invoked. The user name and password specified during a connection or attachment attempt are compared with the valid user name and password combinations on the client node to determine if the user name is permitted access to the instance. No further authentication will take place on the database server.

CLIENT-Level Security is for TRUSTED Clients Only

Trusted clients are clients that have a security system.
Specifically, all UNIX, OS/2, and Windows NT clients are trusted.
Users on untrusted clients must provide a user ID and password on a connect request even when the instance security is CLIENT.

- **DCS** Specifies that authentication occurs at the DRDA Application Server (AS). If the Distributed Database Connection Services (DDCS) for NT product is not being used to access a DRDA AS, the DCS is the same as SERVER, and verification is at the DB2 for NT database server.

2.5.5 Configuring User and Group Security

This section describes how to access local, domain and remote databases. It also covers operational considerations when using authentication and provides sample scenarios showing SERVER and CLIENT authentication on both DB2 for Windows 95 and Windows NT client machines.

2.5.5.1 Local and Domain Database Access

Windows NT always requires a user to log on to a workstation or domain before granting access to the Windows NT Desktop. Once the user has been authenticated by the operating system, the user may attempt to perform database activities.

If the user performs a workstation logon, the user is known only to that workstation. When a DB2 user attempts to access a remote database that has SERVER authentication, the user will be required to supply a user name and password.

The DB2 for Windows NT database manager detects whether a connection is local or remote. For local connections, when authentication is SERVER, additional verification of the user is not performed. For remote connections, the user is validated using the specified user name and password.

2.5.5.2 Remote Database Access

If the remote instance has CLIENT authentication, a user name and password is not required. If the remote instance has SERVER authentication, the user must provide a user name and password although the user has already logged on to the local machine or to the domain.

DB2 for Windows NT introduces the concept of trusted clients for CLIENT authentication to protect against clients whose operating environment has no inherent security. This includes operating systems such as DOS, Windows 3.x and Windows 95.

To protect against unsecured clients, the administrator can change to trusted client authentication. This implies that all trusted platforms can authenticate the user on behalf of the server. Untrusted clients will be authenticated on the Server.

The following utility is provided to support trusted clients:

DB2TCLNT.EXE V | 0 | 1

where:

v View the current setting.

0 Trust only clients that are known to be secure. These are:

- AIX
- OS/400
- HP-UX
- MVS
- OS/2
- Solaris
- VM
- Windows NT

1 Trust all clients. This information is stored as part of the database manager configuration. Once this information has been changed, the DB2 Server must be restarted in order for the change to take effect. To change or view this option, use the DB2TCLNT.EXE utility.

When setting up security for DB2, you must log on with a user name that has administrator authority. The user name you use must follow the naming rules described in 2.6, “The DB2 for NT Environment” on page 61.

When you install Windows NT, you can create two administrator user names:

1. One is called Administrator.
2. The other is a name of your choice. You cannot use the Windows NT Administrator name because that name exceeds the eight-character limit imposed by DB2.

Windows NT supports serial log on to a machine. That is, only one person can log on locally at one time.

The user may log on to the local machine or, when the machine is installed in a Windows NT Advanced Server domain, the user may log on to the domain. DB2 for Windows NT supports both of these options. To authenticate the user, DB2 performs the following sequence:

1. Checks the domain controller for the current domain.
2. Checks any trusted domains known to the domain controller

To illustrate how this works, we'll look at the following example, Figure 47. We assume that the DB2 instance requires SERVER authentication. The configuration is as follows:

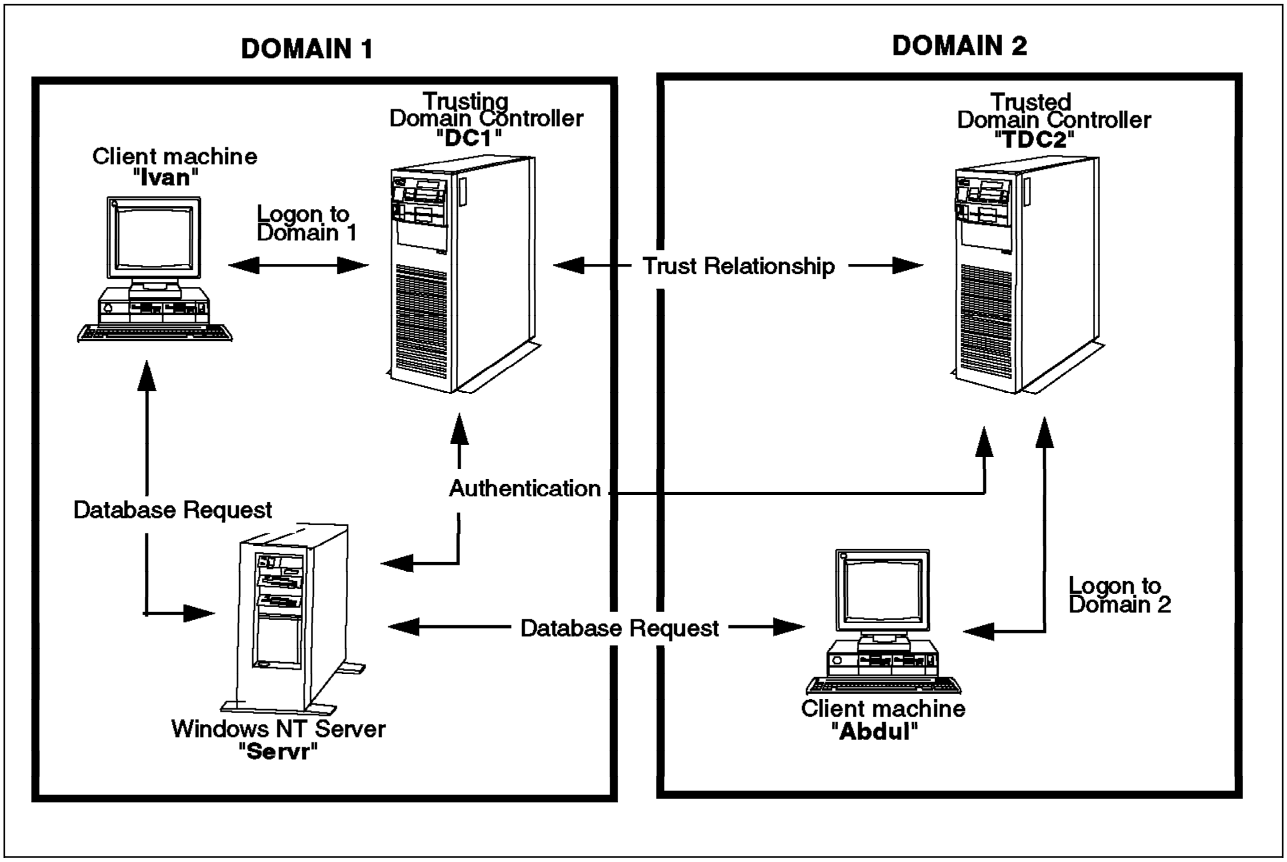


Figure 47. Authentication on Windows 95 and Windows NT

Each machine has a security database, Security Access Manager (SAM), unless one or both of the client machines are running Windows 95. Windows 95 machines do not have a SAM database. DC1 is the domain controller in which the client machine, Ivan, and the DB2 for Windows NT server, Servr, are enrolled. TDC2 is a trusted domain for DC1, and the client machine, Abdul, is a member of TDC2's domain.

2.5.5.3 Server Authentication

1. Abdul logs on to the TDC2 domain (that is, he is known in the TDC2 SAM database).
2. Abdul then connects to a DB2 database that physically resides on Servr:

   `DB2 CONNECT TO remotedb user` **`Abdul`** `using abdulpw`

3. Servr determines where Abdul is known. The API that is used to find this information first searches the local machine (Servr) and then the domain controller (DC1) before trying any trusted domains. User name Abdul is found on TDC2. This search order requires a single namespace for users and groups.
4. The DB2 machine called Servr then does the following:

a. Validates the user name and password with TDC2

b. Finds out whether Abdul is an administrator by asking TDC2

c. Enumerates all Abdul's groups by asking TDC2

2.5.5.4 Client Authentication and Windows NT Client Machine

1. Dale, the administrator, logs on to Servr and changes the authentication for the database instance to *Client*:

```
NET stop myinst
DB2 update dbm cfg using authentication client
NET start myinst
```

2. Ivan, at a Windows NT client machine, logs on to the DC1 domain (that is, he is known in the DC1 SAM database).

3. Ivan then connects to a DB2 database that is cataloged to reside on Servr:

```
DB2 CONNECT to remotedb user Ivan using ivanpw
```

4. Ivan's machine validates the user name and password. The API used to find this information first searches the local machine (Ivan) and then the domain controller (DC1) before trying any trusted domains. User name Ivan is found on DC1.

5. Ivan's machine then validates the user name and password with DC1.

6. The DB2 database server, Servr then:

    a. Determines where Ivan is known

    b. Finds out whether Ivan is an administrator by asking DC1

    c. Enumerates all Ivan's groups by asking DC1

2.5.5.5 Client Authentication and Windows 95 Clients

1. Dale, the administrator, logs on to Servr and changes the authentication for the database instance to client:

```
NET stop myinst
DB2 update dbm cfg using authentication client
NET start myinst
```

2. Ivan, at a Windows 95 client machine, logs on to the DC1 domain (that is, he is known in the DC1 SAM database).

3. Ivan then connects to a DB2 database that is cataloged to reside on Servr:

```
DB2 CONNECT to remotedb user Ivan using ivanpw
```

4. Ivan's Windows 95 machine cannot validate the user name and password. The user name and password are therefore assumed to be valid.

5. Servr then:

    a. Determines where Ivan is known

    b. Finds out whether Ivan is an administrator by asking DC1

    c. Enumerates all Ivan's groups by asking DC1

    Note: Because a Windows 95 client cannot validate a given user name and password, client authentication under Windows 95 is not secure. If the Windows 95 machine has access to a Windows NT security provider, however, some measure of security can be imposed by configuring the Windows 95 system for validated pass-through logon. For details on how to configure your Windows 95 system in this way, refer to the Microsoft documentation for Windows 95.

DB2 also supports global groups. In order to use global groups, you must include global groups inside a local group that is on the security server. When DB2 enumerates all the groups that a person is a member of, it also lists the local groups the user is a member of indirectly (by virtue of being in a global group that is itself a member of one or more local groups).

See 2.6.2, “DB2 for NT Security Server Service” on page 62, for more information.

2.6 The DB2 for NT Environment

This section discusses some of the considerations that you need for logging into a DB2 for NT environment, especially the first time the database manager is started. You'll need to understand the restrictions that DB2 imposes on user and group IDs and passwords. Then, you'll need to understand the default Windows NT environment and what to change before logging into DB2.

2.6.1 User ID and Group ID Limitations

This section explains some of the items you need when first logging into a DB2 for NT environment. The first item is selecting user IDs.

User IDs, group Ids and passwords are limited to a maximum of eight characters. There are other restrictions:

- Cannot start with a digit (0 to 9) or end with a dollar sign ($)
- Can be one to eight characters long and may contain the following characters:
  - Upper or lower case characters A to Z
  - Special characters #, @ or $
  - Digits 0 to 9
- Cannot be PUBLIC, USERS, ADMINS, LOCAL or GUESTS, or a name that starts with IBM, SYS or SQL

2.6.1.1 Starting DB2 for NT

This section covers some important details for starting the database manager for the first time in a Windows NT environment.

To start the database manager or create the sample database, you need to log on to NT with an administrator user ID. The Windows NT default administrator (Administrator) being longer than eight characters, will not allow you to install or access DB2 for NT. Another account must be created that conforms to the DB2 user and group ID limitation. This account must also be a member of the administrator group (SYSADM) on the machine in which DB2 for NT will be installed.

Being an administrator means you are part of the group in NT User Manager called Administrators. All members of the NT Administrators group receive SYSADM privileges, by default, until such time as a group is defined in the database manager configuration file for SYSADM_GROUP.

Also, passwords in Windows NT are case-sensitive, although user IDs are not. This can be a common cause for logon failure.

User IDs and group IDs for DB2 can be a combination of uppercase and lowercase characters. However, they are converted to uppercase when used within DB2. For example, if you connect to a database and create the table "schema1.table1," it is stored as SCHEMA1.TABLE1 within the database.

2.6.2 DB2 for NT Security Server Service

The DB2 for NT Security Service is installed as a part of the Client Application Enabler for Windows NT. Recall that the CAE is a common component that is part of the installation of the DB2 Server, SDK and DDCS products. For more information, see 1.2, "DB2 Products and Components for Windows NT" on page 2. The DB2 for NT Security Service is required to authenticate users on the machine on which it is installed. By default, the service is configured to be manually started. This means that unless the service is reconfigured to start automatically (the recommended method), it has to be manually started each time Windows NT is started on the machine.

The service can be manually started (and stopped) in two ways.

1. The first is to enter the following command from a command window:

   ```
   NET start (stop) DB2NTSECSERVER
   ```

2. Open Services within the Control Panel. Select **DB2 for NT Security Service** by clicking on it and then select the **Start (Stop)** button.

To have the service start automatically with Windows NT, open Services within the Control Panel. Select **DB2 for NT Security Service** by clicking on it and then select **Startup.** A Service window will appear. Click on the **Automatic** radio button under Startup type and click on **OK**. Note that you have to be logged onto the machine with an account that is a member of the Administrators group to change the Services configuration.

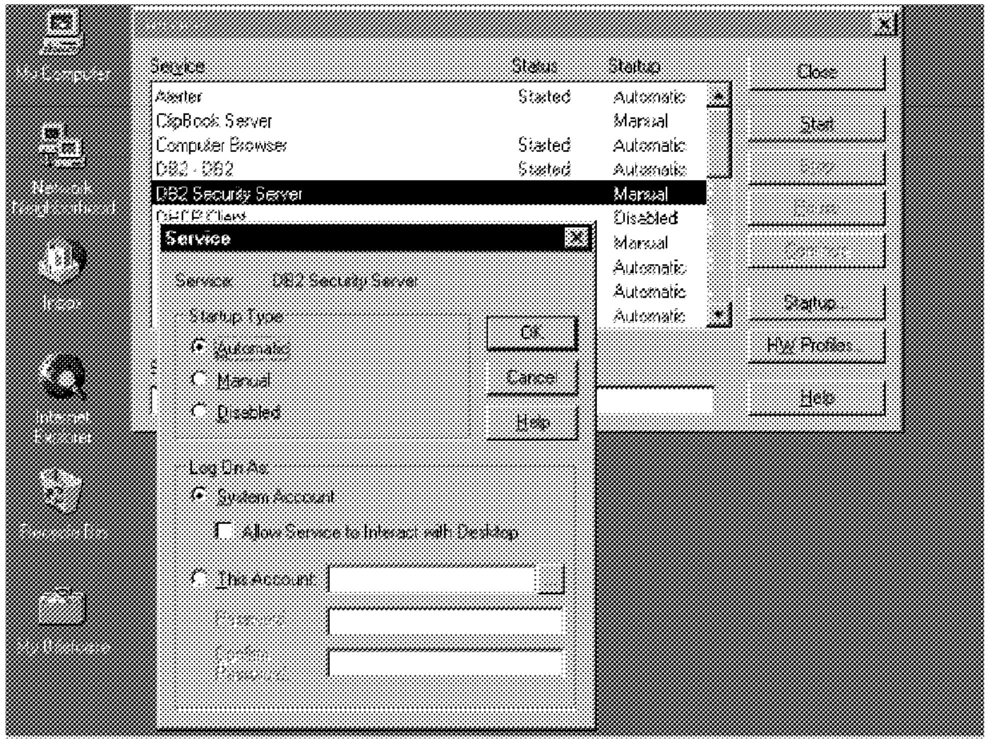


Figure 48. Setting DB2 Security Service to Start Automatically

Notice that in Figure 48 there is an option to Log On As. The default is to log on as System Account. This should not and cannot be changed because the security server requires system privileges.

This service needs to be started on each Windows NT machine, whether NT Workstation or NT Server is being used, where user authentication will occur. It can be run on a client or server machine, but it only needs to be started on a workstation if the database manager configuration file is configured for authentication client.

2.6.2.1 DB2 Instances and Windows NT Services

All DB2 instances should also be configured to start automatically as a service. There will be service for each DB2 instance in the Control Panel | Services. For example, in Figure 49 on page 64, there are two DB2 instances: DB2 (the default instance) and TEST.

Start and stop DB2 instances as a service in DB2 for NT instead of issuing the `DB2START` and `DB2STOP` commands. Not starting DB2 as a service will result in things like DB2 not logging to the Event Viewer logs (For more information on Event Viewer logs, see 6.2.1, “Examining Events in Event Viewer” on page 253).

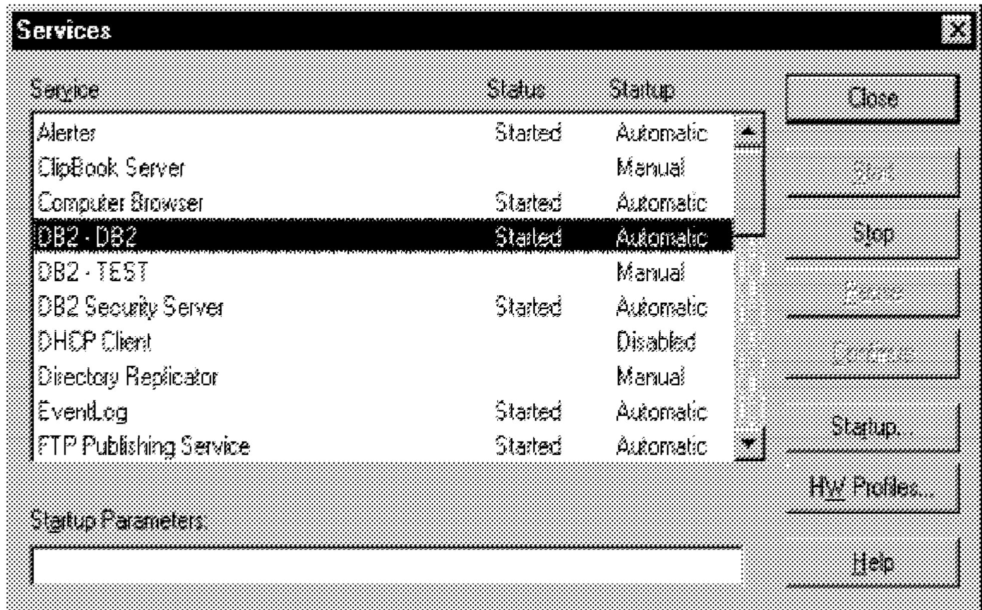


Figure 49. DB2 Instances as Windows NT Services

The DB2 for NT Security Server service will be removed if DB2 is deinstalled using the UNINSTALL.EXE utility.

2.6.3 Planning for DB2 in an NT Environment

This section looks at where you might install DB2 Server in a Windows NT environment and looks in detail at what is required if it is installed on a Backup Domain Controller.

An important point when considering where to install DB2 is the type of authentication that will occur between clients and server. This will partly be driven by the types of clients on the network (see 2.5.5.2, "Remote Database Access" on page 58, for notes on trusted clients). Also consider the types of clients in any trusted domains.

2.6.4 DB2 for NT in a Domain

There are two basic options for installing DB2 for NT in a domain. They are:

1. On a workstation. This server could be NT Server that is not a domain controller or an NT Workstation where Peer-to-Peer services are enabled.
2. On a domain controller. In a Windows NT LAN environment, a user can be authenticated at either a Primary or Backup Domain Controller. This feature is very important in large distributed LANs with one central PDC and one or more BDCs at each site.

2.6.4.1 DB2 for NT on a Workstation

This option could be used with either an authentication client or server. This implementation of DB2 is actually the one used in the examples given for how DB2 authentication occurs in 2.5.5, "Configuring User and Group Security" on page 57.

One of the big advantages of putting DB2 for NT on a dedicated server is that it can be tuned and secured specifically for that task. Also the DB2 application will not have to compete for server resources if it is performing as a domain controller.

On the negative side, regardless of which type of authentication DB2 is configured for, that authentication will have to occur on another server in its own or another domain.

Another disadvantage with this configuration is DB2 for NT's problem with resolving group membership of user accounts. Recall from 2.5.1, "DB2 for NT Group Resolution" on page 50, a user account must be in the same SAM database as any groups to which DB2 has granted authorities in order for those authorities to be applied to the user. Therefore, if DB2 authorities were granted to a group local to a server, then all user accounts must also exist on that server. This defeats the purpose of having a domain.

2.6.4.2 DB2 for NT on a Primary Domain Controller

DB2 could be installed on a Primary Domain Controller. If so, then the default of authentication server should be kept.

The biggest advantage of installing DB2 for NT on a PDC is that authentication can occur on the same machine. Also, there is less chance of encountering difficulties with DB2 for NT's resolutions of group membership for user accounts. Domain accounts can be made members of domain local groups, and DB2 will be able to resolve them.

Implementing DB2 on a PDC also avoids some of the extra configuration that is discussed in the next section, 2.6.4.3, "DB2 for NT on a Backup Domain Controller." However, it is recommended, especially in larger networks, that the Primary Domain Controller be maintained as a dedicated server.

Once installed and configured, DB2 administrators and Windows NT administrators can and should be kept separate. In a larger enterprise or network, it would not be desirable, from a domain administration point of view, to have DB2 administrators logging on and performing DB2 maintenance on the PDC. The opposite is also true from a DB2 viewpoint. You would not want PDC administrators logging into a DB2 instance as SYSADM.

A disadvantage of installing DB2 on a PDC can be poor performance due to heavy utilization of the PDC in a large LAN.

2.6.4.3 DB2 for NT on a Backup Domain Controller

Installing DB2 for NT on a Backup Domain Controller offers all of the same advantages as those for a Primary Domain Controller. Users can be authenticated on the BDC at their site (and, in fact, on the same machine) in a distributed environment instead of requiring a call to the PDC for authentication.

Installing DB2 for NT on a BDC was not addressed in Version 2.1.0 of DB2 for NT because authentication calls had to go to the PDC. With Version 2.1.1 of DB2 for NT, however, authentication can occur at a BDC under the following conditions:

1. The DB2 for NT Server is installed on the BDC.
2. The `DB2DMNBCKCTRL` system environment variable is set appropriately.

If the `DB2DMNBCKCTRL` system environment variable is not set or is set to blank, DB2 for NT performs authentication at the PDC.

If the DB2DMNBCKCTRL system environment variable is set to a question mark (that is, DB2DMNBCKCTRL=?), then DB2 for NT will perform its authentication on the BDC under the following conditions:

1. The Cached Primary Domain under the registry editor is the domain at which the machine is located. (You can find this setting under HKEY_LOCAL_MACHINESoftwareMicrosoftWindows NTCurrent VersionWinLogon.)
2. The Server Manager shows the BDC as active and available.
3. The DB2 for NT server is a BDC on the cached primary domain.

Under normal circumstances the above method of setting DB2DMNBCKCTRL=? will work; however, it will not work in all environments. The information supplied about servers on the domain is dynamic, and the Windows NT Computer Browser must be running to keep this information up-to-date and accurate.

Large LANs may not be running the Windows NT Computer Browser and therefore the information contained in the Server Manager may not be correct. In this case, there is a second method for NT to authenticate at the BDC: by setting DB2DMNBCKCTRL=ABC where ABC is the domain name. With this setting, authentication will occur on the BDC under the following conditions:

1. The Cached Primary Domain under the registry editor is the domain at which the machine is located. (You can find this setting under HKEY_LOCAL_MACHINESoftwareMicrosoftWindows NTCurrent VersionWinLogon.)
2. The machine is configured as a BDC for the specified domain. (If the machine is set up as a BDC but for another domain, the setting will result in an error.)

Chapter 3. Using DB2 Utilities in the NT Environment

This chapter discusses some of DB2 utilities in the Windows NT environment. In this chapter we look at a database object in DB2 called the tablespace. We discuss the kinds of tablespaces, their characteristics and how to create one. We also describe the backup and restore utilities used to safeguard and recover databases in the event of a failure. Moreover, we mention how to move data into a database using the DB2 utilities IMPORT/EXPORT and LOAD. For a complete discussion of all DB2 utilities, refer to the following DB2 manuals: *Administration Guide*, *Command Reference*, *SQL Reference*.

3.1 Tablespaces with DB2 for NT

A database consists of many different kinds of objects. Tablespaces are objects created within databases. Tables are objects created in tablespaces. Tablespaces allow you to control the location of database objects. DB2 supports two different kinds of tablespaces. They are:

System Managed Space (SMS)

The operating system controls the storage space.

Database Managed Space (DMS)

The database manager controls the storage space.

A table holds object data types such as regular data, index, LOB, or long data. These table objects are stored in containers. A container in DB2 is a generic term used to describe the allocation of physical space. DB2 containers in Windows NT can be any of the following.

- Directory
- File
- Device

However, not all platforms support all types of containers.

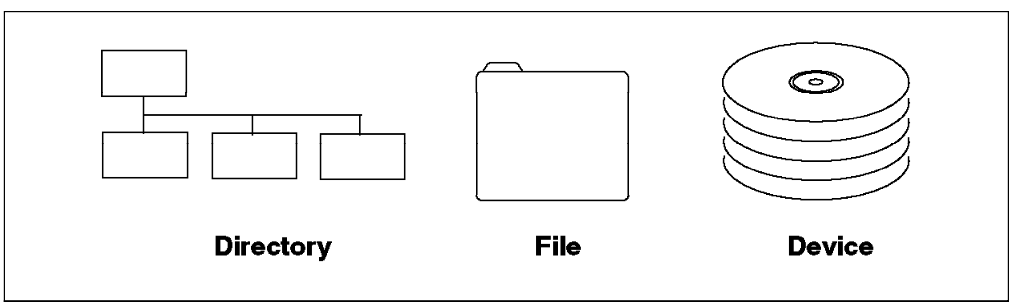


Figure 50. Containers in DB2

For SMS tablespaces, a container can be a directory.

For DMS tablespaces, a container can be a file or a device.

3.1.1.1 Tablespaces and Containers

There is a one-to-many relationship between a tablespace and containers. Multiple containers may be defined for a tablespace. However, a container can only be assigned to one tablespace. Figure 51shows this one-to-many relationship.

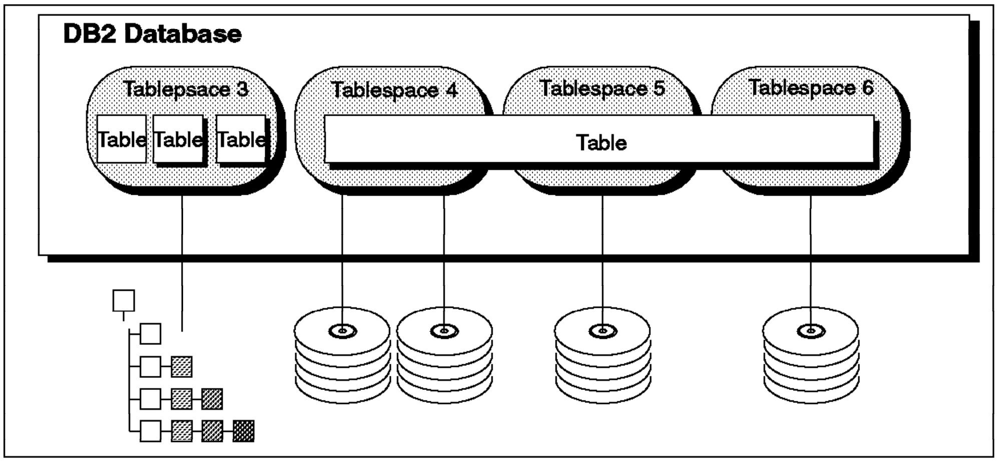


Figure 51. Tablespace and Container, One-to-Many Relationship

Tablespace 3 has only one container assigned to it, a directory. Tablespace 4 has two containers assigned to it. The containers for Tablespace 4, Tablespace 5 and Tablespace 6 are devices. A mixture of containers is possible within a database. You may also mix container types within a tablespace, though it is not recommended for performance reasons. Notice that a table can span multiple tablespaces. In Figure 51, one table spans Tablespace 4, Tablespace 5 and Tablespace 6.

3.1.2 Extents

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

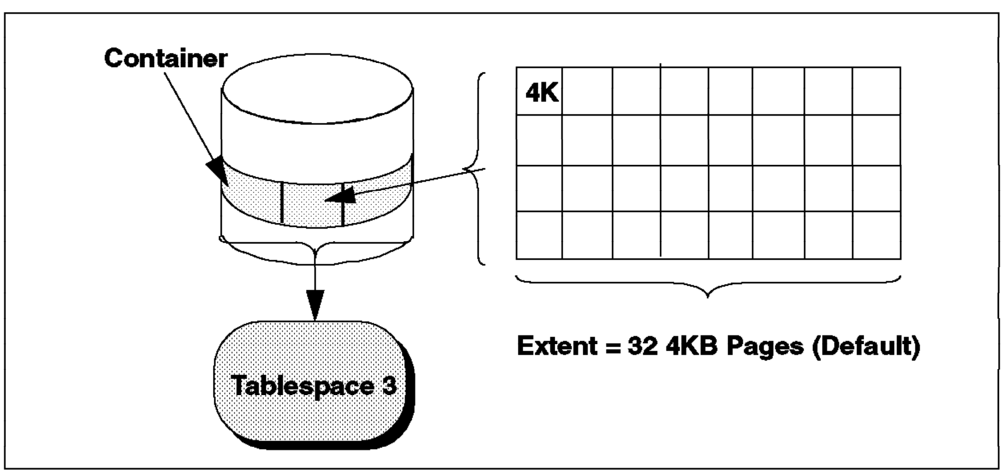


Figure 52. Extent, Container and Tablespace in DB2

Figure 52shows the relationship between an extent, a container and a tablespace. The tablespace is initialized when it is created. As part of this initialization, an allocation size is set for the tablespace. This parameter is called the extent size. The tablespace page size is 4096 bytes (4KB). The default extent size for a tablespace is 32 4KB pages.

3.1.3 SMS Tablespaces

The SMS tablespaces store tables under configured subdirectories. All the data objects are located under the database subdirectory. The database administrator can choose where to create the database subdirectory. The name and location of all the database subdirectories are defined by the database manager. As shown in Figure 53, SMS tablespaces are the default tablespaces when creating a database unless you specify differently. In SMS tablespaces, a table (regular data, index, LOB (long data)) cannot span several tablespaces. You cannot increase the number of containers after creating an SMS tablespace. You can, however, add another SMS tablespace. The total number of containers in SMS tablespaces is determined when the tablespace is created. The containers in SMS tablespaces do not preallocate its storage. There will be some space used during tablespace creation for overhead.

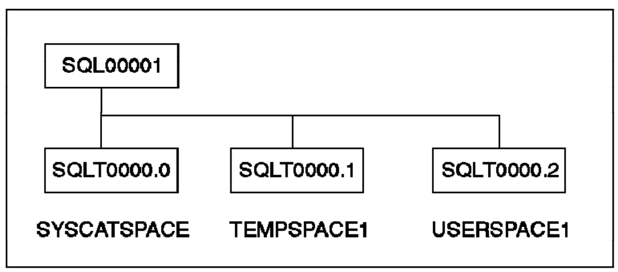


Figure 53. Default SMS Tablespaces

SQLT0000.0 is the directory container that holds the system catalogs. The system catalog tablespace contains all the system catalog tables for the database and cannot be dropped. This tablespace is called SYSCATSPACE.

SQLT0001.0 is the directory container that holds temporary tables that are created and removed during normal processing. This tablespace is called TEMPSPACE1.

SQLT0002.0 is the directory container that holds all user defined tables. This tablespace is named USERSPACE1.

The following files are found within an SMS tablespace directory:

SQLTAG.NAM There is one of these files in each container subdirectory, and they are used by the database manager when you connect to the database to verify that the database is complete and consistent.

SQLxxxxx.DAT Table file. All rows of a table are stored here except for LONG VARCHAR, LONG VARGRAPHIC, CLOB, BLOB, DBCLOB data.

SQLxxxxx.LF File containing LONG VARCHAR or LONG VARGRAPHIC data. This file is only created if LONG VARCHAR or LONG VARGRAPHIC columns exist in the table.

SQLxxxxx.LB Files containing BLOB, CLOB, or DBCLOB data. These files are only created if BLOB, CLOB, or DBCLOB columns exists in the table.

SQLxxxxx.LBA Files containing allocation and free space information about the SQLxxxxx.LB files.

SQLxxxxx.INX Index file for a table. All indexes for the corresponding table are stored in this file. It is only created if indexes have been defined. When an index is dropped, the space is not physically freed from the index file until the index file is deleted. The index file will be deleted if all the indexes on the table are dropped and committed or if the table is reorganized. If the index file is not deleted, the space will be marked free once the drop has been committed, and will be reused for future index creations or index maintenance.

SQLxxxxx.DTR Temporary data file for a REORG of a DAT file. While reorganizing a table, the REORG utility creates a table in one of the temporary tablespaces. These temporary table spaces can be defined to use containers different from those for user-defined tables.

SQLxxxxx.LFR Temporary data file for a REORG of an LF file. Notes for the .DTR file apply here as well.

SQLxxxxx.LXR Temporary data file for a REORG of an INX (Index) file. Notes for the .DTR file apply here as well.

SQLxxxxx.RLB Temporary data file for a REORG of an LB file. Notes for the .DTR file apply here as well.

SQLxxxxx.RBA Temporary data file for a REORG of an LBA file. Notes for the .DTR file apply here as well.

3.1.4 Creating SMS Tablespaces

This section shows an example of creating an SMS tablespace and a table in a database server on Windows NT. The following SQL statements create an SMS tablespace using two directories on two separate drives. The table is then created in the tablespace. The table will have a unique index.

The following SQL statement will create an SMS tablespace for holding either regular data, index or long data.

```
CREATE REGULAR TABLESPACE TABLESPACE1 MANAGED BY SYSTEM USING ('E:TBSP1_1',
'F:\TBSP1_2' )
```

The following SQL statement will create a table in TABLESPACE1.

```
CREATE TABLE TABLE1 (COL1 INTEGER NOT NULL, COL2 BLOB(100K)) IN TABLESPACE1
INDEX IN TABLESPACE1 LONG IN TABLESPACE1
```

The following SQL statement will create a unique index on the table.

```
CREATE UNIQUE INDEX INDEX1 ON TABLE1 (COL1)
```

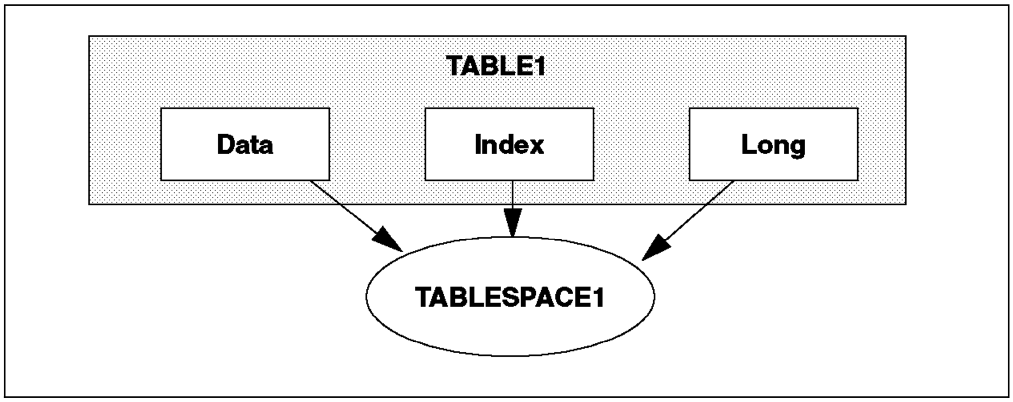


Figure 54. SMS Tablespace Example

As illustrated on Figure 54, all object data types of table, regular data, index, LOB (long data) must be in one SMS tablespace. The table cannot span several tablespaces. Notice that the containers for the SMS tablespace are directories.

3.1.5 DMS Tablespaces

DMS tablespaces are created on preallocated disk partitions or files. This preallocated space can be a file or a device, which is called a container. If the container is a device, it has to be prepared before issuing CREATE TABLESPACE or ALTER TABLESPACE command. The database manager has a responsibility for the management of this space. The table objects for the regular data, indexes and LOB (long data) columns of a table can be stored in the same tablespace or in different tablespaces. That is, in contrast with SMS tablespaces, in DMS tablespaces, a table (regular data, index, LOB (long data)) can span several tablespaces. The size of a DMS tablespace can be increased by adding containers after the tablespace is created.

3.1.6 Creating DMS Tablespaces Using File Containers

This section shows an example of creating DMS file tablespaces and a table in a database on a Windows NT database server. The following SQL statements create three DMS tablespaces, TABLESPACE1 for regular data, TABLESPACE2 for its index, and TABLESPACE3 for LOB (long data)). All tablespaces will use file containers. The table that is then created in those three tablespaces will have a unique index.

The following SQL statement will create a DMS file tablespace for holding regular data only:

```
CREATE REGULAR TABLESPACE TABLESPACE1 MANAGED BY DATABASE USING (FILE
'E:\TBSP1_1\TBSP1_1' 5000)
```

The following SQL statement will create a DMS file tablespace for holding only an index:

```
CREATE REGULAR TABLESPACE TABLESPACE2 MANAGED BY DATABASE USING (FILE
'E:\TBSP2_1\TBSP2_1' 5000)
```

The following SQL statement will create a DMS file tablespace for holding LOB (long data) only:

```
CREATE LONG TABLESPACE TABLESPACE3 MANAGED BY DATABASE USING (FILE
'E:\TBSP3_1\TBSP3_1' 5000)
```

The following SQL statement will create a table with regular data in TABLESPACE1, an index in TABLESPACE2 and LOB (long data) in TABLESPACE3.

```
CREATE TABLE TABLE1 (COL1 INTEGER NOT NULL, COL2 BLOB(100K)) IN TABLESPACE1
INDEX IN TABLESPACE2 LONG IN TABLESPACE3
```

The following SQL statement will create a unique index on the table.

```
CREATE UNIQUE INDEX INDEX1 ON TABLE1 (COL1)
```

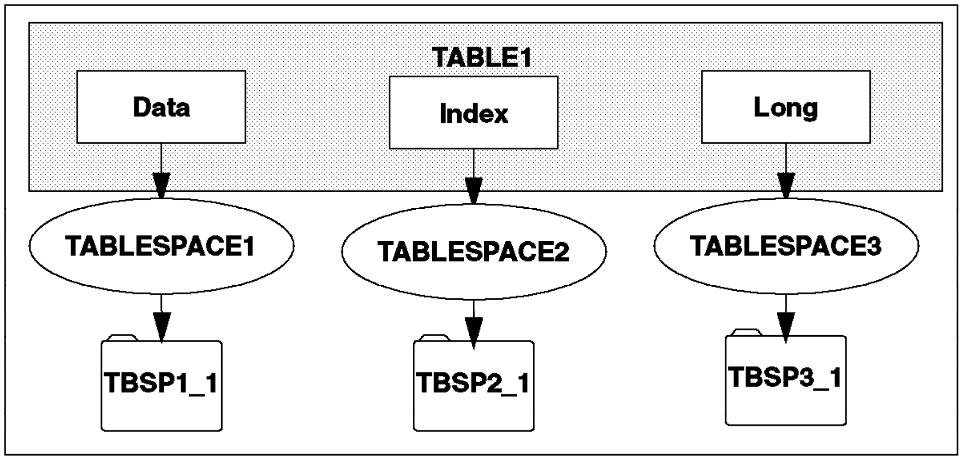


Figure 55. DMS File Tablespaces Example

As illustrated on Figure 55, the table, TABLE1, can span three DMS tablespaces.

3.1.7 Creating DMS Tablespaces Using Device Containers

DB2 for Windows NT supports direct disk access (RAW I/O). This allows you to attach a raw device to a Windows NT system. There are two methods for using RAW I/O. One is using a physical hard drive; the other is using a logical raw partition. The former, using the physical hard drive, allows you to allocate an entire hard drive exclusively for a DMS tablespace. The latter, using the logical raw partition, allows you to allocate only a portion of the hard drive, for instance a logical raw partition, for a DMS tablespace.

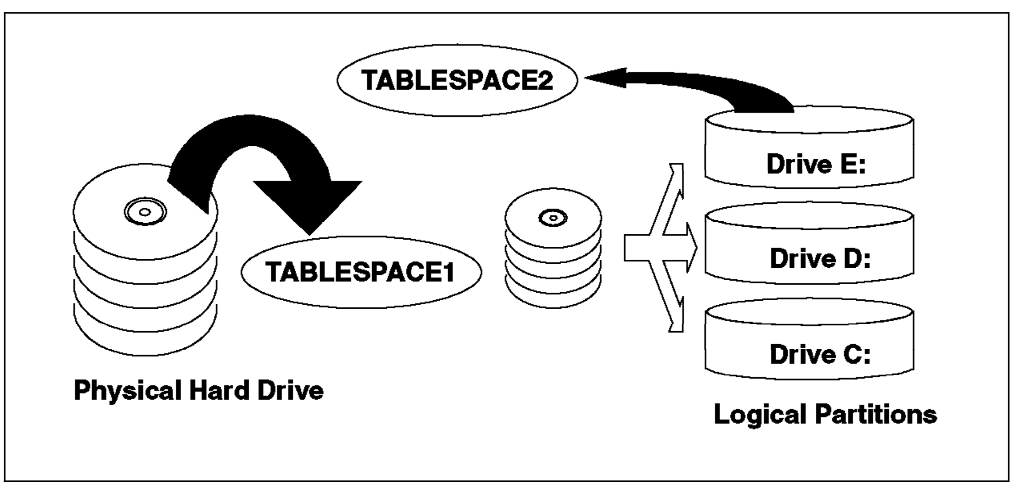


Figure 56. Physical Hard Drive and Logical Raw Partition

In Figure 56, TABLESPACE1 uses the whole physical hard drive, whereas TABLESPACE2 uses only a part of physical hard drive, which is logical drive E:.

When creating DMS device tablespaces, you must pay attention not to use physical devices or logical drives that have been already allocated for use by the operating system.

3.1.7.1 Preparing Disk for DMS Device Tablespaces

One of the tools in Windows NT, the Disk Administrator, is helpful in obtaining the partition information of physical hard drives and logical drives used by your system.

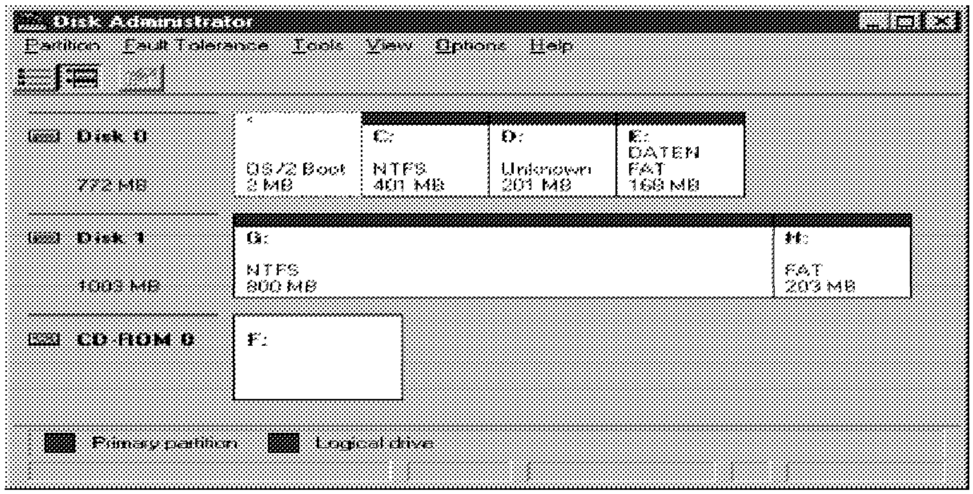


Figure 57. Disk Administrator

Figure 57 shows you that there are two physical hard drives (Disk 0 and Disk 1), one CD-ROM and six logical drives (C:, D:, E:, F:, G:, H:) in the system.

To open a physical hard drive for direct disk access, use the following device naming convention:

.PhysicalDrive*N*

where *N* represents each of the physical drives (0, 1, 2 and continuing) in the system. For example, when physical drive1 is used, .PhysicalDrive1.

To open a logical raw partition, (A raw partition must not be formatted.) use the following:

.*M*:

where *M*: represents the logical drive letter (C:, D:, E: and so on) in the system. For example, when drive E: is used, .E:.

There are advantages and disadvantages when you use raw devices in your I/O configuration:

The advantages are as follows:

- You can attach more than 26 physical drives to system.
- The file I/O pathlength is shorter and less physical I/O occurs since there is no FILE METADATA (such as date, time of file creation, file owner) to be

updated. You should use benchmarks to evaluate if there are measurable benefits for your work load when using raw device instead of files within the file system, such as the NT File System (NTFS).

The disadvantages are as follows:

- The device cannot be shared by other applications. The entire device must be assigned to DB2.
- The device cannot be operated upon by any operating system utility or third-party tool, such as backup or copy.
- You can easily wipe out the file system on an existing drive if you specify the wrong physical drive number.

3.1.7.2 Creating DMS Tablespace (Physical Device)

This section shows an example of creating DMS device tablespaces (Physical Device) and a table in a Windows NT database server. The following SQL statements create three DMS device tablespaces (TABLESPACE1 for regular data, TABLESPACE2 for index, and TABLESPACE3 for LOB (long data)) using physical devices. The table is then created in those tablespaces. It will have a unique index.

The following SQL statement will create a DMS device tablespace (physical device) for holding only regular data.

```
CREATE REGULAR TABLESPACE TABLESPACE1 MANAGED BY DATABASE USING (DEVICE
'\\.\PhysicalDrive1' 100000)
```

The following SQL statement will create a DMS device tablespace (physical device) for holding only an index.

```
CREATE REGULAR TABLESPACE TABLESPACE2 MANAGED BY DATABASE USING (DEVICE
'\\.\PhysicalDrive2' 100000)
```

The following SQL statement will create a DMS device tablespace (physical device) for holding only LOB (long data).

```
CREATE LONG TABLESPACE TABLESPACE3 MANAGED BY DATABASE USING (DEVICE
'\\.\PhysicalDrive3' 100000)
```

The following SQL statement will create a table with regular data in TABLESPACE1 and index in TABLESPACE2 and LOB (long data) in TABLESPACE3.

```
CREATE TABLE TABLE1 (COL1 INTEGER NOT NULL, COL2 BLOB(100K)) IN TABLESPACE1
INDEX IN TABLESPACE2 LONG IN TABLESPACE3
```

The following SQL statement will create a unique index on the table.

```
CREATE UNIQUE INDEX INDEX1 ON TABLE1 (COL1)
```

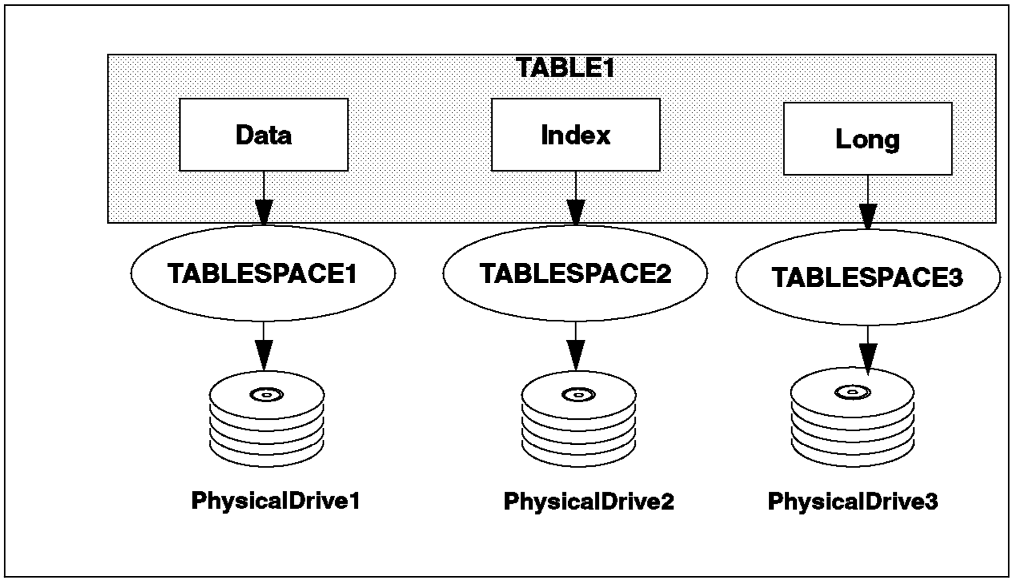


Figure 58. DMS Physical Device Tablespaces Sample

As illustrated on Figure 58, the table can span three tablespaces on separate physical devices.

Note: If using physical device numbers, the entire physical hard drive must be allocated for only one DMS device tablespace. No other tablespace nor file system can coexist on the physical hard drive.

3.1.7.3 Creating DMS Tablespace (Logical Drive)

This section shows an example of creating DMS device tablespaces (logical drive) and a table in a database on a Windows NT database server. The following SQL statements will create three DMS tablespaces (TABLESPACE1 for regular data, TABLESPACE2 for index, TABLESPACE3 for LOB (long data)) using logical devices. The table will then be created in those tablespaces. The table will have a unique index.

The following SQL statement will create a DMS device tablespace (logical drive) for holding regular data only:

```
CREATE REGULAR TABLESPACE TABLESPACE1 MANAGED BY DATABASE USING (DEVICE '.D'
50000)
```

The following SQL statement will create a DMS device tablespace (logical drive) for holding index data.

```
CREATE REGULAR TABLESPACE TABLESPACE2 MANAGED BY DATABASE USING (DEVICE '.E'
50000)
```

The following SQL statement will create a DMS device tablespace (logical drive) for holding exclusively LOB (long data):

```
CREATE LONG TABLESPACE TABLESPACE3 MANAGED BY DATABASE USING (DEVICE '.F'
50000)
```

The following SQL statement will create a table with regular data in TABLESPACE1 and an index in TABLESPACE2 and a LOB (long data) in TABLESPACE3.

```
CREATE TABLE TABLE1 (COL1 INTEGER NOT NULL, COL2 BLOB(100K)) IN TABLESPACE1
INDEX IN TABLESPACE2 LONG IN TABLESPACE3
```

The following SQL statement will create a unique index on the table.

```
CREATE UNIQUE INDEX INDEX1 ON TABLE1 (COL1)
```

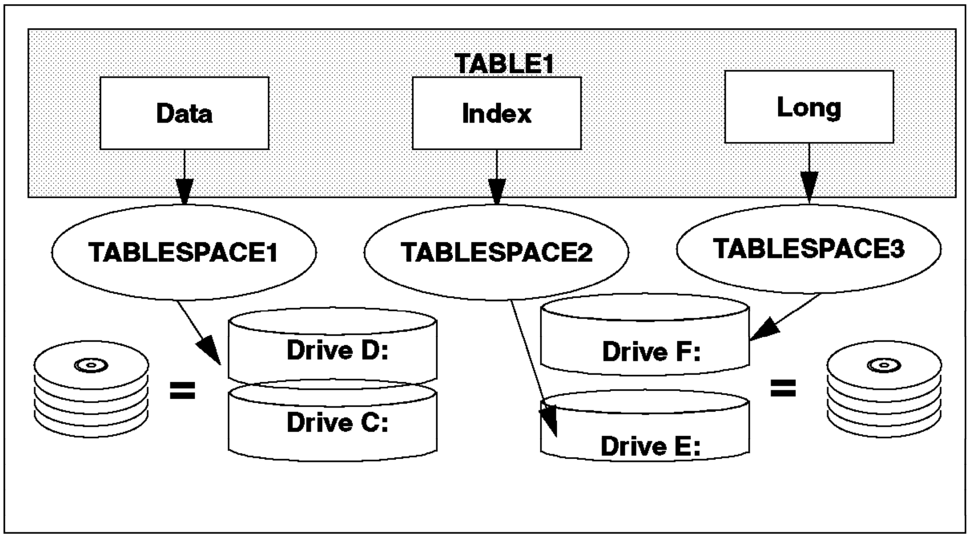


Figure 59. DMS Logical Device Tablespaces Example

As illustrated on Figure 59, the table can span three tablespaces on separate logical drives.

3.1.7.4 Increasing Size of DMS Tablespaces

You can increase the size of a DMS tablespace by adding one or more containers to the tablespace. This section shows an example of adding a container by issuing ALTER TABLESPACE statement. Once the new container has been added and the transaction is committed, the contents of the tablespace are asynchronously rebalanced across all containers.

The following SQL statement will create a DMS device tablespace (physical device) for holding regular data on a raw device:

```
CREATE REGULAR TABLESPACE TABLESPACE1 MANAGED BY DATABASE USING (DEVICE
'\\.\PhysicalDrive1' 100000)
```

The following SQL statement will add a device container to TABLESPACE1:

```
ALTER TABLESPACE TABLESPACE1 ADD (DEVICE '.PhysicalDrive2' 100000)
```

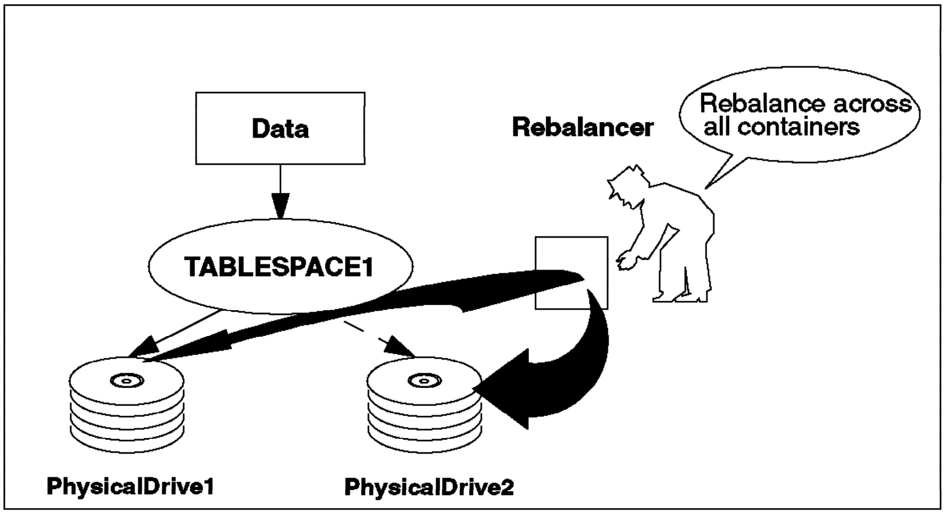


Figure 60. Increasing the Size of DMS Tablespaces

The following points should be considered when changing the size of a DMS tablespace:

1. If you need to add more than one container, it is recommended that they be added at the same time (either in the same ALTER TABLESPACE statement, or within the same transaction) so that the cost of rebalancing is incurred only once.

2. While a rebalance of a tablespace is in progress, if you attempt to add more containers to the tablespace, you will receive an error (SQL0258N, SQLCODE=-258, or SQLSTATE=55041). You need to wait until the rebalance has completed.

3.1.8 Choosing between SMS or DMS Tablespaces

When creating a tablespace or a database, you can choose the type of tablespace (SMS or DMS). When creating a database, unless you specify the type of tablespaces, your tablespaces (SYSCATSPACE, TEMPSPACE1, USERSPACE1) will be SMS tablespaces. Once you create a tablespace, you cannot change the type of tablespace (from SMS to DMS, from DMS to SMS).

There are a lot of points to consider when determining which type of tablespace you should use to store your table data. For the system catalog tablespace (SYSCATSPACE), an SMS tablespace is recommended unless the database will be very large. (Here, very large is defined as a file that exceeds the size limitations imposed by the operating system.) From an administrative point, SMS tablespaces are easier to maintain. For temporary tablespaces (TEMPSPACE1), SMS tablespace offers the best choice. The size of the temporary tablespace can peak to a high value while sorting large tables. Its peak size is difficult to predict, and if using DMS, you would have to allocate a larger amount of space that you would not be able to recover. The following table shows points briefly to help you determine whether your user tablespaces should be SMS or DMS. You may even decide that a mix of tablespaces, some of them SMS tablespaces and some of them DMS tablespaces, is the best alternative for your database.

Table 4. DMS vs. SMS Tablespace Considerations

Points	SMS Tablespaces	DMS Tablespaces
Space is not allocated by the system until it is required	Yes	No
Increase number of containers in tablespace	No	Yes
Store index data in separate tablespaces	No	Yes
Store LOB(long data) in separate tablespaces	No	Yes
One table (regular data, index, LOB (long data) can span several tablespaces	No	Yes
Containers can be a raw device	No	Yes

3.1.9 Tablespace Considerations

There are some additional considerations to keep in mind when creating tablespaces.

3.1.9.1 Performance Considerations

If you are using Database Managed Space (DMS) device containers for your tablespaces, you need to understand the following items so you can effectively administer your environment:

- Buffering of data

  For tablespaces using System Managed Space (SMS), when the database manager subsequently requests the page from the file system, the file system may still have that page in its own space. This can eliminate I/O operation that would otherwise have been required.

- Tablespaces using Database Managed Space (DMS) device do not use the Windows NT file system nor its cache.

  As a result, you may wish to reduce the size of the file system cache. This will allow you to increase the size of the buffer pool for added performance.

3.1.9.2 Error Messages with Containers in Use (SQL0294N)

An error (SQL0294N, SQLCODE=-294, SQLSTATE=42730) can be returned from the create database or create or alter tablespace operations. Normally, this situation indicates a specification error on the operating system resource name since, apparently, the container is already in use by another tablespace. If you are a system administrator or database administrator who finds that the database that had last used the containers has been deleted, yet somehow the container's tag was not removed, you can use the db2untag tool. This tool will display a container tag's information so that you can check the database to which the container belongs. If you decide to release the container, you should also take the following steps:

- For SMS containers: Remove the directory and contents using the usual operating system file delete command.
- For DMS containers: Either delete the file or device or use the db2untag utility to remove the container tag. The db2untag tool does not remove the DMS container.

3.2 Backup and Restore Utilities

In this section, backing up and restoring utilities are discussed. For those utilities, DB2 for Windows NT provides the BACKUP DATABASE command and the RESTORE DATABASE and ROLLFORWARD DATABASE commands. The backup, restore and rollforward operations are not performed at the operating system level.

The BACKUP DATABASE command can be used on two levels:

- Database-Level Backup - Backup each database on a regular basis. The backup may be directed to fixed disk, tape, ADSM, or other vendor products enabled for DB2. It can be online or offline. An online backup can be performed if rollforward recovery is enabled. To enable rollforward recovery, you must change the database configuration parameters LOGRETAIN, USEREXIT, or both, setting them from OFF to ON.
- Tablespace-Level Backup - Tablespace-level backup contains one or more tablespaces for a database. It can be take online or offline. To ensure that the restored tablespaces are synchronized with the rest of the database, tablespaces must be rolled forward to the end of logs. For this reason, a tablespace-level backup and restore can be performed only if rollforward recovery is enabled. To enable rollforward recovery, you must change the database configuration parameters LOGRETAIN, USEREXIT, or both from OFF to ON.

  The RESTORE DATABASE command can be used on two levels:

- Database-Level Restore - A database-level restore will rebuild the entire database by using a copy or image made previously and using the BACKUP DATABASE command.
- Tablespace-Level Restore - A tablespace restore, the most recent image of the database was made using the BACKUP DATABASE command and specifying only one or more tablespaces to be backed up.

  The ROLLFORWARD DATABASE command can be used in two modes:

- Point in time - If performing a database-level recovery with archive logging (LOGRETAIN or/and USEREXIT enabled), you can specify the point in time to which all committed transactions are to be rolled back.

  Note the following regarding point-in-time rollforward:

  – There is no point-in-time recovery for tablespace recovery.

  – The format of point-in-time is *yyyy-mm-dd-hh.mm.ss.nnnnnn* (year, month, day, hour, minutes, seconds, microseconds), which is Coordinated Universal Time (CUT).

- End of Logs - All committed transactions from the archive log files are to be applied.

  Note: A loss of an archive log file causes the end-of-logs recovery to fail. This is fatal for tablespace-level recovery.

3.2.1 USEREXIT and LOGRETAIN Parameters

As mentioned in the previous section, if you want to perform a tablespace-level backup or a online backup of a tablespace or database, the USEREXIT parameter or LOGRETAIN parameter, or both, must be enabled.

To enable these parameters, you should issue the UPDATE DATABASE command as follows:

```
UPDATE DATABASE CONFIGURATION FOR database_alias USING LOGREATIN ON
UPDATE DATABASE CONFIGURATION FOR database_alias USING USEREXIT ON
```

The following should be noted about LOGRETAIN and USEREXIT:

1. After enabling LOGRETAIN, USEREXIT or both, you have to take an offline database-level backup.
2. You need SYSADM , SYSCTRL, or SYSMAINT authorization to issue the UPDATE DATABASE command.

The following table shows the relationship among values of LOGRETAIN and USEREXIT parameters, database logging, backup levels, and types that you can choose, and rollforward modes.

Table 5. USEREXIT and LOGRETAIN Parameters

LOGRETAIN (OFF/ON)	USEREXIT (OFF/ON)	Type of Logging	Type of Backup Permitted	Roll forward allowed (Yes or No, If Yes, specify roll forward time for logs)	Description
OFF	OFF	Circular logging	Database offline	No	Default (LOGRETAIN=OFF,USEREXIT=OFF)
ON	OFF	Archive logging	Database online	Yes, roll forward to point in time/end of logs	Log files remain in SQLOGDIR, unless manually moved.
			Database offline	Yes, roll forward to point in time/end of logs	
			Tablespace offline	Yes, roll forward to end of logs	
			Tablespace online	Yes, roll forward to end of logs	
ON	ON	Archive logging	Database online	Yes, roll forward to point in time/end of logs	Log files in SQLOGDIR directory archived by db2uexit
			Database offline	Yes, roll forward to point in time/end of logs	
			Tablespace online	Yes, roll forward to end of logs	
			Tablespace offline	Yes, roll forward to end of logs	

3.2.2 Backup File Format

If you perform a database or tablespace backup to a Windows NT file system, the backup file is placed in a three-level subdirectory tree as follows:

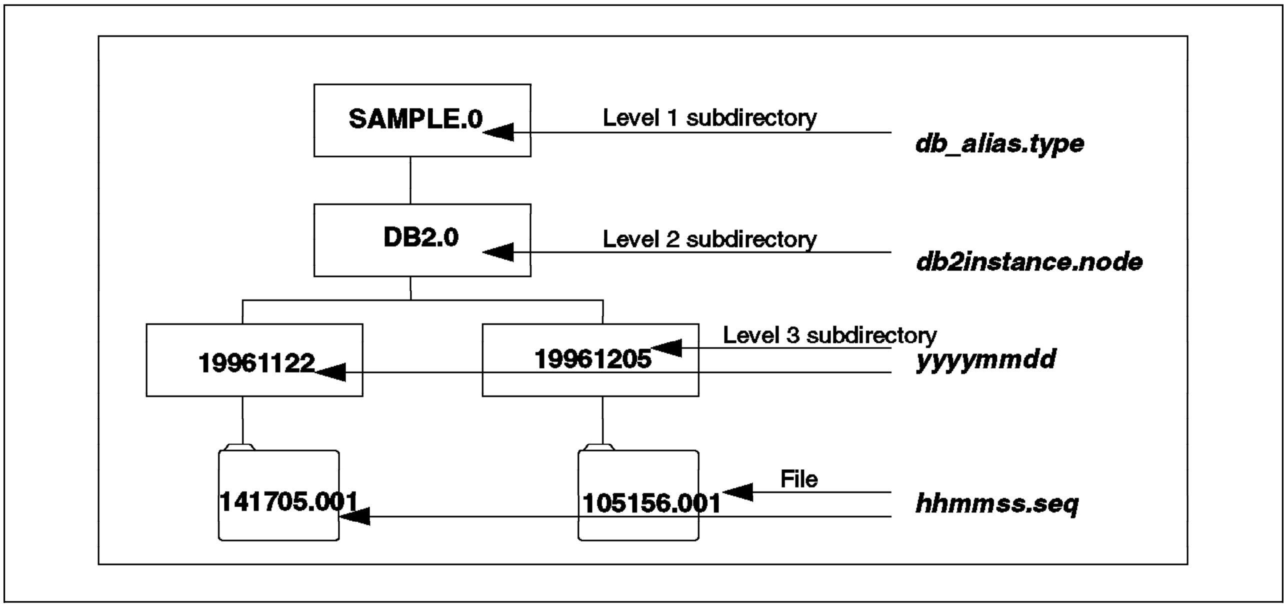


Figure 61. Backup File Format Example

The format of the backup file shown in Figure 61 can be explained as follows:

db_alias.typedb2instance.nodeyyyymmddhhmmss.seq

db_alias	1 to 8 character database alias.
type	Type of backup taken. **0** for full database backup, **3** for tablespace level backup, **4** for copy from a table load.
db2instance	1 to 8 character database instance name.
node	Reserved for future use.
yyyymmdd	Date(year month day).
hhmmss	Time(hour minute second).
seq	A file extension consisting of a 3-digit sequence number.

Note that tape images for backups are not named, but contain the same information in the backup header for verification purposes.

In Figure 61, there are two database-level backup files for the database alias **SAMPLE** of the instance **DB2**. One of them is taken at 14:17:05 on 11/22/1996, which is placed in SAMPLE.0DB2.019961122141705.001. Another one is done at 10:51:56 on 12/05/1996, which is placed in SAMPLE.0DB2.019961205105156.001.

Backup history provides key information in an easy-to-use format and is kept in the recovery history file. The recovery history file is updated automatically with summary information whenever you perform a backup or restore of a full database or tablespace, or load to a table. The contents of the history file are useful to support management of the backup images. This information includes:

- The part of the database that has been copied by a backup, load or copy operation.
- When the database was copied.
- The location of the copy.
- Time of the last restore.

3.2.3 DB2 for Windows NT Tape Support

DB2 for Windows NT supports backup and restore to streaming tape devices. The following DB2 command line processor commands can be used for tape initialization and positioning. These commands are supported for DB2 for Windows NT only.

```
REWIND TAPE [ON device]
SET TAPE POSITION [ON device] TO position
INITIALIZE TAPE [ON device] [USING blockingfactor]
```

device This must be a valid NT tape device name. For example, .TAPE0 is the first tape device. This parameter is optional and defaults to the first device, TAPE0.

position This is the tape mark position. DB2 for Windows NT supports the writing of TAPEMARKS and SETMARKS. DB2 writes a tape mark after every backup. Thus, position 1 is the tape mark after the first archive. Position 2 is the tape mark after the second archive, and so forth. If the tape is positioned at tape mark 1, archive 2 is positioned to be restored.

blockingfactor This is the block size to be used for the device. It is checked against the range of blocking sizes valid for the device. It must also be a proper factor of, or a multiple of 4096. This parameter is optional and defaults to the default blocking factor for the hardware you are using.

The following should be noted for DB2 for Windows NT tape support:

1. We recommend a blocking factor of 4096 and a buffer size of 16 for both backup and restore.
2. Initializing tape, rewinding tape and setting tape position are supported only by using Command Line Processor. The Database Director does not support these functions.

3.2.3.1 Backing Up To Tape

This section contains sample scenarios for backing up DB2 databases to tape.

In backing up the first database, the following command initializes a tape with block size 4096.

```
INITIALIZE TAPE ON .TAPE0 USING 4096
```

DB2 for Windows NT opens the tape, rewinds it, and queries the device to see if the blocking factor is acceptable.

Assuming that the tape is rewound, if the blocking factor had not been specified, DB2 for Windows NT would have queried the default block size of the device and used that.

The following command takes the backup of the first database, NTDB1.

```
BACKUP DATABASE NTDB1 TO .TAPE0 BUFFER 16
```

Taking the defaults for the block size, the backup is performed to tape device 0. At close time, a tape mark is written, and the tape will not be rewound (you may use the REWIND TAPE command).

In backing up the second database, the following command positions the tape to the mark after the first archive.

```
SET TAPE POSITION ON .TAPE0 TO 1
```

This positions the tape to the first tape mark after the first archive. DB2 for Windows NT determines whether the device supports direct positioning. If not, the tape will be rewound first and then positioned.

The following command takes the backup of the second database, NTDB2.

```
BACKUP DATABASE NTDB2 TO .TAPE0 BUFFER 16
```

The following items should be noted when backing up to tape on Windows NT:

1. During the backup operation, if there is an overflow situation such that the backup image is going to span over multiple tapes, DB2 prompts for a new tape unless you specify the WITHOUT PROMPTING option for use in the BACKUP DATABASE command.
2. After changing the tape during backup or restore operations, allow enough time for the tape device to reinitialize before entering c to continue the backup or restore. This time will vary for different tape devices. This hardware initialization may take as long as three to five minutes. If this is not done properly, the backup or restore will not complete successfully.
3. You cannot partially erase a tape if you have multiple backup images on a tape. If you want to reuse the tape at all, the entire tape must be erased.
4. We recommend putting only one backup image on a tape cartridge. Therefore, if a backup fails for any reason, you can erase the entire tape and start again.

3.2.3.2 Restoring from Tape

This section contains sample scenarios for restoring DB2 database backups from tape.

In restoring the second archive, the following command positions the tape to the second archive.

```
SET TAPE POSITION FOR .TAPE0 TO1
```

The following command restores the second database, NTDB2.

```
RESTORE DATABASE NTDB2 ON .TAPE0 BUFFER 16
```

In restoring the first archive, the following command rewinds the tape.

```
REWIND TAPE ON .TAPE0
```

The following command restores the first database, NTDB1.

```
RESTORE DATABASE NTDB1 ON.TAPE0 BUFFER 16
```

The following items can be noted when performing a restore from tape on DB2 for NT:

1. The tape is assumed to be positioned. The device is opened and read until the end of file (the next tapemark) is detected. The tape is rewound and ejected if the device supports it.

2. During restore, when there is an overflow situation such that the backup image spans over multiple tapes, DB2 prompts for a new tape unless you specify the `WITHOUT PROMPTING` option for `RESTORE DATABASE` command.

3.2.4 Database Level Backup

This section discusses taking a database level backup. There are two kinds of backups. One of them is an offline backup; the other is an online backup.

3.2.4.1 Database-Level Backup (Offline)

A database-level backup (offline) provides you with a complete snapshot of the data at a fixed time. You can take the database-level backup (offline) as long as the `BACKUP DATABASE` command can get exclusive access to the database. The parameters `LOGRETAIN`, `USEREXIT` may be either `ON` or `OFF`. During the database-level backup (offline), any attempt to access the database by a DB2 application or an user results in an error (SQL1035N, SQLCODE=-1035, SQLSTATE=57019). Also, if there is a DB2 application or a user accessing the database while the database-level backup (offline) is initializing, the backup will fail.

The advantages of taking the database-level backup (offline) are the following:

- You can recover the database from the database-level backup (offline) with or without archive log files. If there are archive log files, you may use the `RESTORE DATABASE` command and not specify the `WITHOUT ROLLING FORWARD` option. This will avoid having to perform a roll forward using the archive log files. This option may be useful if you unfortunately lose an archive log file but need to recover the database.

- You can create a new database from the database-level backup.

The disadvantages of taking the database-level backup (offline) are the following:

- Taking the database-level backup (offline) uses the database in exclusive mode. No other application or user can access the database during the backup.

- As long as there is an active application or a user connecting to the database, the backup operation fails. You may use the `FORCE APPLICATION` command to forcibly remove the applications before beginning the backup.

- The database-level backup may need a longer time to complete than a tablespace-level backup.

The following command takes a database-level backup (offline) to drive C:.

```
BACKUP DATABASE NTDB1 TO C:
```

3.2.4.2 Database-Level Backup (Online)

If either `LOGRETAIN`, `USEREXIT` or both is enabled (that is, roll forward recovery is enabled), a database-level backup (online) can be performed. In contrast with the database-level backup (offline) mentioned in the previous section, during the database-level backup (online), access to the database by DB2 applications or other users is permitted. When restoring from the database-level backup (online), the data is not consistent until the archive logs are applied during roll forward recovery, through the point in time where the backup completed.

The advantages of taking the database level backup (online) as follows:

- An application or a user can continue to connect to the database while the database-level backup (online) task is running.
- You can create a new database from the database-level backup.

The disadvantages of taking the database-level backup (online) as follows:

- After restoring from the database-level backup (online), you must execute the ROLLFORWARD DATABASE command to apply the archive logs. This means that you must maintain archive log files as well as the backup image.

The following command takes a database-level backup (online) to drive C: specifying ONLINE option.

```
BACKUP DATABASE NTDB1 ONLINE TO C:
```

3.2.5 Tablespace-Level Backup

This section discusses taking a tablespace-level backup. Tablespace-level backups can be performed either offline or online.

3.2.5.1 Tablespace-Level Backup (Offline)

If LOGRETAIN, USEREXIT, or both are enabled, you can perform a tablespace-level backup (offline). This will reduce the time when the database is not available. During the tablespace-level backup (offline), accessing the database by a DB2 application or a user results in an error (SQL1035N, SQLCODE=-1035, SQLSTATE=57019). Also, if there is a DB2 application or a user accessing the database while you are beginning the tablespace-level backup (offline), an error will result. The tablespace-level backup can include a single tablespace or multiple tablespaces. However, restore is not selective. You must restore all the tablespaces in the image.

The advantages of taking the tablespace-level backup (offline) are the following:

- Except for restoring the tablespace containing the system catalogs, if you invoke the RESTORE DATABASE command with the TABLESPACE ONLINE option, tablespaces can be restored online.
- Except for rolling forward the tablespace containing the system catalogs, if the ROLLFORWARD DATABASE command is invoked with the TABLESPACE ONLINE option, tablespaces can be rolled forward online.

The disadvantages of taking the tablespace-level backup (offline) are the following:

- During the tablespace-level backup (offline), no other application or user can use the database.
- You cannot create a new database from the tablespace-level backup.
- There is no point-in-time recovery. After restoring from the tablespace-level backup (offline), you must execute the ROLLFORWARD DATABASE command with the END OF LOGS option.
- A loss of an archive log file causes the tablespace recovery to fail.

The following command takes a tablespace-level backup (offline) to drive C:.

```
BACKUP DATABASE NTDB1 TABLESPACE USERSPACE1 TO C:
```

3.2.5.2 Tablespace-Level Backup (Online)

If LOGRETAIN and/or USEREXIT is enabled, you can perform a tablespace-level backup (online) instead of a database-level backup to reduce the time needed for the backup. The tablespace-level backup can include a single tablespace or multiple tablespaces. However, restore is not selective.

The advantages of taking a tablespace-level backup (online) are the following:

- During taking the tablespace-level backup (online), other applications and users can use the database.
- Except for restoring the tablespace containing the system catalogs, if you invoke the RESTORE DATABASE command with the TABLESPACE ONLINE option, tablespaces can be restored online.
- Except for rolling forward the tablespace containing the system catalogs, if the ROLLFORWARD DATABASE command is invoked with the TABLESPACE ONLINE option, tablespaces can be rolled forward online.

The disadvantages of taking the tablespace-level backup (online) are the following:

- You cannot create a new database from the tablespace-level backup.
- There is no point-in-time recovery. After restoring from the tablespace level backup (online), you must execute the ROLLFORWARD DATABASE command with the END OF LOGS option.
- A loss of an archive log file causes the tablespace recovery to fail.

The following command takes a tablespace-level backup (online) to drive C:.

```
BACKUP DATABASE NTDB1 TABLESPACE USERSPACE1 ONLINE TO C:
```

3.2.6 Database-Level Restore

This section discusses restoring database-level backups. There are some differences between restore from offline backup and restore from online backup.

3.2.6.1 Restore from Database-Level Backup (Offline)

For the duration of the database-level backup (offline), the BACKUP DATABASE command uses the database in exclusive mode, and there is no connection to the database by other DB2 processes. Hence, the ROLLFORWARD DATABASE command is not necessary. But, if archive logging is enabled and you want to perform roll forward recovery after restoring the database, the ROLLFORWARD DATABASE command must be issued.

The following command restores the database from the database-level backup (offline) from drive C:.

```
RESTORE DATABASE NTDB1 FROM C:
```

To prevent the database from entering a roll forward pending state after it has been successfully restored in an archive logging environment, you may execute the RESTORE DATABASE command as follows:

```
RESTORE DATABASE NTDB1 FROM C: WITHOUT ROLLING FORWARD
```

The above command is the same as issuing the following two commands:

```
RESTORE DATABASE NTDB1 FROM C:
ROLLFORWARD DATABASE NTDB1 STOP
```

3.2.6.2 Restore Database-Level Backup (Online)

A restore of an online database-level backup allows other DB2 applications to access and update the database while the backup process is running. However, the database must be brought to a consistency point with the log files by performing a roll forward operation. The RESTORE DATABASE command followed by the ROLLFORWARD DATABASE command is used. You cannot use the RESTORE DATABASE command unless the roll forward operation is also specified.

The following commands restore a database-level backup (online) and roll forward to the end of logs.

```
RESTORE DATABASE NTDB1 FROM C:
ROLLFORWARD DATABASE NTDB1 TO END OF LOGS AND STOP
```

3.2.7 Tablespace-Level Restore

This section discusses restoring a tablespace-level backup. You can restore either an offline or online backup image of a tablespace.

3.2.7.1 Tablespace-Level Restore (Offline or Online)

When performing a recovery using a tablespace-level backup (offline or online), you must complete the roll forward until the end of logs have been reached. This ensures that the tablespace is recovered to the same level as the rest of the database. In contrast with the database-level roll forward, the STOP parameter is not required for a tablespace-level roll forward. From the tablespace-level backup (either of offline or online), the online restore and roll forward recovery can be performed. This means that other processes can access the database except for the tablepsace involved in the restoration. This tablespace will not be available until the restore and roll forward operations have successfully completed.

The following commands perform a restore of a tablespace-level backup and a roll forward to the end of logs.

```
RESTORE DATABASE TABLESPACE ONLINE NTDB1 FROM C:
ROLLFORWARD DATABASE NTDB1 TO END OF LOGS TABLESPACE ONLINE
```

Note: You cannot restore to a tablespace which has been dropped or does not exist.

3.2.8 Backup and Restore Considerations

There are several considerations when using the BACKUP DATABASE, RESTORE DATABASE and ROLLFORWARD DATABASE commands and APIs. They are as follows:

- Authorizations

Either SYSADM, SYSCTRL, or SYSMAINT authority is needed for executing the BACKUP DATBASE, RESTORE DATABASE and ROLLFORWARD DATABASE commands.

- LOB (Large Object) column with NOT LOGGED option

  When recovering a database-level or a tablespace-level backup image using the RESTORE and ROLLFORWARD commands, LOB data that was NOT LOGGED and was written since the last backup will be replaced by binary zeros.

- Online backup and LOAD command with COPY NO option

If the LOAD utility was executed with the default COPY NO option, the tablespace or tablespaces are put into a backup pending state. Before backing up the loaded tablespaces, all connections to the database must be terminated. This can be performed by issuing the FORCE APPLICATION ALL and/or DEACTIVATE DATABASE command. If an online backup is in progress, do not start the LOAD utility.

- Copy the logs before the restore and roll forward recovery

  It is recommended that you copy the logs before you begin the restore and roll forward recovery in case the logs are lost or damaged during the recovery.

- DUOW restrictions

  BACKUP DATABASE, RESTORE DATABASE, ROLLFORWARD DATABASE commands and APIs are not supported in a Distributed Unit Of Work (DUOW, CONNECT Type2). Using these in the DUOW will result in an error (SQL30090N, SQLCODE=-30090, SQLSTATE=25000, reason code=02).

3.3 Moving Data Utilities

In this section, we look at how to move data into DB2 for NT Version 2 databases with the following utilities:

- IMPORT
- EXPORT
- LOAD

3.3.1 IMPORT

The IMPORT utility inserts data from an input file with a supported file format into a table or view. You need SYSADM or DBADM authorization to execute the IMPORT utility and CONTROL privilege on the table or view; INSERT and SELECT privilege on the table or view are also needed. When data is imported from PC/IXF file format, you can specify the CREATE or REPLACE_CREATE option to create a table without issuing any DDL (Data Definition Language). When these specifications are used, you can specify target tablespaces.

The IMPORT utility and API are not supported in a Distributed Unit Of Work (DUOW CONNECT Type2). Using these in the DUOW will result in an error(SQL30090N, SQLCODE=-30090, SQLSTATE=25000, reason code=02).

3.3.1.1 Input Data File Format

Four file formats for an input file are supported by the IMPORT utility. They are PC/IXF (PC version of Integrated Exchange Format), DEL (delimited ASCII), ASC (non-delimited ASCII,) and WSF (worksheet format for Lotus-1-2-3, Symphony).

PC/IXF This file format is supported. Refer to 3.3.2, “EXPORT” on page 89 for more details.

DEL This file format is supported. Refer to 3.3.1, “IMPORT.”

ASC Non-delimited ASCII files are used for loading data from other applications that create flat text files with aligned column data, such as word processors. Each ASCII file is a stream of ASCII characters consisting of data values organized by row and column. Rows in the data stream are separated by a line feed(or a carriage return/line feed). An example of an ASC file is:

Smith, Bob 4973 15.46

Jones, Suzanne 12345 16.34

Shakespeare, William 1564 16.16

WSF This file format is supported. Refer to 3.3.2, "EXPORT."

3.3.1.2 Import Example

The following command creates the PHOTO_IMG table using three tablespaces (REG01, IND01 (index tablespace), LOB01 (long tablespace)) and imports data from the PHOTO.IXF file with PC/IXF format to the table. LOB data is read from LOB files.

```
IMPORT FROM PHOTO.IXF OF IXF MODIFIED BY lobsinfile CREATE INTO PHOTO_IMG IN
REG01 INDEX IN IND01 LONG IN LOB01
```

3.3.2 EXPORT

The EXPORT utility exports data from a database to an output file. Users can specify the data to be exported by supplying an SQL SELECT statement. To execute the EXPORT utility, you need SYSADM or DBADM authorization. Moreover, CONTROL or SELECT privilege on each participating table or view is needed.

The EXPORT command and API are not supported in a Distributed Unit Of Work (DUOW, CONNECT Type2). Using these in the DUOW will result in an error (SQL30090N, SQLCODE=-30090, SQLSTATE=25000, reason code=02).

3.3.2.1 Output Data File Format

The EXPORT utility supports three file formats for an output file. They are PC/IXF (PC version of Integrated Exchange Format) and DEL (delimited ASCII) and WSF (worksheet format for Lotus-1-2-3, Symphony). There is no support for ASC (non-delimited ASCII).

PC/IXF This file format is a database manager adaptation of the Integrated Exchange Format and is the preferred method for exchange between database managers. The IXF architecture is designed to enable exchange of relational database structure and data. You can export a data file using a Distributed Database Connection Services (DDCS) gateway from a host database to the DB2 Server. The advantage of this the elimination of having to create a table using DDL (Data Definition Language) and related indexes into a new or empty table. In general, a PC/IXF file consists of an unbroken sequence of variable-length records. The file will have the following types of records in the order given:

- One header record of record type H.
- One table record or record type T.
- Multiple column descriptor records of record type C (one record for each column of data in the table).
- Multiple data records of record type D. Each row in the table is represented by one or more D records.

DEL If you do not have a DDCS connection with a remote database, or you are transferring data from some other source, it is likely to be in delimited ASCII format. Delimited ASCII is used for exchanging files with a wide variety of industry applications, especially other database products. This is a commonly used way of storing data that separates column values with a special delimiting character. An example of a DEL file is:

"Smith, Bob",4973,15.46

"Jones, Suzanne",12345,16.34

"Shakespeare, William",1564,16.16

where (") is a character string delimiter, (,) is a column delimiter and (.) is a decimal point. Alternatively, (;) can be used as a column delimiter, with (') as the character string delimiter.

WSF Lotus 1-2-3 uses this file format. The database manager supports the subset of the worksheet records that are the same for all the Lotus products. That is, for the releases of Lotus 1-2-3 supported by the database manager, all file names with any three-character extension are accepted, for example: WKS, WK1, WRK, WR1, WJ2. To comply with the correct release of Lotus/Symphony, the WSF file format should use the filetype-mod (1,2 or 3) parameter followed by the `MODIFIED BY` option.

1 Creates a WSF file that is compatible with Lotus 1-2-3 Release 1 or Lotus 1-2-3 Release 1a. This is the default.

2 Creates a WSF file that is compatible with Lotus Symphony Release 1.0.

3 Creates a WSF file that is compatible with Lotus 1-2-3 Version 2 or Lotus Symphony Release 1.1. These files can also be directed to a specific product by specifying an L for Lotus 1-2-3 or an S for Symphony in the filetype-mode parameter string.

The following can be noted regarding the WSF file format:

1. Data in the WSF files uses a Lotus codepoint mapping that is not necessarily the same as existing code pages supported by DB2 Common Server. As a result, when importing or exporting a WSF file, data is converted from the Lotus code points to/from the code points used by the application code page. DB2 Common Server supports conversion between the Lotus code points and code points defined by code pages 437, 819, 850, 860, 863, and 865. For multi-byte characters users, no conversions are performed.

2. Do not use the WSF file format to transfer data between DB2 Common Server databases since a loss of data may occur. Use PC/IXF file format instead.

3.3.2.2 Sample Usage

The following command exports data from the EMP_PHOTO table to the PHOTO.IXF file with the PC/IXF format. LOB data is stored in LOB files such as IMAGE1.001, IMAGE1.002, and so on.

```
EXPORT TO PHOTO.IXF OF IXF LOBNAME IMAGE1 MODIFIED BY lobsinfile SELECT * FROM
EMP_PHOTO
```

3.3.3 LOAD

The LOAD utility is intended for the initial load or append of a table where large amounts of data are inserted. There are no restrictions on the data types used by the LOAD utility. You may include LOBs (Large Objects) and User-Defined Types (UDTs) as data that is going to be loaded. The LOAD utility is faster than performing an IMPORT because LOAD writes formatted pages directly into the database, while import performs SQL inserts. Also the LOAD utility does not log each write of data during a load operation. There are three phases to the LOAD process:

Load Where the data is inserted into the table. During the load phase, data is loaded into the table, and index keys are collected. Save points or consistency points are established at intervals specified by you in the LOAD command. Messages let you know how many input rows have been successfully loaded at the time of the save point. If a failure occurs, you should use the `restartcount` option set to the value indicated by the last load consistency/save point indicated in the messages file. If the failure occurs near the beginning of the load, you could consider restarting the load again from the beginning.

Build Where the indexes are created. During the build phase, indexes are created, based on the index keys collected in the load phase. The index keys are sorted during the load phase. If a failure occurs, the build is restarted from the beginning of the build phase.

Delete Where the rows that caused a unique key violation are removed from the table. During the delete phase, all rows causing a unique key violation are deleted. If a failure occurs, this phase should be restarted by you from the beginning of the delete phase. Once the database indexes are rebuilt, information about the rows containing the invalid keys is stored in an exception table, (if you have created one before the load began and identified it in the LOAD command). Unique key violations are placed into the exception table, and messages on rejected rows are put into the message file. Finally duplicated keys are deleted.

All phases of the load process are part of one operation which is completed only after all three phases complete successfully. The LOAD utility will generate messages during the progress of each phase. Should a failure occur during one of the phases, then these messages can assist you in deciding on recovery actions.

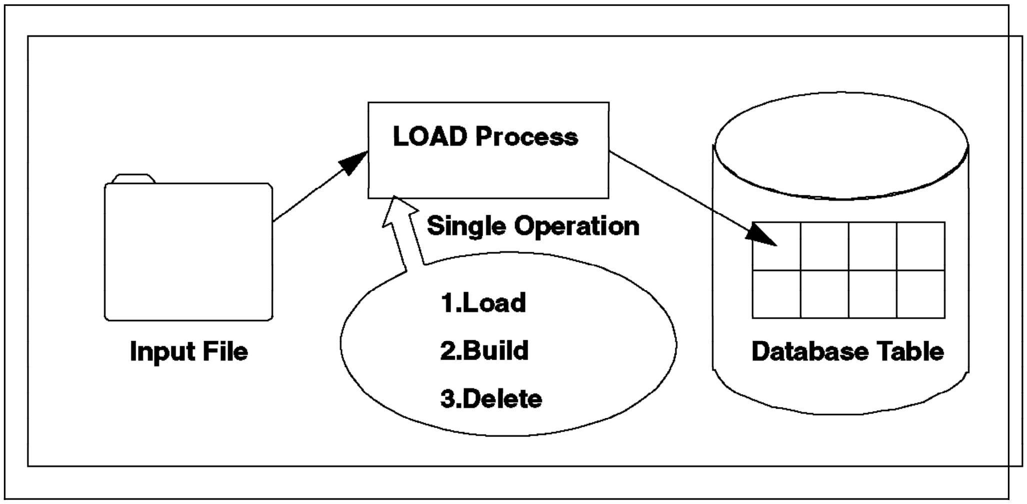


Figure 62. Three Phases of Load

Figure 62 illustrates the three phases of the load process, and shows that they are part of a single operation.

The following can be noted about the LOAD utility:

1. The `LOAD` command and API are not supported in a Distributed Unit Of Work (DUOW, CONNECT Type2). Using these in the DUOW will result in an error (SQL30090N, SQLCODE=-30090, SQLSTATE=25000, reason code=02).
2. In DB2 for NT, loading from a tape device is not supported.

3.3.3.1 Input Data File Format

Three file formats for an input file are supported by the LOAD utility. They are PC/IXF (PC version of Integrated Exchange Format) and DEL(delimited ASCII) and ASC (non-delimited ASCII). WSF (worksheet format for Lotus-1-2-3, Symphony) is not supported.

PC/IXF This file format is supported. Refer to the description of EXPORT.

DEL This file format is supported. Refer to the description of EXPORT.

ASC This file format is supported. Refer to the description of IMPORT.

3.3.3.2 Example of the LOAD Utility

The following command loads data from the PHOTO.IXF file with the PC/IXF format to the PHOTO_IMG table. Archive logging is enabled.

```
LOAD FROM PHOTO.IXF OF IXF MODIFIED BY lobsinfile REPLACE INTO PHOTO_IMG
STATISTICS YES WITH DISTRIBUTION AND DETAILED INDEXES ALL COPY YES TO
C:\LOADCOPY
```

The specification of `lobsinfile` in the `MODIFIED BY` parameter tells DB2 that all LOB data is to be loaded from files.

The specification of `STATISTICS YES` is for gathering statistics for the table and for any existing indexes.

The specification of `COPY YES` saves changes to a copy file.

3.3.4 Differences Between IMPORT and LOAD

Though both IMPORT and LOAD are used to put data into a database, it is important to understand the differences between the two utilities.

Table 6. Differences between IMPORT and LOAD Utilities

IMPORT	LOAD
Creation of table and indexes supported with IXF format.	Table and indexes must exist.
WSF format is supported.	WSF format is not supported.
Can import into aliases, some views and tables.	Can load into aliases, tables.
The tablespace(s) that contain the table and its indexes are online for the duration of import.	The tablespace(s) that contain the table and its indexes are offline for the duration of the load.
Triggers will be fired.	Triggers are not supported.
If an import is interrupted and a commit count had been specified, the table is usable and will contain the rows that were loaded up to the last commit. The user has the choice to restart the import or use the table as is.	If a load is interrupted and a SAVECOUNT option had been specified, the table remains in a load pending state and cannot be used until the load is restarted to or until the tablespace is restored from a backup image created before the load.
All constraints are validated during an import.	Uniqueness is verified during a load, but all other constraints must be checked using the SET CONSTRAINTS statement.
The keys of each row are inserted into the index one at a time during the import.	During a load, all the keys are sorted, and the index is built after the data has been loaded.
If up-to date statistics are required after an import, RUNSTATS must be executed.	Statistics can be gathered during the load.
You can import into a host database with DDCS.	You cannot load into a host database.
A file that is imported must reside on the node where import is invoked.	A file/pipe that is loaded must reside on the node where the database resides.
No backup image is required.	When archive logging, the backup image should be taken with the COPY YES option during the load or by issuing the BACKUP DATABASE command after the load.

Chapter 4. DB2 for NT Server Communication

This chapter looks at the steps needed to configure a DB2 for NT database server for remote client access. The installation of the communication product is not covered. It is assumed that the basic steps to enable communication using a specific protocol have already been performed. The communication protocols discussed in this chapter are TCP/IP, APPC, NetBIOS, and APPC. Using the DB2 for NT database server as a LAN gateway is discussed. Also discussed is how to configure a Windows NT gateway to access a DB2 for MVS database.

4.1 Configuring a Windows NT DB2 Server

DB2 supports most of the common LAN communication protocols, including NetBIOS, TCP/IP, IPX/SPX, and APPC. The protocols supported by DB2 also depend on the operating system.

The DB2 for NT Server product supports all of the protocols supported by DB2 Common Server, including NetBIOS, TCP/IP, IPX/SPX, and APPC. However, you need to decide which communications protocol (or combination of protocols) needs to be used based on the client operating system. DB2 NT Server can communicate with remote clients by using any of the supported communications protocol as long as the client supports that protocol. Refer to Table 7 for a quick reference of the supported communication protocols.

Table 7. DB2 NT Client/Server Connectivity Chart

DB2 Clients (CAE / SDK)	NetBIOS	IPX/SPX	TCP/IP	APPC
DOS	Yes	Yes	Yes	No
Windows 3.X	Yes	Yes	Yes	No
Windows 95	Yes	Yes	Yes	No
Windows NT	Yes	Yes	Yes	Yes
OS/2	Yes	Yes	Yes	Yes
AIX	No	No	Yes	Yes

You need to ensure that before enabling the communications support with DB2 Server, the corresponding communications product has been correctly configured on the NT system. Once the communication protocol has been configured, then the steps for configuring the DB2 for NT Server are as follows:

1. Update the necessary communication variables. This will always be DB2COMM variable. Depending on the protocol used, other environment variables may also need to be set or changed.
2. Update the database manager configuration file.
3. Stop and start the instance.

The next sections provide the detailed communication configuration steps for DB2 for NT Server.

4.2 NetBIOS

For remote client access using the DB2 NT Server product and NetBIOS, you must first have installed and configured the required communications protocol support. For NetBIOS support, ensure that you have installed the NetBEUI/NetBIOS network software component.

To verify the installation of Windows NT for NetBIOS support:

1. Go to the Control Panel on the Windows NT machine.
2. Double-click on the **Network icon.**
3. Click on **Protocols** and the following window appears:

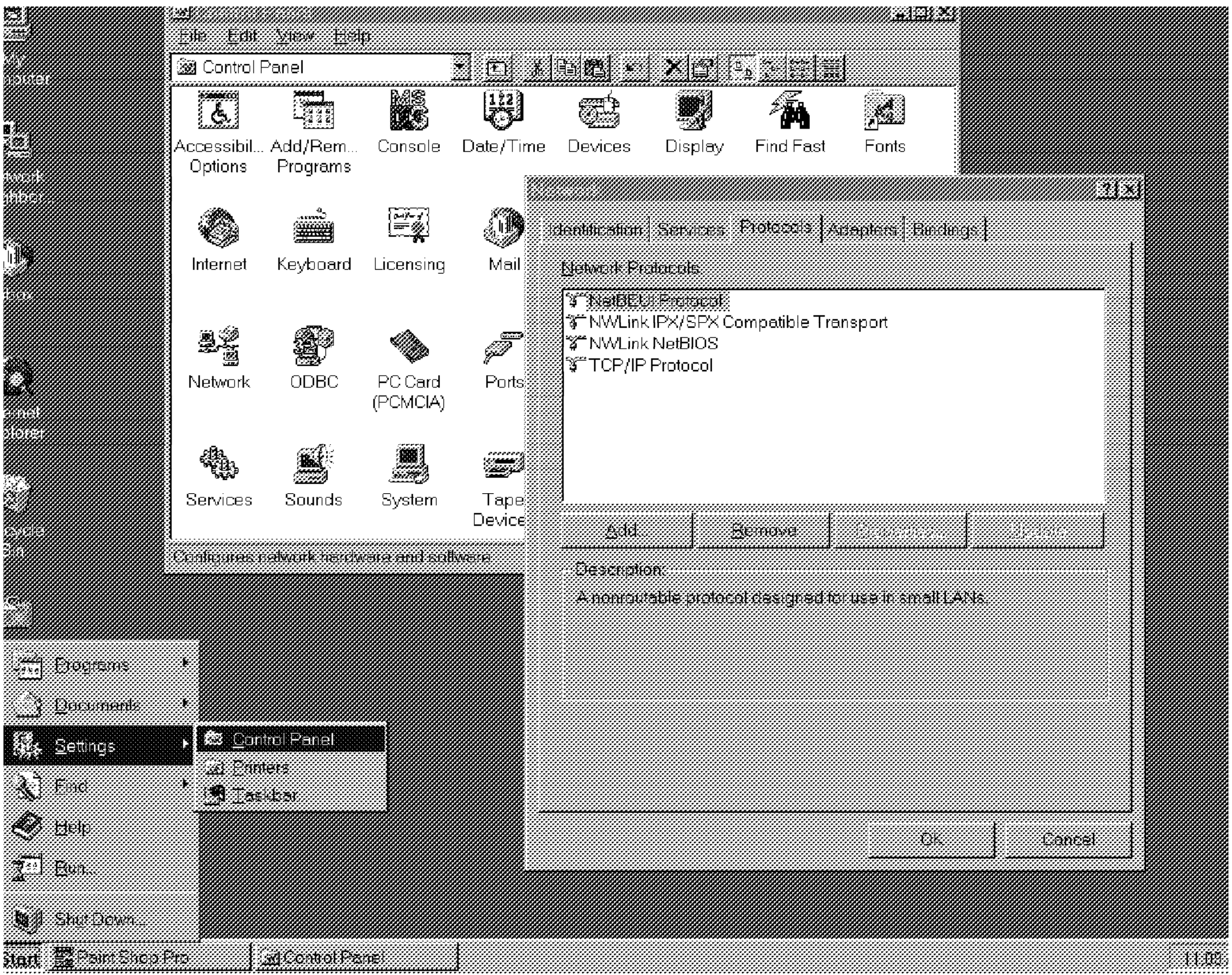


Figure 63. Verifying NetBIOS Support on Windows NT

4.2.1 Setting the Environment Variable DB2COMM on the Server

There is one communication environment variable that must be set for all protocols. The environment variable is DB2COMM. DB2COMM determines which protocol(s) will be enabled when the database manager is started. DB2COMM can be set to multiple values if you need the NT database server to support more than one communications protocol simultaneously.

In order to set the DB2COMM variable on the DB2 for NT database server to support remote NetBIOS clients, you need to edit the Systems Environment Variables section. There, you must add an entry with a variable name of DB2COMM whose values is set to NetBIOS.

To set the DB2COMM variable in the Windows NT registry:

1. Go to the Control Panel icon on the Windows NT machine.
2. Select the **System icon** by double-clicking.
3. In the Systems Control panel, in the System Environment Variables section, do the following:
    - If the environmental variable DB2COMM does not exist:
        - Select any environment variable and change the name in the Variable box to DB2COMM.
        - Enter NetBIOS in the value box.
        - Click on the **Set** button. The variable will be added in alphabetical order.
    - If the environmental variable DB2COMM already exists:
        - Select DB2COMM from the System Environment Variable section.
        - In the value box, add NetBIOS to the already existing value(s), separated by a comma(,).
        - Click on the **Set** button. (The variable will be updated to the new value.)
4. Select **OK** to exit.

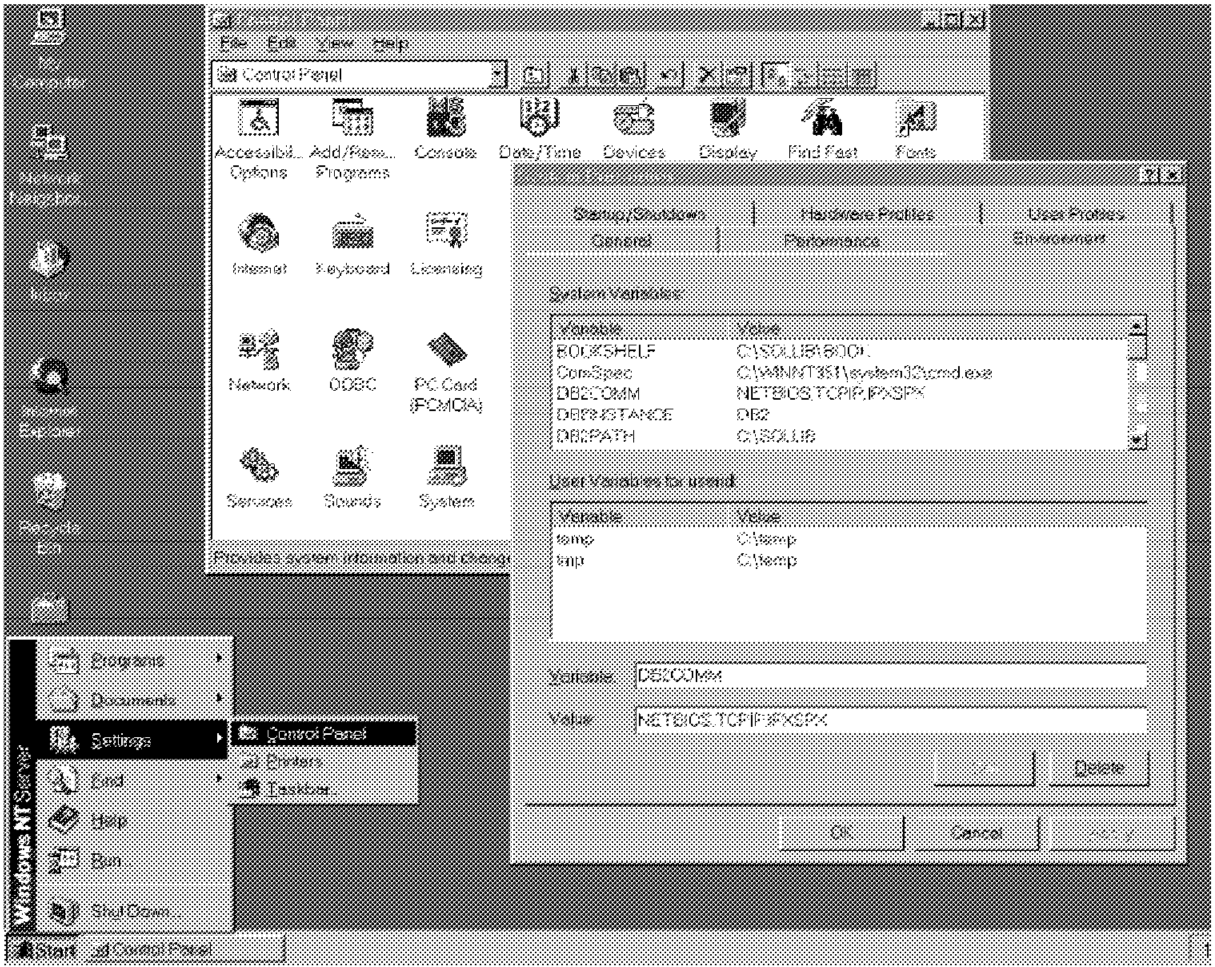


Figure 64. Setting the DB2COMM Variable

Figure 64 shows setting the DB2COMM variable to enable communications through NetBIOS, TCP/IP and IPX/SPX.

***** Tip *****

You can determine the setting for the DB2COMM variable by using the SET command. To check the value of the DB2COMM variable, enter the following on the command prompt:

```
SET DB2COMM
```

This will tell you the value of DB2COMM whether or not it has been set. If DB2COMM has been set, you can use the echo command to check its value:

```
echo %DB2COMM%
```

4.2.2 Changing the NetBIOS Configuration on the Server

NetBIOS support for DB2 for NT Server is implemented by using NetBIOS frames and default LAN Adapter 0. In order for NetBIOS to be correctly supported using the DB2 product, you must either:

- Configure the NetBIOS interface to the defaults expected by DB2 NT Server.
- Catalog the NetBIOS node directory to match the Windows NT NetBIOS configuration you are using.

4.2.2.1 Configuring the NetBIOS Interface Configuration

To change or view the NetBIOS interface configuration:

1. Select the **Network icon** in the Control Panel program group.
2. Click on the **Services** Tab.
3. Select **NetBIOS Interface** under Installed Network Services.
4. Select **Properties.**
5. Find the Network Route: Nbf->... (physical Adapter specific)
6. Identify the Logical LAN Adapter Number (Lana Number) associated with the Network route Nbf ->... This value needs to be changed to 000, (if it is not already set) if you want DB2 to use the default value for LAN Adapter Numbers.

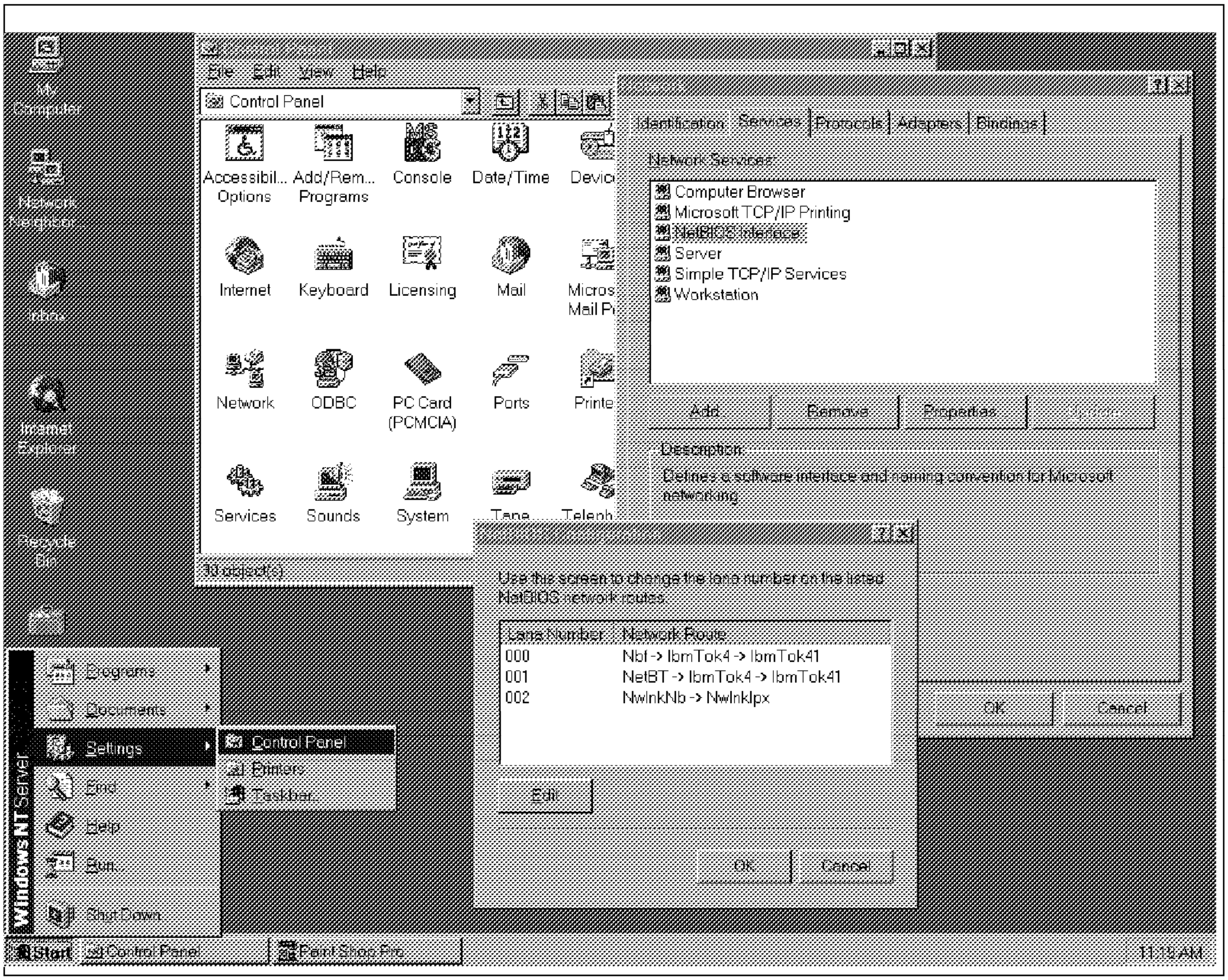


Figure 65. Windows NT NetBIOS Interface Configuration

The figure shows Windows NT NetBIOS Configuration where the Lana Number of the Network Route: Nbf->... is set to 000.

4.2.2.2 Catalog Node to Match the Windows NT NetBIOS Configuration

As an alternative to configuring the NetBIOS interface, you can catalog the NetBIOS Node Directory. To catalog the NetBIOS node directory to correspond with the Windows NT NetBIOS interface configuration, you must:

- Identify the LAN Adapter Number associated with the Network route Nbf ->. entry and perform the following step:
  - On the Server, set the environment variable DB2NBADAPTERS equal to the Lana number associated with the Network Route Nbf. If you want to change DB2NBADAPTERS from the default (000), the environment variable DB2NBADAPTERS must be defined as a System Environment Variable, and Windows NT must be restarted before the change is implemented.

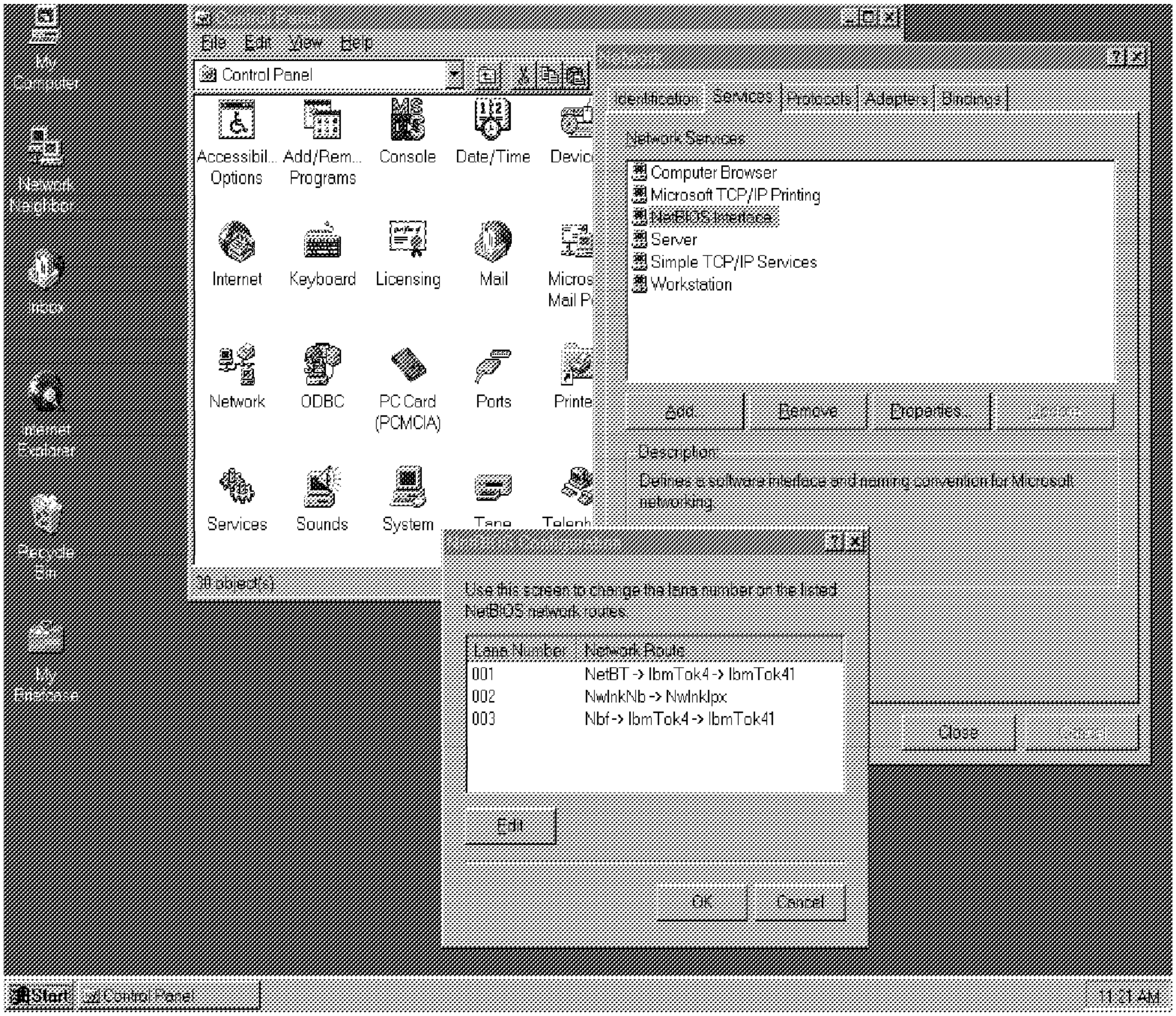


Figure 66. NetBIOS Interface Configuration

Figure 66 shows the Windows NT NetBIOS Configuration where the Lana Number of the Network Route: Nbf->... is 003. In this case, you will need to set an additional environment variable, DB2NBADAPTERS, equal to 003 (The value 003 here is the LANA number) in the System Environment Variable. For more details on how to set the System Environment variables, refer to 4.2.4.1, “Setting Environment Variables” on page 109.

4.2.3 Updating the Database Manager Configuration File on the Server

The DB2 NT Server must also be updated with the workstation name variable (NNAME) which has to be unique on the network. The client will use the NNAME to make contact with the server through NetBIOS.

***** Configuration Tip*****

A NetBIOS error will occur if the workstation name is not unique on the network.

4.2.3.1 Using the DB2 Client Setup

This section covers setting the workstation name variable, NNAME, on the NT database server.

- Open the DB2 Client Setup program.
- Select the **Client** menu, then select **Configure**. The Client Configuration window opens, as shown below:

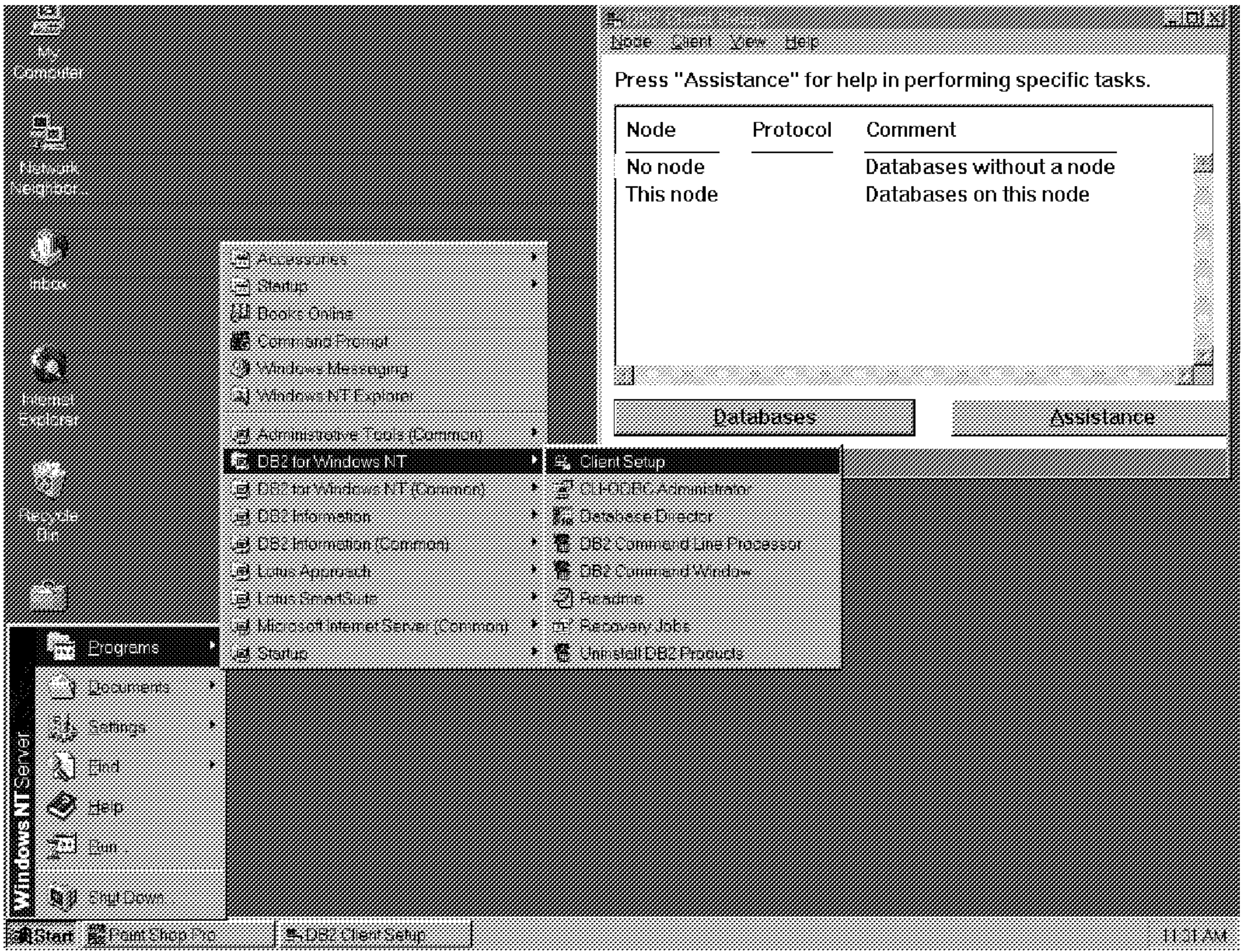


Figure 67. DB2 Client Configuration

- Select the **Communications** option under Settings.
- Enter the value for Node Name. This is the NNAME variable. Figure 68 on page 103 shows the value being set to DB2NTSRV.
- Select **OK**.

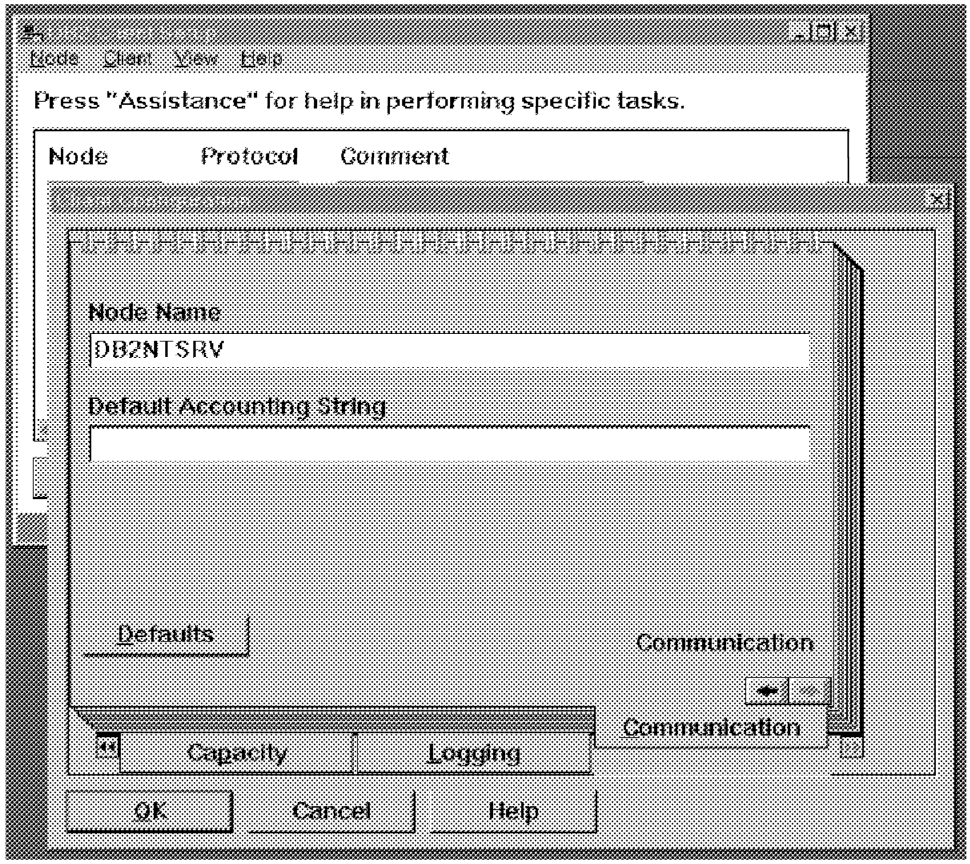


Figure 68. Update NetBIOS NName Using DB2 Client Setup

4.2.3.2 Using the DB2 Command Line Processor

You can also use the Command Line Processor to set or change the `NNAME` value.

- Open the DB2 Program group.
- Double-click on the **Command Line Processor icon**. This will put you in the Command Line Processor Interactive mode.
- Enter the command to update the database manager configuration file. For example, if the `NNAME` is `db2ntsrv`, either form of the command could be used:

```
UPDATE DATABASE MANAGER CONFIGURATION USING NNAME db2ntsrv
UPDATE DBM CFG USING NNAME db2ntsrv
```

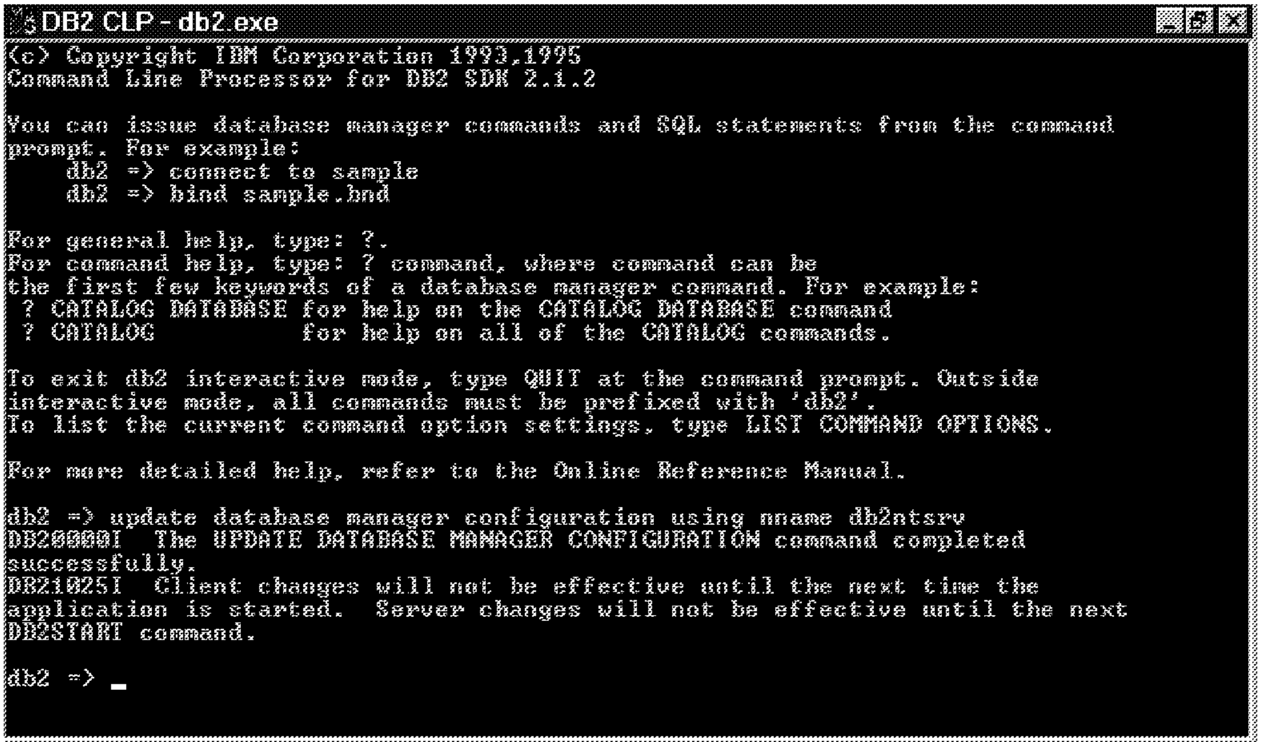


Figure 69. Update NetBIOS NName Using DB2 CLP

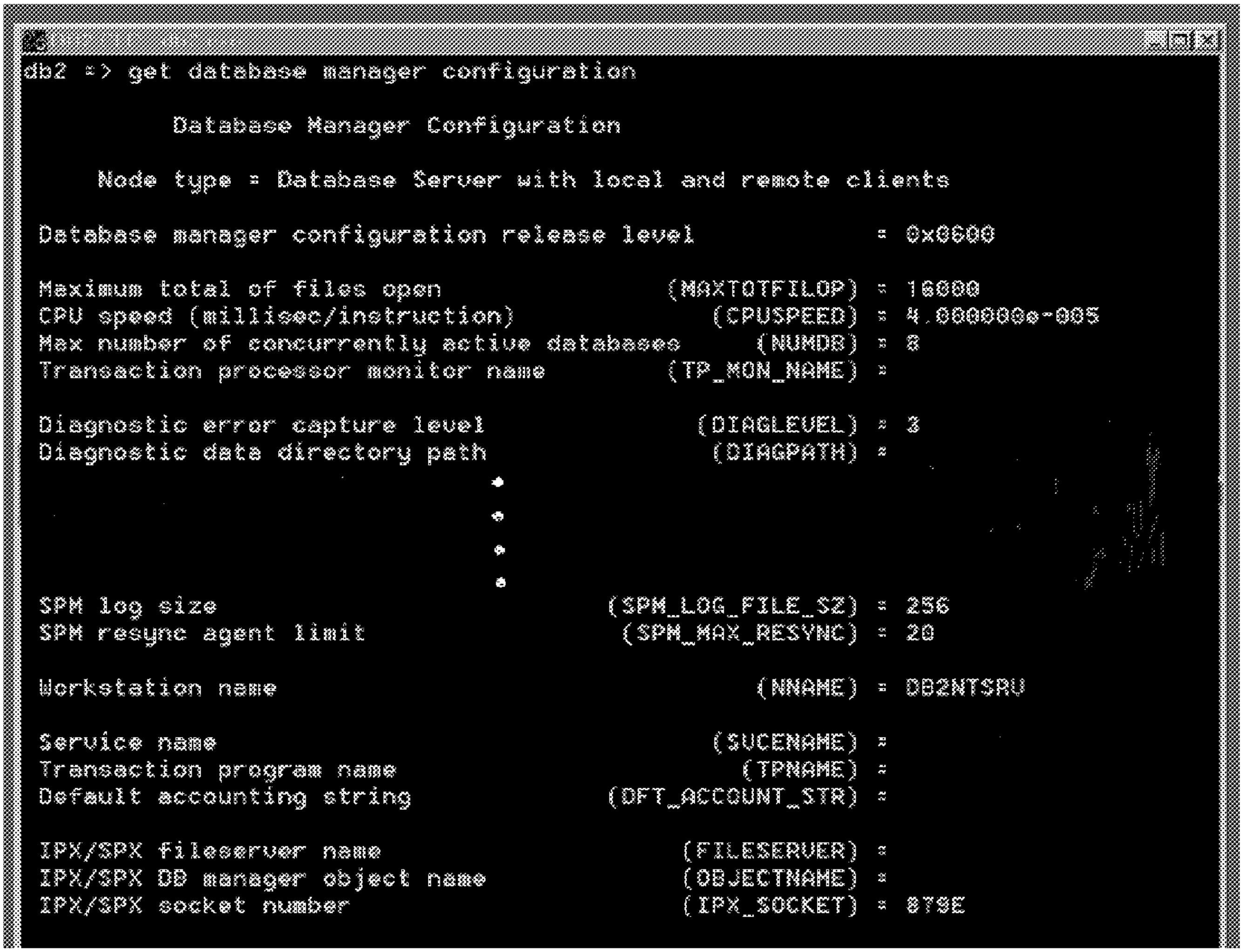
Figure 69 is an example of updating the NetBIOS name in the Database Manager configuration file using the DB2 Command Line Processor.

To verify that the database manager has updated the change, issue one of the following commands from a DB2 CLP window:

```
GET DATABASE MANAGER CONFIGURATION
GET DBM CFG
```

Figure 70 on page 105 shows the parameters in the configuration file. Notice that the workstation name (NNAME) has been updated with the value DB2NTSRV.

.



```
db2 => get database manager configuration

          Database Manager Configuration

     Node type = Database Server with local and remote clients

 Database manager configuration release level            = 0x0600

 Maximum total of files open               (MAXTOTFILOP) = 16000
 CPU speed (millisec/instruction)             (CPUSPEED) = 4.000000e-005
 Max number of concurrently active databases     (NUMDB) = 8
 Transaction processor monitor name        (TP_MON_NAME) =

 Diagnostic error capture level              (DIAGLEVEL) = 3
 Diagnostic data directory path               (DIAGPATH) =
                                 .
                                 .
                                 .
                                 .
 SPM log size                          (SPM_LOG_FILE_SZ) = 256
 SPM resync agent limit                 (SPM_MAX_RESYNC) = 20

 Workstation name                                (NNAME) = DB2NTSRV

 Service name                                 (SVCENAME) =
 Transaction program name                       (TPNAME) =
 Default accounting string             (DFT_ACCOUNT_STR) =

 IPX/SPX fileserver name                    (FILESERVER) =
 IPX/SPX DB manager object name             (OBJECTNAME) =
 IPX/SPX socket number                      (IPX_SOCKET) = 879E
```

Figure 70. Verifying Database Manager Configuration

To make the changes in the database manager configuration file take effect, you must stop and restart the database manager. To start a DB2 instance, log on with a user that has administrative authority. You can start or stop an instance on the DB2 NT Server in one of two ways:

- Select the **START / STOP** button for the instance you want from the Services dialog in the Control Panel.
- Use `net start <instance name>` and `net stop <instance name>` in any command window.

  ```
  net start <instance_name>
  net stop <instance_name>
  ```

  When starting the database manager, you must also start the DB2 for NT Security Service by using the following command:

  net start DB2NTSECSERVER

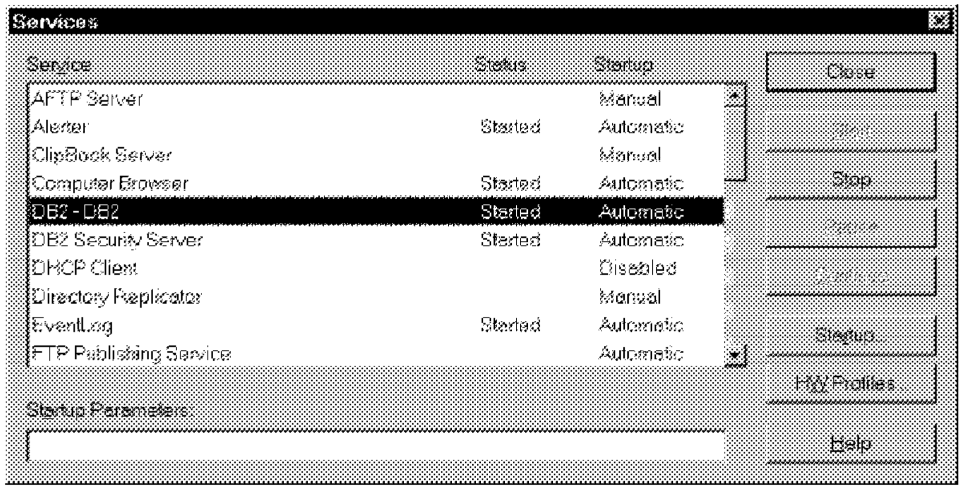


Figure 71. Start / Stop DB2 Instance

Figure 71 illustrates the Services window and shows that the DB2 Instance (by the name DB2) has been started. You can also see that the DB2 Security Server which will be used for validating users is running. Also notice that the startup options for both DB2-DB2 and DB2 Security Server have been modified from `Manual` (which is the default) to `Automatic`. This results in both services starting automatically when the server is booted. For more details on modifying Windows NT Services refer to Windows NT documentation or consult a Windows NT Administrator.

Figure 72 on page 107 and Figure 73 on page 108 are for reference. Figure 72 on page 107 shows a summary of the necessary steps to configure a DB2 for NT database server using NetBIOS. The items that are checked represent only those steps necessary for the server configuration. Figure 73 on page 108 shows the correlation between client and server when using NetBIOS for a communication protocol with DB2. The following is a summary of those tasks:

- CAE or SDK for OS/2, DOS, Windows, Windows 95, or Windows NT clients, along with DB2 for OS/2 and DB2 for Windows NT clients may use the NetBIOS protocol to communicate with a DB2 for Windows NT Server.
- On a DB2 client, as well as the DB2 server, NetBIOS support must be installed and configured correctly before performing the DB2 communication steps.
- On both the client and the server, there are environment variables that can be updated. Setting the `DB2COMM` environment variable to include NetBIOS is required on the server. The other environment variables on the client and the server are optional. You may take the default for them.
- The workstation name (`NNAME`) on both the client and the server must also be updated. This parameter is found in the database manager configuration file. By default, it is blank.
- When cataloging the client node directory, the server's DBM workstation name (`NNAME`) must be known. The client will use this name to make contact with the server through NetBIOS.
- When cataloging the system database directory on the client, the alias that the server machine uses in its system database directory when referencing the DB2 database server must be known.

NetBIOS Connectivity Checklist

- [x] Install NetBIOS support on both client and server
- [x] Update environment variables on server:
  * Server must be set DB2Comm= NetBIOS
- [x] Update DBM CFG file on server:
  * Specify workstation name (NNAME)
- [x] Change NetBIOS interface configuration on server:
  * Windows NT platform
- [ ] Update DBM CFG file on client:
  * Specify workstation name (NNAME)
- [ ] Change NetBIOS interface configuration on client:
  * Windows 95 and NT platform only
- [ ] Catalog Node Directory on client:
  * Choose NetBIOS protocol
  * Must know server's DBM workstation name (NNAME)
- [ ] Catalog System Directory Database on client:
  * Must know server's system database directory alias name

Figure 72. NetBIOS Connectivity Server Checklist

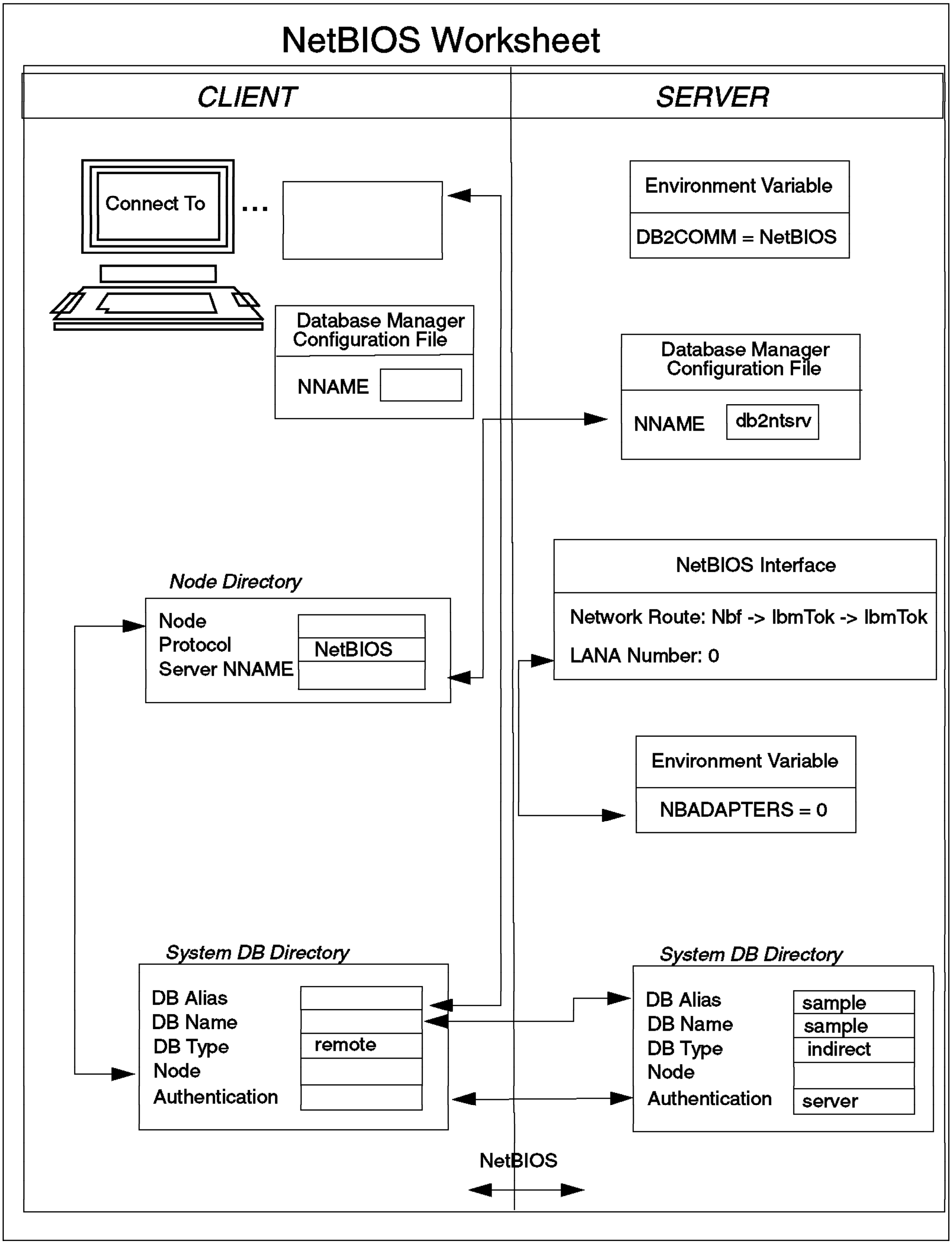


Figure 73. Completed NetBIOS Server Worksheet

4.2.4 Tuning the NetBIOS Configuration for DB2

This section discusses some of the variables that can be customized to use with DB2. The variables can be altered by modifying the System Environment section using the steps described below. The parameters and their significance is discussed in 4.2.4.2, "DB2 NetBIOS Environment Variables" on page 110.

4.2.4.1 Setting Environment Variables

The use of NetBIOS resources by DB2 is controlled by the environment variables. These variables can be added or altered by modifying the System Environment section using the following steps:

1. Go to the Control Panel icon on the Windows NT machine.
2. Select **System icon** by double-clicking.
3. In the Systems Control Panel, in the System Environment Variables section, do the following:
    - If the environmental variable does not exist:
        - Select any environment variable.
        - Change the name of the variable in the Variable box to the new variable name (for example, DB2NBADAPTERS).
        - Enter the appropriate value in the value box.
        - Click on the **Set** button. The variable will be added in alphabetical order.
    - If the environmental variable already exists:
        - Select the variable to be updated from the System Environment Variable section.
        - Alter the value, as required.
        - Click on the **Set** button. The variable will be updated to the new value.
4. Select **OK** to exit.

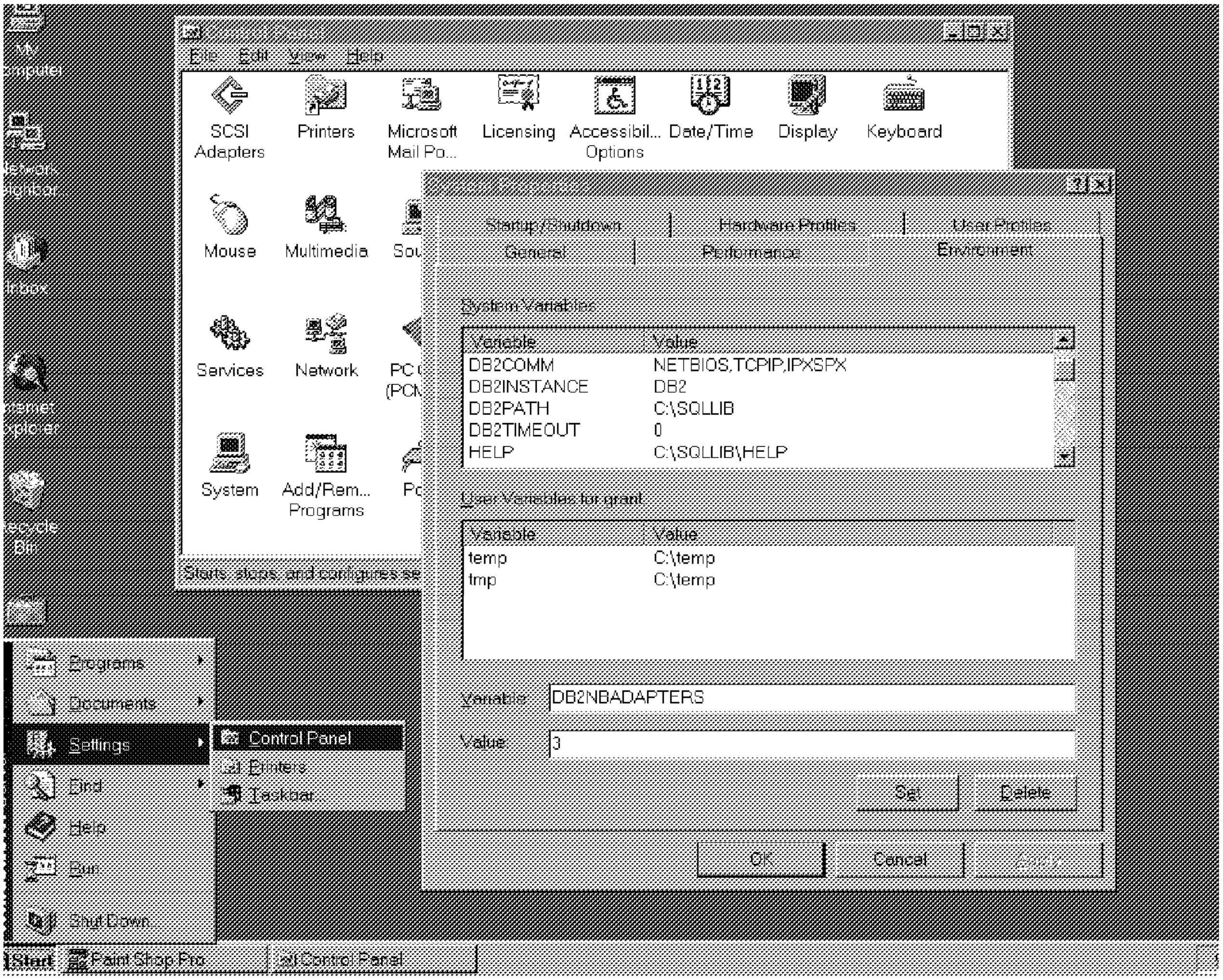


Figure 74. Setting Environment Variables

4.2.4.2 DB2 NetBIOS Environment Variables

There are seven other NetBIOS-related parameters that can be modified although it is not mandatory to do so. The variables and their significance is discussed in this section for your reference:

- **DB2NBADAPTERS** - Specifies which adapter(s) to use for DB2 NetBIOS LAN Communications. It allows the selection of multiple adapters, by specifying their Logical Adapter Number. Multiple adapters may be selected by specifying their logical adapter number. For example, to use logical adapters 0 and 2, use `DB2NBADAPTERS= 0,2`.
- **DB2NBSESSIONS** - Specifies the number of NetBIOS sessions to be reserved for DB2. This variable enables you to make specific session requests for each adapter that was specified in the `DB2NBADAPTERS` variable. For example, to request 12 and 30 sessions for the logical adapters 0 and 2 respectively, use `DB2NBSESSIONS= 12,30`. The value specified represents the total DB2 sessions required per adapter and can be calculated as the sum of the following:
  - Maximum expected concurrent NetBIOS client sessions.

- Sessions required for interrupt - usually 1, unless you have specified a different value for the DB2NBINTRLISTENS variable.
- Sessions required for DB2 overhead - always 3.
- **DB2NBINTRLISTENS** - Specifies the number of NetBIOS listen commands (ncbs) that will be asynchronously issued in readiness for remote client interrupts. This flexibility is provided for interrupt active environments to ensure that the interrupt calls from the remote clients will be able to establish connections when the DB2 Server is busy servicing other remote interrupts. Specifying lower values conserves NetBIOS sessions and ncbs at the server while higher values may be essential where client interrupts are common. For example, to request 4 and 3 interrupt sessions for the logical adapters 0 and 2 respectively, use DB2NBINTRLISTENS= 4,3.
  - Changing this value affects your session calculation when specifying the DB2NBSESSIONS value.
  - The server will automatically adjust its NetBIOS ncb reservation request based on the value specified here.
- **DB2NBRECVNCBS** - Specifies the number of NetBIOS receive any ncbs that the server will issue and maintain during operation. This value may be adjusted depending on the number of remote clients your server is connected to. Lower values will conserve server resources. Higher values may be required to prevent clients from reissuing requests (when high traffic results in the server being busy). Each adapter can have its own receive ncb value. For example, to set the receive ncb value to 10 and 12 for the logical adapters 0 and 2 respectively, use DB2NBRECVNCBS= 10,12.
  - The DB2DIAG.LOG file can be used to determine if the set value is adequate or not.
  - The server will automatically adjust its NetBIOS ncb reservation request based on the value specified here.
- **DB2NBSENDNCBS** - Specifies the number of send commands (ncbs) that the server will reserve for use. This value may be adjusted depending on the number of remote clients that are connecting to your server. Lower values will conserve server resources. Higher values may be required to prevent the server from waiting to send to a remote client when all other send commands are in use. Only one value can be specified for this parameter, and it is a server-wide value (independent of the number of adapters used). For example, to set the number of send commands to 4, use DB2NBSENDNCBS= 4.
  - The DB2DIAG.LOG file can be used to determine if the set value is adequate or not.
  - The server will automatically adjust its NetBIOS ncb reservation request based on the value specified here.
- **DB2NBRECVBUFFSIZE** (SVRIOBLK) - Specifies the size of the DB2 NetBIOS receive buffers. These buffers are assigned to the NetBIOS receive ncbs. The receive buffer size should be adjusted to the size of the data transfer from the client workstation. It is recommended to have server receive buffers large enough to completely receive the data sent from the client so that only one server receive operation is needed. Lower values will conserve server resources. Higher values may be

required if client transfers are larger. Only one value can be specified for this parameter, and it is a server-wide value (independent of the number of adapters used). For example, to set the DB2 NetBIOS receive buffer size to 8192, use `DB2NBRECVBUFFSIZE= 8192`.

  - The DB2DIAG.LOG file can be used to determine if the set value is adequate or not.
  - This parameter must be set to a value larger than the Client I/O Block size (for example, RQRIOBLK) of all the clients. If this value is smaller, then the client requests may be broken down into a number of sends. This may cause NetBIOS failure in some cases.

- **DB2NBCHECKUPTIME** - Specifies the time interval between each invocation of the NetBIOS protocol checkup procedure. Checkup time is specified in minutes. Lower values will ensure that the NetBIOS protocol checkup routine runs more often, freeing up system resources (including memory) left around when unexpected session termination occurs. This ensures that the server remains consistent. Higher values may be used when unexpected occurrences are rare, hence reducing the time spent determining that NetBIOS protocol manager is operational. Only one value can be specified for this parameter, and it is a server-wide value. For example, to set the NetBIOS protocol checkup routine to run once in 4 minutes, use `DB2NBCHECKUPTIME= 4`.

  - Specifying a value of 0 will prevent the checkup routine from running and is not recommended.

Table 8 is a summary of the seven additional NetBIOS DB2 environment variables discussed in this section.

Table 8. DB2 NetBIOS Environment Variables

Parameter	Description	Default	Range
db2nbadapters	Specifies which adapter(s) to use for DB2 NetBIOS LAN Communications	0	0 - 15
db2nbsessions	Specifies the number of NetBIOS sessions to be reserved for DB2	32	5 - 254
db2nbintrlistens	Specifies the number of NetBIOS listen ncbs that will be asynchronously issued in readiness for remote client interrupts	1	1 - 10
db2nbrecvncbs	Specifies the number of NetBIOS 'receive any' commands (ncbs) that the server will issue and maintain during operation	10	1 - 99
db2nbsendncbs	Specifies the number of send commands (ncbs) that the server will reserve for use	6	1 - 99
db2nbrecvbuffsize	Specifies the size of the DB2 NetBIOS receive buffers	4096	4096 - 65536
db2nbchecktime	Specifies the time interval between each invocation of the NetBIOS protocol checkup procedure	1	1 - 720

4.3 TCP/IP

For remote clients to access the DB2 for NT database server using TCP/IP, you must first have installed and configured the required communications protocol support. For TCP/IP support, ensure that you have installed the TCP/IP network software component.

To verify the installation of Windows NT for TCP/IP support:

1. Go to the Control Panel on the Windows NT machine.
2. Double-click on the **Network icon.**
3. Click on **Protocols,** and when the following window appears, ensure TCP/IP is in the list:

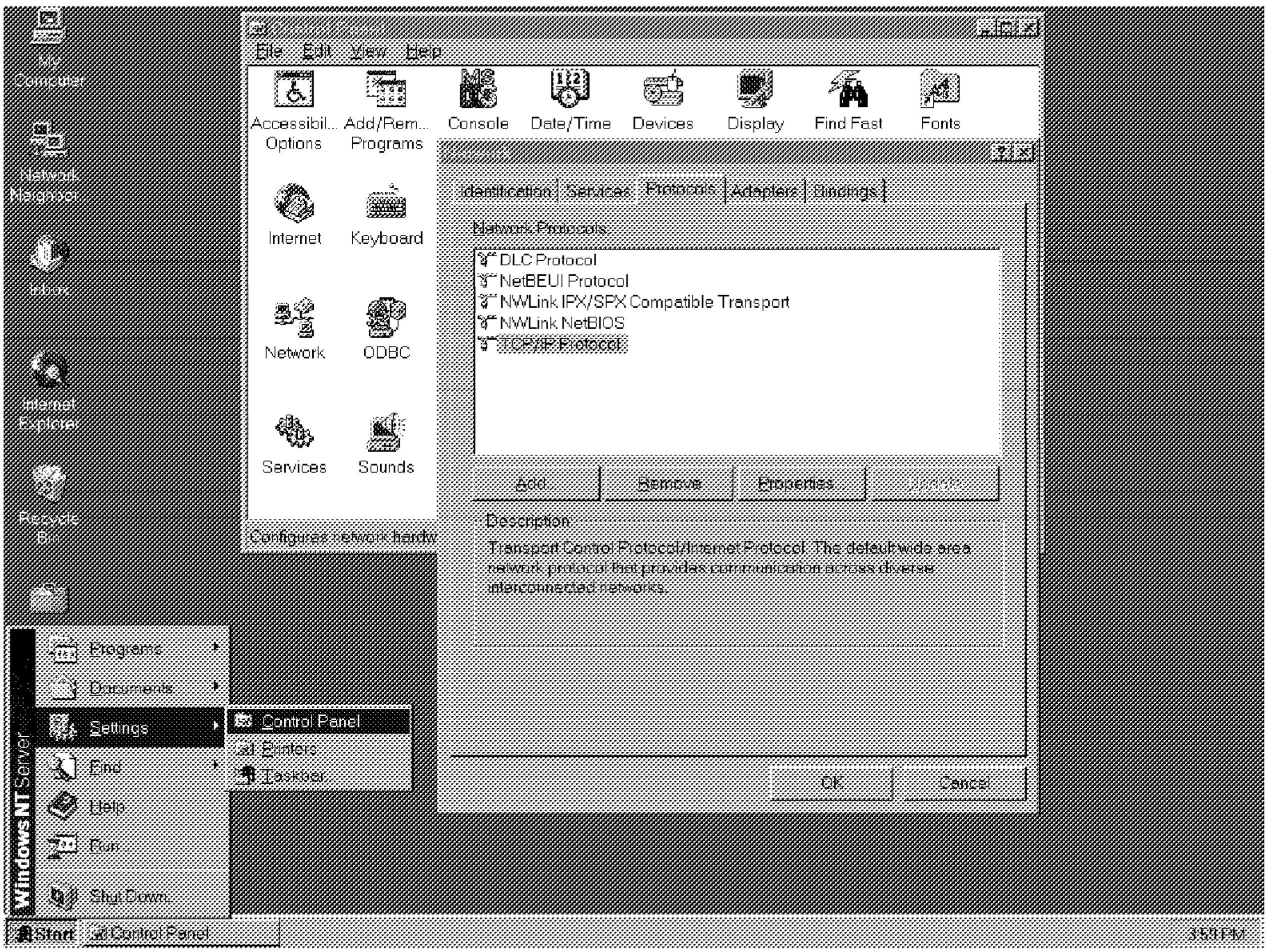


Figure 75. Verifying TCP/IP Support for Windows NT

4.3.1 Setting the DB2COMM Environment Variable on the Server

There is one communication environment variable that must be set for all protocols. The environment variable is DB2COMM. DB2COMM determines which protocol(s) will be enabled when the database manager is started. DB2COMM can be set to multiple values if you need the database server to support more than one communications protocol simultaneously.

In order to set the DB2COMM variable on DB2 NT Server to support TCP/IP clients, you need to edit the System Environment Variables section. Add an entry with a variable name of DB2COMM and set its value to TCPIP.

To set the DB2COMM variable in the Windows NT registry:

1. Go to the Control Panel icon on the Windows NT machine.
2. Select **System icon** by double-clicking.
3. In the Systems Control panel in the System Environment Variables section, do the following:
   - If the environmental variable DB2COMM does not exist:
     - Select any environment variable and change the name in the Variable box to DB2COMM.
     - Enter TCPIP in the value box.
     - Click on the **Set** button. The variable will be added in alphabetical order.
   - If the environmental variable DB2COMM already exists:
     - Select DB2COMM from the System Environment Variable section.
     - In the value box, add TCPIP to the already existing value(s), separated by a comma(,).
     - Click on the **Set** button. (The variable will be updated to the new value.)
4. Select **OK** to exit.

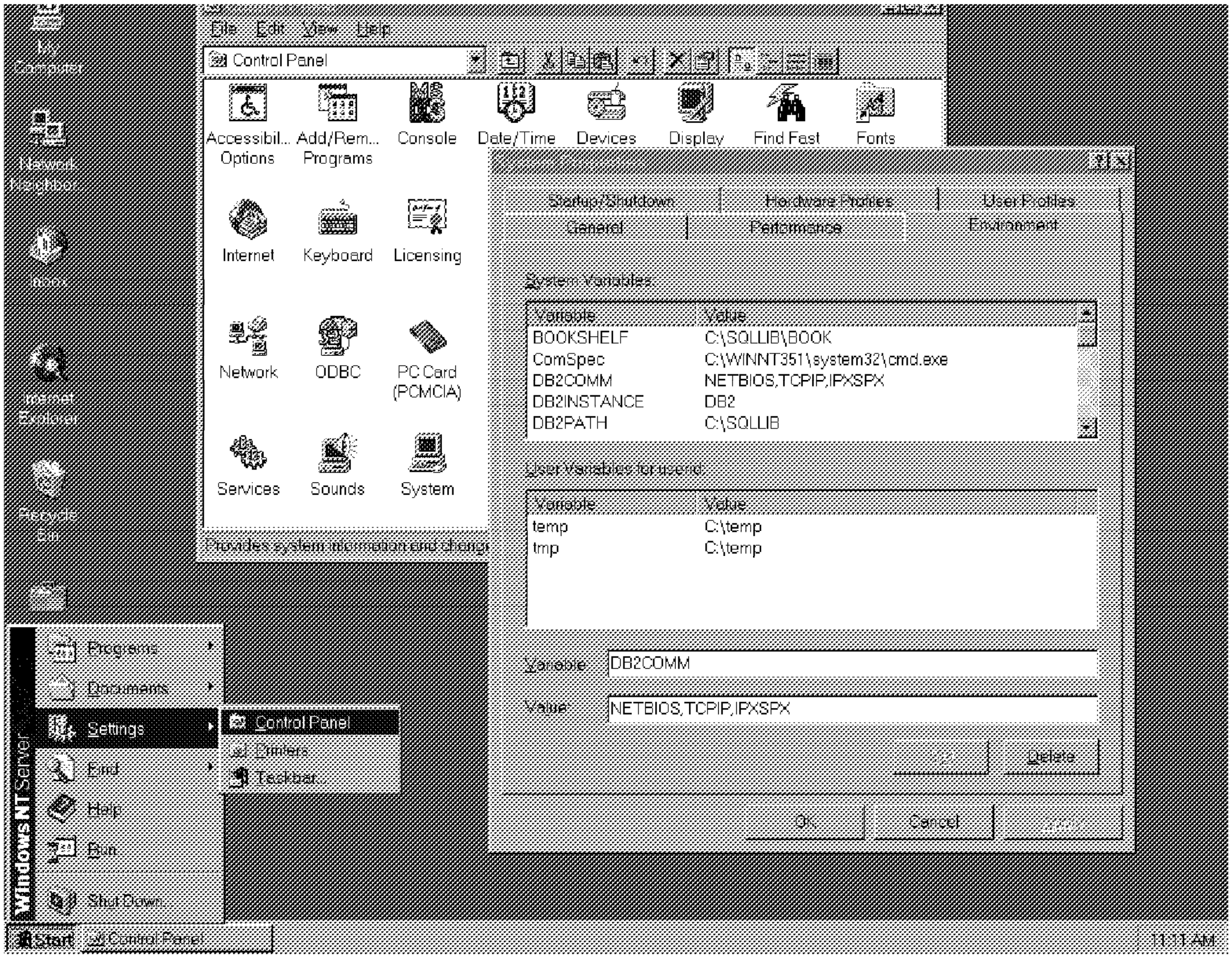


Figure 76. Setting the DB2COMM Variable

Figure 76 shows setting the DB2COMM variable to enable communications through NetBIOS, TCP/IP and IPX/SPX.

***** Tip *****

You can determine the setting for the DB2COMM variable by using the echo command. To check the value of the DB2COMM variable, enter the following on the command prompt:

```
echo %DB2COMM%
```

4.3.2 Updating the Services File on the Server

To enable TCP/IP protocol support on the DB2 NT Server, you'll also need to update the services file on the database server. The services file is updated using an editor and is not part of any of the DB2 configuration files.

Add two entries to the X:<winnt-dir>system32driversetcservices file (where X: is the drive on which Windows NT Operating System is installed and <winnt-dir> is the directory name) using two consecutive port numbers for every instance of DB2 on a machine.

- Port n is the main connection port.
- Port n+1 is the interrupt port.

Note: The interrupt port is only used by Version 1 clients.

For example, the entries in the c:winnt351system32driversetcservices file should be similar to:

```
db2instc 5000/tcp #DB2 main connection port
db2instci 5001/tcp # DB2 interrupt port
```

where db2instc and db2instci are unique service name entries for the connection port and 5000 and 5001 are the port numbers.

The following should be noted regarding the entries in the services file:

- The service name is always case-sensitive.
- The port numbers must be a decimal number greater than 1000.
- The port numbers must be unique for each DB2 instance within the services file (on the server and all the clients).
- The port numbers used by the server will also be added to the services file on all the clients wishing to connect to the server through TCP/IP. You need to ensure that there is no conflict on any of the client systems with the port number you have selected.
- To ensure that the entries in the services file are read correctly, the port number must not be the last character in the file. It is recommended that you press **Enter** at the end of the line or end the last line with a comment (#).

4.3.3 Updating the Database Manager Configuration File on the Server

The parameter in the database manager configuration file used for TCP/IP communication is SVCENAME. By default, this parameter is blank. The service name is assigned as the main connection port name for the instance and defined in the server's services file. This name in the services file references the main connection port number that clients must use to connect to that instance on the database server through TCP/IP.

For example, if the name defined in the services file is db2instc1, the command would be:

```
UPDATE DATABASE MANAGER CONFIGURATION USING SVCENAME db2instc1
UPDATE DBM CFG USING SVCENAME db2instc1
```

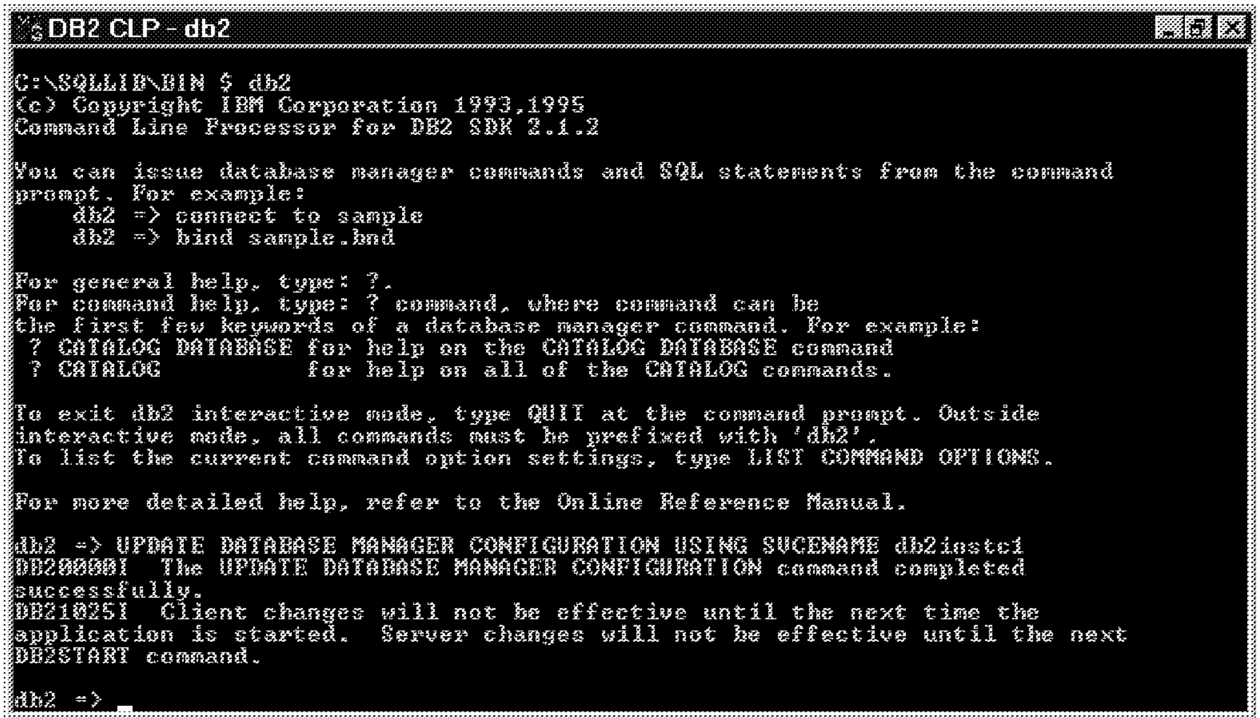


Figure 77. Updating Database Manager Configuration using DB2 CLP

Figure 77 is an example of successfully updating the database manager configuration file for enabling TCP/IP communication on the DB2 for NT Server.

To verify that the database manager has updated the change, issue one of the following commands from a DB2 CLP:

```
GET DATABASE MANAGER CONFIGURATION
GET DBM CFG
```

Figure 78 on page 118 shows the parameters in the configuration file. Notice that the service name parameter (SVCENAME) has been updated with the value db2instc1.

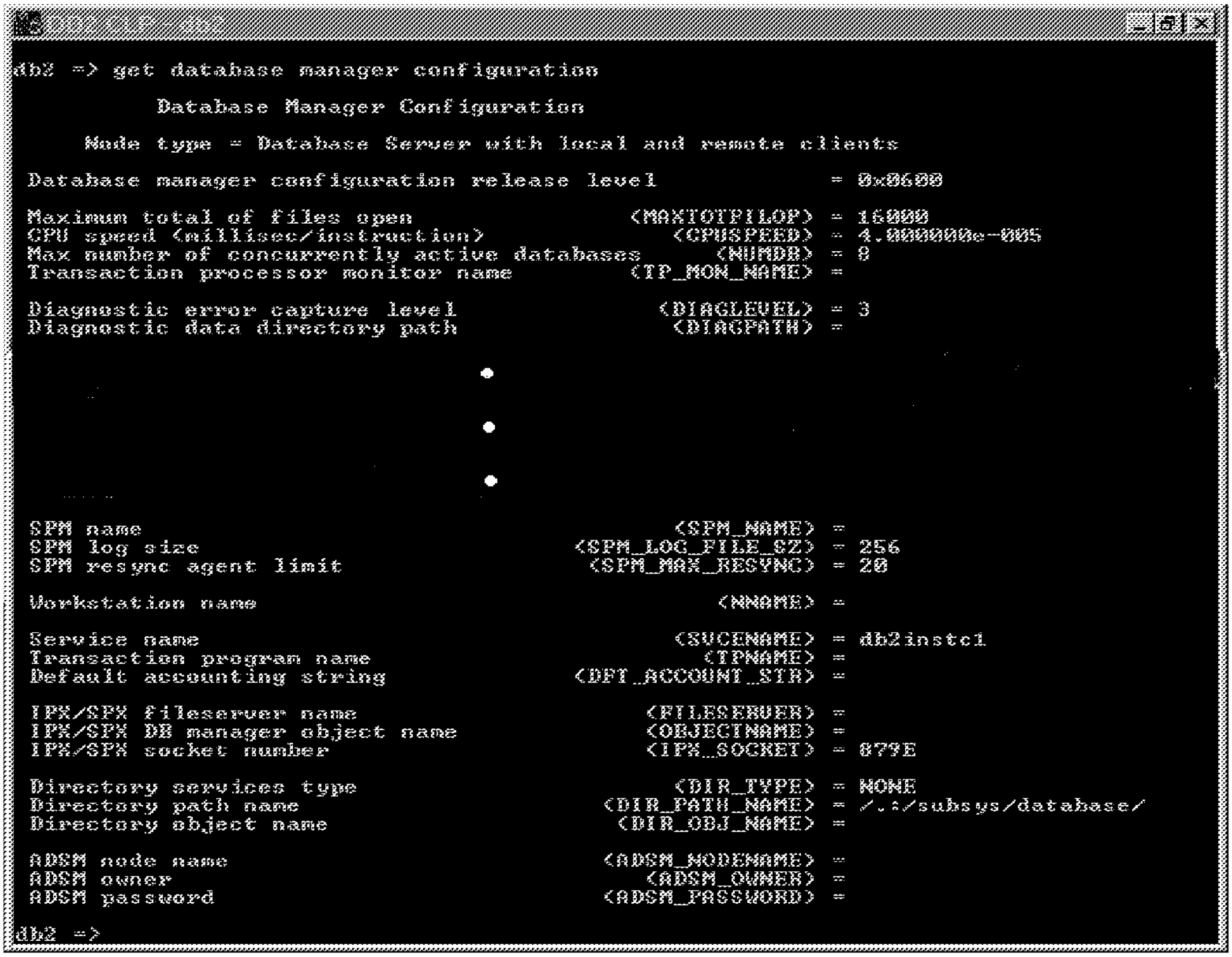


Figure 78. Verifying Database Configuration

To make the changes in the database manager configuration file take effect, you must stop and start the instance. To start a DB2 instance, log on with a user that has administrative authority. You can start or stop an instance on the DB2 NT Server using one of the two ways:

- Select the **START / STOP** button for the instance you want in the Services dialog in the Control Panel.
- Use `net start <instance name>` and `net stop <instance name>` in any command window.

```
net start <instance-name>
net stop <instance-name>
```

When you start the database manager, the DB2 Security Service is automatically started by DB2.

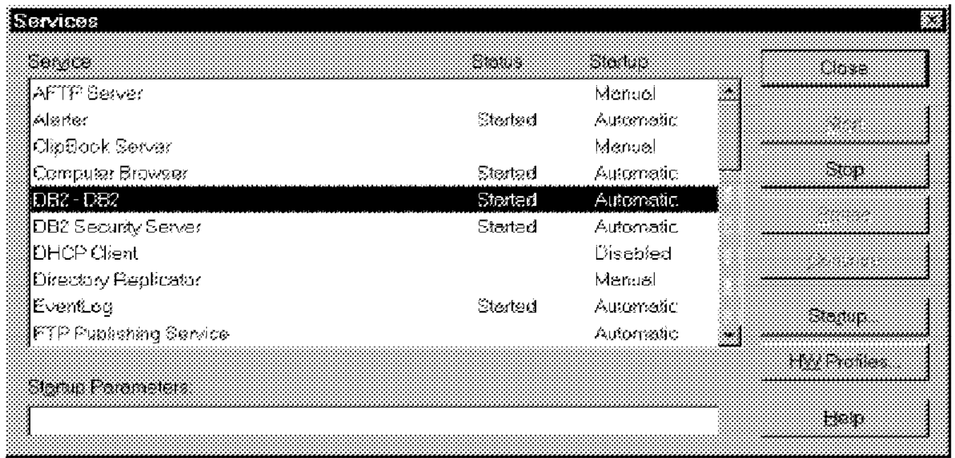


Figure 79. Start / Stop DB2 Instance

Figure 79 illustrates the Services window and shows that the DB2 instance (by the name DB2) has been started. You can also see that the DB2 Security Server which will be used for validating users is running. Also notice that the startup options for both DB2-DB2 and DB2 Security Server have been modified from `Manual` (which is the default) to `Automatic`. This results in both services starting automatically when the server is booted. For more details on modifying Windows NT Services, refer to Windows NT documentation or consult a Windows NT Administrator.

When the database server is re-started, two listen processes will be started. The listen processes will be waiting to be contacted by remote clients using the port numbers specified in the services file.

Figure 80 on page 121 and Figure 81 on page 122 are for reference. Figure 80 on page 121 shows a summary of the necessary steps to configure a DB2 for NT database server and remote client by using TCP/IP. The items that are checked indicate only those steps necessary to configure the NT database server. Figure 81 on page 122 shows the correlation between client and server when using TCP/IP for a communication protocol with DB2. The following is a summary of those tasks:

- CAE or SDK for OS/2, AIX, DOS, Windows, Windows 95, Windows NT, HP-UX, Solaris, SINIX, or Macintosh clients, along with DB2 for OS/2, AIX, Windows NT, HP-UX, Solaris, or SINIX clients may use the TCP/IP protocol to communicate with a DB2 for NT Server.
- On a DB2 client, as well as on a DB2 Server, TCP/IP support must be installed and configured correctly.
- On both the client and the server, there are environment variables that can be updated. Setting `DB2COMM=TCPIP` is required on the server. The other environment variables on the client and server are optional.
- The service name (`SVCENAME`) on the server must be updated. This parameter is found in the database manager configuration file. By default, it is blank.
- The services file on both the client and server must be updated. On the server, the port names and numbers that will be used to service clients for this instance are defined. The main connection port name must match the `SVCENAME` specified in the database manager configuration file. On the client,

the port numbers specified in its services file must match the port numbers defined for each instance in the server's services file that it wants to access.

- The client's hosts file must be updated to include the IP address of the server, as well as a host name (an alias) for the server which is associated with the IP address specified. (This step may have already been done.) The IP address cannot be used when cataloging the TCP/IP node directory.
- When cataloging the client node directory, the client's main connection port name in its services file must be specified. This will be used to reference the main connection port number for the server's instance. When cataloging the client node directory, the host name of the server, as specified in the client's hosts file, is also designated.
- When cataloging the system database directory on the client, the alias that the server machine uses in its system database directory when referencing the DB2 database must be known.

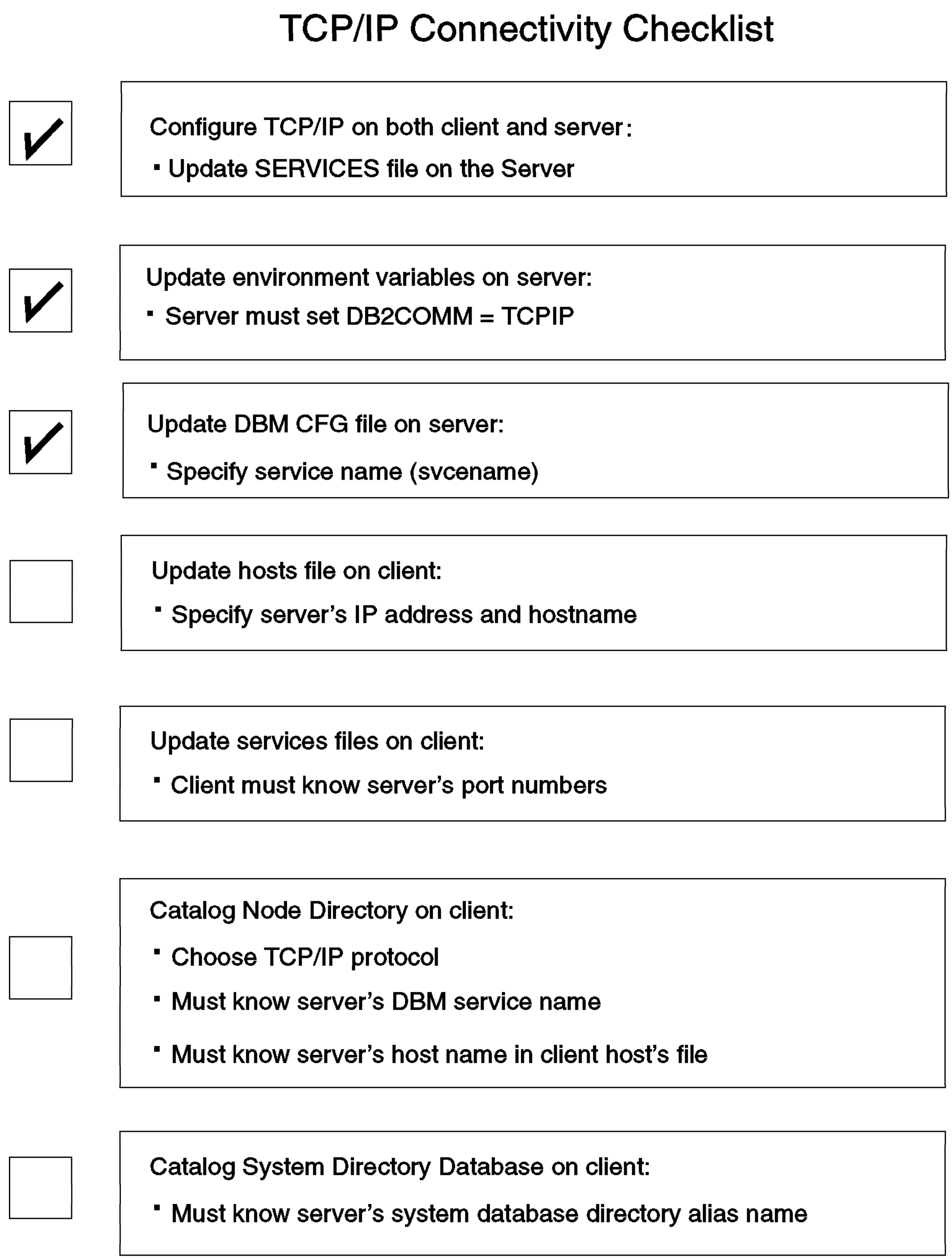


Figure 80. TCP/IP Connectivity Server Checklist

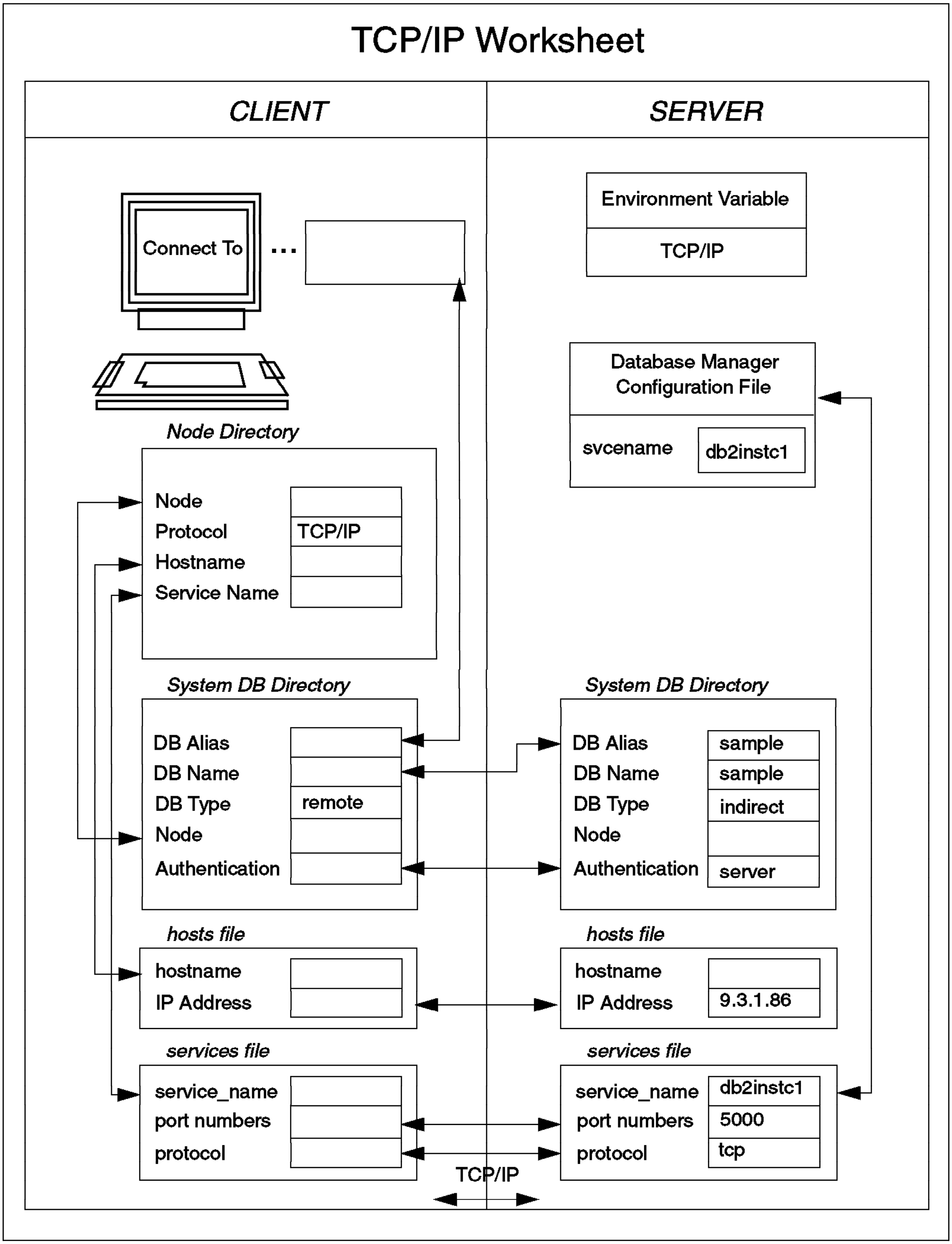


Figure 81. Completed TCP/IP Server Worksheet

4.3.4 Tuning the TCP/IP Configuration for DB2

This section discusses a number of TCP/IP configuration parameters that can affect the performance of applications communicating to a DB2 Server through TCP/IP.

4.3.4.1 SO_KEEPALIVE

The TCP/IP protocol has been developed so that one host may not be notified of the failure of its partner on another host. As a result, an application accessing the remote DB2 Server through TCP/IP may appear to be hung. DB2 uses the TCP/IP **SO_KEEPALIVE** socket option to alleviate this problem. A message is transmitted periodically to determine if the partner is still alive. If the partner fails to respond to the message, the connection is considered to be broken, and the application will be notified. The frequency at which the messages are transmitted can be modified using the `KeepAliveTime` TCP/IP configuration parameter in the Windows NT Registry. The parameter should be added to:

```
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesTcpipParameters
```

The default value of this parameter is 2 hours.

It is important to understand that this parameter (if modified) will affect all TCP/IP socket applications, not just DB2 NT applications.

4.3.4.2 DB2SOSNDBUF and DB2SORCVBUF

By default, DB2 NT Server uses 32767 as the send and receive buffer size used by TCP/IP (which is specified by the tcpip `SetSockOpt` call). This value can be altered by setting the environment variables `DB2SOSNDBUF` (for sends) and `DB2SORCVBUF` (for receives) as System Environment Variables.

4.4 IPX/SPX

For remote clients accessing the DB2 NT Server using IPX/SPX, you must first have installed and configured the required communications protocol support. For the IPX/SPX support, ensure that you have installed the NWLink IPX/SPX-compatible transport network software component.

To verify the installation of Windows NT for IPX/SPX support:

1. Go to the Control Panel on the Windows NT machine.
2. Double-click on the **Network icon.**
3. Click on **Protocols** and the following window appears:

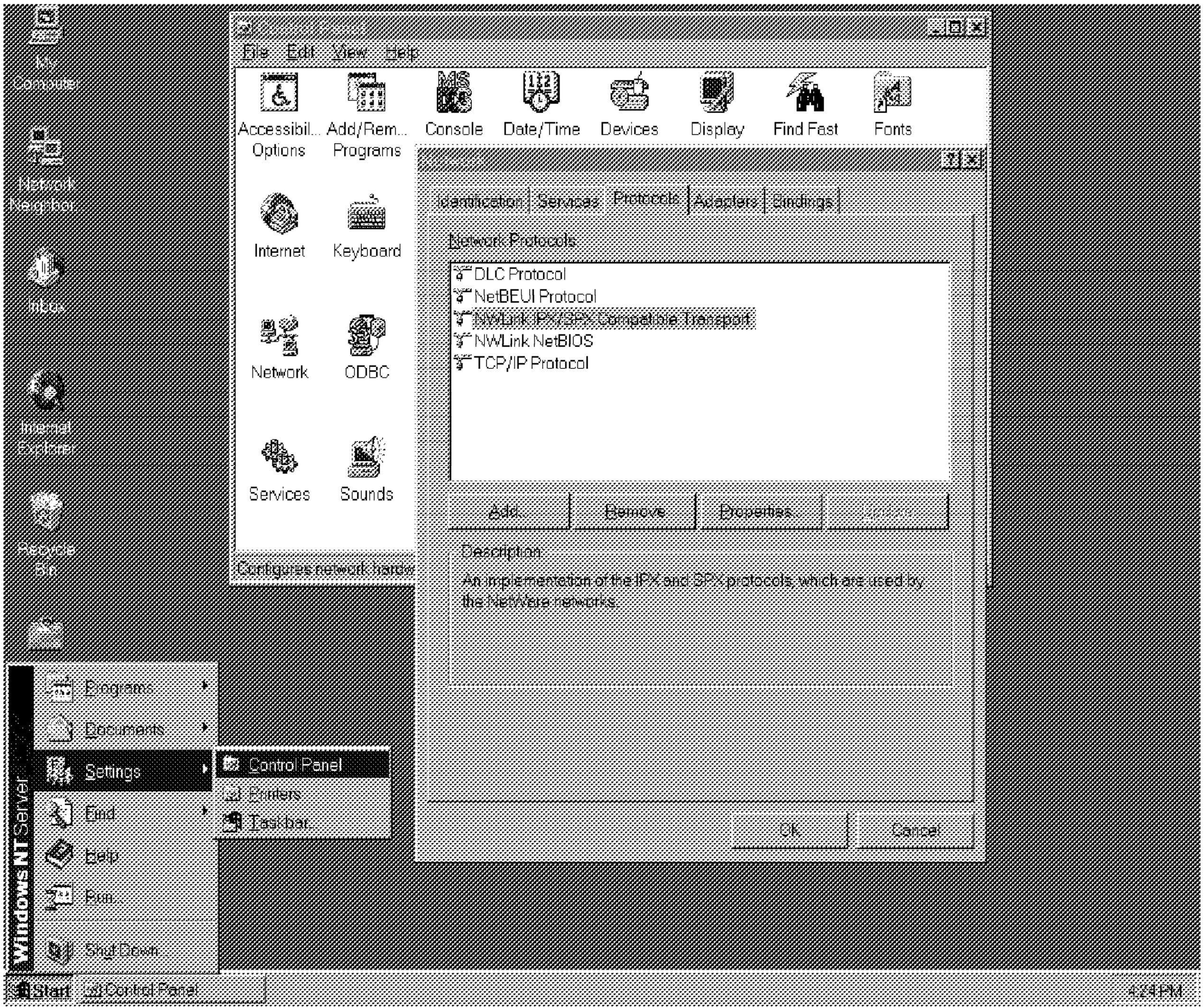


Figure 82. Verifying the IPX/SPX Support

4.4.1 Setting the Environment Variable DB2COMM on the Server

There is one communication environment variable that must be set for all protocols. The environment variable is DB2COMM. DB2COMM determines which protocol(s) will be enabled when the database manager is started. DB2COMM can be set to multiple values if you need the database server to support more than one communications protocol simultaneously.

To enable IPX/SPX protocol support on the DB2 NT Server, you need to set the DB2COMM variable in the System Environment Variable section to include IPX/SPX. IPX/SPX is a part of the base Windows NT operating system. Ensure that you have installed the NWLink IPX/SPX network software.

To set the DB2COMM variable in the Windows NT registry:

1. Go to the Control Panel icon on the Windows NT machine.
2. Select the **System icon** by double-clicking.
3. From the Systems Control panel in the System Environment Variables section, do the following:

- If the environmental variable DB2COMM does not exist:
  - Select any environment variable and change the name in the Variable box to DB2COMM.
  - Enter IPXSPX in the value box.
  - Click on the **Set** button. The variable will be added in alphabetical order.
- If the environmental variable DB2COMM already exists:
  - Select **DB2COMM** from the System Environment Variable section.
  - In the value box, add IPXSPX to the already existing value(s), separated by a comma(,).
  - Click on the **Set** button. The variable will be updated to the new value.

4. Select **OK** to exit.

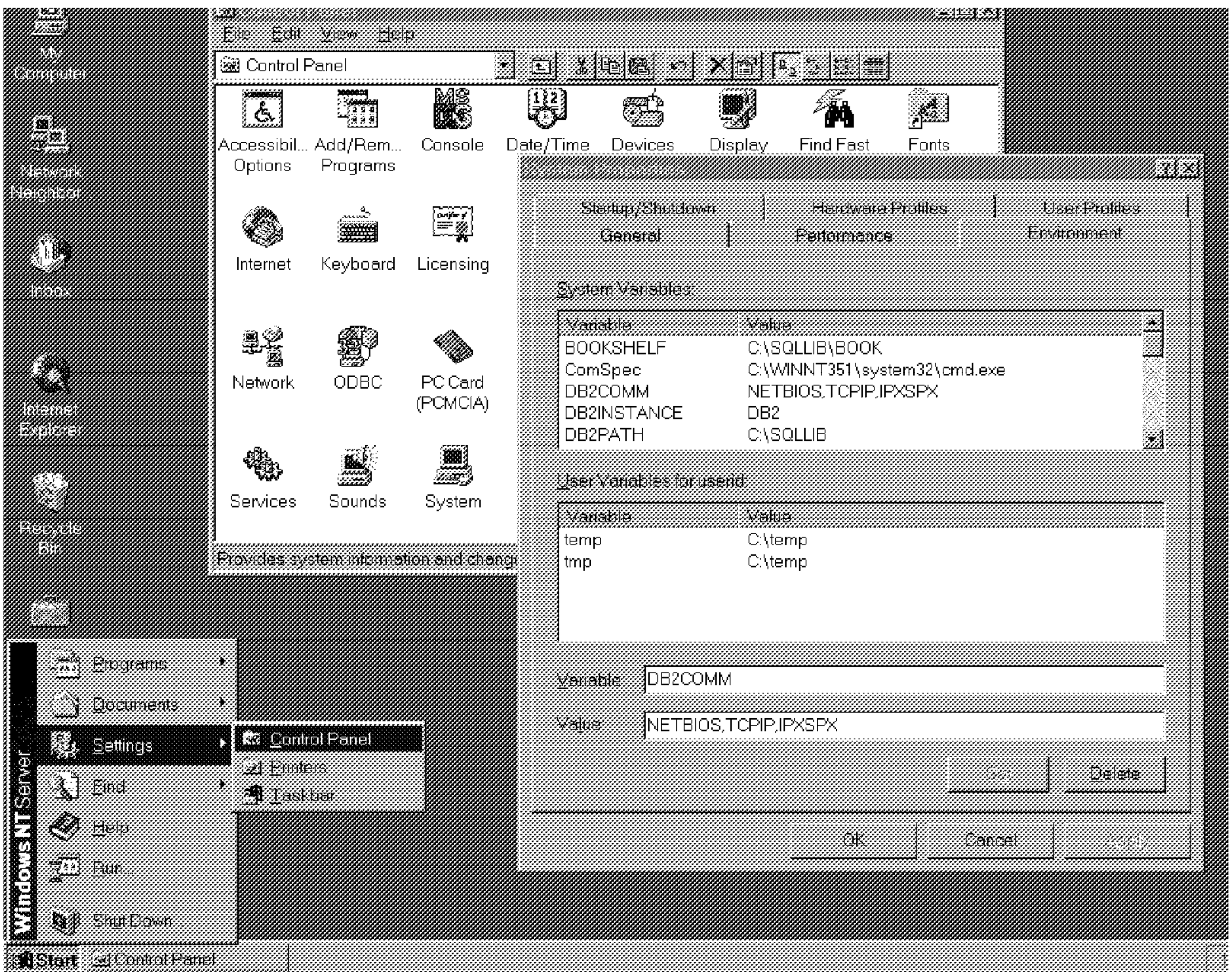


Figure 83. Setting the DB2COMM Variable

Figure 83 shows setting the DB2COMM variable to enable communications through NetBIOS,TCP/IP and IPX/SPX.

***** Tip *****

You can determine the setting for DB2COMM variable by using the echo command. To check the value of DB2COMM variable, enter the following on the command prompt:

```
echo %DB2COMM%
```

4.4.2 Updating the Database Manager Configuration File on the Server

This section covers the steps you need to take to update the database manager configuration file on the DB2 for NT Server when using IPX/SPX.

4.4.2.1 Determining the NetWare Internetwork Address

DB2 NT Server supports IPX/SPX support through Direct Addressing. When Direct Addressing is used, DB2 clients can communicate directly with the DB2 Server, bypassing the NetWare File Server (if it is present on the network). When Direct Addressing is used, the FILESERVER and OBJECTNAME in the database manager configuration should be set to '*'. (The FILESERVER and OBJECTNAME parameters are used only when you want to use IPX/SPX support using File Server Addressing, which is not supported by DB2 NT Server). The db2ipxad utility should be executed on the DB2 Server to determine the Internetwork Address for the server machine. Figure 84 on page 127 is an example of using the db2ipxad utility to determine the Internetwork Address for a DB2 Server.

***** Note *****

A NetWare Internetwork Address has the following form:

<8 byte net id>.<12 byte node id>.<4 byte socket number>

For example - 00007007.0004AC947CF4.879E

The NetWare Internetwork Address can be retrieved by running the db2ipxad utility at the DB2 Server. The utility is located in %DB2PATH%miscdb2ipxad.exe

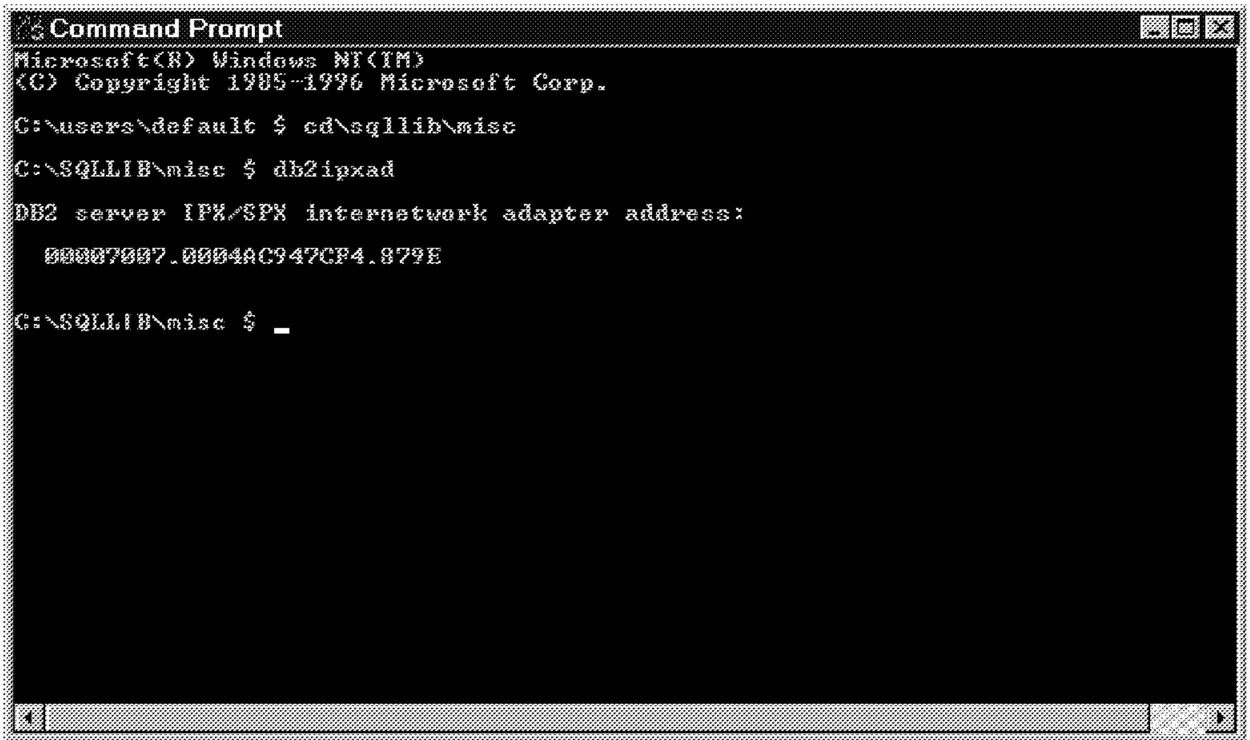


Figure 84. NetWare Internetwork Address

This address will be used at each DB2 client (which uses IPX/SPX) to locate the DB2 Server. The internetwork address will be supplied as the server name during the CATALOG NODE command at each client workstation.

4.4.2.2 Selecting IPX_SOCKET Value

The IPX_SOCKET parameter in the database manager configuration defines the IPX/SPX socket number used for communication. You select a hex number representing the connection end-point identifier that must be unique for each DB2 server instance and unique among all IPX/SPX applications running on one DB2 server. If you have more than one instance on the server, select unique socket values for each instance in the range of 879E (default) to 87A2. DB2 has registered well-known sockets with Novell in that range. If you run more than five instances on the server machine, you must prevent socket collisions by choosing socket numbers that are not 0X000 or which are in the dynamic socket range 0X5000 to 0X7FFF.

For example, to configure a DB2 Server using Direct Addressing, use one of the following:

```
UPDATE DATABASE MANAGER CONFIGURATION USING FILESERVER * OBJECTNAME *
IPX_SOCKET 87A0
UPDATE DBM CFG USING FILESERVER * OBJECTNAME * IPX_SOCKET 87A0
```

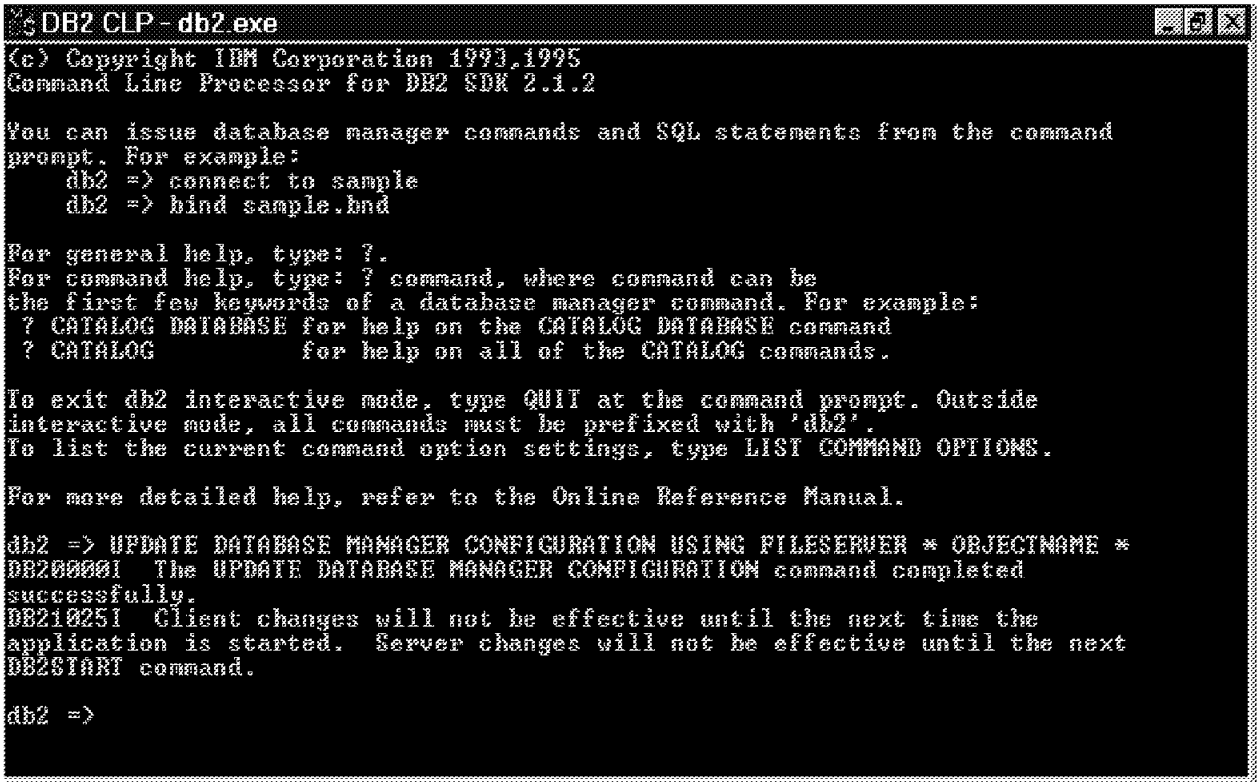


Figure 85. Updating the Database Configuration using DB2 CLP

Figure 85 is an example of successfully updating the database manager configuration file to enable IPX/SPX communication (using Direct Addressing) for the DB2 Server.

To verify that the change has been updated, issue:

```
GET DATABASE MANAGER CONFIGURATION
GET DBM CFG
```

Figure 86 on page 129 shows the parameters in the configuration file. Notice that the IPX/SPX File Server name parameter (FILESERVER) and IPX/SPX DB manager object name (OBJECTNAME) have been updated to reflect their new values.

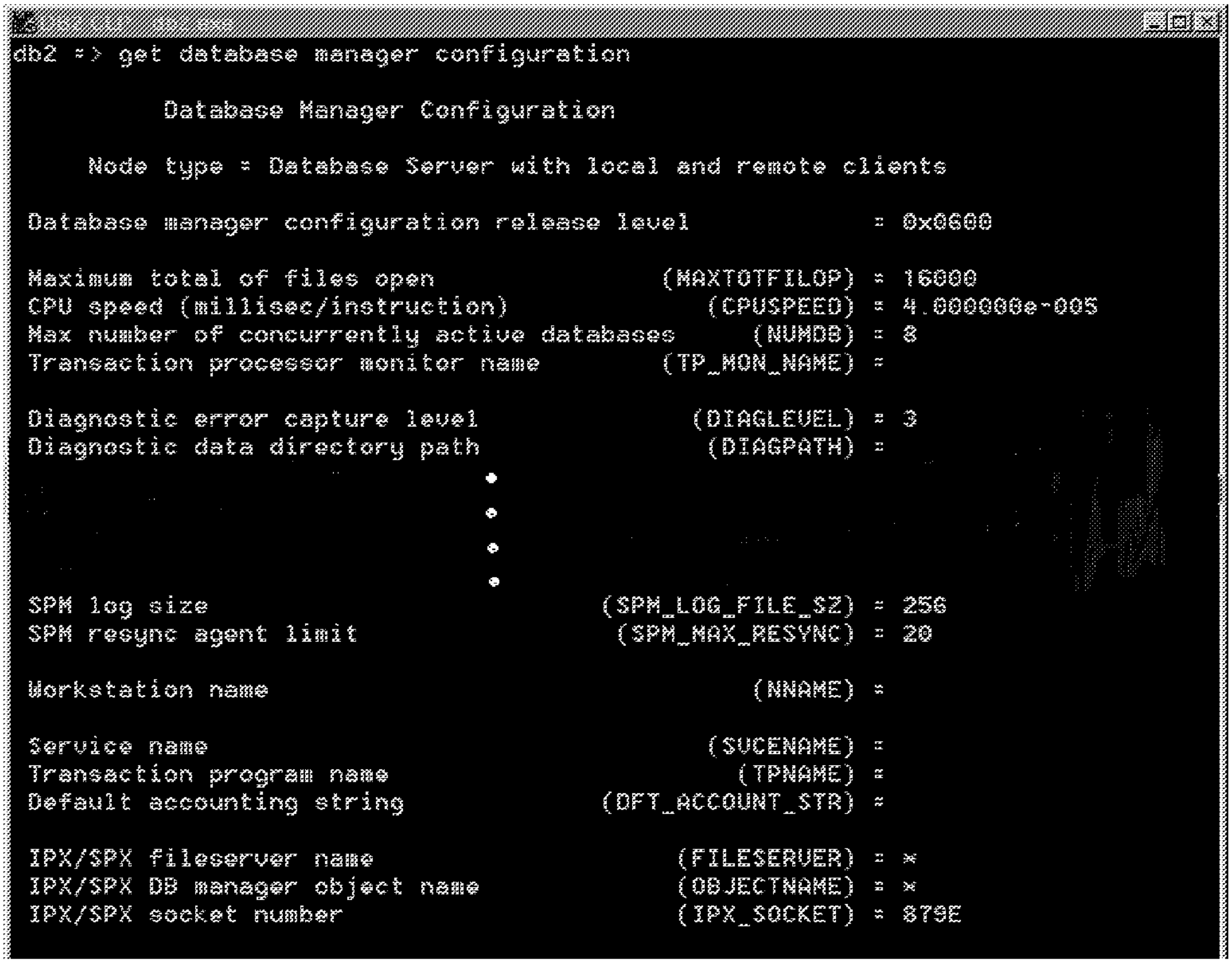


Figure 86. Verifying Database Manager Configuration

To make the changes to the database manager configuration file take effect, you must stop and restart the database manager (instance). To start a DB2 instance, log on with a user that has administrative authority. You can start or stop an instance on the DB2 NT Server in one of two ways:

- Select the **START / STOP** button for the instance you want in the Services dialog in the Control Panel.
- Use `net start <instance name>` and `net stop <instance name>` in any command window.

```
net start <instance-name>
net stop <instance-name>
```

When you start the database manager, you must also start the DB2 Security Service using the following:

```
net start DB2NTSECSERVER
```

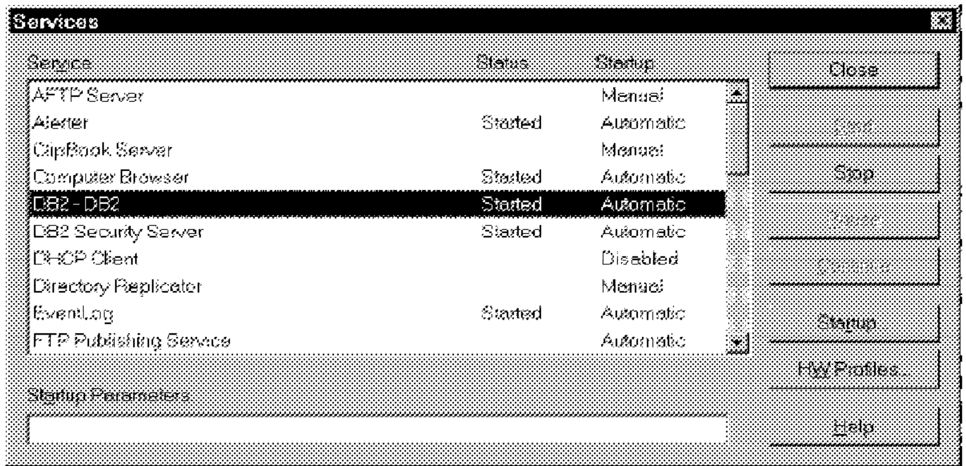


Figure 87. Start / Stop DB2 Instance

Figure 87 illustrates the Services window and shows that the DB2 instance (by the name DB2) has been started. You can also see that the DB2 Security Server which will be used for validating users is running. The startup options for both DB2-DB2 and DB2 Security Server have been modified from `Manual` (which is the default) to `Automatic`. This results in both services starting automatically when the server is booted. For more details on modifying Windows NT Services, refer to Windows NT documentation or consult a Windows NT Administrator.

Figure 88 on page 131 and Figure 89 on page 132 are for reference. Figure 88 on page 131 shows a summary of the necessary steps to configure a DB2 for NT database server using IPX/SPX. The items that are checked indicate those steps necessary on the database server. Figure 89 on page 132 shows the correlation between client and server when using IPX/SPX for a communication protocol with DB2. The following is a summary of those tasks:

- CAE or SDK or OS/2, DOS, WIndows, Windows 95, or Windows NT, along with DB2 for OS/2 and DB2 for Windows NT clients may use the IPX/SPX protocol to communicate with a DB2 for NT Server.
- On the server, the `DB2COMM` environment variable must be updated to include IPX/SPX. (`DB2COMM=IPXSPX`)
- The IPX/SPX fileserver name (`FILESERVER`), IPX/SPX DB manager object name (`OBJECTNAME`), and IPX/SPX socket number (`IPX_SOCKET`) on the server must also be updated. These parameters are found in the database manager configuration file. Be default they are all blank, except for the `IPX_SOCKET`, which is set to `879E` by default.
- If file server addressing is being used (not available on Windows NT platform), then the DB2 Server must be registered on the NetWare File Server. This is NOT required if all the clients accessing the server are using Direct Addressing.
- When cataloging the client node directory, the server's database manager IPX/SPX fileserver name and IPX/SPX database manager object name must be specified in order to connect with the server through IPX/SPX.
- When cataloging the system database directory on the client, the alias that the server machine uses in its system database directory when referencing the DB2 database must be known.

IPX/SPX Connectivity Checklist

Update environment variables on server and client

- Server must set DB2COMM = IPXSPX

Update DBM CFG file on server:

- Specify file server name
- Specify object name
- Specify socket number

Catalog Node Directory on client:

- Specify IPX/SPX protocol
- Client must know server's DBM file server name
- Client must know server's DBM object name
- Client must know server's DBM socket number

Catalog System Directory Database on client:

- Client must know server's system database directory alias name

Figure 88. IPX/SPX Connectivity Server Checklist

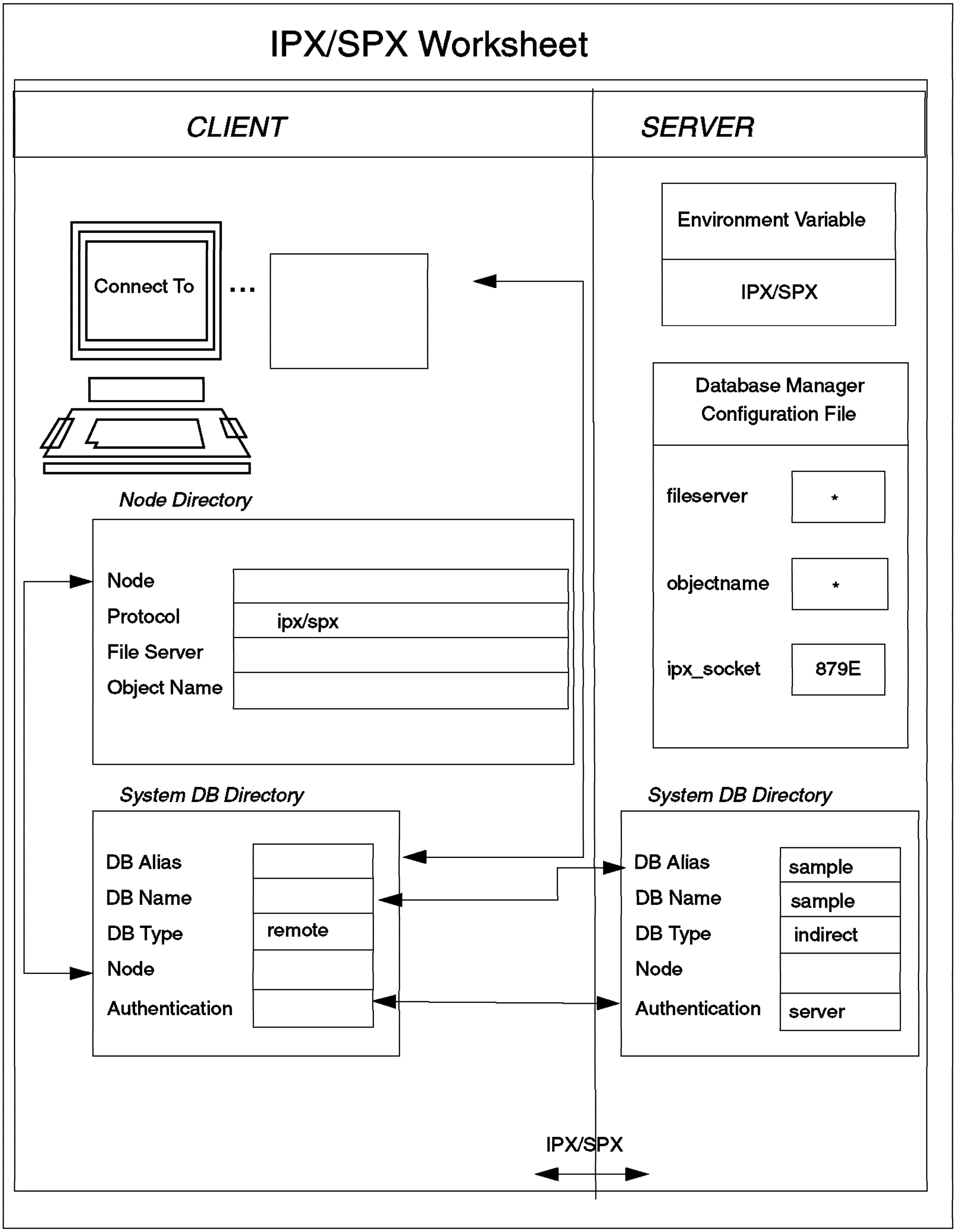


Figure 89. Completed IPX/SPX Worksheet

4.5 Using the NT Server as a LAN Gateway

It is possible for DB2 clients to use the DB2 NT Server as a gateway to access databases on other DB2 servers on a LAN. This may be useful if the client and the target server support different communication protocols.

To understand this in a better way, let's consider a scenario where an organization has two DB2 Servers in the network, one running on Windows NT and the other on AIX. DB2 clients on the different platforms access the DB2 NT server using NetBIOS, which is the only supported protocol available on the client workstations. The Windows NT Server is configured to support both NetBIOS and TCP/IP protocol. If the DB2 clients (which only have NetBIOS support installed) need to access a database on the AIX Server (which only has TCP/IP as the installed communications protocol), they can route the connections through the DB2 NT Server. In short, the Windows NT Server acts as a gateway for the DB2 clients to access database on the DB2 AIX Server. Figure 90 illustrates the scenario.

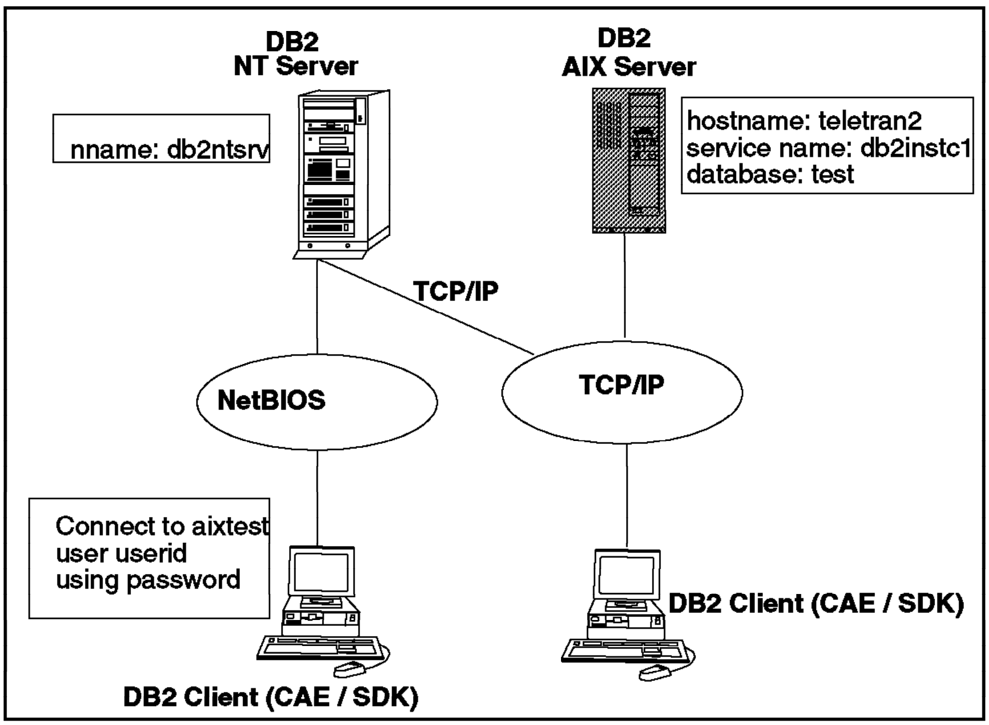


Figure 90. Using the DB2 NT Server as a Gateway

The clients route the connection through the DB2 for Windows NT server by creating the appropriate directory entries. By doing this, they will be able to access the data on the DB2 AIX Server transparently. This is also referred to as a *double hop*.

The double hop is achieved by creating appropriate directory entries. In the case of the above example, the following values are used:

```
DB2 for Windows NT nname: db2ntsrv
DB2 for AIX hostname: teletran2
DB2 for AIX service name: db2inst1c
Target database on AIX: test
```

The following entries need to be cataloged:

At the DB2 NT Server:

```
catalog tcpip node aixsrv remote teletran2 server db2inst1c
catalog database test as test at node aixsrv
```

At the DB2 Client:

```
catalog NetBIOS node aixsrv remote db2ntsrv adapter 0
catalog database test as aixtest at node aixsrv
```

For more details on how to catalog a remote node and database, refer to Chapter 5, “Enabling Remote Clients” on page 151.

When the client connects to the *aixtest* database, the connection request will be passed through the DB2 NT Server (*db2ntsrv*) using NetBIOS, and then on to the DB2 AIX Server (*teletran2*) using TCP/IP.

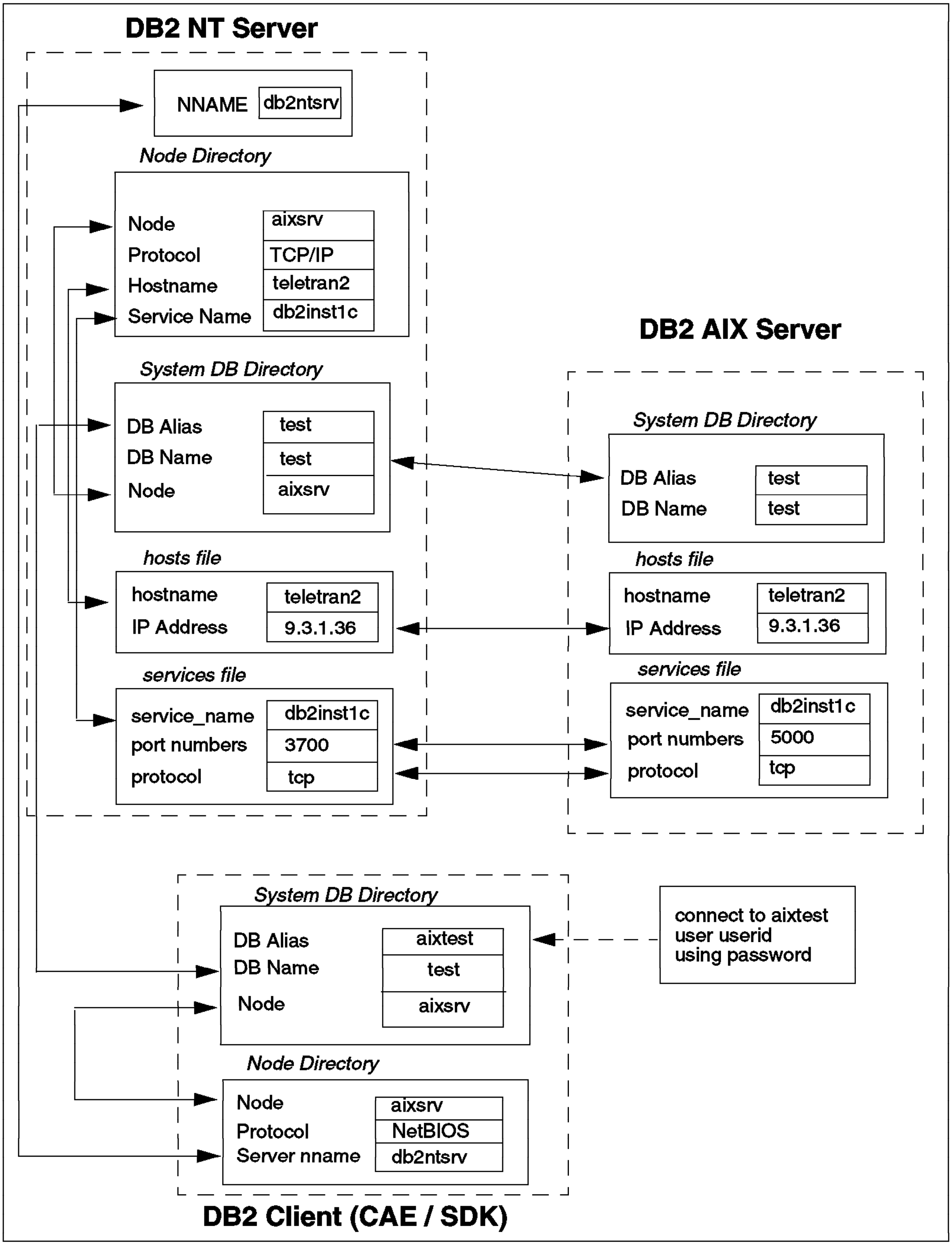


Figure 91. Directory Entries for DB2 NT Server Acting as Gateway

4.6 DDCS Gateway for Windows NT

Many organizations manage large volumes of data using DB2 on host and mini-computer environments such as MVS, VM, VSE, and AS/400. The users of the systems are more than likely to be using personal computers and UNIX workstations for day-to-day jobs. This caused a problem until Distributed Database Connection Services (DDCS) was available. DDCS can be used for connecting personal computers and UNIX-based workstations to enterprise systems.

DDCS allows applications running in environments that include OS/2, DOS, Windows, and UNIX to access data stored in DB2 and other non-relational databases that are running on MVS, VSE, VM, and AS/400. Client applications can work with data on enterprise systems transparently, as if the data was contained within a DB2 Common Server environment. DDCS makes this possible by implementing the DRDA protocol to allow personal computer applications to work with DB2 Servers on enterprise systems.

DDCS for NT is available in two versions, single-user and multi-user gateway. The single-user version allows only local clients on the DDCS workstation to connect to the data. The multi-user gateway version allows multiple clients, both local and remote, to connect to the host. Configuring a multi-user gateway is the subject of this chapter.

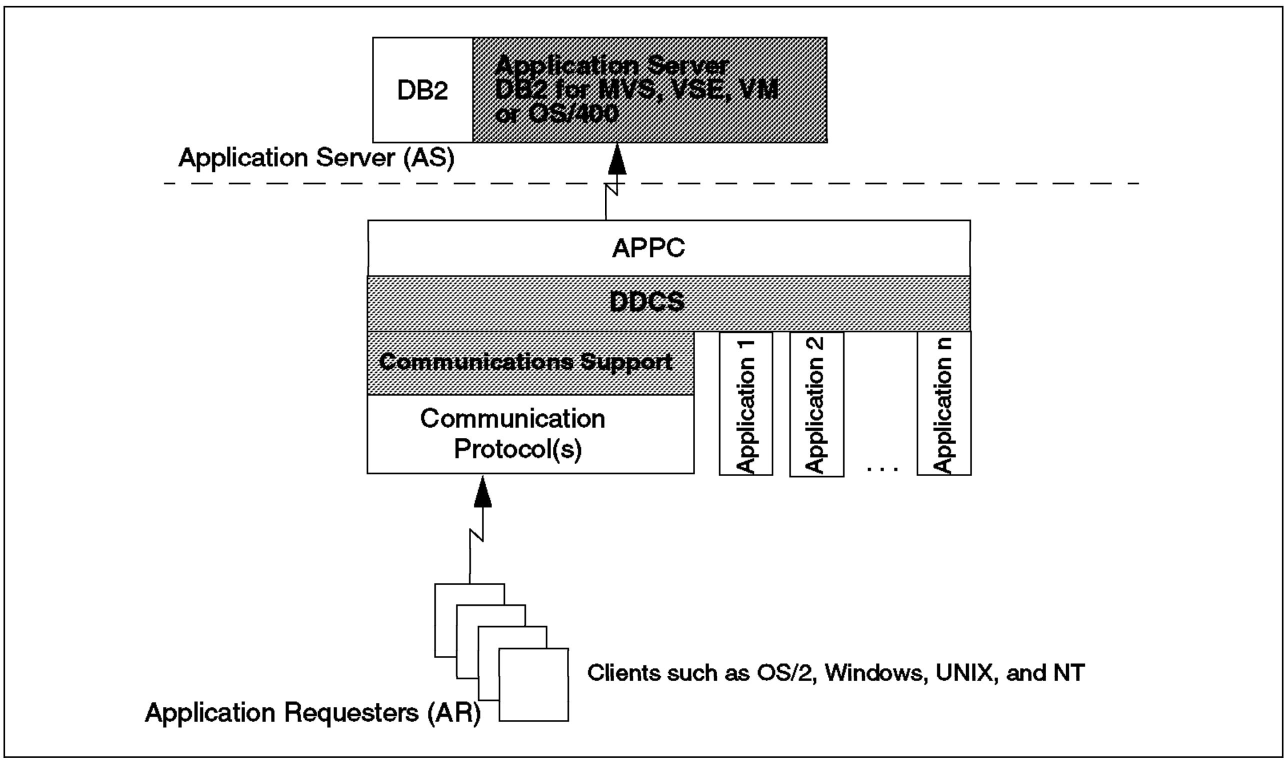


Figure 92. DDCS Multi-User Gateway

4.7 Configuring a Windows NT Gateway

Before using DDCS Multi-User Gateway for NT, the following needs to be installed:

- Windows NT 3.50 or higher
- Microsoft SNA Server 2.11 or higher

To use DDCS, you first need to configure SNA Server for NT to enable APPC communication between the DDCS Gateway and the host machine you wish to access. You will need to collect certain information from the host machine and also a number of communication definitions defined for your workstation.

This section supplies step-by-step details on how to configure DDCS Multi-User Gateway for NT to communicate with an MVS system.

The following assumptions are made:

- A token-ring is being used for communications.
- The DDCS Multi-User Gateway product is installed.
- SNA Server is installed.
- Communication definitions are available for use in SNA Server.
- The DRDA Application Server (AS) is a host machine rather than a PC or UNIX-based system. If a PC or UNIX-based system, it is recommended that the CAE component, not DDCS, be used.
- The network you are using is fairly straightforward.
- You have access to a Virtual Telecommunications Access Method (VTAM) administrator and/or a database administrator.

4.7.1 DB2 for MVS

To use DDCS to communicate with MVS, the DRDA Application Server (MVS) must be configured. We will assume that in this case, MVS is already configured for use as a DRDA AS.

To help you with the configuration, we suggest you fill out the following worksheet prior to attempting the configuration. The values in the worksheet are obtainable from both the VTAM administrator and the database administrator.

4.7.1.1 NT Configuration Worksheet

This worksheet can be used to map your NT configuration values to the appropriate parameter.

Item	Parameter	Example	Your Value
1	Network ID	USIBMSC	
2	Control Point Name	SC03112	
3	Local Node ID	05D 03112	
4	Mode Name	IBMRDB	
5	Local LU Name	SC03112I	
6	Local LU Alias	SC03112I	

4.7.1.2 MVS Server Worksheet

Your MVS configuration values can be planned using this worksheet.

Item	Parameter	Example	Your Value
1	NETID	USIBMSC	
2	Controller Address	400008210210	
3	MODEENT	IBMRDB	
4	SSCP Name	SCG20	
5	APPL	NDCDB201	
6	LU	SC031112	
7	PU	SC03112	
8	IDBLK and IDNUM	05D 03112	
9	Remote TP Name	07F6C4C2 (default)	
10	Database Location	NDCDB201	

Some of the parameters used in the configuration worksheet have names that correspond to other, differently named parameters. For example *Network ID* specified in the NT Configuration Worksheet is called *NETID* in the MVS Server worksheet. Another example is the NT Configuration Worksheet *Mode Name* is the parameter name, whereas in the MVS Server Worksheet this is called *MODEENT*. SNA Server and VTAM use different names to represent the same parameter. When configuring DDCS be aware of these differences in parameter names.

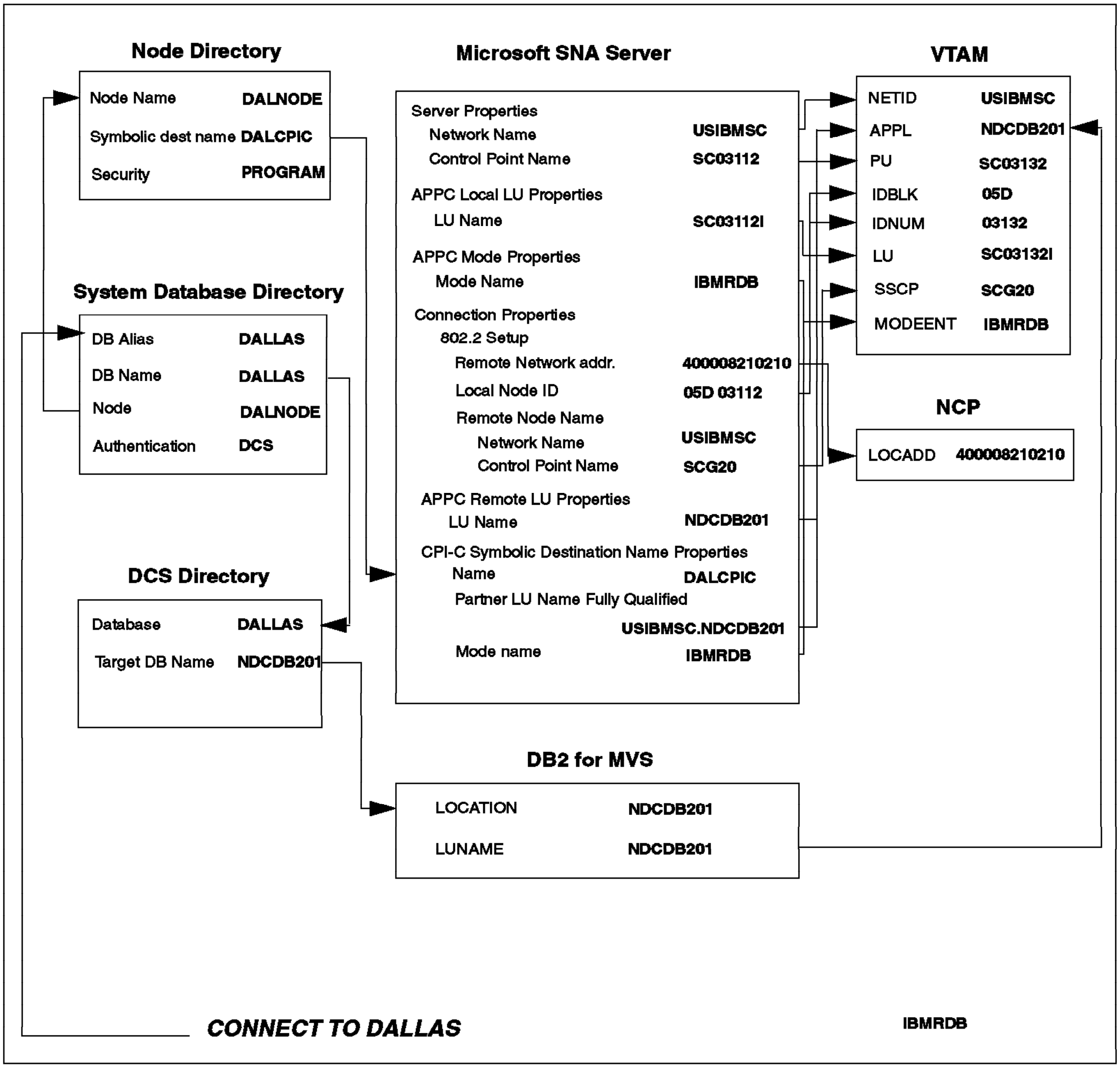


Figure 93. DDCS for NT Connectivity Diagram

Figure 93 illustrates the relationship between the different parameters and where the parameters are referenced: SNA Server, DDCS directories, DRDA AS, and VTAM:

By using the information in Figure 93 and referencing the worksheet, you can see where each parameter is defined and referenced.

If you already have a SNA Server configuration, you should have some of the configuration parameters already available to you. The following table outlines who should be able to supply you with the relevant information you will need:

Item	Parameter	Example	Contact
1	Network ID (NETID)	USIBMSC	VTAM Administrator
2	Local Node Name (PU)	SC03112	VTAM Administrator
3	Local Node ID (IDBLK and IDNUM)	05D 03112	VTAM Administrator
4	Mode Name (MODEENT)	IBMRDB	Recommended Name
5	Local LU Name	SC03112I	VTAM Administrator
6	Local LU Alias	SC03112I	You Choose
7	Lan Dest. Address (Controller Address)	400008210210	VTAM Administrator
8	Partner Node Name (SSCP Name)	SCG20	VTAM Administrator
9	Partner LU Name (APPL and LUNAME)	NDCDB201	MVS DB Administrator
10	LOCATION	NDCDB201	MVS DB Administrator

***** Note *****

Be aware that although in our example the Partner LU name and LOCATION name are the same, this may not be the case in your settings.

4.7.1.3 Configuring the DDCS Workstation

The following steps take you through the configuration:

1. Ensure that you have a Link Service setup within SNA Server. If not, please refer to SNA Server documentation on how to create one.
2. Open the SNA Server Admin program.

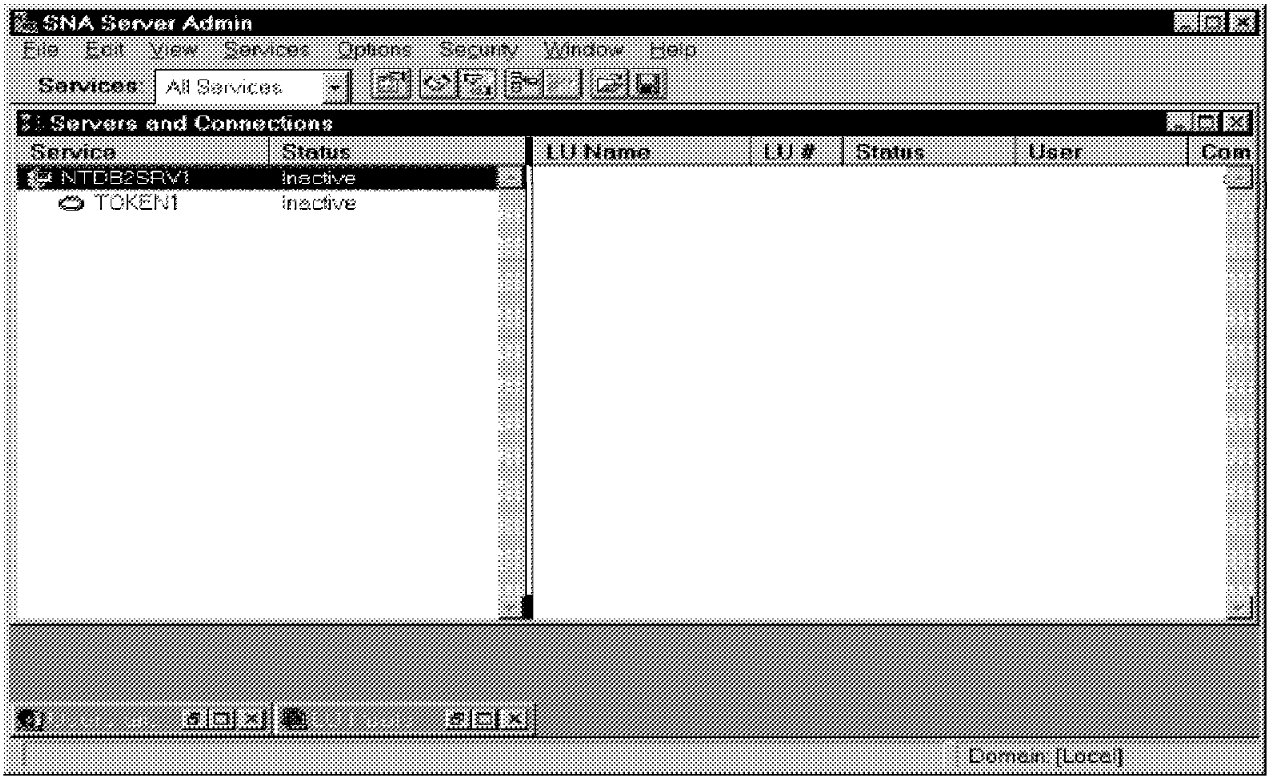


Figure 94. SNA Server Admin Panel

3. From the Server and Connections panel, select the server entry:
   - Now go to the Services menu and select **Properties**. You will be presented with the Server Properties panel.
   - In the Server Properties panel, enter the value you selected in item 1 from the NT Configuration Worksheet in the Network Name box and item 2 in the Control Point Name. Click on the **OK** button.

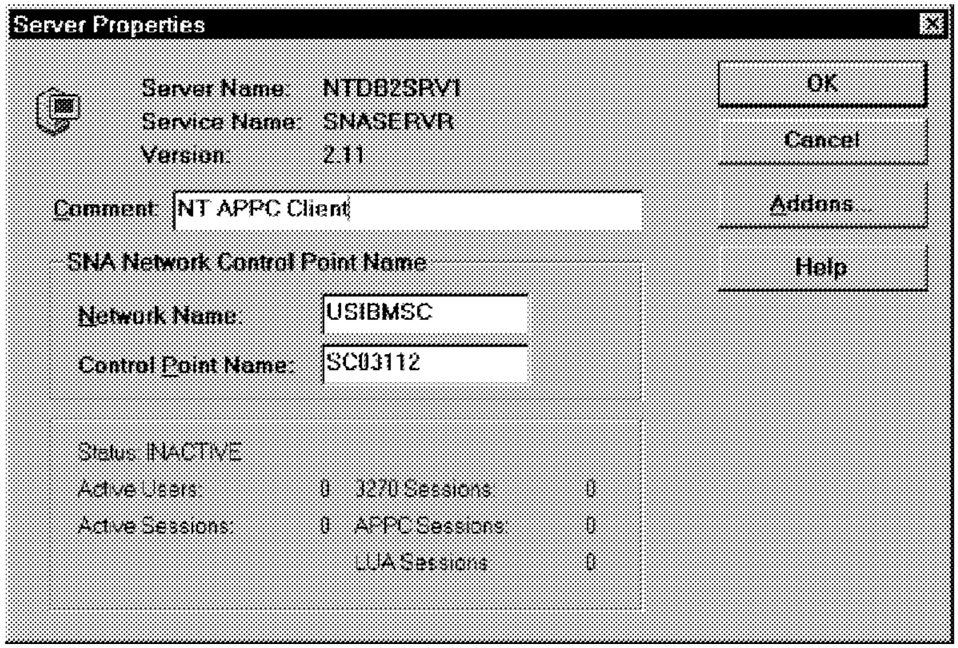


Figure 95. Server Properties Panel

4. The next step is to assign a local APPC LU to the Server:
   - From the Services and Connections panel, select the server.
   - From the Services menu, select **Assign LUs**; from the Insert LU panel, select **APPC** [local] and click on the **OK** button.
   - The New APPC LU Properties panel will appear.
   - Select an LU 6.2 Type of Independent or Dependent. This should usually be Independent. Check with the VTAM Administrator.
   - For the LU Alias, enter the value you selected in item 6 of the 4.7.1.1, "NT Configuration Worksheet" on page 137.
   - For Network Name, enter the value you selected in item 1 of the NT Configuration Worksheet.
   - For the LU Name, enter the value you selected in item 5 of the NT Configuration Worksheet.
   - Check the Enable Automatic Partnering check box. Once complete, click on the **OK** button.

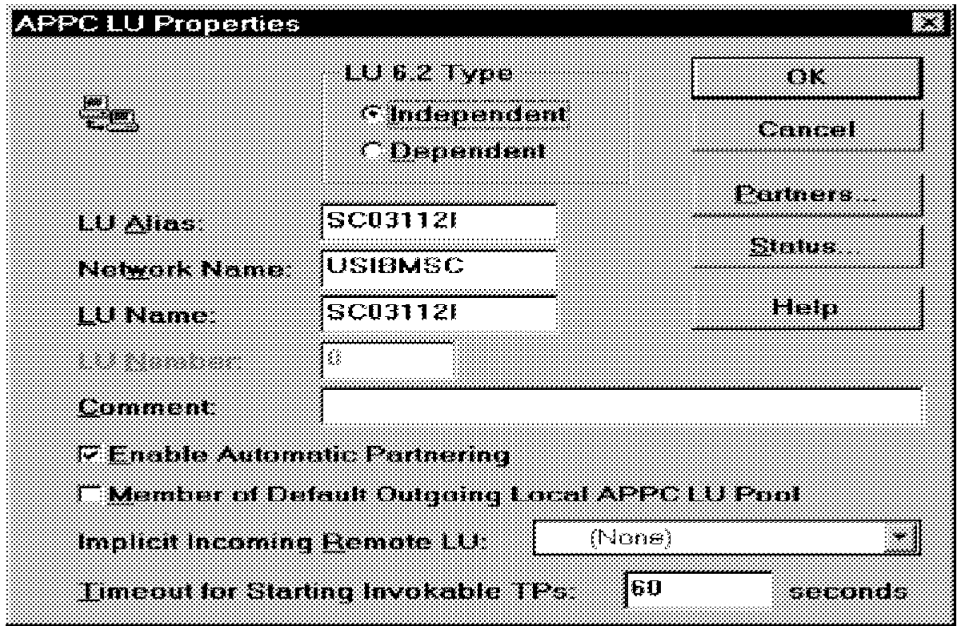


Figure 96. APPC LU Properties Panel

5. Next you need to specify the local LU the client will use when connecting through APPC. This can be done in one of two ways:

   - **Assigning a Default APPC LU to a User or Group -** Ensure the User/Group has an account on the SNA Server. To add a User/Group to the list used by SNA Server:

     – In the Users and Groups window, select any line.

     – From the View menu, choose **Configured Users**.

     – From the Users menu, choose **New User**; the Add Users and Groups panel will appear.

     – In the List Names From box, select the domain from which to list users or groups who have network accounts, or select the name of the local server.

     – In the Names box, from the list of users and groups who have accounts, select the one you want to add.

     – Click on the **Add** button.

     – Click on the **OK** button.

     – Save the configuration.

     To assign a default APPC Local LU to the user or group:

     – In the Users and Groups window, from the View menu, choose **Configured Users**.

     – Select the user of group.

     – From the Users menu, choose **Properties**; the User Properties or Group Properties panel will appear.

     – In the Default Local APPC LU box, select a local APPC LU.

     – Click on the OK button.

     – Save the configuration.

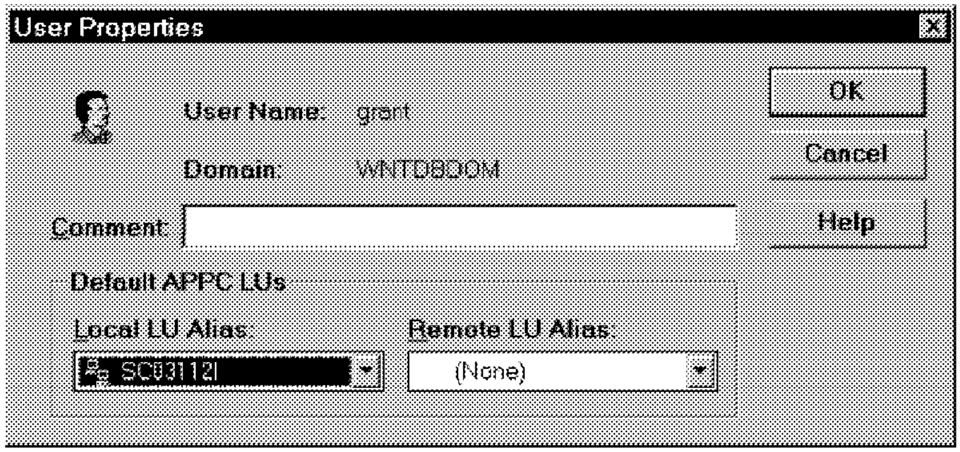


Figure 97. User Properties Panel

6. **Setting up a Default Outgoing Local APPC LU Pool:**

   - To add a local APPC LU to the default outgoing local APPC LU pool, check the box for Member of Default Outgoing Local APPC LU Pool when configuring the APPC Local LU Properties.

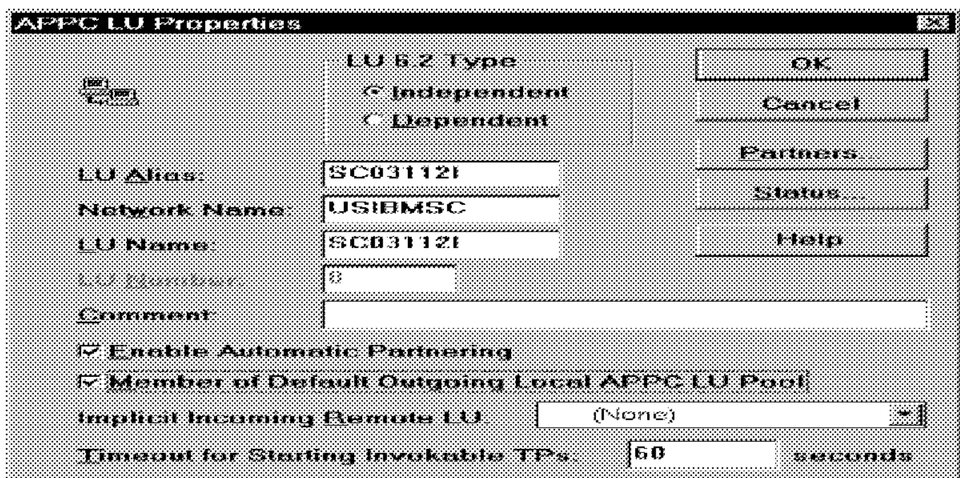


Figure 98. APPC LU Properties Panel

7. The last step in this section is defining a mode. You can use an IBM-defined mode or one appropriate to your specific application:

   - From the Server and Connections panel, select either a local or remote APPC LU with which the mode is to be associated.
   - From the Services menu, select **Properties**. One of the following panels will appear:
     – APPC LU Properties
     – APPC Remote LU Properties
   - From the panel, click on the **Partners** button. The LU 6.2 Partner LUs panel will appear
   - From this panel, select the **Modes** button.The APPC Mode Properties panel will appear.

- In this panel, in the Mode Name box, enter the value you selected for item 4 in the NT Configuration Worksheet. For the other parameters in this panel, select the values that are best for your application. The values shown in the following figure are acceptable for use and were used in our configuration.

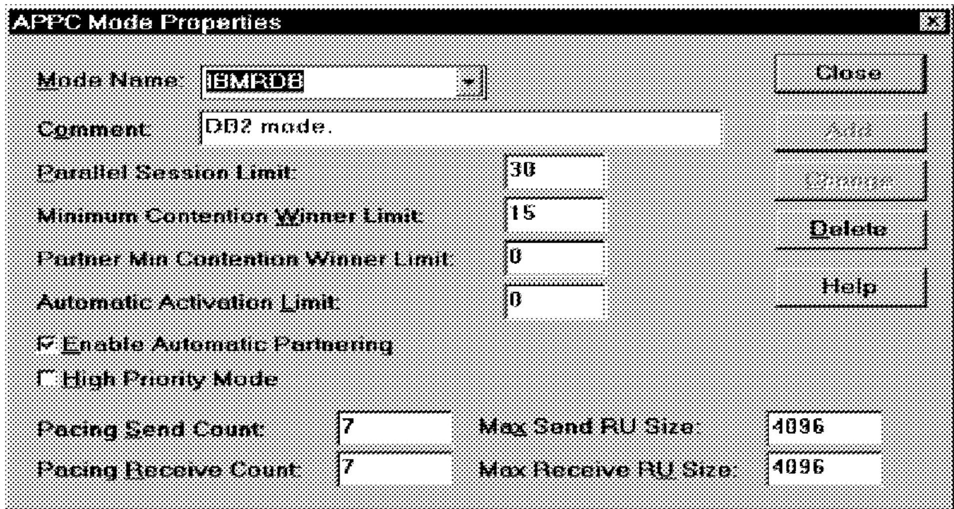


Figure 99. APPC Mode Properties

This completes the configuration of the DDCS workstation. The next two sections discuss connecting to a server and enabling remote clients for connection.

4.7.2 Server Connection Configuration

This section shows how to configure DDCS for connection to remote DRDA Application Servers. As in the previous section, it is recommended you complete the following worksheet before attempting the configuration.

4.7.2.1 NT to Server Connection Worksheet

This woksheet can be used to map your Server configuration values to the appropriate parameter.

Item	Parameter	Example	Your Value
1	Link Name	TOKEN1	
2	LAN Destination Address	400008210210	
3	Symbolic Destination Address	DALCPIC	
4	Control Point Name	SCG20	
5	Partner LU Name	NDCDB201	
6	TP Name	07F6C4C2 (default)	
7	Target Database Name	NDCDB201	

The following steps take you through the configuration for connection to a remote DRDA AS:

1. From the Servers and Connections panel, select the connection.
2. From the Services menu, choose **New Connection**. The Insert Connection dialog box will appear.
3. Select the type of connection; usually, this should be 802.2. Click on the **OK** button.
4. The next panel is the Connections Properties panel. Specify the value you selected in item 1 of the name for Connection Name and set Link Service to the appropriate link service.
5. Check the following check boxes:
   - Host System
   - On Server Startup (On Demand can also be used)
   - Out Going Calls

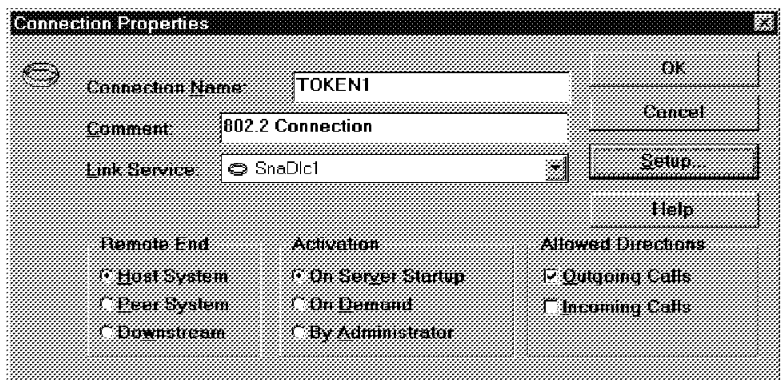


Figure 100. Connection Properties Panel

6. Click on the **Setup** button and the 802.2 Setup panel is displayed.
7. On the 802.2 Setup panel, specify the following values:
   - For Remote Network Address, enter the value you selected in item 2 of the NT to Server Configuration Worksheet.
   - For Local Node ID, enter the value you selected in item 3 of the 4.7.2.1, “NT to Server Connection Worksheet” on page 145.
   - For the Network Name, enter the value you selected in item 1 from the 4.7.1.2, “MVS Server Worksheet” on page 138.
   - For the Control Point Name, enter the value you selected in item 4 of the NT to Server Configuration Worksheet.
   - Click on the **OK** button

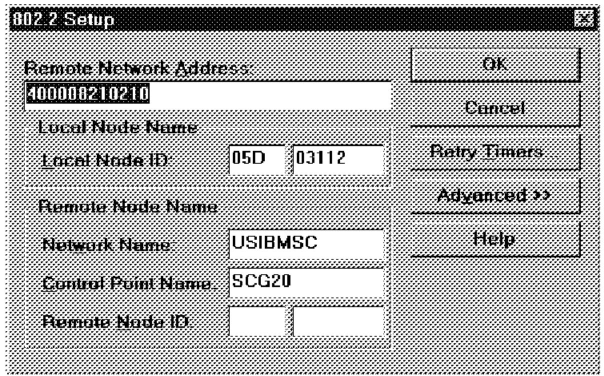


Figure 101. 802.2 Setup Panel

8. Now you must assign a remote LU to the connection.
9. From the Servers and Connections panel, select the connection.
10. From the Services menu, select **Assign LUs**. The Insert LU panel will appear.
11. From the Insert LU panel, select **APPC** [**Remote**]. Click on the **OK** button.
12. You will now be presented with the New APPC Remote LU Properties panel:
    - For LU Name, enter the value you selected in item 5 of the NT to Server Configuration Worksheet. Also choose a value for the LU Alias.
    - For Network Name, enter the value you selected in item 1 of the NT Configuration Worksheet.
    - Check the **Enable Automatic Partnering** checkbox.
    - Click on the **OK** button

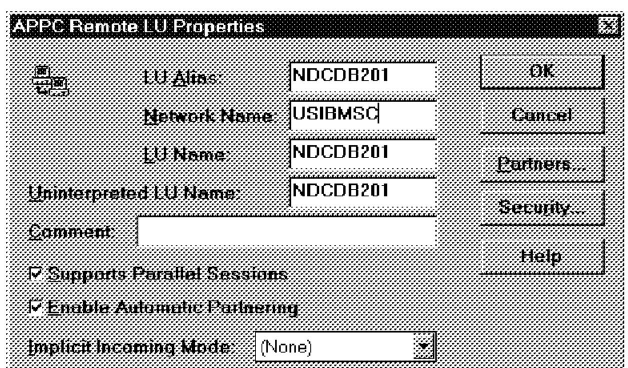


Figure 102. APPC Remote LU Properties Panel

13. The final step is to configure the CPI-C properties:
    - From the options menu choose **CPI-C**, the Configure CPI-C Names panel will appear.
    - Click on the **Add** button and you will be presented with the CPI-C Symbolic Destination Name Properties panel.
    - For Name, enter the value you selected in item 3 of the NT to Server Connection Worksheet.

- For Partner TP Name, click on the **SNA Service TP** button, and enter the value you selected in item 6 of the NT to Server Connection Worksheet.
- For Partner LU Name, enter the remote LU alias in the Alias box after selecting the **Alias** button.
- For Mode Name, choose the mode created in the Configuring the DDCS Workstation section previously.
- Conversation security is best set to None. This does not mean you will have no security, but indicates that security will be determined from the node directory entry.
- Click on the **OK** button

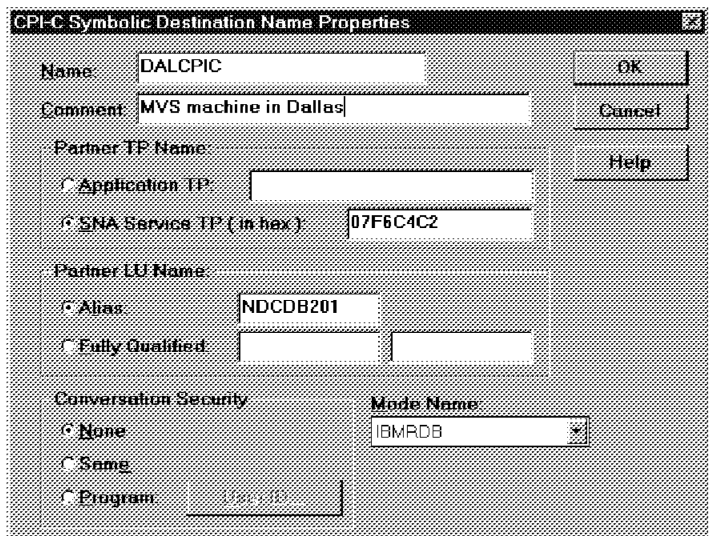


Figure 103. CPI-C Symbolic Destination Name Properties

This completes the configuration for connection to DRDA servers. The next section discusses the configuration of DDCS to enable remote clients to connect.

4.7.2.2 Configuring DDCS for Remote Clients

This section shows how to configure DDCS for connections from remote clients. As in the previous sections, it is recommended that you complete the following worksheet before attempting the configuration.

1. If you are using APPC remote clients, please note the following:

> **Note**
>
> Only one instance of DB2 can support down-level DB2 for OS/2 Version 1 clients.
> These clients use the hard-coded service TP name x'07'6DB.
> The instance is specified by setting the DB2SERVICETPINSTANCE environment variable to the instance name.

2. TCP/IP Connections

Item	Parameter	Example	Your Value
1	Connection Service Name	db2inst1c	
2	Interrupt Service Name	db2inst1i	
3	Connection Port	5000	

If you wish to use TCP/IP connections from remote clients, you need to add two entries in the TCP/IP Services file.

- The first entry is the connection service name, as specified in item 1 of the TCP/IP Connections Worksheet. This is followed by the connection port and TCP.
- The second entry is the interrupt service name, as specified in item 2 of the TCP/IP Connections Worksheet. This is followed by the connection port plus 1 (in our example, this would be 5001) and TCP.
- Example:
- db2inst1c 5000/tcp
- db2inst1i 5001/tcpi

1. The next step is to configure the database manager:

- For APPC, issue the following CLP command where SERVER is the value you selected in the APPC Connections Worksheet:
- `update database manager configuration using tpname SERVER`
- For TCP/IP, issue the following CLP command where db2inst1c is the value you selected in item 1 of the TCP/IP Connections Worksheet:
- `update database manager configuration using svcename db2inst1c`
- For IPX/SPX, issue the following CLP command where netwsrv, db2inst1c and 879E are items on the IPX/SPX worksheet in our configuration.
- update database manager configuration using fileserver * objectname *
- ipx_socket 879E
- For NetBIOS, issue the following CLP command where db2inst1 is shown on the NetBIOS worksheet in our configuration.
- update database manager configuration using nname db2inst1

1. The final step is to set the DB2COMM environment variable to the protocols you plan to use:

- Stop Database Manager if it is running.
- Set DB2COMM to the protocols you plan to use. For example,
- DB2COMM=APPC,IPXSPX,NETBIOS,TCPIP
- Start the Database Manager.

4.7.2.3 Completing the Configuration

Once the configuration is complete you will need to:

1. Stop and start SNA Server.
2. Update the node directory, system database directory and DCS directory:

```
CATALOG APPC NODE DALNODE REMOTE DALCPIC SECURITY PROGRAM
CATALOG DATABASE DALLAS AS DALLAS AT NODE DALNODE AUTHENTICATION DCS
CATALOG DCS DATABASE DALLAS AS NDCDB201
```

3. Configure any remote clients
4. Connect to DRDA Server and bind the utilities and applications.

Chapter 5. Enabling Remote Clients

This chapter discusses how to set up remote clients who want to access a remote DB2 for NT database server. The chapter describes the steps to configure the node for communicating with a DB2 for NT remote server using one of the supported communication protocols. After configuring the communication protocol, you will need to catalog remote node and database. Also included in this chapter are the necessary steps the client uses in the DB2 Client Setup or the Command Line Processor.

Once remote communication is enabled for DB2, a connection between an application on the client workstation and a target database can be established automatically when the application issues a CONNECT statement. The location of the database (local or remote) and the communications protocol to be used (if remote) are transparent to the user application.

The administrator of the client system sets up the system to communicate with the DB2 for NT database server. This setup will depend on the operating system and the communication protocol that the DB2 database server is using for the instance. The following is a table of supported communications protocol from a remote client to a DB2 for NT database server.

Table 9. DB2 NT Client/Server Connectivity Chart

DB2 Clients (CAE / SDK)	NetBIOS	IPX/SPX	TCP/IP	APPC
DOS	Yes	Yes	Yes	No
Windows 3.X	Yes	Yes	Yes	No
Win'95	Yes	Yes	Yes	No
Win NT	Yes	Yes	Yes	Yes
OS/2	Yes	Yes	Yes	Yes
AIX	No	No	Yes	Yes

Setting up the communications between a node and a server requires that certain tasks be performed at the server. You need to ensure that the tasks for each protocol have been carried out and that you have been provided with the necessary information before you proceed with the client configuration. Configuration for the server needs to be done only once for each protocol to be used. For more detail on configuring a DB2 for NT database server to support remote clients, see Chapter 4, “DB2 for NT Server Communication” on page 95.

After the communication configuration has been completed on the remote client workstation, the remote clients accessing the DB2 Server must also do the following:

1. Catalog a remote node.
2. Catalog the remote database.

***** Tip *****

To catalog a database or node directory, you must either have a SYSADM or SYSCTRL authority.

5.1 Connectivity Steps on Remote Clients

This section discusses in detail the configuration of a remote client for the following protocols:

- NetBIOS
- TCP/IP
- IPX/SPX
- APPC

5.1.1 NetBIOS

For remote client to access a DB2 for NT database server through NetBIOS, you need to ensure that you have installed and configured the required communications support.

5.1.1.1 Communications Protocol Requirement

The following section lists the necessary communication products to enable NetBIOS support on different platforms.

DOS / Windows 3.X: Ensure that you have the NetBIOS component installed and configured. The client should have one of the following products installed and configured on the workstation:

- IBM LAPS 1.2.1 or higher
- DOS LAN Requester / DOS LAN Services
- Banyan Vines NetBIOS 5.52 or later
- NetBEUI (shipped with Windows for Workgroups 3.11)

Refer to the corresponding product documentation for more details.

5.1.1.2 Windows 95 / Windows NT

For the remote client workstation to access the DB2 for NT database server using NetBIOS, you must first have installed and configured the required communications protocol support. For NetBIOS support, ensure that you have installed the NetBIOS/NetBEUI network software component which is a part of the base Windows 95 operating system and Windows NT.

5.1.1.3 OS/2

For the remote client workstation to access the DB2 for NT database server through NetBIOS, you must first have installed and configured the required communications protocol support. For NetBIOS support, ensure that you have configured NetBIOS support in LAN Adapter and Protocol Support (LAPS) or Multi- Protocol Transport Services (MPTS).

5.1.1.4 Information Required for NetBIOS Client Workstation

The following table is a list of the information that the remote client needs to obtain from the DB2 for NT server:

Table 10. NetBIOS Client Setup Worksheet

Parameter	Value	Description
nname	client1	This is the client workstation name and must be unique among all the NetBIOS workstations in this DB2 network. The value of this parameter is always stored in uppercase text regardless of how it was entered.
db2node	db2ntnb1	This is the local nickname that a user chooses to identify the remote database server node. This db2node name has to be unique among all db2nodes in the client's node directory. This value also is always stored in uppercase text regardless of how it was entered.
server name	db2ntsrv	This is the nname of the remote database server node. This name has to be given to the user by the administrator. This value can be determined by viewing the Database Manager configuration on the server.
adapter	0	This is the logical LAN Adapter (Lana) Number of the adapter at your client workstation that you want to use when a connection is attempted with the server.
database name	sample	Name of the database on the server you will be accessing

Figure 104 on page 154 is for reference. Figure 104 on page 154 shows a summary of the necessary steps to configure a DB2 for NT database server and using NetBIOS. The items that are checked represent only those steps necessary for the server configuration.

NetBIOS Connectivity Checklist

- [x] Install NetBIOS support on both client and server
- [x] Update environment variables on server:
  - Server must set DB2COMM = NETBIOS
- [x] Update DBM CFG file on server:
  - Specify workstation name (nname)
- [x] Change NetBIOS interface configuration on server:
  - Windows NT platform
- [ ] Update DBM CFG file on client:
  - Specify workstation name (nname)
- [ ] Change NetBIOS interface configuration on client:
  - Windows 95 and NT platform only
- [ ] Catalog Node Directory on client:
  - Choose NetBIOS protocol
  - Must know server's DBM workstation name, for instance nname
- [ ] Catalog System Directory Database on Client:
  - Must know server's system database directory alias name

Figure 104. NetBIOS Connectivity Server Checklist

5.1.1.5 Updating the Client Workstation

The client must be updated with the workstation name variable (nname). The client nname will be used to identify this NetBIOS node on the network. This value must be unique among all the NetBIOS workstations in this DB2 network.

Tip

A NetBIOS error will occur if the workstation name is not unique on the network.

5.1.1.6 Using the DB2 Client Setup

- Open the DB2 Client Setup program.
- Go to the Client menu, then select **Configure**. The Client Configuration window appears:

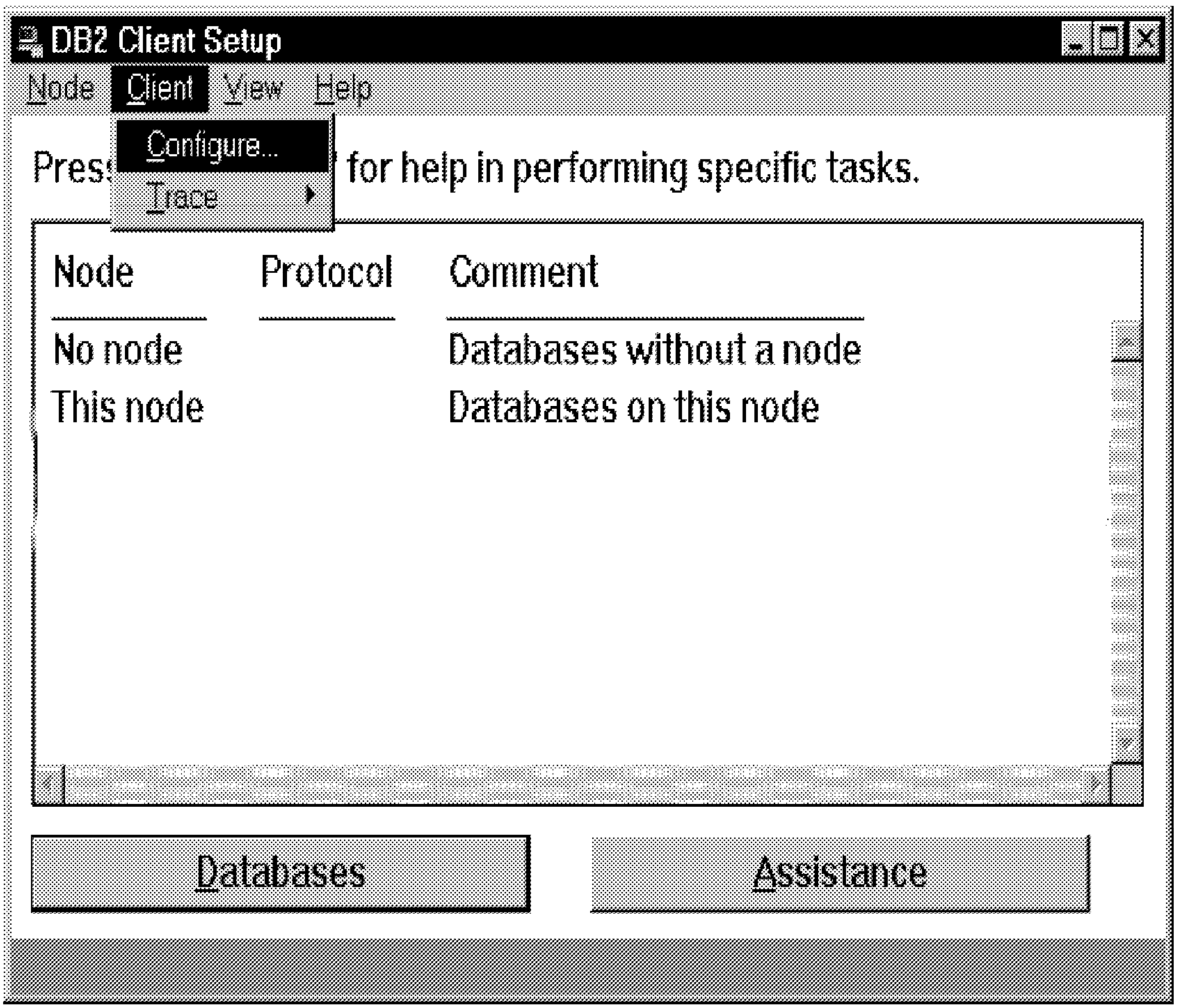


Figure 105. DB2 Client Configuration

- Select the **Communications** option under Settings.

- Enter the value for node name (nname). For example, the value can be Node Name, client1
- Select **OK**.

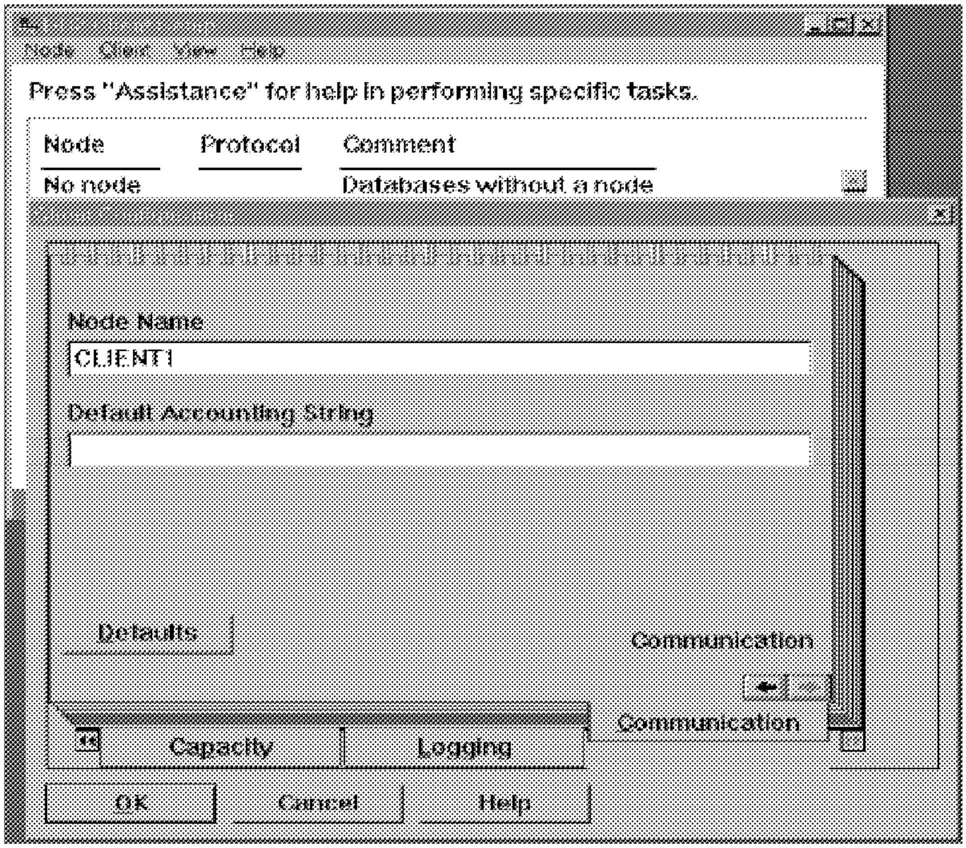


Figure 106. Updating NetBIOS Name Using DB2 Client Setup

5.1.1.7 Using the DB2 Command Line Processor

- Open the DB2 Program group.
- Double-click on the **Command Line Processor icon**. This will put you in the Command Line Processor Interactive mode.
- Enter the command to update the database manager configuration file:

For example, if the nname is client1, the command would be:

```
UPDATE DATABASE MANAGER CONFIGURATION USING NNAME client1
UPDATE DBM CFG USING NNAME client1
```

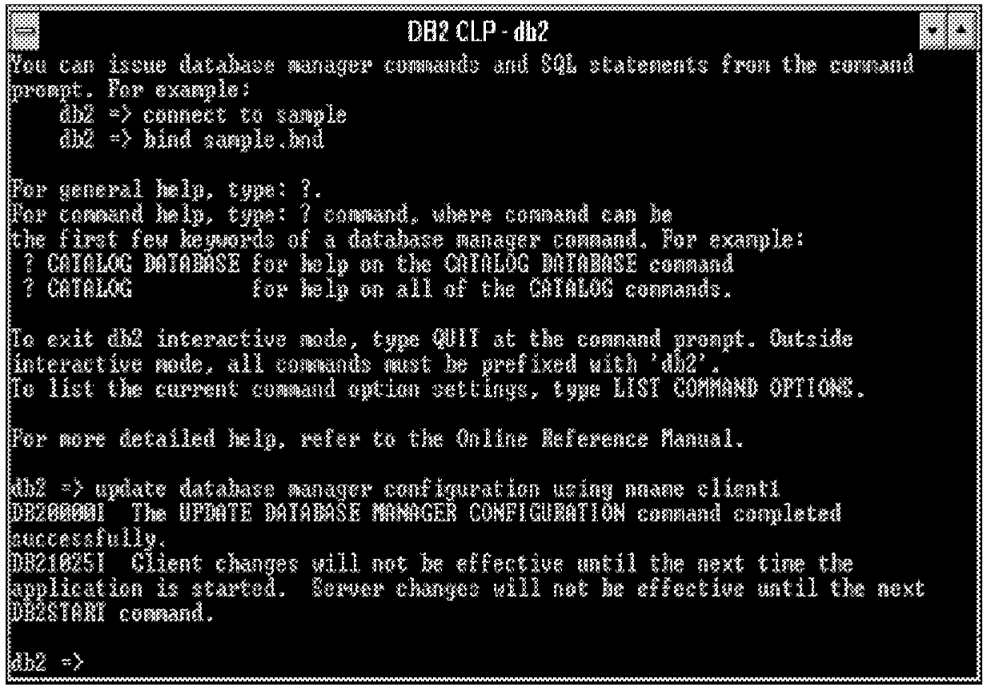


Figure 107. Updating NetBIOS Name Using DB2 CLP

To verify that the database manager on the client workstation has been updated the change, issue:

```
GET DATABASE MANAGER CONFIGURATION
GET DBM CFG
```

The following is verification that the client workstation instance has been updated:

```
DB2 CLP - db2
db2 => get database manager configuration

          Database Manager Configuration

     Node type = Client

 Database manager configuration release level            = 0x0600

 Transaction processor monitor name        (TP_MON_NAME) =

 Diagnostic error capture level              (DIAGLEVEL) = 3
 Diagnostic data directory path               (DIAGPATH) =

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

 Directory cache support                     (DIR_CACHE) = YES

 Max requester I/O block size (bytes)         (RQRIOBLK) = 32767
 DOS requester I/O block size (bytes)     (DOS_RQRIOBLK) = 4096
 DRDA services heap size (4KB)            (DRDA_HEAP_SZ) = 128

 Transaction manager database name         (TM_DATABASE) =


 Diagnostic error capture level              (DIAGLEVEL) = 3
 Diagnostic data directory path               (DIAGPATH) =

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

 Directory cache support                     (DIR_CACHE) = YES

 Max requester I/O block size (bytes)         (RQRIOBLK) = 32767
 DOS requester I/O block size (bytes)     (DOS_RQRIOBLK) = 4096
 DRDA services heap size (4KB)            (DRDA_HEAP_SZ) = 128

 Transaction manager database name         (TM_DATABASE) =

 Workstation name                                (NNAME) = CLIENT1

 Default accounting string             (DFT_ACCOUNT_STR) =


db2 =>
```

Figure 108. Verifying NetBIOS Configuration on Client Workstation

To make the changes to the configuration file effective, you must stop and start the instance on the client workstation.

5.1.1.8 Changing the NetBIOS Configuration on the Client

You need to perform this step only if you are installing a DB2 client for Windows 95 or Windows NT.

NetBIOS support for DB2 products for Windows 95 and Windows NT is implemented using NetBIOS frames and default LAN Adapter 0.

5.1.1.9 Windows 95

In order for NetBIOS communication support to function correctly, you must initialize and configure the NetBEUI protocol:

1. Select the **Network icon** in the Control Panel Program Group.
2. Click on the **Configuration** tab.
3. Highlight the NetBEUI protocol in the list of installed Network Components.
4. Click on **Properties**.
5. On the Advanced tab sheet, ensure that **Set this Protocol to be the default Protocol** is checked.
6. Select **OK** on both the NetBEUI window and Network window.
7. Reboot your system (even if the system does not prompt you to do so).

5.1.1.10 Windows NT

In order for the DB2 NetBIOS support to function correctly, you must either:

- Configure the NetBIOS interface to the defaults expected by DB2 NT.
- Catalog the NetBIOS node directory to match the Windows NT NetBIOS configuration you are using.

To change or view the NetBIOS interface configuration:

1. Select the **Network icon** in the Control Panel program group.
2. Click on the **Services** tab .Select **NetBIOS Interface** under installed Network Services.
3. Select **Properties**.
4. Find the Network Route: Nbf->... (physical Adapter specific)
5. Identify the Logical LAN Adapter Number (Lana Number) associated with the Network route Nbf ->... This value needs to be changed to 0 (zero) (if it is not already set) if you want DB2 to use the default value for LAN Adapter Numbers.

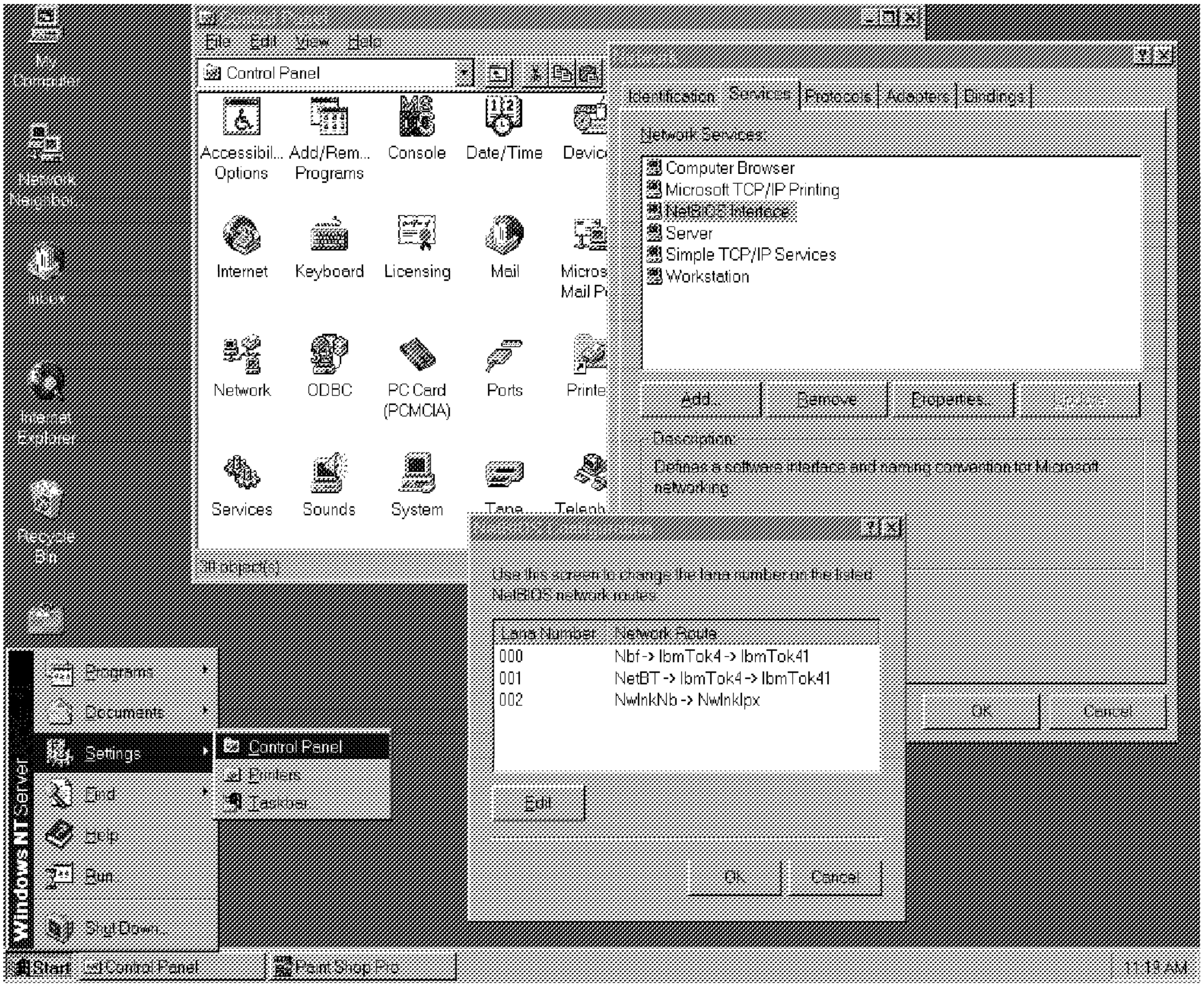


Figure 109. Windows NT NetBIOS Interface Configuration

Figure 109 illustrates Windows NT NetBIOS interface configuration where the Lana number of the NetBIOS interface `Nbf -> IbmTok4->IbmTok41` is set to `000`.

Cataloging Node to Match Windows NT NetBIOS Configuration: As an alternative to configuring the NetBIOS interface, you can catalog the NetBIOS node directory. To catalog the NetBIOS node directory to correspond with the Windows NT NetBIOS interface configuration, you must:

- Identify the LAN Adapter Number associated with the network route
- Enter the Nbf ->... entry on the client machine.
- On the client, when cataloging the NetBIOS node, explicitly specify the adapter number using the Lana Number associated with network route Nbf.

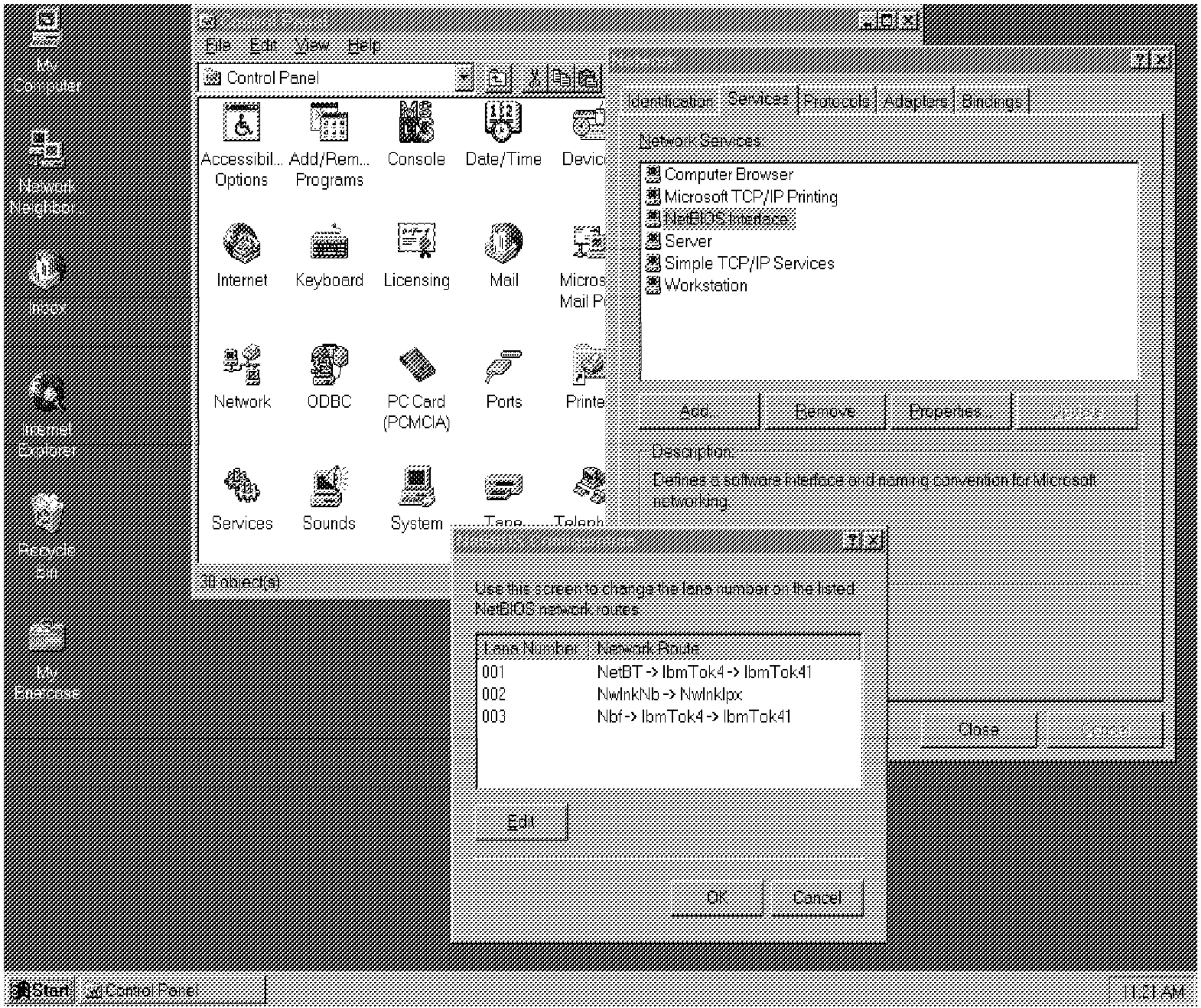


Figure 110. NetBios Interface Configuration

Figure 110 illustrates Windows NT NetBIOS interface configuration where the Lana number of the NetBIOS interface `Nbf -> IbmTok4->IbmTok41` is `003` (and not `000` which is the expected default value). Hence, in this case, while cataloging the NetBIOS node, you will have to use the Adapter Number as `003`.

5.1.1.11 Cataloging a NetBIOS Node on Client

The information obtained from the remote DB2 for NT database server must be cataloged in the node directory on the client workstation. This adds an entry in the client's node directory that points to the remote database server and the local Logical LAN Adapter Number that will be used to access the remote node.

You can catalog NetBIOS node using the DB2 Client Setup or the Command Line Processor (CLP).

Using the DB2 Client Setup

- Open the DB2 Client Setup program.
- Go to the Node menu.
- Select **New** menu item. This will open the New Node window.

- Select the **NetBIOS** radio button.
- Enter the Name (db2node), Comments, Workstation Name (nname) and Adapter.

  Name: `db2ntnb1`

  Comment: `DB2 Server on NT via NetBIOS`

  Workstation Name: `db2ntsrv`

  Adapter: `0`
- Click on **OK**.

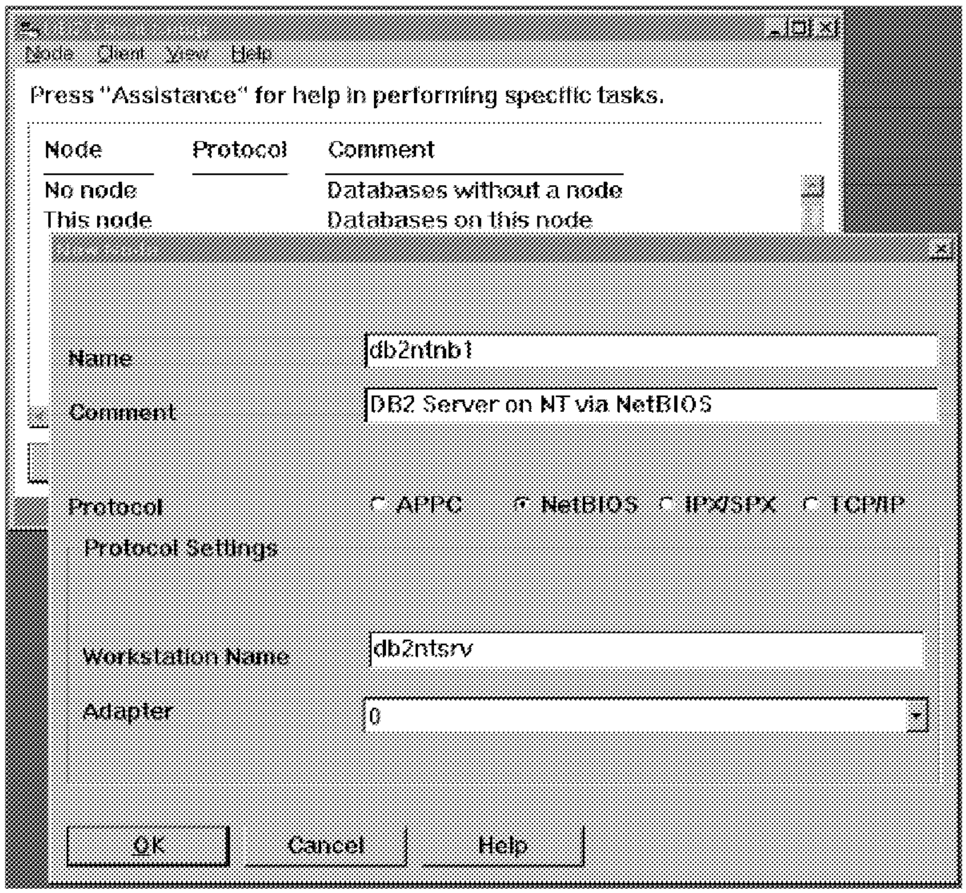


Figure 111. Catalog NetBIOS Node Using DB2 Client Setup

Using the Command Line Processor

- Open the DB2 Command Line Processor.
- Enter the following command to catalog a NetBIOS node:

```
catalog netbios node db2ntnb1 remote db2ntsrv adapter 0
```

```
CATALOG NETBIOS NODE nodename REMOTE server-nname ADAPTER adapter-number
[WITH "comment-string"]
```

Figure 112. Command to Catalog NetBIOS Node

5.1.1.12 Cataloging the Database on the Client

After the remote database server node has been configured, the remote database must also be cataloged on the client node. The database manager uses the information in the database directory along with the information in the node directory to establish a connection to the database.

You can catalog database using the DB2 Client Setup or Command Line Processor (CLP).

Using the DB2 Client Setup

- Open the DB2 Client Setup program.
- Select the node on which the database resides to highlight that node.

  (For example, DB2NTNB1)
- Select the **Database** pushbutton.
- The DB2 Client Setup - Databases window opens.
- Go to the Database menu and select **New**. The New Database window opens.
- Enter the following information:

  Name: **sample**

  Alias: **nbsamp**

  Comment: **Sample Database on NT**

  Note the following:

  - The Alias and the Comments are optional.
  - If you are connecting to a DB2 Version 1 Server, you must also select the Authentication type for the database you are cataloging. Click on the **Authentication Type (DB2 1.X Database)** check-box. Then select the value from the Authentication type drop down list. Select either **SERVER**, **CLIENT** or **DCS**. The value you select must match the authentication type of the database defined at the database directory at the server.
- Click on **OK**. (If you have not already logged on, a logon window appears prompting you for a user Id and password to log on to the database server).

Figure 113 on page 164 shows how to catalog a remote database using DB2 Client Setup.

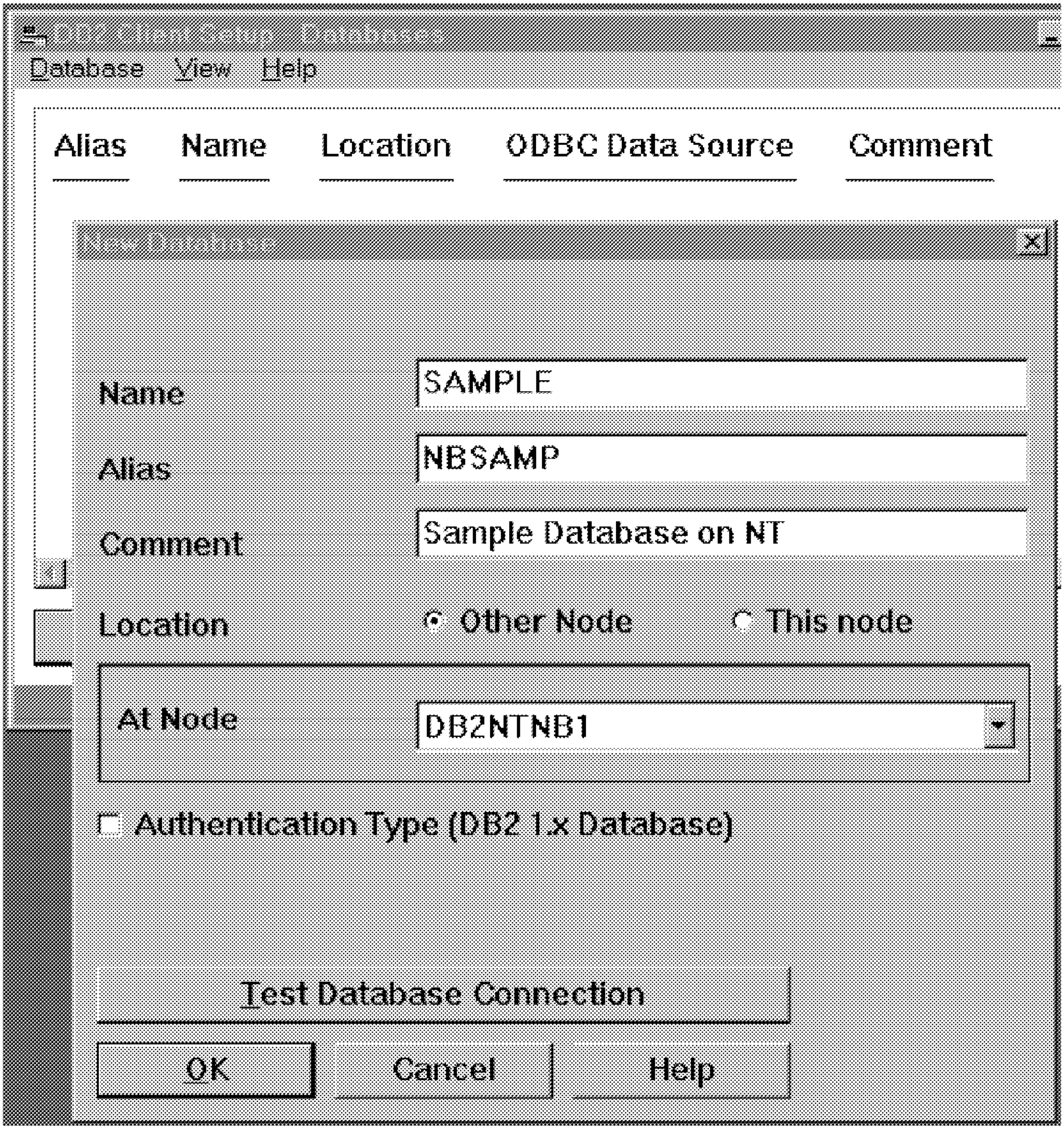


Figure 113. Catalog a Remote Database Using DB2 Client Setup

> ***** Tip *****
>
> Once you have added the node and the database, you can test the settings you specified by selecting the **Test Database Connection** push button.

Using the Command Line Processor

- Open the DB2 Command Line Processor.
- Enter the following command to catalog a database:

```
catalog database sample as nbsamp at node db2ntnb1
```

```
CATALOG DATABASE database-name [AS alias] [ON drive | AT NODE nodename]
[AUTHENTICATION {SERVER | CLIENT | DCS}] [WITH "comments"]
```

***** Tip*****

A database can be cataloged before cataloging the node on which it resides. A warning that the node has not yet been cataloged will be displayed.

Figure 114. Command to Catalog Database

Figure 115 on page 166 and Figure 116 on page 167 are for reference. Figure 115 on page 166 shows a summary of the necessary steps to configure both a DB2 for NT database server and remote client workstation using NetBIOS. Figure 116 on page 167 shows the correlation between client and server when using NetBIOS for a communication protocol with DB2.

NetBIOS Connectivity Checklist

- [x] Install NetBIOS support on both client and server

- [x] Update environment variables on server:
  - Server must set DB2COMM = NetBIOS

- [x] Update DBM CFG file on server:
  - Specify workstation name (nname)

- [x] Change NetBIOS interface configuration on server:
  - Windows NT platform

- [x] Update DBM CFG file on client:
  - Specify workstation name (nname)

- [x] Change NetBIOS interface configuration on client:
  - Windows 95 and NT platform only

- [x] Catalog node directory on client:
  - Choose NetBIOS protocol
  - Must know server's DBM workstation name, for instance nname

- [x] Catalog system directory database on client:
  - Must know server's system database directory alias name

Figure 115. NetBIOS Client and Server Connectivity Checklist

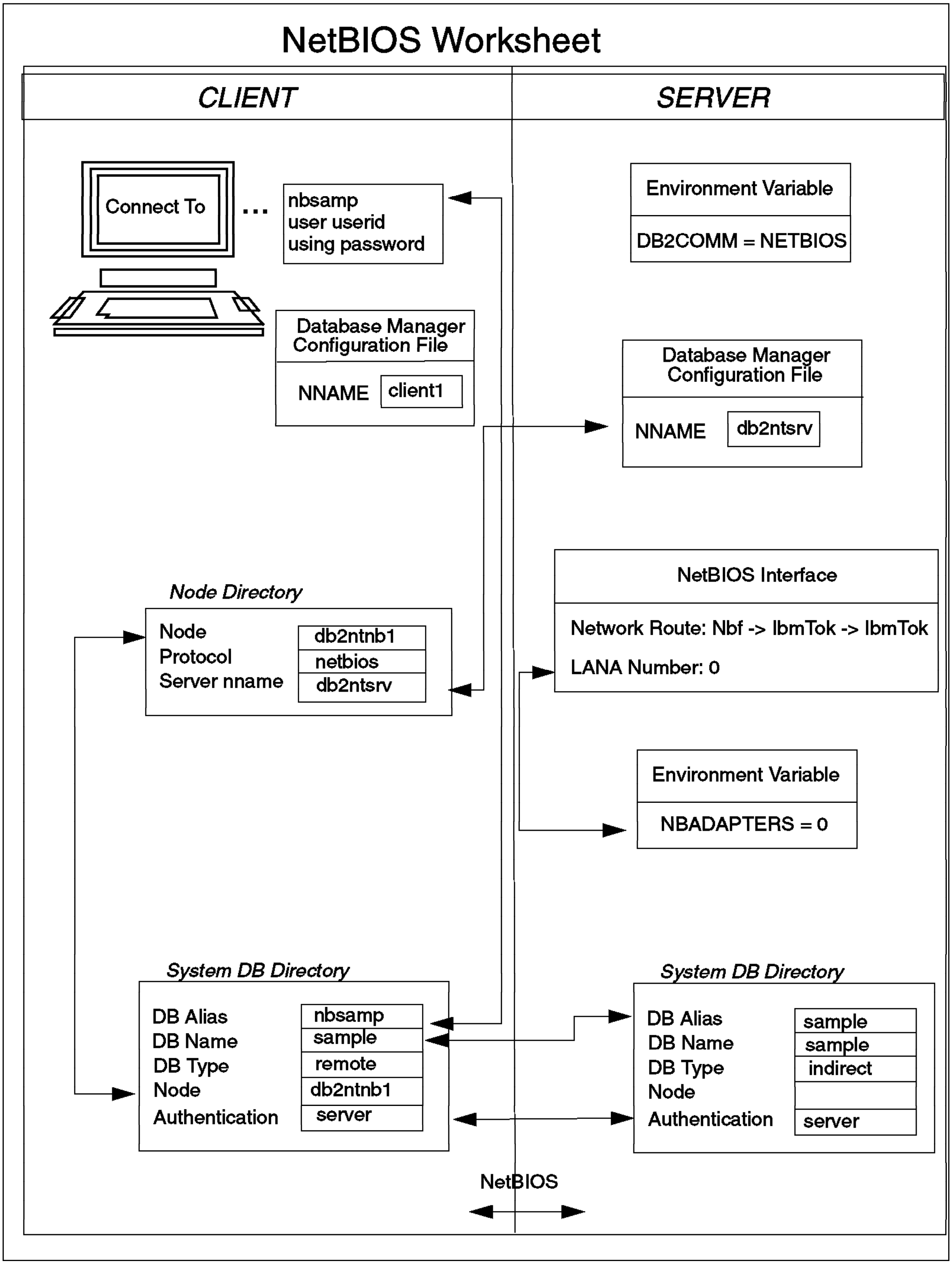


Figure 116. NetBIOS Worksheet for Client and Server Configuration

5.1.2 TCP/IP

For a remote client workstation to access DB2 for NT database server using TCP/IP, you need to ensure that you have installed and configured the required TCP/IP communication support.

5.1.2.1 Communications Protocol Requirement

The following sections detail the necessary steps to enable basic TCP/IP communication support on the client workstation.

DOS / Windows 3.X: Ensure that you have the TCP/IP component installed and configured. The client should be assigned the appropriate IP address, subnet mask and router or gateway address. Refer to the TCP/IP product documentation for more details.

Windows 95 / Windows NT: For the client workstation to access the DB2 for NT database server through TCP/IP, you must first have installed and configured the required communications protocol support. For the TCP/IP support, ensure that you have installed TCP/IP network software component which is a part of the base Windows 95 operating system and Windows NT.

OS/2: For the client workstation to access the DB2 for NT database server through TCP/IP, you must first have installed and configured the required communications protocol support. For the TCP/IP support, ensure that you have installed TCP/IP for OS/2 Version. 2.0 or later.

UNIX: For the client workstation to access the DB2 for NT database server through TCP/IP, you must first have installed and configured the required communications protocol support. For the TCP/IP support, ensure that you have installed TCP/IP component of the operating system.

***** Tip *****

To check if the TCP/IP support is configured correctly, use a TCP/IP utility like ping, Telnet or FTP to connect to the server on which DB2 is installed.

5.1.2.2 Information Required for Configuration of TCP/IP Client

The following table is a list of the information that the remote client needs to obtain from the DB2 for NT Server to enable TCP/IP communication support:

Table 11. TCP/IP Client Setup Worksheet

Parameter	Value	Description
nodename	db2nttcp	This is the local nickname that a user chooses to identify the remote database server node. This nodename has to be unique among all nodes in the client's node directory. This value also is always stored in uppercase text regardless of how it was entered
host name	ntdb2srv1	This is the TCP/IP name of the remote database server node. The hostname of the remote database server can be obtained by running the hostname command on the server. This parameter is case-sensitive.
svcename	db2instc1	This parameter is the TCP/IP service name for the server instance. The service name for the server can be obtained from the Database Manager configuration on the remote Database Server (and reading the value of the SVCENAME parameter). This parameter is case-sensitive.
database name	sample	Name of the database on the server you will be accessing

Figure 117 on page 170 is for reference. Figure 117 on page 170 shows a summary of the necessary steps to configure a DB2 for NT database server and using TCP/IP.

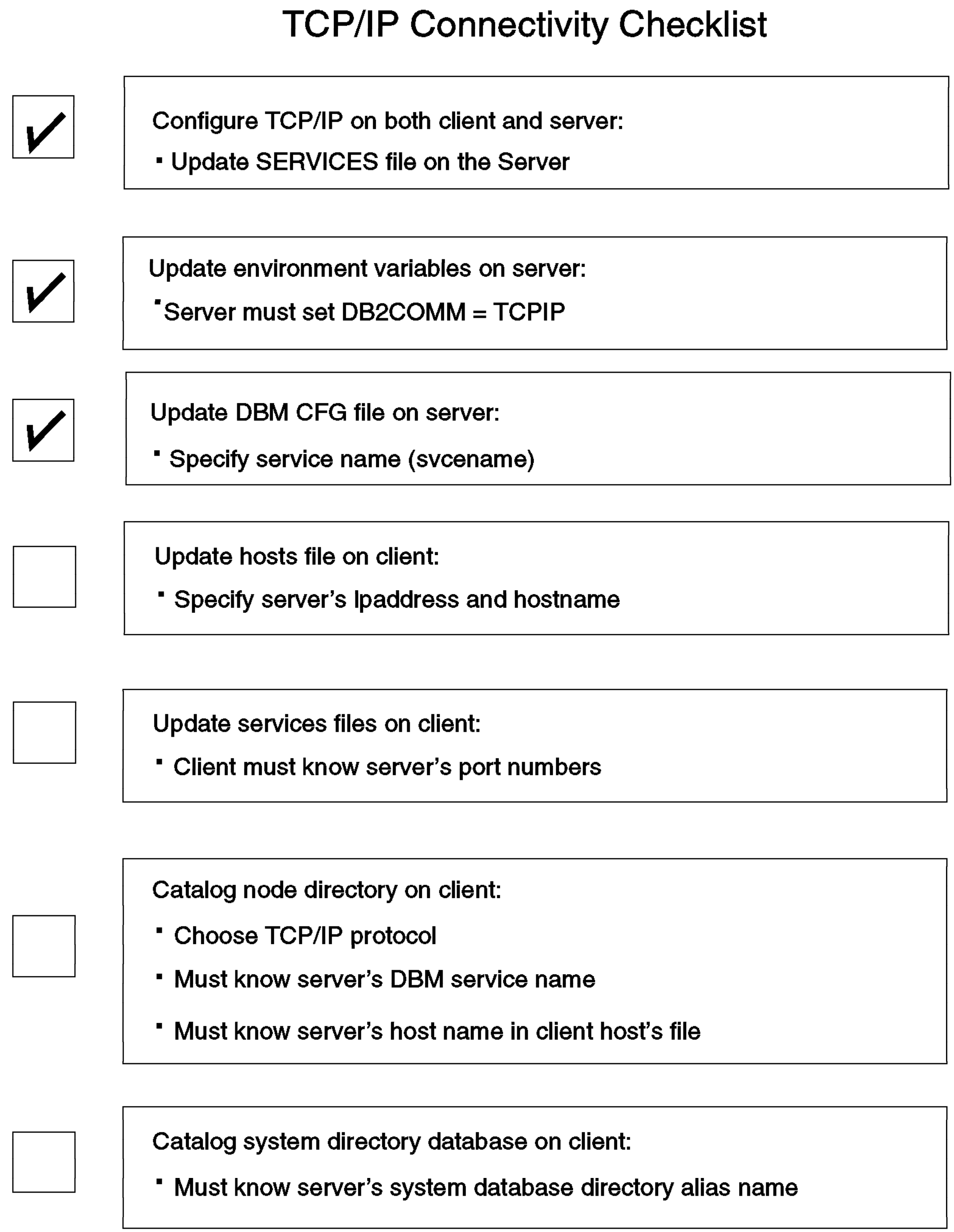


Figure 117. TCP/IP Connectivity for Client Configuration

5.1.2.3 Resolving Host Address on the Client

The client workstation must know the address of the remote database server to which it is attempting to connect. The client will refer to this address using an alias. The IP address cannot be used when cataloging a TCP/IP node. There are two ways to resolve an IP address of the host:

- By using a Name Server on the network

If you are using a Name Server, you can skip this step and proceed to the next section. Refer your TCP/IP documentation about how to configure TCP/IP to use a Name Server.

- By specifying the host address in the local hosts file

Add an entry in the hosts file on the client. The location of the hosts file may vary depending on the client operating system. For example on a Windows NT client, the hosts file is located in <winnt-dir>system32driversetchosts.

An entry in the hosts file might be similar to the following:

```
9.3.1.86 ntdb2srv1 # DB2 Server on Windows NT
```

where `9.3.1.86` is the IP address and `ntdb2srv1` is the hostname.

***** Tip *****

> If the remote database server does not reside in the same domain as the client, the hostname must be a fully specified domain name. For example, `db2ntsrv1@itsorus.austin.ibm.com`

5.1.2.4 Updating the Services File on the Client

You'll also need to update the services file on the client. The services file is updated using an editor and is not part of any of the DB2 configuration files.

Add an entry to the services file giving the Service Name and the connection port. The location of the services file may vary depending on the client operating system.

For example, on a Windows NT client, the entry in the c:winnt351system32driversetcservices should be similar to:

`db2instc1 5000/tcp # DB2 connection port`. The Service Name entry, `db2instc1`, is unique for the connection port. The port number is `5000`.

Note the following:

- The Service Name is always case-sensitive.
- The port numbers must be an integer number greater than 1000.
- The port numbers must be unique within the services file on the server. All clients must match services entries to connect to a particular instance on a particular database server.
- To ensure that the entries in the services file are read correctly, the port number must not be the last character in the file. It is recommended that you press **Enter** at the end of the line or end the last line with a comment (#).

5.1.2.5 Cataloging a Remote TCP/IP Server on the Client

The remote information about the remote DB2 for NT database server must be cataloged in the node directory. This adds an entry in the client's node directory that points to the remote database server.

When cataloging the client node directory, the client's main connection port name in its services file must be specified. This will be used to reference the main connection port number at the server. When cataloging the client node directory, the hostname of the remote database server will also be used.

You can catalog the TCP/IP node using the DB2 Client Setup or Command Line Processor (CLP).

Using the DB2 Client Setup

- Open the DB2 Client Setup program.
- Click on the Node menu.
- Select **New** menu item. This will open the New Node window.
- Select the **TCP/IP** radio button.
- Enter the Name (db2node), Comments, File Server and Object Name information of the remote server.

  Name: `db2nttcp`

  Comment: `DB2 Server on NT via TCP/IP`

  Hostname: `ntdb2srv1`

  Service Name: `dn2instc1`

- Click on **OK**.

Figure 118 on page 173 shows how to catalog a TCP/IP node using DB2 Client Setup.

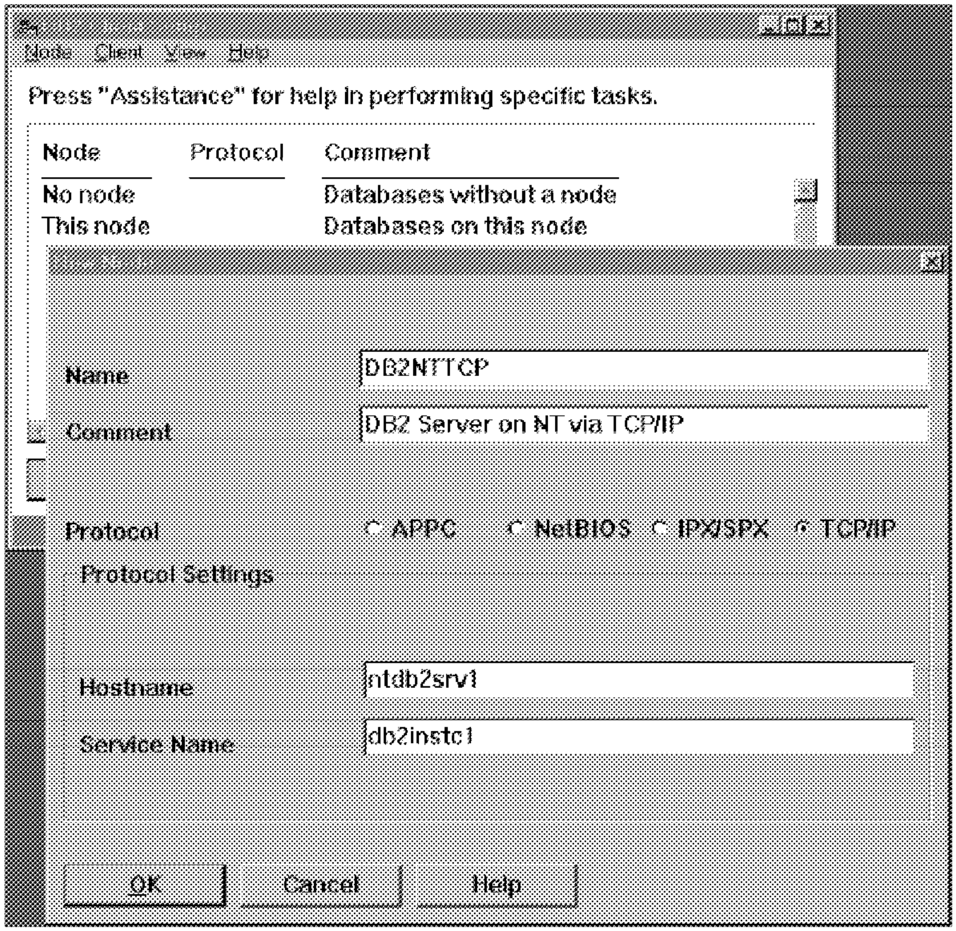


Figure 118. Catalog a TCP/IP Node Using DB2 Client Setup

Using the Command Line Processor

- Open the DB2 Command Line Processor.
- Enter the following command to catalog a TCP/IP node:

```
catalog tcpip node db2nttcp remote * server 00007007.0004AC947CF4.879E
```

5.1.2.6 Cataloging a Remote Database on the Client

After the remote database server has been configured, the remote database must be cataloged on the client node. The client uses the information in the database directory along with the information in the node directory to establish a connection to the database.

When cataloging the database directory on the client, the alias that the server machine uses in its system database directory when referencing the DB2 database must be known.

You can catalog the database using the DB2 Client Setup or the Command Line Processor (CLP).

Using the DB2 Client Setup

- Open the DB2 Client Setup program.
- Select the node on which the database resides to highlight that node.

(For example, DB2NTTCP)

- Select the **Database** pushbutton.
- The DB2 Client Setup - Databases window opens.
- Go to the Database menu and select **New**. The New Database window opens.
- Enter the following information:

  Name: **sample**

  Alias: **tcpsamp**

  Comment: **Sample Database on NT**

  Note the following:

  – The Alias and the Comments are optional.

  – If you are connecting to a DB2 Version 1 Server, you must also select the Authentication Type for the database you are cataloging. Click on the **Authentication Type (DB2 1.X database)** check box. Then select the value from the Authentication Type drop-down list. Select either **SERVER**, **CLIENT** or **DCS**. The value you select must match the authentication type of the database defined at the database directory at the server.

- Click on **OK**. (If you have not already logged on, a logon window appears prompting you for a user Id and password to log on to the database server).

Figure 119 on page 175 shows how to catalog a remote database using DB2 Client Setup.

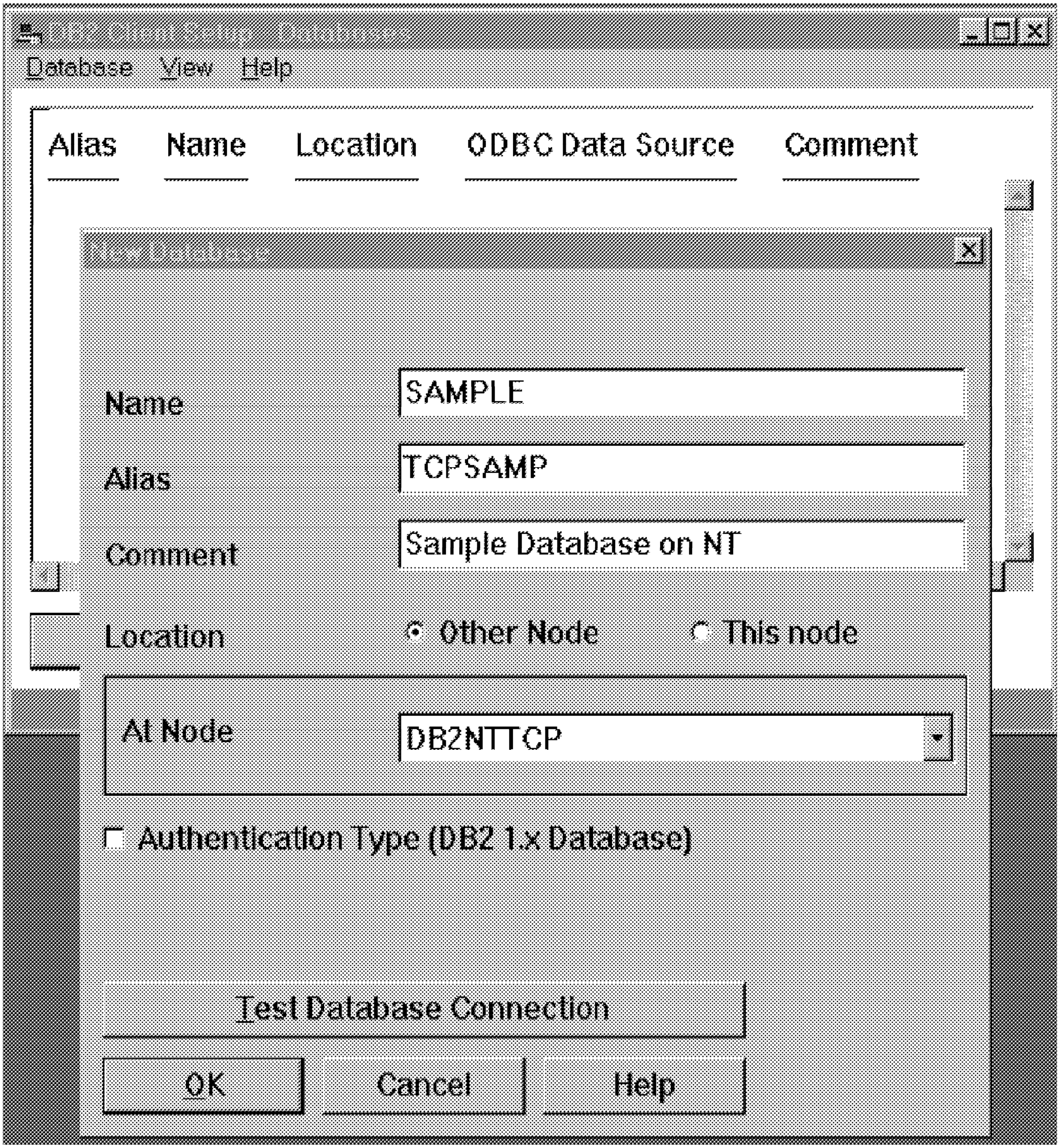


Figure 119. Catalog Remote Database Using DB2 Client Setup

***** Tip *****

Once you have added the node and the database, you can test the settings you specified by selecting the **Test Database Connection** pushbutton.

Using the Command Line Processor

- Open the DB2 Command Line Processor.
- Enter the following command to catalog a database:

```
catalog database sample as tcpsamp at node db2nttcp
```

```
CATALOG DATABASE database-name [AS alias] [ON drive | AT NODE nodename]
[AUTHENTICATION {SERVER | CLIENT | DCS}] [WITH "comments"]
```

Figure 120. Command to Catalog Remote Database

***** Tip*****

A database can be cataloged before cataloging the node on which it resides. A warning that the node has not yet been cataloged will be displayed.

Figure 121 on page 177 and Figure 122 on page 178 are for reference. Figure 121 on page 177 shows a summary of the necessary steps to configure both a DB2 for NT database server and remote client workstation using NetBIOS. Figure 122 on page 178 shows the correlation between client and server when using NetBIOS for a communication protocol with DB2.

TCP/IP Connectivity Checklist

Configure TCP/IP on both client and server:

- Update the SERVICES file on the Server

Update the environment variables on server:

- Server must set DB2COMM = TCPIP

Update the DBM CFG file on server:

- Specify service name (svcename)

Update the services files on client and server:

- Client must know server's port numbers

Update the hosts file on client:

- Specify server's Ipaddress and hostname

Catalog the node directory on client:

- Choose TCP/IP protocol
- Must know server's DBM service name
- Must know server's host name in client host's file

Catalog the system directory database on client:

- Must know server's system database directory alias name

Figure 121. TCP/IP Connectivity Checklist for Client and Server Configuration

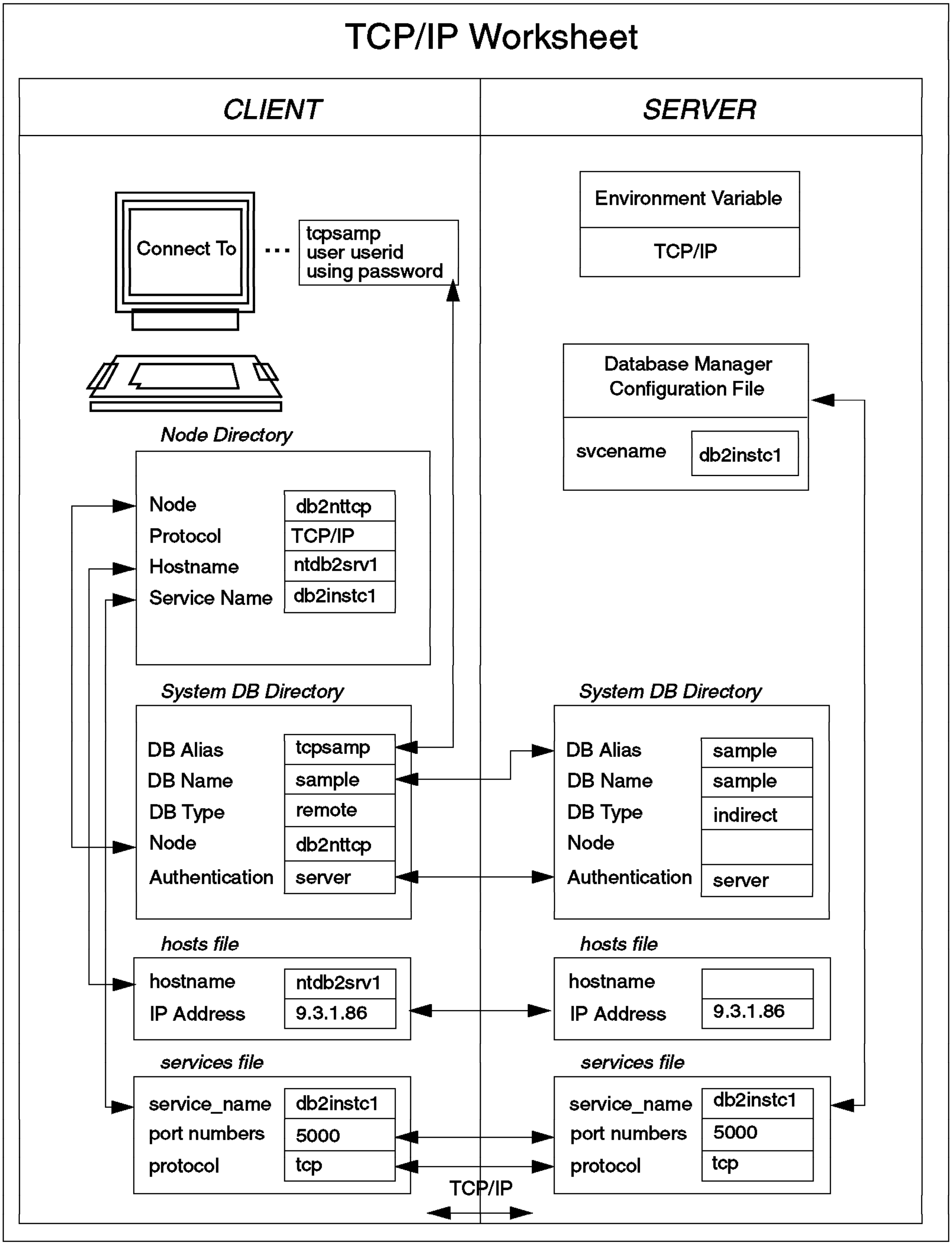


Figure 122. TCP/IP Connectivity Worksheet for Client and Server Configuration

5.1.3 IPX/SPX

For the remote client workstation to access a DB2 for NT database server through IPX/SPX, you need to ensure that you have installed and configured the required communications support.

5.1.3.1 Communications Protocol Requirement

The following sections detail the necessary steps to enable basic IPX/SPX communication support on the client workstation.

DOS / Windows 3.X: Ensure that you have the IPX/SPX component installed and configured. The client should have the Novell NetWare client installed and configured. Refer to the corresponding product documentation for more details.

Windows 95 / Windows NT: For the node to access the DB2 Server through IPX/SPX, you must first have installed and configured the required communications protocol support. For the IPX/SPX support, ensure that you have installed the NWLink IPX/SPX network software component that is a part of the base Windows 95 operating system and Windows NT.

OS/2: For the node to access the DB2 Server through IPX/SPX, you must first have installed and configured the required communications protocol support. For the IPX/SPX support, ensure that you have installed Novell NetWare Client for OS/2 Version. 2.1 or later.

5.1.3.2 Information Required for Setting Up IPX/SPX Client

The following table is a list of information that the remote client needs to obtain from the DB2 for NT Server to enable IPX/SPX communication support.

Table 12. IPX/SPX Client Setup Worksheet

Parameter	Value	Description
nodename	db2ntipx	This is a alias (nickname) defined by the user to identify the server on the network. This value must be unique among all DB2 nodes in the client's node directory. DB2 stores this value in uppercase regardless of how you enter it. The name you choose should comply to the naming conventions for the database manager.
objectname	00007007. 0004AC94 7CF4.879E	This is the NetWare Internetwork address of the DB2 Server. This address will be of the form: <8 byte net id>.<12 byte node id>.<4 byte socket number>.
database name	sample	Name of the database on the server which you will be accessing

Figure 123 on page 180 is for reference. Figure 123 on page 180 shows a summary of the necessary steps to configure a DB2 for NT database server and using IPX/SPX.

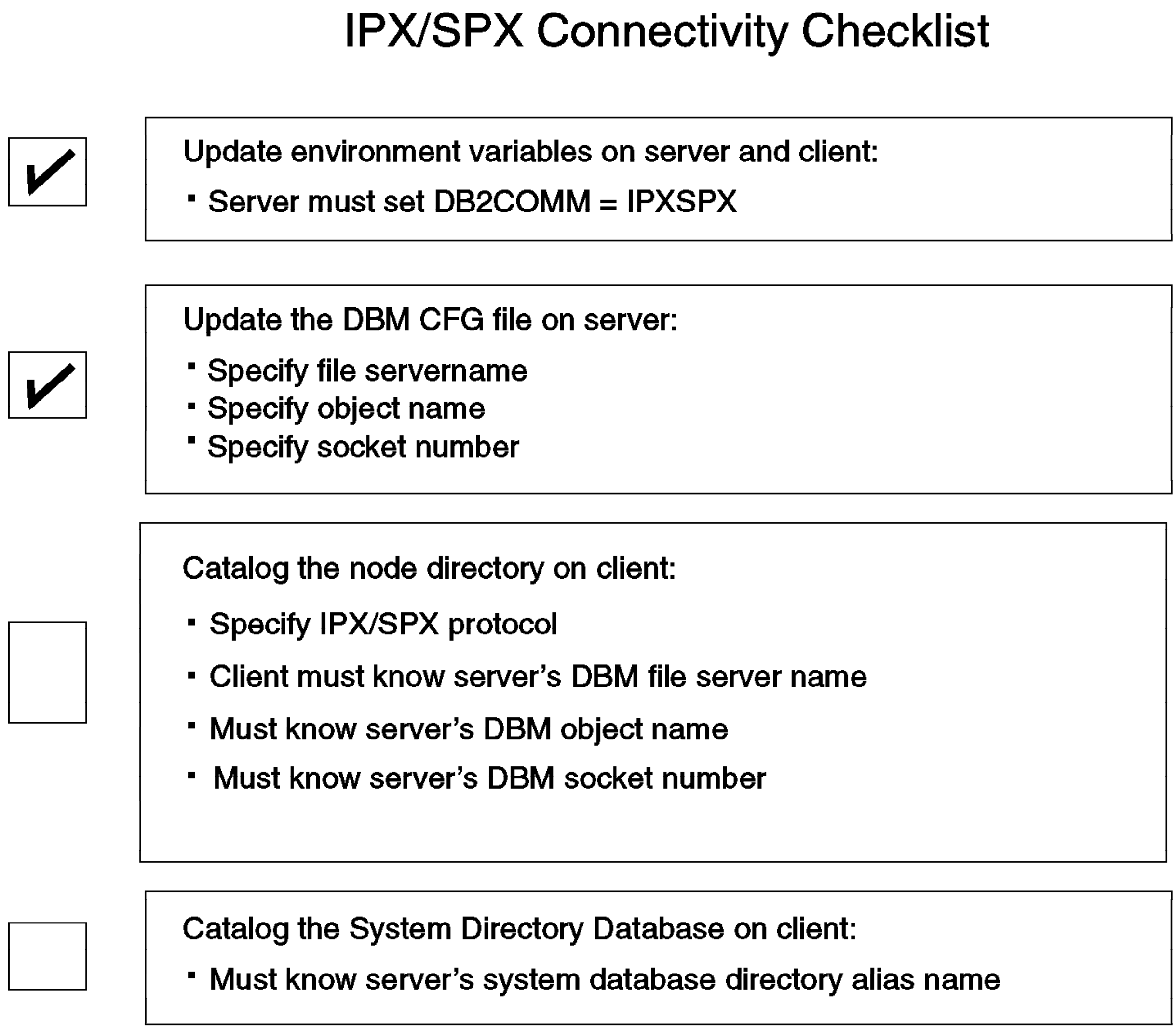


Figure 123. IPX/SPX Connectivity Checklist for Client Configuration

5.1.3.3 Cataloging a IPX/SPX Node on the Client

The information from the remote DB2 for NT Server must be cataloged in the node directory. This adds an entry in the client's node directory that points to the remote database server.

DB2 NT supports IPX/SPX clients through Direct Addressing, and hence you can use * (asterisk) for the value of the file server and the NetWare Internetwork address of the remote database server as the object name.

You can catalog an IPX/SPX node using the DB2 Client Setup or Command Line Processor (CLP).

Using the DB2 Client Setup

- Open the DB2 Client Setup program.
- Go to the Node menu.
- Select **New** menu item. This will open the New Node window.
- Select the **IPX/SPX** radio button.

- Enter the Name, Comments, File Server and Object Name information of the remote server.

  Name: db2ntipx

  Comment: DB2 Server on NT via IPX

  File Server: *

  Object Name: 00007007.0004AC947CF4.879E

- Click on **OK**.

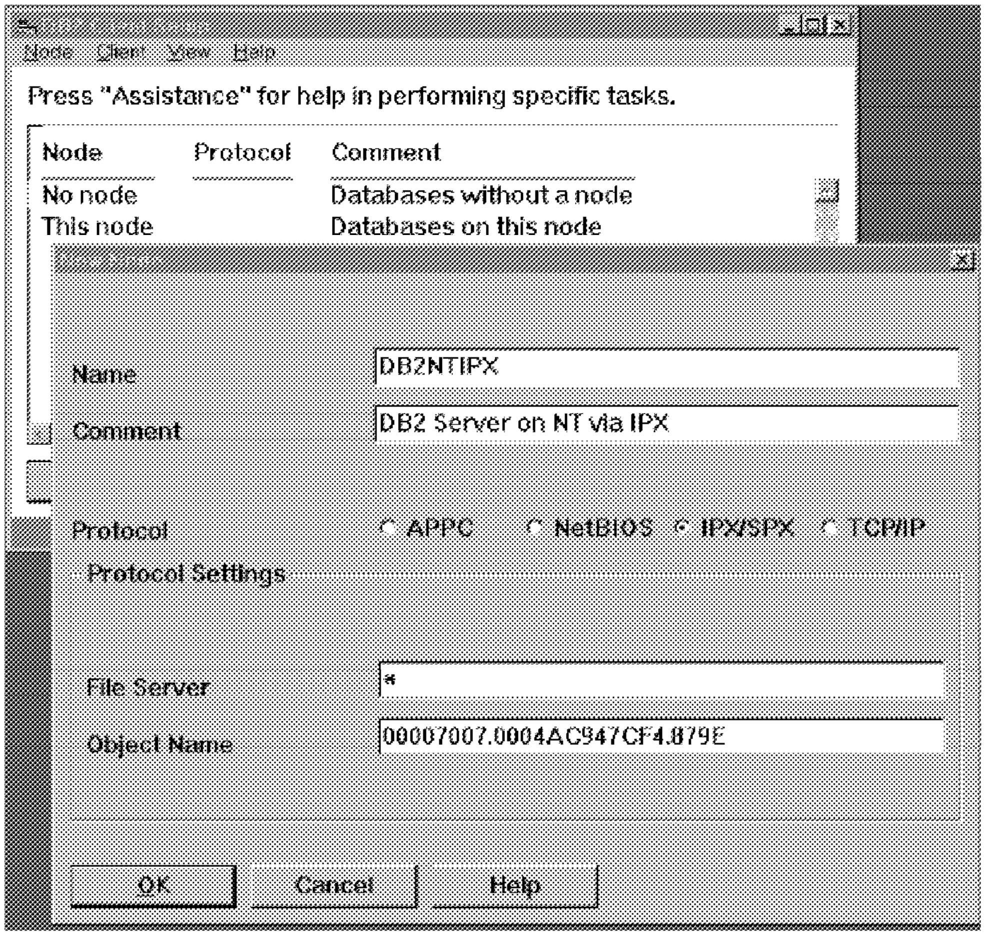


Figure 124. Catalog IPX Node Using DB2 Client Setup

Using the Command Line Processor

- Open the DB2 Command Line Processor.
- Enter the following command to catalog a IPX/SPX node:

  `catalog ipxspx node db2ntipx remote * server 00007007.0004AC947CF4.879E`

```
CATALOG IPXSPX NODE db2node REMOTE fileserver SERVER objectname [WITH
"comment-string"]
```

Figure 125. Command to Catalog IPX/SPX Node

5.1.3.4 Cataloging Database on the Client

After the remote database server has been configured, the remote database must be cataloged on the client node. The client uses the information in the database directory along with the information in the node directory to establish a connection to the database.

You can catalog a remote database either by using the DB2 Client Setup or Command Line Processor (CLP).

Using the DB2 Client Setup

- Open the DB2 Client Setup program.
- Select the node on which the database resides to highlight that node.

  (For example, DB2NTIPX)
- Select the **Database** push-button.
- The DB2 Client Setup - Databases window opens.
- Go to the Database menu and select **New**. The New Database window opens.
- Enter the following information:

  Name: `sample`

  Alias: `ipxsamp`

  Comment: `Sample Database on NT`

  Note the following:

  – The Alias and the Comments are optional.

  – If you are connecting to a DB2 Version 1 Server, you must also select the Authentication Type for the database you are cataloging. Click on the **Authentication Type (DB2 1.X database)** checkbox. Then select the value from the Authentication Type drop-down list. Select either **SERVER**, **CLIENT** or **DCS**. The value you select must match the authentication type of the database defined at the database directory at the server.
- Click on **OK**. (If you have not already logged on, a logon window appears prompting you for a user Id and password to log on to the database server).

Figure 126 on page 183 shows how to catalog a database using DB2 Client Setup.

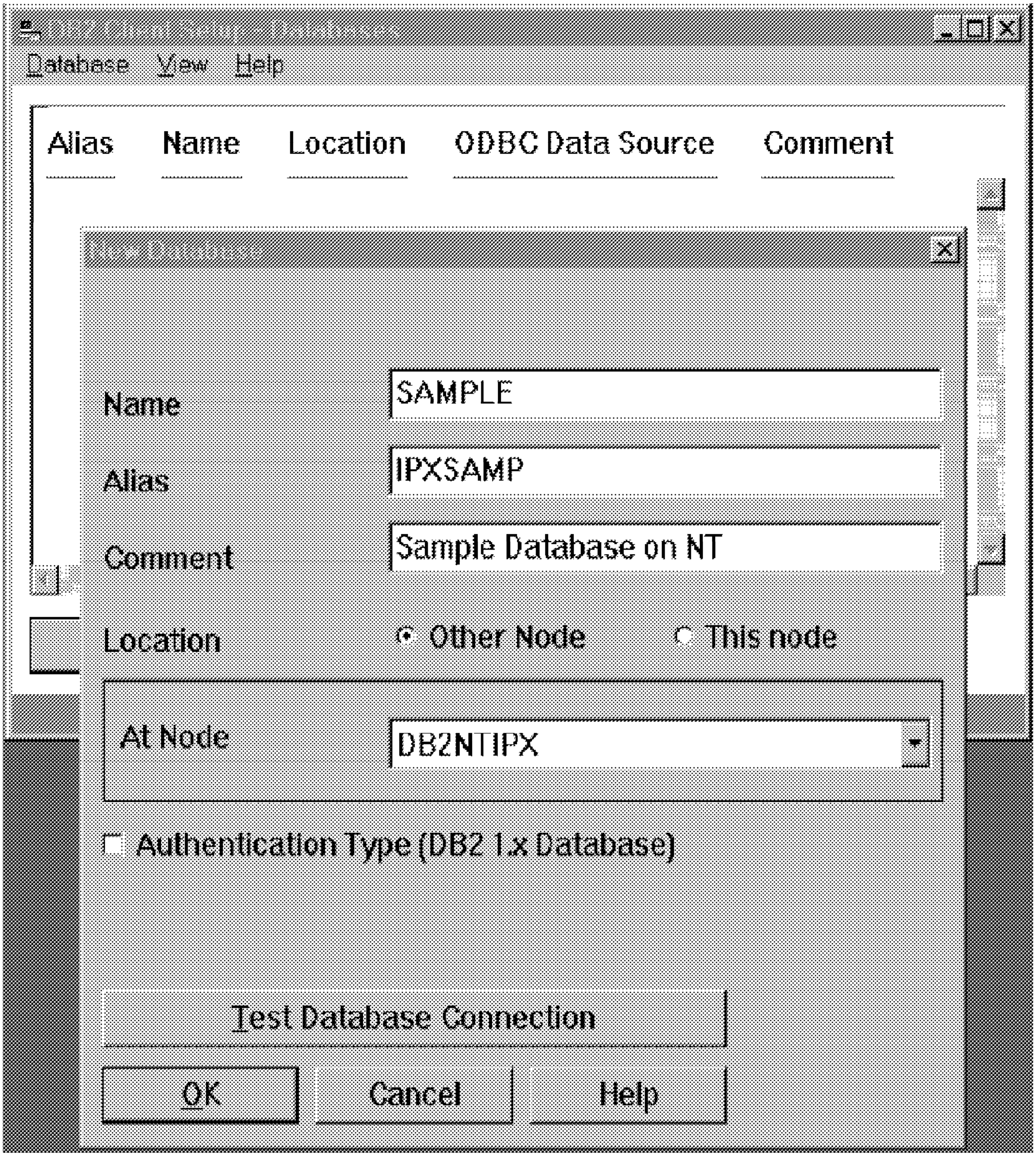


Figure 126. Catalog Database Using DB2 Client Setup

***** Tip *****

Once you have added the node and the database, you can test the settings you specified by selecting the **Test Database Connection** push button.

Using the Command Line Processor

- Open the DB2 Command Line Processor.
- Enter the following command to catalog a database:

```
catalog database sample as ipxsamp at node db2ntipx
```

```
CATALOG DATABASE database-name [AS alias] [ON drive | AT NODE nodename]
[AUTHENTICATION {SERVER | CLIENT | DCS}] [WITH "comments"]
```

Figure 127. Command to Catalog Database

***** Tip*****

A database can be cataloged before cataloging the node on which it resides. A warning that the node has not yet been cataloged will be displayed.

***** Note *****

If you are connecting from a Windows 95 client to a DB2 Server using IPX/SPX protocol, and the machine hangs, it is likely that this is because of improper setting of the KeepAliveTimeout NT registry parameter, which is set to a default of 0XC (6 seconds). It is recommended that you increase this value to 0X20 which should solve your problem.

The location of Registry Key is as follows:

```
HKEY_LOCAL_MACHINE
    SYSTEM
        CurrentControlSet
            Services
                NwlnkSpx
                    Parameters
                        KeepAliveTimeout
```

Figure 128 on page 185 and Figure 129 on page 186 are for reference. Figure 128 on page 185 shows a summary of the necessary steps to configure both a DB2 for NT database server and remote client workstation using IPX/SPX. Figure 129 on page 186 shows the correlation between client and server when using IPX/SPX for a communication protocol with DB2.

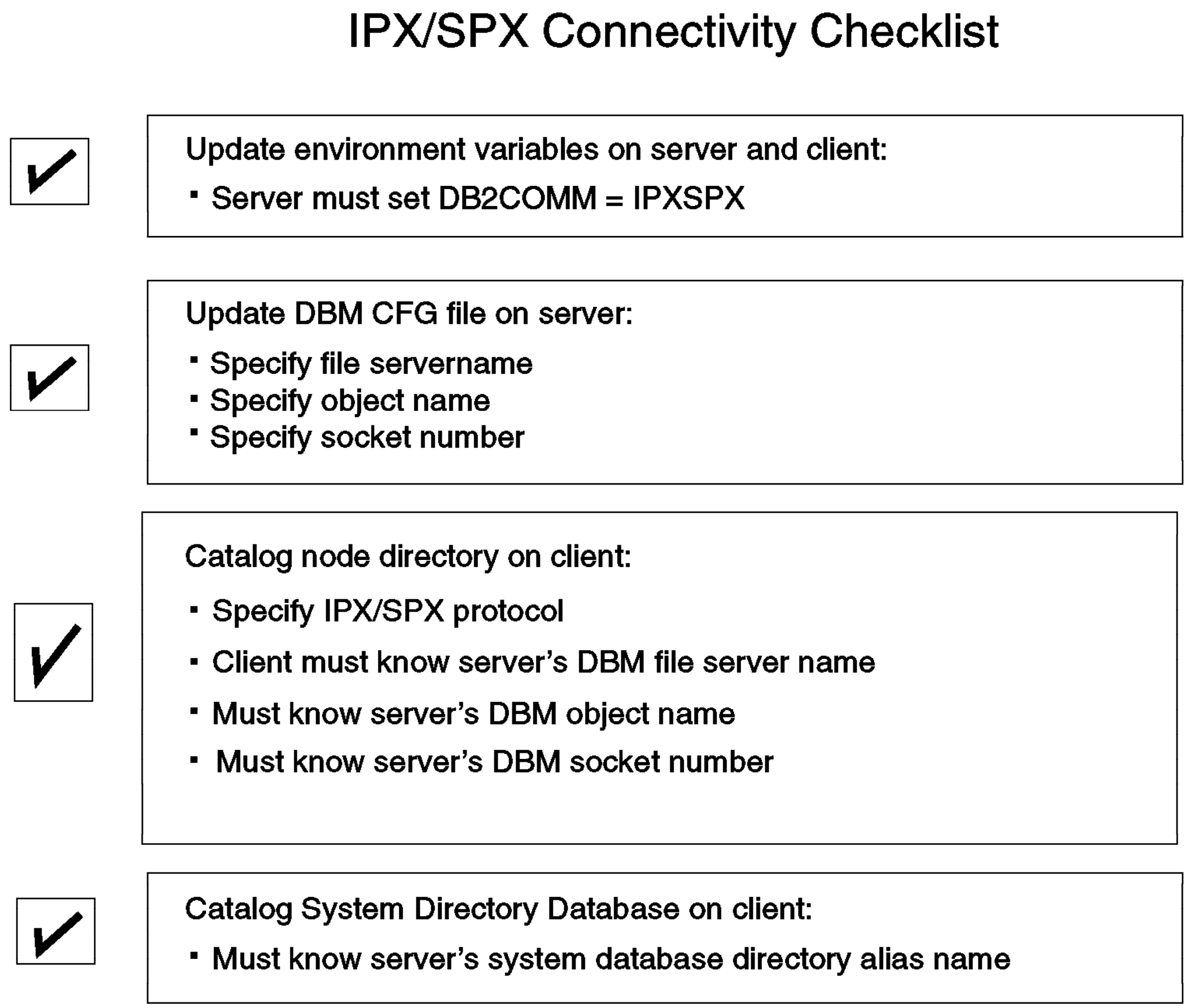


Figure 128. IPX/SPX Connectivity Checklist for Client and Server Configuration

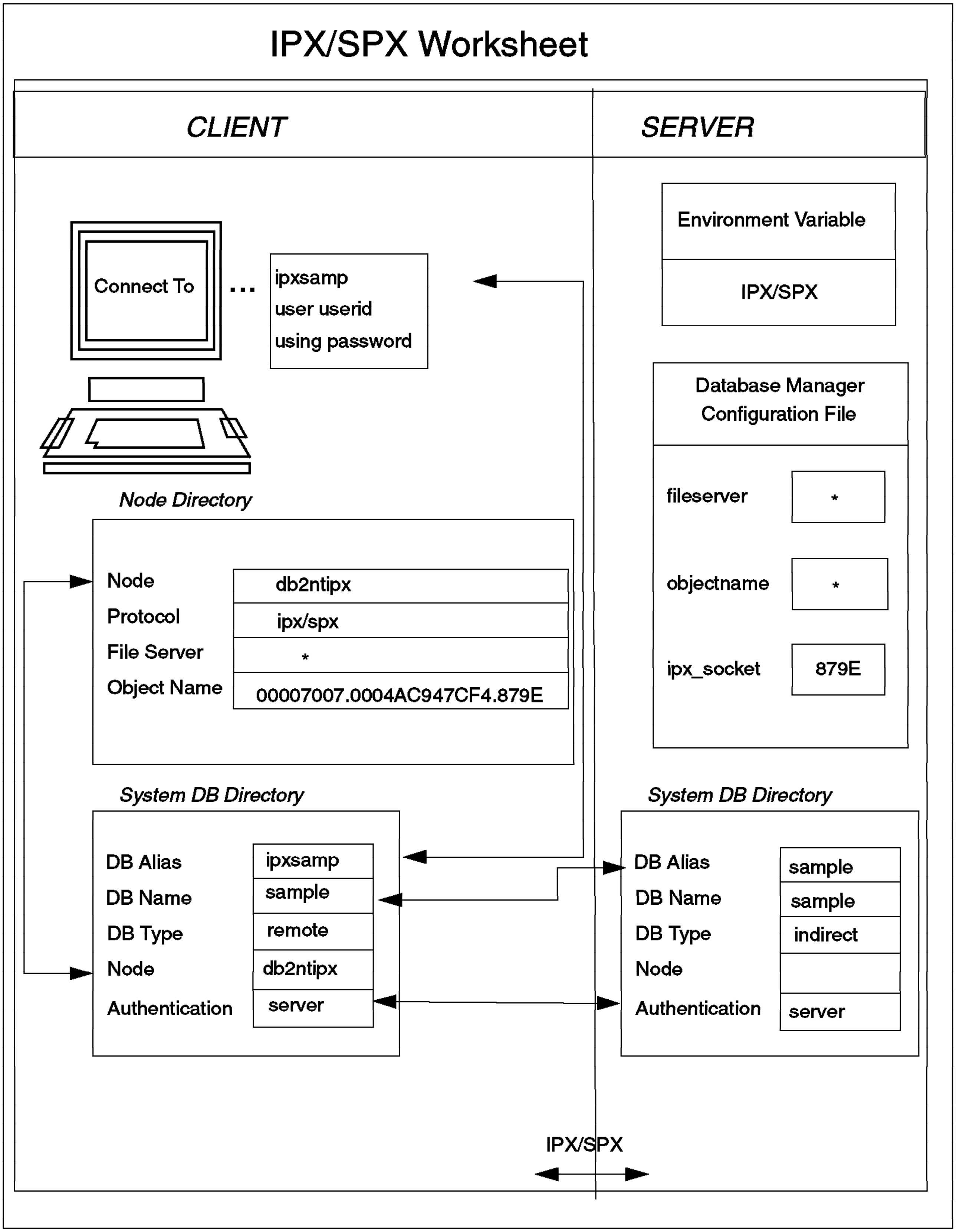


Figure 129. IPX/SPX Connectivity Worksheet for Client and Server Configuration

5.2 Verifying the Connection

To connect to either a local or remote database, you use the CONNECT statement. After a connect is issued, all SQL requests are executed against the database to which you are connected.

When a client connects to a DB2 NT Server, there are two forms of CONNECT statements that can be used. One form specifies the user name and password to be used and is referred to as an explicit CONNECT statement. The syntax of the statement to be used is as follows:

```
CONNECT TO <database-name> USER <userid> using <password>
```

For example:

```
CONNECT TO TCPSAMP USER samit using win2warp
```

where TCPSAMP is the database alias in the database directory, samit is the user name and win2warp is the password.

The other form of CONNECT statement is the implicit CONNECT statement. The syntax of this statement is as follows:

```
CONNECT TO <database-name>
```

For example:

```
CONNECT TO TCPSAMP
```

where TCPSAMP is the database alias. Depending on the authentication and the client operating system, the userid and password will be automatically picked up, you will be prompted to log on (if you have not logged on), or you will be denied the connection.

***** Important *****

When a client connects to a DB2 for NT Server (and the Authentication is Server) the userid and password is sent to the DB2 NT Server for verification. The password is passed as it is typed, and the DB2 NT Server is case sensitive for verifying the password. There must be an exact match.

For more information on DB2 and Windows NT security, refer to Chapter 2, "DB2 and Windows NT Security" on page 27.

5.3 Logon Considerations from the Remote Client to DB2 NT Server

When a remote client wants access to a database server where the operating system of the database server is different than that on the client system, the client utilities must be bound to each database on the server that the client wishes to access. For example, CAE or SDK for DOS, OS/2 or AIX would need to have their utilities bound at a DB2 for NT database server. The utilities that need to be bound include IMPORT, EXPORT, the Command Line Processor, REORG and so on. To simplify this process, the bind files are grouped together in different .lst files. The client utilities only need to be bound once per database from each type of client.

To bind the client utilities from a client, enter the following set of commands from a command line:

```
db2 connect to <database-name> user <userid> using <password>
db2 bind <path>@db2ubind.lst blocking all grant public
```

where `<path>` is the full path name of the directory where the bind files are located, such as c:sqllibbnd on OS/2. The @ (at sign) is used to imply to the binder that the file `db2ubind.lst` is a list of bind files and not the bind file itself, and `grant public` grants execute and bind privileges to public.

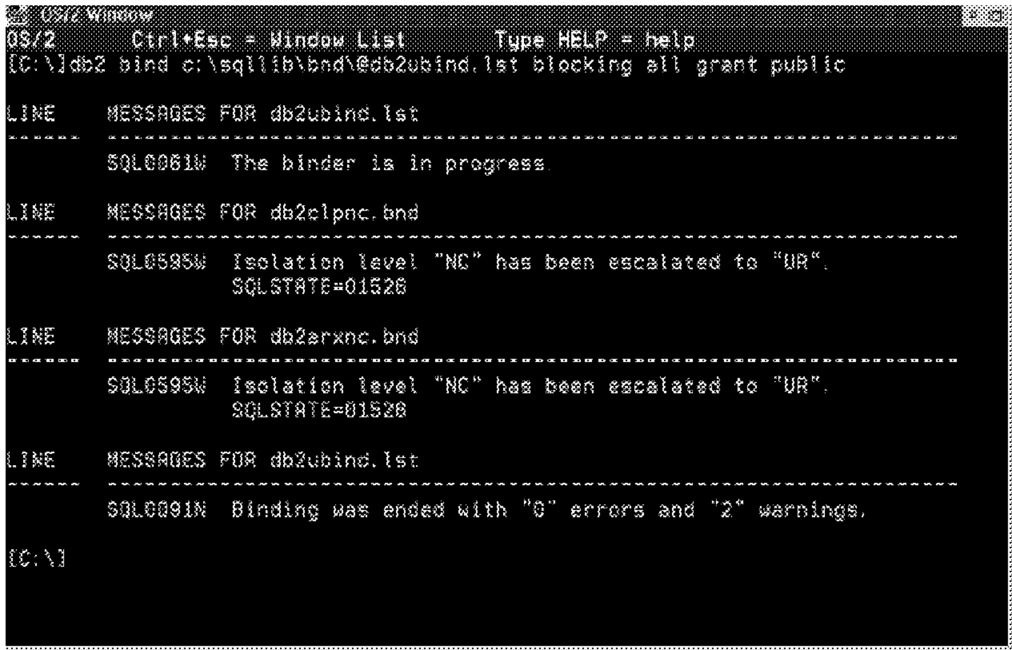


Figure 130. Binding Utilities to Database

The db2ubind.lst file contains the list of bind files required to create the package for the database utilities.

Similarly, to use the DB2 CLI (Call Level Interface) client utilities at a DB2 Server on a different platform, packages must be bound at each database on the DB2 Server that the client wishes to access. The client will use either the DB2 CLI or the ODBC driver.

To bind the CLI utilities, enter the following set of commands from a command prompt:

```
db2 connect to <database-name> user <userid> using <password>
db2 bind <path>@db2cli.lst blocking all grant public
```

where `<path>` is the full path name of the directory where the bind files are located, such as c:sqllibbnd on OS/2. The @ (at sign) is used to imply to the binder that the file db2ubind.lst is a list of bind files and not the bind file itself, and `grant public` grants execute and bind privileges to public.

***** Tip *****

You can also use the DB2 Client Setup program to bind the packages to each database. For more information on this refer to *Installing and Using DB2 Clients* for each platform.

5.4 Using the Command Line Processor

The Command Line Processor is installed with all the DB2 products. Using the CLP, you can manage a database environment by entering DB2 commands and SQL statements in an interactive mode or from command files. Both of these methods are discussed in this section.

The Command Line Processor supports two modes of operation. You can either use a DB2 Command window or an interactive DB2 Command Line Processor. A command window can be invoked by clicking on the CLP command window option or by entering the command `db2cmd`. When using a command window, you need to prefix all CLP commands with `db2`. To operate in the interactive mode, invoke the DB2 Command Line Processor from the DB2 for Windows NT group. In the interactive mode, you can enter CLP commands without prefixing them with `db2`. Unlike DB2 for OS/2, CLP in DB2 for Windows NT also allows you to recall your previously entered commands.

When using the CLP, you need to ensure that Windows NT does not mistake the database commands for operating system commands. It is recommended that you enclose any database command within quotation marks (″ ″), which would ensure that they are not interpreted as operating system commands. From the CLP, you can issue operating system commands by prefacing them with '!'. If your commands on the CLP exceeds the limit allowed by Windows NT window, use a backslash '/' as the line continuation character.

It is also possible to obtain the syntax and information for all of the DB2 commands from the Command Line Processor by using one of the following options:

- `db2 ?` - list of all DB2 commands
- `db2 ?` command - information / syntax of a specific command
- `db2 ? SQLnnnn` - information about a specific SQLCODE. SQLCODE must be 4 to 5 digits in length.
- `db2 ? DB2nnnn` - information about an error generated by the CLP

The CLP has some parameters that can be viewed and also altered. To determine the current CLP settings, issue the following command:

```
DB2 LIST COMMAND OPTIONS
```

The following figure illustrates the output of this command. The values shown are the default settings of the CLP.

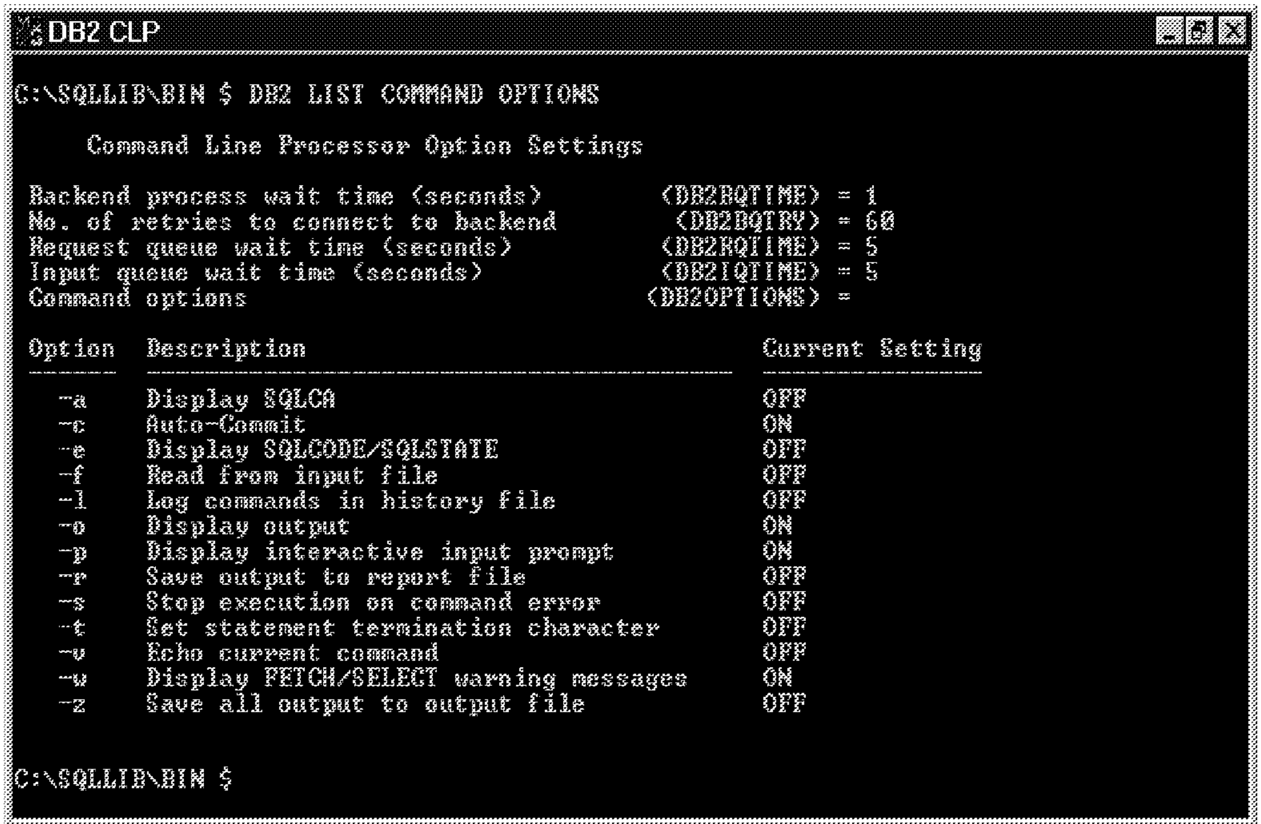


Figure 131. CLP Option Settings

The values of the CLP Option Settings can be altered for the interactive session by using the following command:

```
DB2 UPDATE COMMAND OPTIONS USING <options>...
```

To modify the CLP options for every session, you need to set the DB2OPTIONS system environment variable.

It is also possible to determine all the available options, the possible values for each option and the usage of the command by using the following command:

```
DB2 ? UPDATE COMMAND OPTIONS
```

It is also possible to create a file with SQL statements and DB2 commands and execute this file using CLP. For example, you can create a file called *test.clp* whose contents are as follows:

```
-- Sample File: test.clp
CONNECT TO SAMPLE;
CREATE TABLE names (n1 CHAR (25));
SELECT * FROM names;
COMMIT WORK;
CONNECT RESET;
```

The semi-colon ; is the default terminating character and can be changed by using the -t option. The -- represents a comment line.

To execute the CLP input file, test.clp, execute the following command:

```
DB2 -SVTF test.clp
```

The Command Line Processor consists of two processes, a front-end process and a back-end process. Both processes are essential for maintaining the database connection between each CLP invocation. The front-end process is called DB2, and the back-end is called DB2BP. These two processes communicate through three message queues: a request queue, an input queue and an output queue.

DB2BQTIME, DB2BQTRY, DB2RQTIME and DB2IQTIME offer a means of configuring communication between the two processes. When the CLP is invoked, the front-end process checks to see if the back-end process is already active. If it is, the front-end process reestablishes the connection with it. If not, the front-end process waits for the duration of time specified by the DB2BQTIME variable and then checks again. The front-end process will try to perform this checking for the number of times specified by the DB2BQTRY variable.Once the back-end process is activated, it waits on its request queue for a request from the front-end process. It also waits on a request queue in between requests initiated from the command prompt. The DB2RQTIME variable specifies the length of time the back-end process waits for a request from the front-end process.

When the back-end process receives a request from the front-end process, it sends an acknowledgment indicating that it is ready to receive input through the input queue. The back-end process then waits on its input queue. The DB2IQTIME variable specifies the length of time the back-end process waits on the input queue for the front-end process to pass commands.

The back-end process maintains the connection to the database and the connection can be released by using the terminate command. The interactive CLP session can be closed by issuing the quit command. (This does not release the database connection.)

***** Note *****

DB2 for Windows NT does not allow Command Line Processor commands to be used in every operating system window (as in case os OS/2). A DB2 command window must be used to issue any CLP statement.

5.4.1 Using the ATTACH Command

In DB2 you can perform certain tasks, such as creating a database, forcing off user connections, monitoring a database, and updating the database manager configuration at the instance level. There are other tasks, such as issuing DML, using the LOAD utility or BIND command, which require a database connection. DB2 provides the facility to remotely administer an instance (not remote administration of client nodes) on the server by using the ATTACH command. By attaching to an instance, you can perform the following functions:

- Create / drop databases
- Get / update / reset database manager configuration
- Get / update / reset database configuration
- Monitor database
- Backup / restore database
- Roll forward a database
- Force applications

The syntax of the ATTACH command is:

```
ATTACH [TO nodename] [USER surname [USING password]]
```

If the ATTACH command is executed without any arguments, the current attachment status is returned. If you attach to an instance while you are already attached to a different instance, the current instance is detached before the new attach is attempted. It is only possible to attach to one instance at a time.

It is important to note that database connections are independent of instance attachments. You can have an application that has multiple database connections simultaneous to an instance attachment. However, only one instance attachment can be maintained at any one time.

When attached to another instance, DB2 still uses the directory services of the local instance. Because of this, it is not possible to perform catalog changes or list, update or reset commands on database, node or DCS directories.

If you create a new database, that database will be created at the attached instance and a catalog entry will be created in the local instance directories which refers to the remote databases. To drop a database at the remote instance, the database needs to be cataloged locally.

To remove the logical instance attachment and terminate the physical communication connection, you can use the DETACH command.

5.4.1.1 ATTACH to Local Node

If you are running multiple instances on the local node and don't wish to change the environment variables, then you need to catalog the local node. To catalog the local node, use the following command:

```
CATALOG LOCAL NODE nodename INSTANCE instancename [WITH "comment string"]
```

where nodename is a user-defined nodename, and the instancename is the name of the instance to which you want to connect.

Let's assume that you have a DB2 Server running two instances, INST1 and INST2. You want to connect to both the instances without changing the environment. You can do the following:

1. Set the DB2INSTANCE variable to one of the instances.

   ```
   DB2INSTANCE=INST1
   ```

2. Catalog the local node.

   ```
   CATALOG LOCAL NODE DB2SERV2 INSTANCE INST2
   ```

3. It is possible to attach to both instances without modifying the system environment by using the following commands:

   ```
   ATTACH TO inst1
   DETACH
   ATTACH TO DB2SERV2
   ```

5.4.1.2 ATTACH to Remote Node

If a remote node is cataloged, then applications can attach to the instance name associated with the remote node name. The ATTACH command will refer to the node directory and use the information to determine the necessary information.

Let's assume that you have cataloged a NetBIOS node, db2ntnb, which refers to a DB2 for NT database server with instance INST1. You can attach to the instance by using the following command:

```
ATTACH TO db2ntnb USER userid USING password
```

5.5 Enabling ODBC Applications

This section discusses enabling ODBC applications to use against at DB2 for NT database server.

5.5.1 Overview of ODBC

Microsoft has developed a callable SQL interface called ODBC (Open Database Connectivity). ODBC implements and extends the standard callable SQL Interface from the SQL Access Group (SAG) and X/Open. The ODBC architecture allows applications to access data in several database management systems by using SQL. The ODBC interface gives the user a certain degree of flexibility. An application can access different database management systems, by using different database drivers that link the application to the specific database management system at run time.

5.5.1.1 ODBC Architecture

The ODBC architecture consists of four parts. The relationship between the components is illustrated in Figure 132 on page 194. Each component is discussed in this section.

Application: The application is responsible for making an ODBC function call to submit SQL requests and retrieve results. The application requests for a connection to a database, sends SQL requests to it and obtains the results. The application checks for any errors and reports them to the user. The application also takes care of transaction processing by requesting a `commit` or a `rollback`. Finally, on completion of the operation, the application will terminate the connection.

Driver Manager: The ODBC standard incorporates a driver manager in its design which provides a layer of indirection between the application and the database-specific ODBC drivers. When an application requests to connect to a data source (such as a database), the driver manager will load the necessary driver. The driver will then handle all further requests for that data source. The driver is responsible for managing multiple connections between applications and databases. It uses the ODBC.INI to associate a database name to a specific driver, i.e DLL.

Driver: In an ODBC environment, the driver manager provides the interface to the application. It also dynamically loads the necessary driver for the database server to which the application connects. It is the driver that implements the ODBC function set, with the exception of some extended functions implemented by the driver manager. It is the driver which interacts with a particular database.

Data Source: The data source is a particular database management system. It consists of data needed to perform the SQL requests and returns the result back to the application.

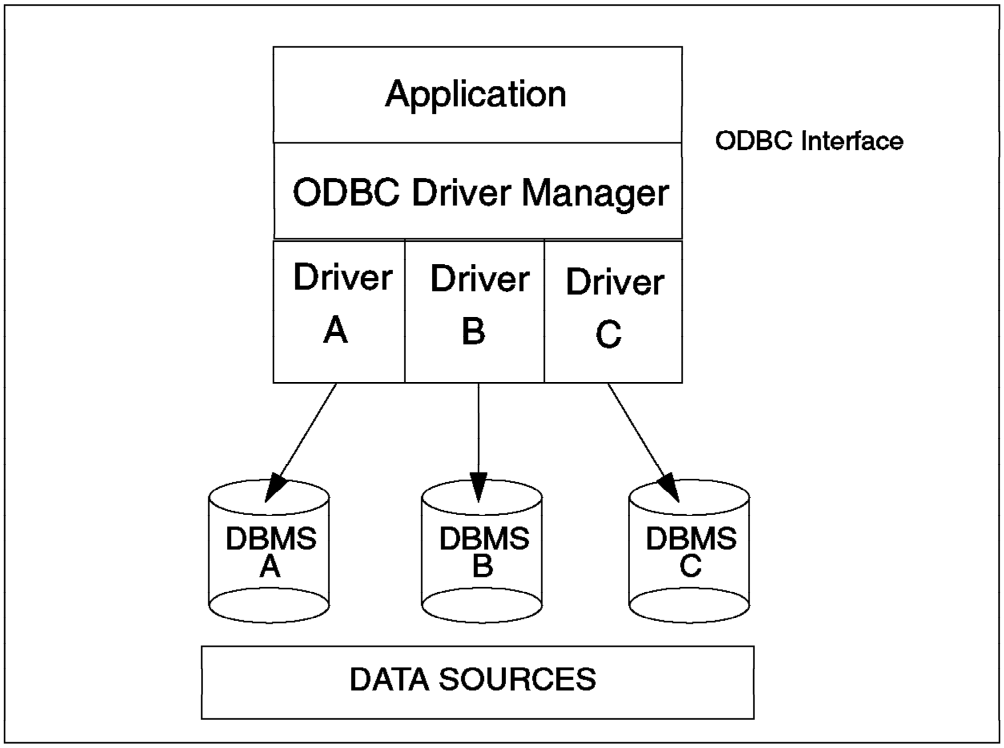


Figure 132. ODBC Architecture

Figure 132 illustrates the ODBC Architecture and its four components.

The advantage of using ODBC is the ease of porting. Also, the application can access more than one database source. Therefore, you could develop an application which accesses data from multiple data sources quite simply with the ODBC interface where it would be impossible with the DB2 CLI driver.

5.5.1.2 DB2 and ODBC

DB2 CLI is a callable SQL interface that is supported on the UNIX and Intel platforms. In Windows, it operates as an ODBC driver (a DLL that implements ODBC function calls and interacts with a DBMS). The DB2 CLI interface incorporates Microsoft's Open Database Connectivity (ODBC) Version 2 specifications. DB2 CLI provides full ODBC Level 1 support, as well as most level 2 functions (with the exception of functions like SQLBrowseConnect(), SQLDescribeParam(), SQLSetPos(), and some X/Open C supported functions). DB2 CLI also contains extensions to access features in IBM DBMS products such as the double byte (graphic) data types, compound SQL, stored procedures, large object (LOB) datatypes, user-defined types, user-defined functions, Distributed Unit of Work, and two-phase Commit. DB2 Version 2.1.1 and higher also supports Asynchronous ODBC.

Asynchronous ODBC means that the DB2 CLI/ODBC driver can now return control (SQL_STILL_EXECUTING) to an application rather than waiting until a function call completes. The application can execute the function, then poll it while executing other functions that do not act on the same SQL statement handle (hstmt) or connection handle.

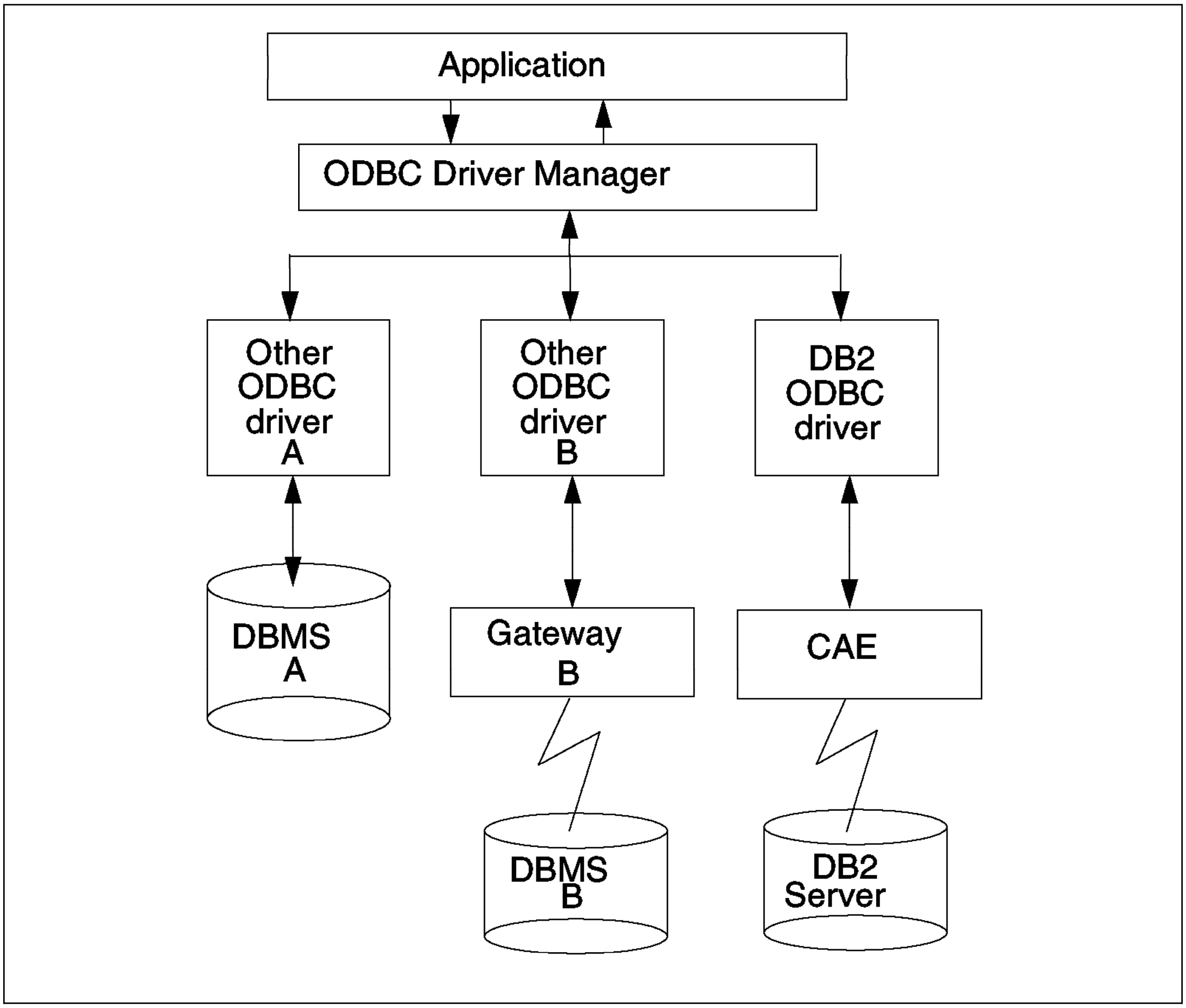


Figure 133. DB2 and ODBC Driver Manager Environment

5.5.2 32-bit and 16-bit ODBC Support

The ODBC Installation delivered with DB2 for Windows NT will allow you to use 16-bit ODBC applications as well as 32-bit ODBC applications.

5.5.3 Setup

In order for an application to access DB2 database through ODBC, the following steps must be performed.

5.5.3.1 Installation of the ODBC Driver

The standard installation of DB2 for Windows NT will do the following:

- Copy both the setup (DB2ODBC.DLL) and the driver (DB2CLI.DLL) to the Windows System directory.
- Adds IBM DB2 ODBC DRIVER to the ODBC drivers section in the ODBCINST.INI file. (See Figure 134 on page 196.)

- Adds a new section name IBM DB2 ODBC DRIVER and adds both the Driver and Setup entries for this new section in the ODBCINST.INI file, which configures the ODBC driver manager to be able to use the DB2 CLI driver.
- Install the ODBC Driver Manager, if necessary.

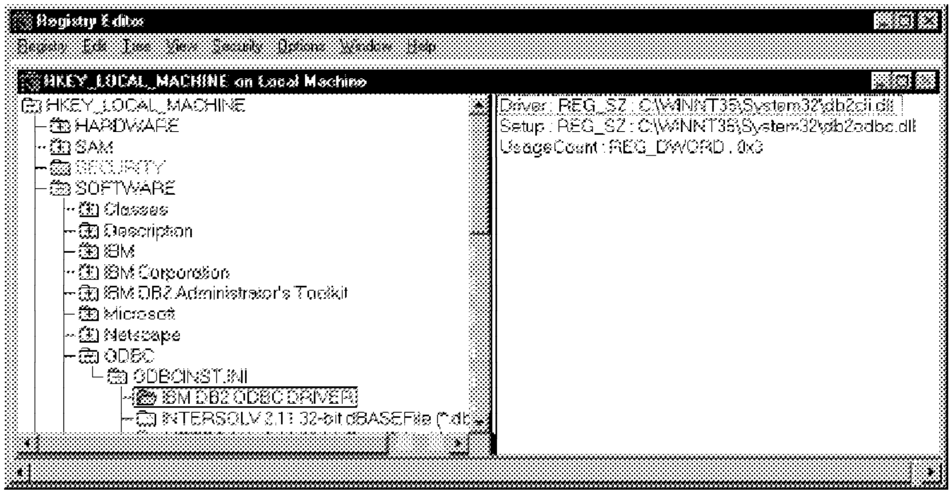


Figure 134. ODB2 Entry in the RegistryEntry HKEY_LOCAL_MACHINE

Note that if the ODBC tools do not display IBM DB2 as an additional source, perform the following:

- Execute the `<DB2PATH>bindb2odbc` command from the command window where `<DB2PATH>` is the path where DB2 product is installed. For example, C:SQLLIB.

5.5.3.2 Verifying the DB2 ODBC Installation

You can manually verify the ODBC driver information, if required. However, it is recommended that you do not edit the .INI file unless you are familiar with it. You can use the ODBC Administrator to edit all the parameters in this file.

You can verify if the IBM DB2 ODBC driver has been installed on your machine. To check this:

- Open the Control Panel.
- Double-click on the **32-bit ODBC Administrator icon** and the Data Sources window appears.
- Select the **Drivers** button, and the Drivers window appears. Verify that IBM DB2 ODBC Driver is shown.
- Close the Data Sources and Drivers Window.

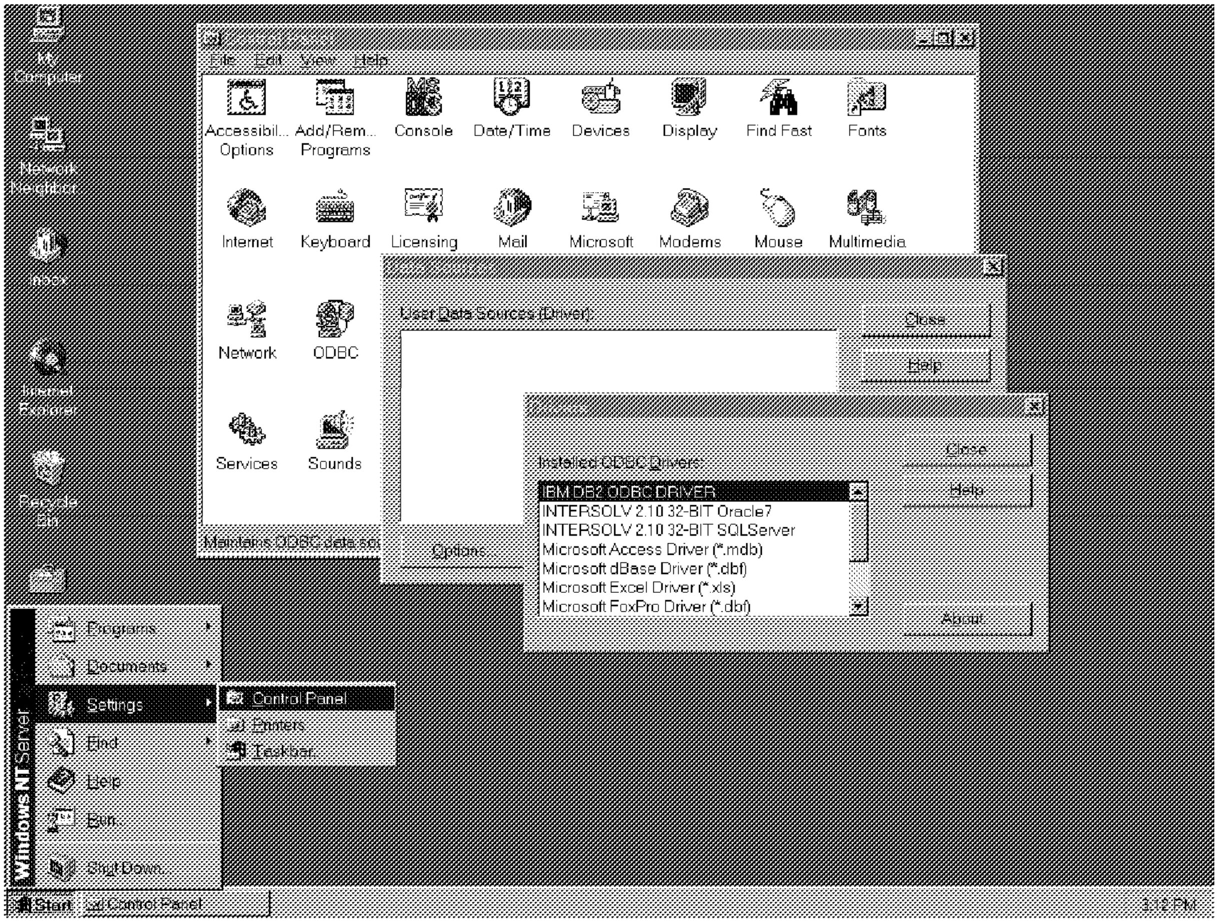


Figure 135. ODBC Driver Manager

5.5.3.3 Cataloging a Database

The DB2 Client setup utility supports you in cataloging a database. You will need a node or server to catalog. The local node is referred to as **This node.** If you want to catalog a database from another node you first have to catalog the node and the protocol you use. The next step is to catalog the database on the node and register it as an ODBC data source.

5.5.3.4 Binding CLI (ODBC) and Database Utilities

The ODBC driver will automatically bind the necessary utilities on the first connect to the database, provided you have the appropriate privilege and authorization. The administrator may need to perform the first connect or explicitly bind the files listed in the DB2CLI.LST list file.

5.5.3.5 Registering the Database with ODBC Driver Manager

The ODBC Driver Manager does not read the information in the DB2 catalog information. It references its own list of data sources contained in the ODBC.INI file. You need to register a data source with the ODBC Driver Manager in order for an ODBC application to access a DB2 database.

Use the DB2 Client Setup program to register the database with the ODBC Driver Manager as a data source by following these steps:

- Run the DB2 Client Setup program. The DB2 Client Setup program appears.

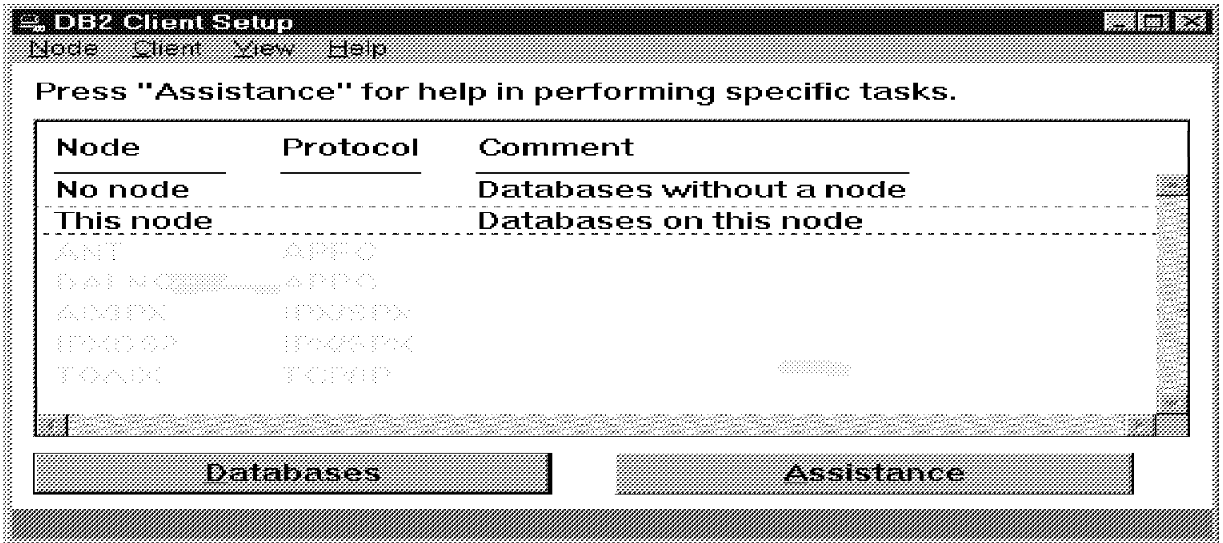


Figure 136. DB2 Client Setup Program

- Click on the **Assistance** push button and the DB2 Client Setup -Assistance window opens.

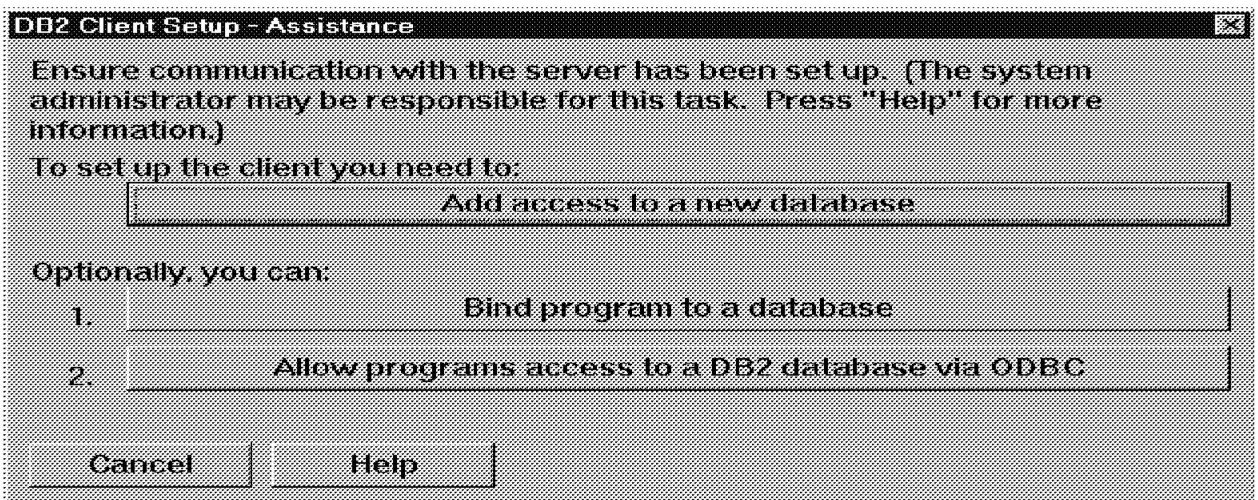


Figure 137. DB2 Client Setup - Assistance Panel

- Click on the **Allow programs access to a DB2 database via ODBC** pushbutton.
- Select the database from the list that you want to enable for ODBC. Click on the **Next** pushbutton to proceed with the setup.

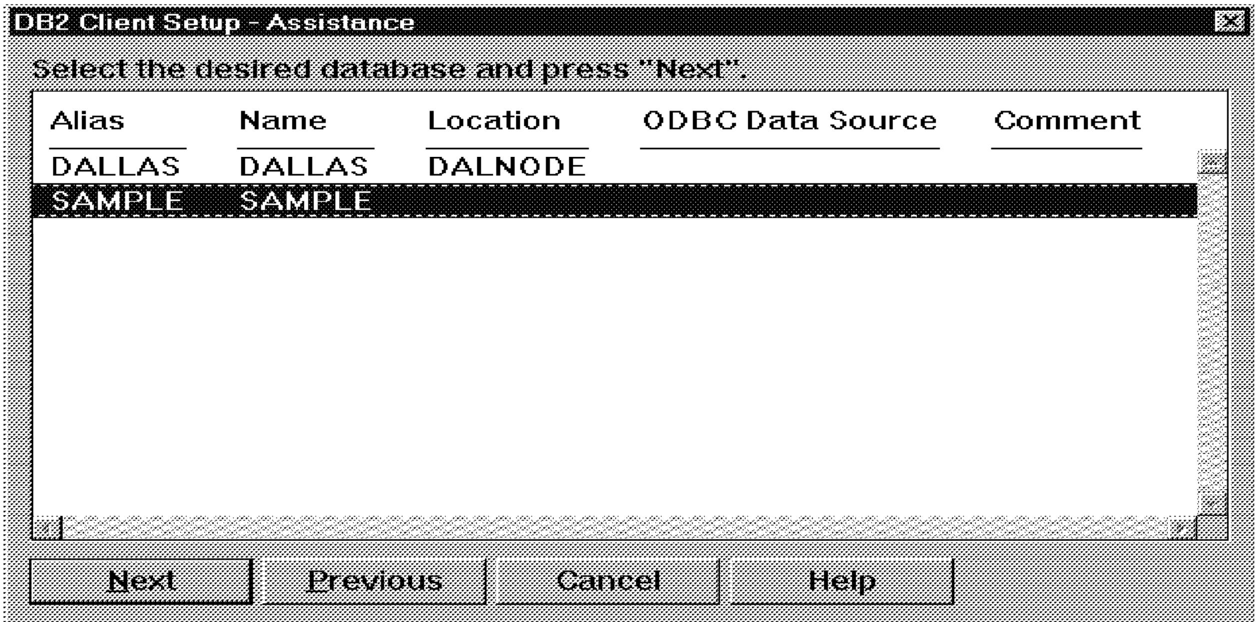


Figure 138. DB2 Client Setup - Assistance Window with Databases

- On the configure ODBC datasource panel, select Data Source and Description from the Setting from item list. Enter a description from the data source and click on **OK**.

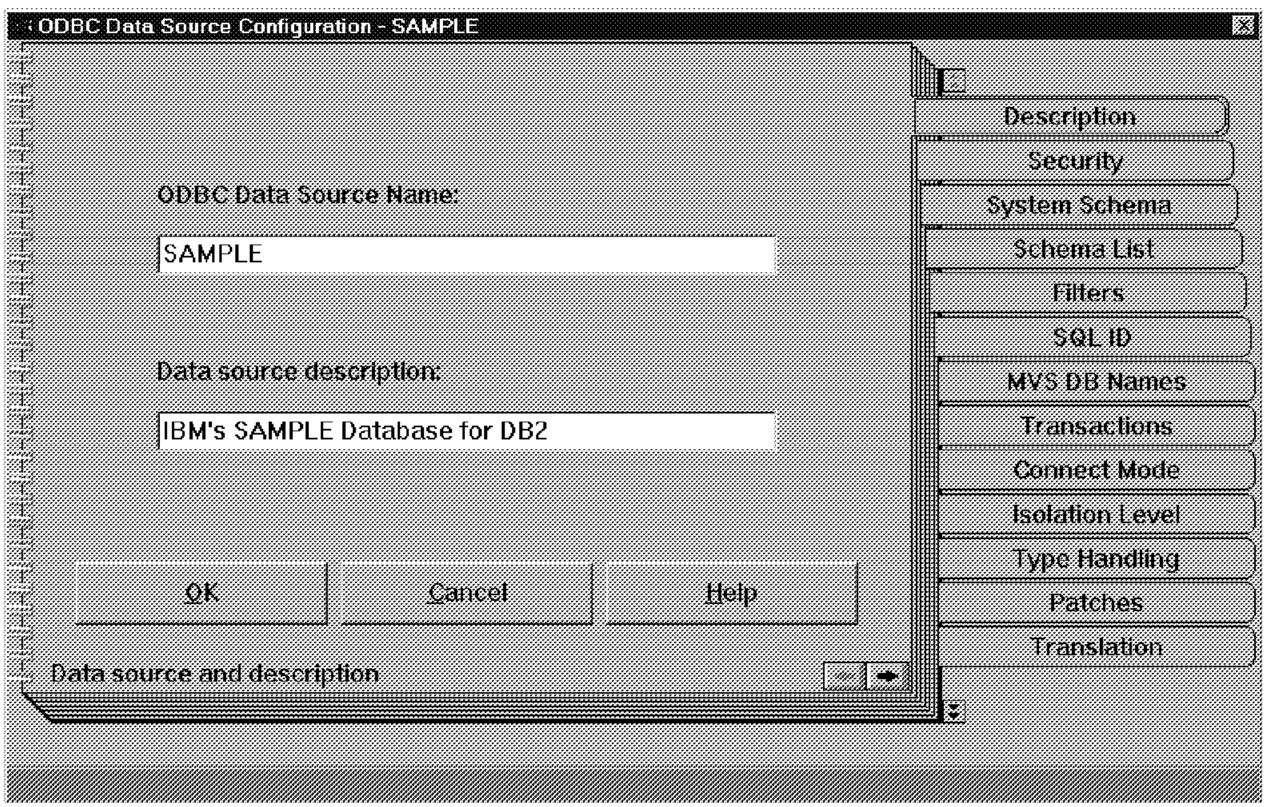


Figure 139. ODBC Database Configuration Window (SAMPLE Database)

- Click on **OK** to save the changes to the DB2CLI.INI file and to return to the Assistance panel.

As shown in Figure 139 on page 199, there are other additional tabs that represent other parameters which can be specified in the DB2CLI.INI file. These parameters are discussed in the following section. The same panel displayed can also be used to set these parameters.

***** Note *****

It is also possible to register data sources using a 32-bit Administrator icon in the Control Panel. For more information on how to register a data source using an ODBC Administrator, refer to ODBC documentation.

You can now start any application that supports ODBC such as Lotus Approach. When you open an ODBC database from within the application, you will be presented with a list of the databases you can connect to, including the one just registered.

5.5.3.6 Configuring the ODBC Driver (Optional)

It is usually not necessary to modify the DB2 CLI configuration file (DB2CLI.INI). However, it is important to understand that the file exists. It may require small modifications. Some of the reasons for modifying the CLI configuration file include:

- Increased CLI application performance
- Enable workarounds for specific applications

The IBM ODBC Driver can be further configured by using either the DB2 Client Setup Tool or by editing the <DB2PATH>db2cli.ini file. This file contains various keywords and values that can be used to modify the behavior of DB2 CLI and the applications using it. The keywords are associated with the database alias name and affect all DB2 CLI ODBC applications that access the database.

In the DB2CLI.INI file, there is one section for each data source (for example, database) the user wishes to configure. Each section always begins with the name of the database alias between square brackets:

[<u>database-alias</u>]

This is referred to as the section header.

The parameters are set up by specifying a keyword with its associated keyword value in the form:

KeywordName = <u>keywordValue</u>

There are some considerations which you need to be aware of while modifying DB2CLI.INI:

- All the keywords and the associated values for each database must be located below the database section header.
- The keyword settings in each section apply only to the database alias named in that section header.
- The keywords are not case-sensitive; however, their value can be if the values are character-based.

- If a database is not found in the .ini file, the default values for these keywords are in effect.
- Comment lines use a semicolon in the first position of a new line.
- If duplicate entries for a keyword exist, the first entry is used.
- Blanks are permitted

The following is a sample .ini file with database alias section:

```
; This is a comment line
[TCPSAMP]
DBALIAS=TCPSAMP
uid=userid
pwd=password
AUTOCOMMIT=0
CURSORHOLD=0
TABLETYPE="'TABLE','ALIAS','VIEW','SYSTEM TABLE'"
```

There are many parameters that can be specified in the configuration file. Here we will discuss some of the important parameters which can drastically affect the execution of an application.

Table 13. Important Keywords in DB2CLI.INI

Keyword	Description
AUTOCOMMIT	0=OFF / 1=ON (Default = 1) This means that every successful SQL statement will be automatically committed and therefore no need to issue the SQLTransact() API to commit the Unit of Work. Many CLI/ ODBC applications have been designed assuming this behavior; so be careful if you decide to change this parameter. You may want to turn this off so that you can control the commit scope of your transactions.
CURSORHOLD	0=CURSOR NOHOLD / 1=CURSOR HOLD (Default = 1) This means that the cursors are maintained across Units of Work. This is quite different from embedded SQL as all cursors exhibit the cursor without hold behavior unless the `DECLARE CURSOR` statement includes the phrase `WITH HOLD`. You may want to turn off the cursor hold behavior if do not need to maintain cursor position across transactions.
MULTICONNECT	0=FALSE / 1=TRUE (Default = 0) The parameter is used to identify if the SQLConnect API will establish a new physical connection (like CONNECT TO in embedded). You may want to set this value to 1 if you want all SQL statements to be mapped to a single database connection. This means that `ROLLBACK` will roll back all statements on all connections. It is recommended that you verify if your application is designed to work with `MULTICONNECT` before doing so or the application may not function correctly.
TXISOLATION	1=Uncommitted Read 2=Cursor Stability (Default) 8=Repeatable Read 16=Repeatable Stability 32=No Commit (DB2 for OS/400 only) This identifies the Isolation Level used for concurrency.
ASYNCENABLE	0=DISABLE/1=ENABLE Disabling this function will disable the new asynchronous feature. The DB2 CLI/ODBC driver will act as it did with previous versions that did not support asynchronous ODBC.

For the complete list of keywords, their syntax and the possible values, refer to *DB2 Installing and Using Clients for Windows NT*.

Note that there is also an additional entry in the HKEY_Current that is user specific for the user who did the registration as the ODBC data source. When different users are working on one workstation using ODBC, tools provide the information in the logon script.

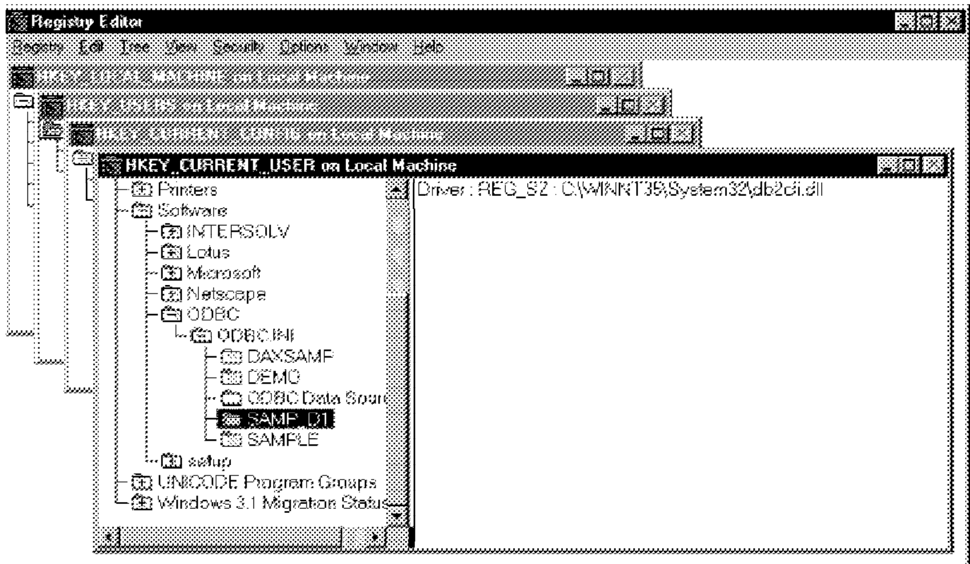


Figure 140. ODBC Entry in HKEYCURRENT_USER

5.5.4 MS-Access

MS-Access is an example of ODBC end-user tools that uses the ODBC setup information provided in the ODBC.INI. This entry will be automatically generated when you do the setup.

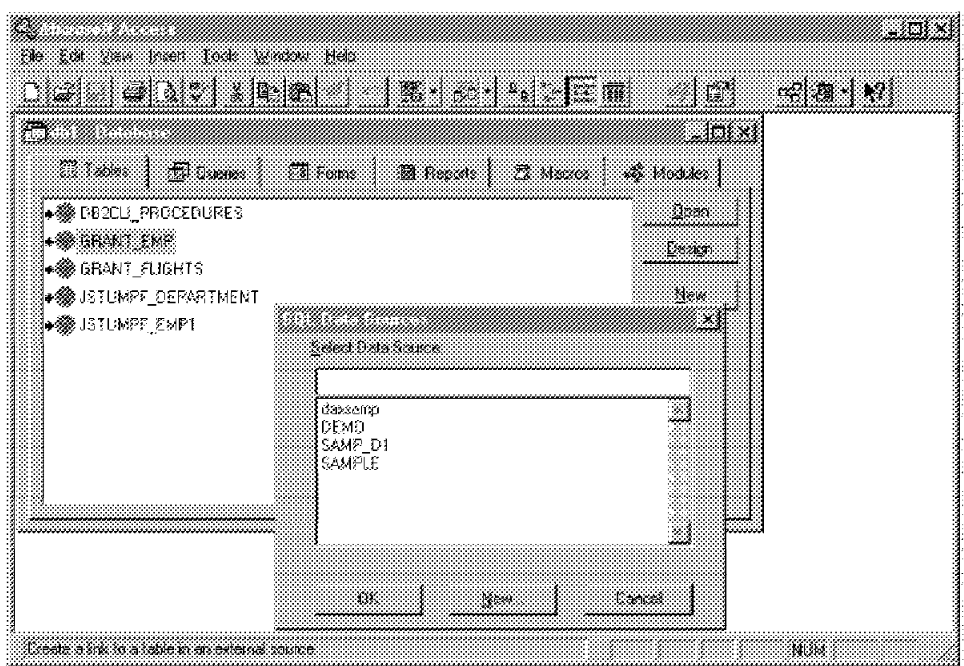


Figure 141. MS-ACCESS Linking Table of a Database

You can include a DB2 table in a MS-Access database by executing the following steps:

Open a database with clicking on **FILE**. Choose **OPEN DATABASE**. I you have several, you can choose one of them.

You will be able to set links to DB2 tables in your MS-Access database.

- Click on **File** in the task bar of ACCESS and choose **Get External Data Link Tables...**You will see a window with the title LINK in which you find a field with a selection list called Files of type.
- Select **ODBC Databases()** out of the item list. MS-Access comes up with a new Window named SQL Data Sources.
- Select a database (for example, SAMPLE) and click on the button **OK.** The window IBM DB2 ODBC Driver appears.
- Type in your user Id and password if different from the one you used for your Windows NT logon. Click on the **OK** button**.**
- MS-Access will connect to the database and bring up a new window called Link Tables. This window shows you all the tables of the database.
- Select one or more tables or use the additional buttons of the window to select all the tables you want to use in MS-Access. Click on the **OK** button. If your table has a primary key or unique index, you will see the table as a linked table in your MS-Access database. If it doesn't, MS-Access comes up with a window titled Select Unique Record identifier. Here you should choose one or more fields of the table that are building together a unique identifier for a row and click on **OK.** The table name is a combination of the creator and the original table name (For example, the creator GRANT tablename EMP ACCESS tablename GRANT_EMP.)
- Double-click on the tablename. MS-Access will come up with the window titled IBM DB2 ODBC Driver to connect again, or it will directly display the table if the connect still exists.

You can also export MS-Access tables to DB2 in a similar way by selecting **File SaveAs/Export** and selecting the desired option.

5.5.5 Lotus Approach

Lotus Approach also uses the information provided through setup in the ODBC.INI. You can open a DB2 table as an Lotus Approach database. This has to be done in the following sequence:

- Select **File** in the taskbar.
- Click on **Open** in the selection list. You will receive an additional Window similar to the window shown in Figure 142 on page 205.
- Open the selection list for Files of type.You can select either **ODBC Data Sources(*)** or **IBM DB2(*)** to connect to a DB2 database.
- Choose **IBM DB2(*).** Approach starts to connect to the database. You get the IBM DB2 ODBC Driver connection window.
- Open the database item list by clicking on the button beside the database name, and select a database out of a selection list.
- Click on **OK** if your userId has the right to access the database; otherwise enter your user Id and password and then click on **OK**. There should be a connection to a database.
- Double-click on the **connect <USERID>@DB2**. The window displays all the different prefixes or collections for tables.
- Double-click on a collection, such as **<USERID>**. You will get a similar window to Figure 142 on page 205 with all the tables created with that <USERID>.
- Double-click on a table name. You should get a form and a worksheet of the table.

Note that the table should have a primary key or a unique index that enables Approach to write back. Otherwise Approach will copy the table in an Approach table.

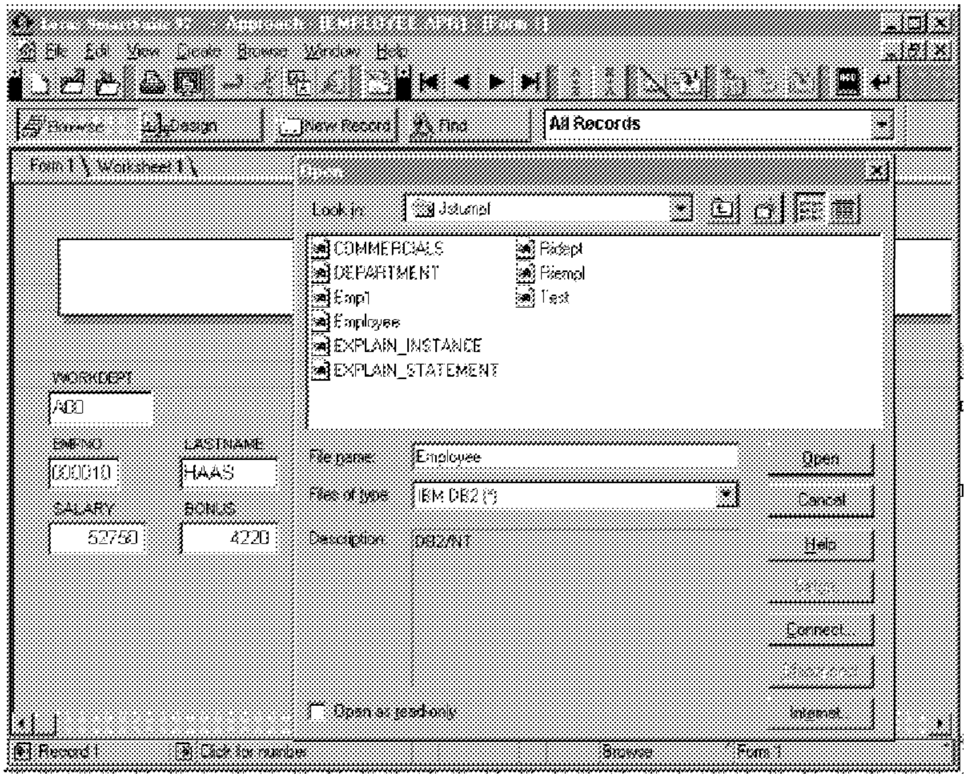


Figure 142. Lotus Approach Displaying Contents of Table

5.6 Application Considerations

There are two main ways of writing applications that access DB2 databases:

1. Dynamic SQL. There are two ways to implement:

- Code function calls to the DB2 Call Level Interface to invoke SQL statements.
- Use embedded dynamic SQL. This means SQL statements are included in your code.

1. Static SQL in the form of embedded static SQL statements in your application.

DB2 also provides the following programming options:

- Application programming interfaces (APIs)
  Your application can call DB2 APIs to perform functions such as backing up and restoring databases. Refer to the *IBM DATABASE 2 API Reference* for more information about APIs.

- Stored procedures
  Your application, executing on a client workstation, can call procedures stored at the database server by using the SQL CALL statement. The procedures access the database and return information to your application. Your client application can be on one platform, and the stored procedure on

the server can be on another platform. Refer to the *Application Programming Guide* for more information about stored procedures.

- User-Defined Functions (UDFs)
  You can develop your own scalar functions that are specific to your database, and you can store them on the server. Your client application can then call the functions as required. You can use UDFs to enhance the use of User-Defined Types (UDTs). Refer to the *Application Programming Guide* for more information about UDFs and UDTs.

5.6.1 CLI

To understand DB2 CLI or any callable SQL interface, it is helpful to understand what it is based on, and to compare it with existing interfaces.

The X/Open Company and the SQL Access Group jointly developed a specification for a callable SQL interface referred to as the X/Open Call Level Interface. The goal of this interface is to increase the portability of applications by enabling them to become independent of any one database vendor′s programming interface. Most of the X/Open Call Level Interface specification have been accepted as part of the ISO Call Level Interface Draft International Standard (ISO CLI DIS).

Microsoft developed a callable SQL interface called Open Database Connectivity (ODBC) for Microsoft operating systems based on a preliminary draft of X/Open CLI. The Call Level Interface specifications in ISO, X/Open, ODBC, and DB2 CLI continue to evolve in a cooperative manner to provide functions with additional capabilities.

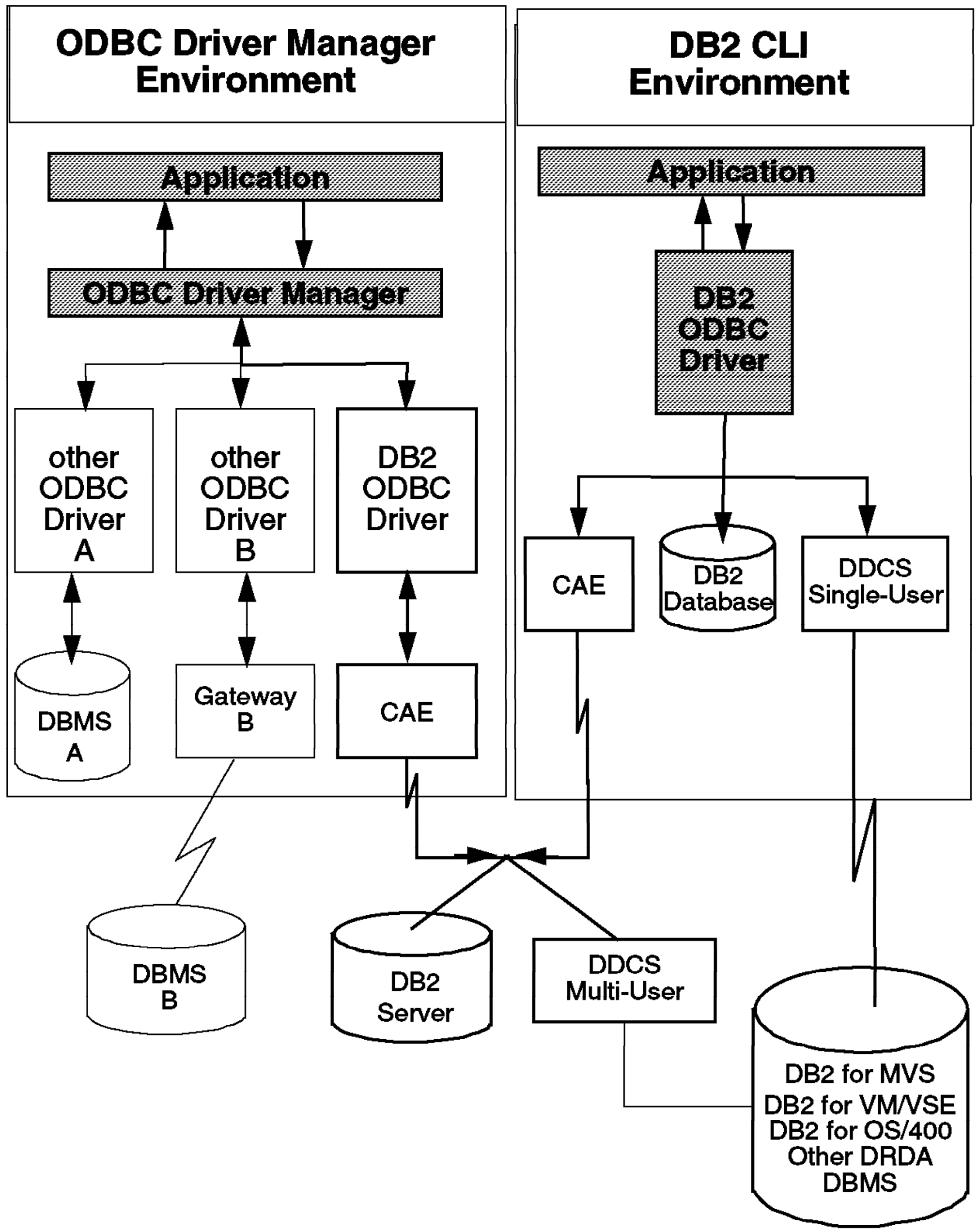


Figure 143. Overview of DB2 CLI

DB2 Call Level Interface (CLI) is IBM's callable SQL interface for the DB2 family of database servers. It is a C and C++ Application Programming Interface for relational database access. It uses function calls to pass dynamic SQL statements as function arguments. It is an alternative to using embedded dynamic SQL, but unlike embedded SQL, it does not require host variables or a precompiler.

DB2 CLI is based on the Microsoft Open Database Connectivity (ODBC) specification, and the X/Open Call Level Interface specification. The DB2 CLI driver can either be invoked by an ODBC Driver Manager or directly invoked. Both methods are shown in Figure 143.

5.6.1.1 Configuring DB2 CLI

The configuration for DB2 CLI is the same as for the ODBC driver. You have to do the following steps:

- Catalog the node.
- Catalog the database on the node.
- Register the database as an ODBC data source.

This will update the DB2CLI.INI file.

5.6.1.2 CLI Program Structure

The following section gives you an overview of the program structure of a CLI program.

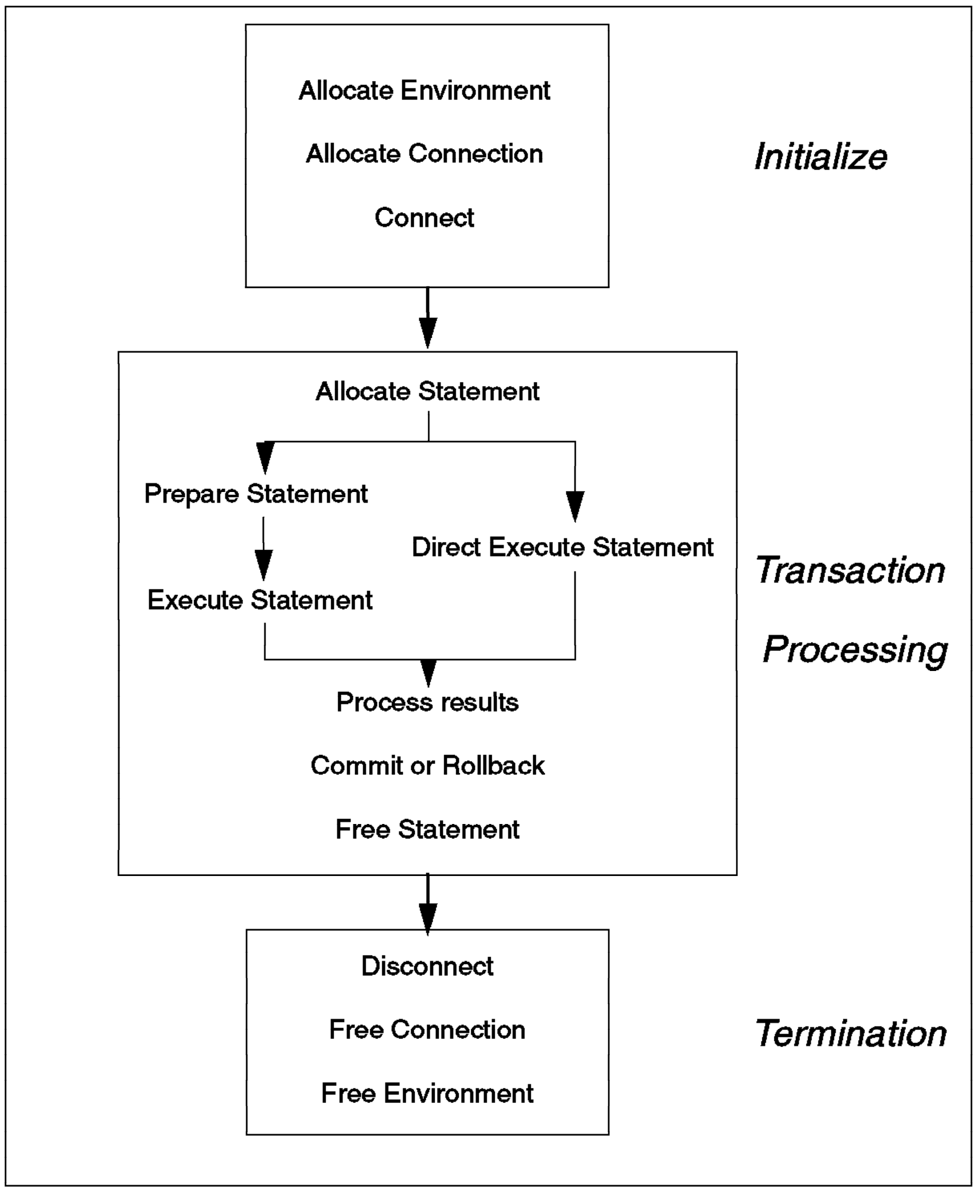


Figure 144. Conceptual View of CLI Application

Steps of the programming specifics, as shown in Figure 144, can be divided into three main steps:

1. **INITIALIZE**
   In this phase, the resources such as the environment for the process and the handle for the connection will be allocated, and a connection will be issued. A handle is a variable that refers to a data object controlled by the DB2 CLI. There are environment, connection and statement handles. Using handles frees your application from having to allocate and manage global variables or data structures such as the SQL Communication Area (SQLCA) used in embedded SQL applications.

2. **TRANSACTION PROCESSING**
   In this phase, the statement handle will be allocated, and SQL statements will run against the database. Once a statement handle has been allocated, there are two methods of specifying and executing SQL statements:

   a. Prepare and execute statement
      This method splits the preparation of the statement from its execution. It is used when the statement will be executed repeatedly (usually with different parameter values). This avoids having to prepare the same statement more than once. The subsequent executions make use of the access plans already generated by the prepare. The application requires information about the columns in the result set prior to statement execution.

      - Call SQLPrepare() with an SQL statement as an argument.
      - Call SQLBindParameter(), or SQLSetParam() if the SQL statement contains parameter markers
      - Call SQLExecute().

   b. Execute direct
      This method combines the prepare step and the execute step into one. This method is used when the statement will be executed only once. This avoids having to call two functions to execute the statement. The application does not require information about the columns in the result set before the statement is executed.

      - Call SQLBindParameter() or SQLSetParam() if the SQL statement contains parameter markers.
      - Call SQLExecDirect() with an SQL Execute direct statement as an argument.

   c. When either method (prepare and execute statement or execute direct) is done, data can be used for further processing.

   d. Commit or rollback
      Your application can either perform a commit or rollback of the entire transaction done between this point-in-time and your last commit.

   This is the end of one transaction process inside the environment.

   a. Free statement handle will end the usability of a statement and free the resources.

   The transaction ends at this point or the next transaction process will start within the same environment.

3. **TERMINATION**
   This step consists of disconnecting and releasing DB2 CLI resources. The step is further divided as follows:

- Disconnect and release the connection handle.
- Release the environment handle.

5.6.1.3 CLI Program

The following is an example of the structure of a CLI program. It illustrates keywords that you will find in a CLI application program. It also provides an overall view of the programming sequence in a CLI program.

```
/* include header files */
#include ..

int main(

/*********************************************************** INITIALIZE ******
*  Allocate an environment and a connection and
* then try to connect to database.
**********************************************/
rc = SQLAllocEnv( &hEnv );
.
    rc = SQLAllocConnect( hEnv, &hDbc );
.
    rc = SQLConnect(..)
*********************************************************** Start TP1 *****
* Allocate a statement and create the sample table.
***************************************************/
    rc = SQLAllocStmt( hDbc, &hStmt );     /*

    rc = SQLExecDirect( hStmt, szCreate, SQL_NTS );
.
    rc = SQLTransact( hEnv, hDbc, SQL_COMMIT ); /********** END TP1 *******/
/********************************************************* START TP2 ******
 *  Prepare an insert statement.
 *******************************/
    rc = SQLPrepare( hStmt, szInsert, SQL_NTS );
/*
* Set input parameter.
 */
    rc = SQLSetParam(..)
/*
 *  Insert a couple of rows into the database.
 */
    rc = SQLExecute( hStmt );
.
    rc = SQLExecute( hStmt );
/*
 *  Commit the changes.
 */
    rc = SQLTransact( hEnv, hDbc, SQL_COMMIT );
/********************************************************* END TP2 *******
 *  Reset input parameter.
 */
    rc = SQLFreeStmt( hStmt, SQL_RESET_PARAMS );
/*************************************************** START TP3.1 *******
 *  Execute the select statement.
 */
    rc = SQLExecDirect( hStmt, szSelect, SQL_NTS );
/*
 *  Return number of columns and describe result set.
 */
    rc = SQLNumResultCols( hStmt, &cCol );

    rc = SQLDescribeCol(......)

    printf( .. );
/*
 *  Bind the output parameter.
 */
    rc = SQLBindCol(...)
/*
 *  Fetch data.
 */
    rc = SQLFetch( hStmt ); /* repeate if sqlerror != 100
/******************************************************** END TP3.1 ******
 *  Close cursor and free bound columns.
 */
    rc = SQLFreeStmt( hStmt, SQL_CLOSE );
```

```
    rc = SQLFreeStmt( hStmt, SQL_UNBIND );
/***************************************************** START TP3.2 ******
 *  Drop table.
 */
    rc = SQLExecDirect( hStmt, szDrop, SQL_NTS );
/*
 *  Commit work.
 */
    rc = SQLTransact( hEnv, hDbc, SQL_COMMIT );
/*
 *  Disconnect and free up CLI resources.************ TERMINATION *********
 */
    rc = SQLFreeStmt( hStmt, SQL_DROP );
    rc = SQLDisconnect( hDbc );
    rc = SQLFreeConnect( hDbc );
    rc = SQLFreeConnect( hDbc );
    rc = SQLFreeEnv( hEnv );
/*
 *  Retrieve last error.
 */
    rc = SQLError(..)
```

5.6.1.4 Executing the CLI Sample Programs

There are several different types of sample programs included with the DB2 SDK product. The sample programs for DB2 CLI are in the subdirectory $(DB2PATH)samplescli. To use these sample programs, you should view the contents of the included MAKEFILE to add the following information:

- Change the entry regarding the compiler if using Visual C++ or VisualAge for C++. The entry should be set to COMPILER=IBM
- Check the entry for datasource, USERID and PASSWORD
  Use your user Id and password and the database name, for example, the SAMPLE database, as the data source.

To successfully execute the CLI sample program in a Windows NT environment, you will need the following products:

- Windows NT Server V 4.0
- DB2 for Windows NT V2.1.2 Server
- DB2 SDK V2.1.2
- IBM VisualAge for C++ for Windows NT V3.5

The following is an example of how to create a specific execution file to use with the sample CLI programs. Note that the following was executed from an NT DB2 command line window:

```
C:SQLLIBsamplescli>nmake adhoc.exe

IBM(R) Program Maintenance Utility for Windows(R)
Version 3.50.000 Feb 13 1996
Copyright (C) IBM Corporation 1988-1995
Copyright (C) Microsoft Corp. 1988-1991
All rights reserved.

        icc /c /Ti /W1 /I:C:\SQLLIB\include adhoc.c
IBM(R) VisualAge(TM) for C++ for Windows(R), Version 3.5
- Licensed Materials - Program-Property of IBM
(C) Copyright IBM Corp. 1991, 1996 - All Rights Reserved.

        ilink /MAP /DEBUG /ST:64000 /PM:VIO -out:adhoc.exe adhoc.obj samputil.obj
    C:\SQLLIB\lib\db2cli.lib

IBM(R) Linker for Windows(R), Version 02.00. r2_960215
Copyright (C) IBM Corporation 1988, 1996.
Copyright (C) Microsoft Corp. 1988-1989.
All rights reserved.
```

5.6.2 Embedded SQL

The following sections provides a discussion on using embedded SQL in your application programs.

5.6.2.1 Overview

Embedded SQL is currently the most popular programming interface used to access databases. It allows programmers to write SQL statements into application programs that are written in a languages such as C or COBOL. The steps involved in processing embedded SQL within your application program is shown in Figure 145 on page 213:

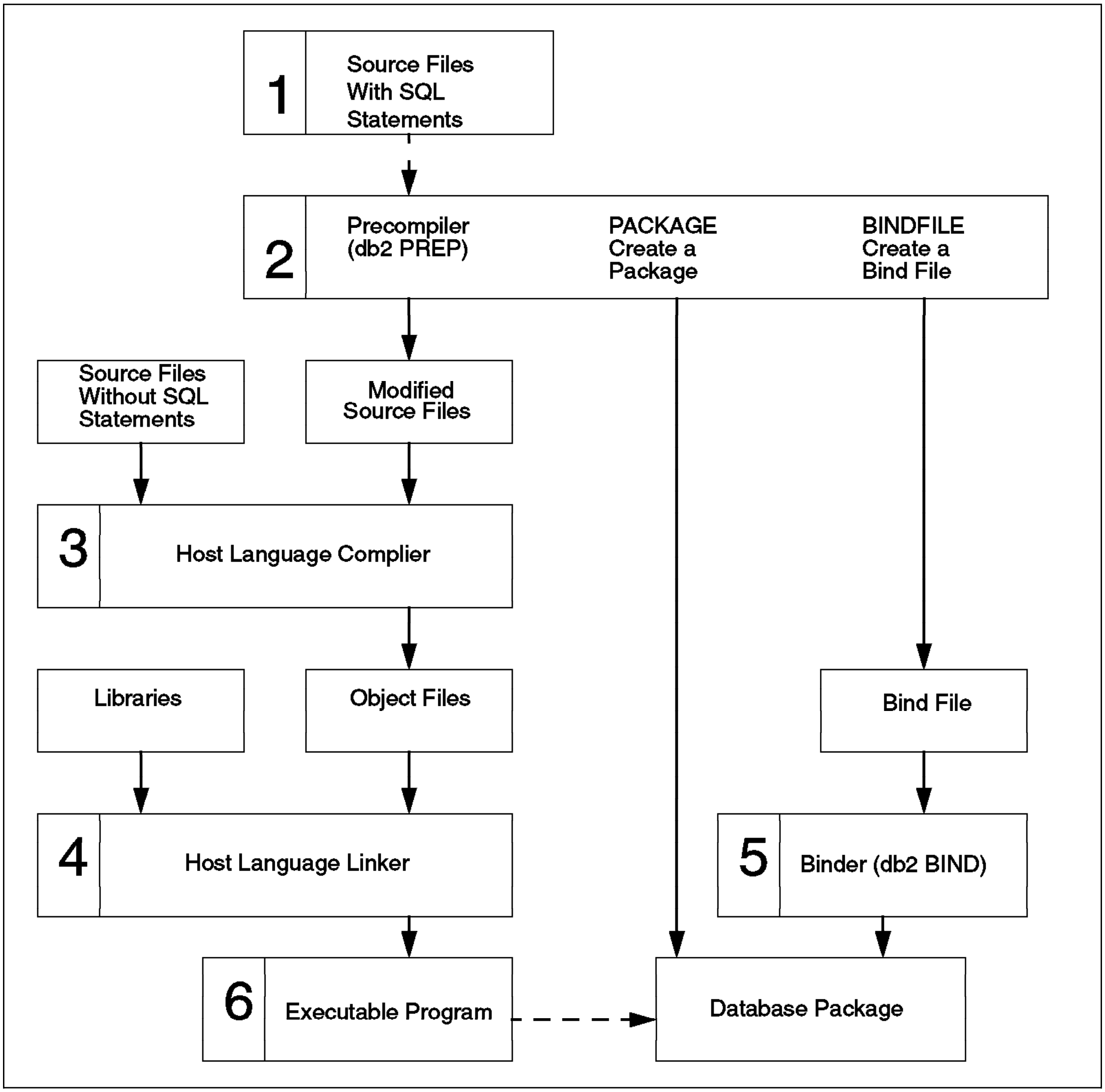


Figure 145. Processing Embedded SQL

The following steps are common to both types of embedded SQL programming against a database. The steps shown in Figure 145 are described as follows:

1. Create a source file. You create a source file including SQL as discussed in 5.6.2.3, "Dynamic Embedded SQL" on page 215, or in 5.6.2.7, "Static Embedded SQL Program" on page 221.

2. Precompile your source file. This precompile step is considered complete in terms of static SQL when all of the information for the SQL statement is known. In terms of dynamic embedded SQL, this is the first part of preparing a statement. The complete prepare must be done during run time with the PREPARE statement. The precompiler (PREP) must be invoked. The source code will be modified during the precompile stage in that the SQL statements are commented in the source code, and information will be stored in one of the following methods:

- As a package in a database. This is the default value for the prepare step. It assumes that you are connected to the database and that you have the privilege to issue a BIND statement against the database

  As a bind file (.BND). In this case, you have to explicitly specify that you want to generate a bind file by adding the option BINDFILE. The command format is similar to the following:

  ```
  DB2 PREP <filename> BINDFILE
  ```

  You can be connected to any database.This may be necessary in one of the following:

  1. You do not have bind authority to execute the BIND statement.
  2. You have created an application which executed a Distributed Unit of Work.

  The bind file will be processed in Step 5.

1. Compile. The modified source files can now be compiled together with the other source files.

2. Link. The compiled files will now be linked together with additional library files to an executable program. The executable file can be either a file with an extension of (.EXE) or a Dynamic Link Library (.DLL). Stored procedures are examples of a DLL.

3. Bind. In this step, a user with BIND privilege against the database will be connected to the particular database and perform the bind operation. The package will now be created in the database with the collection ID of the user who did the precompile. Note that using a collection ID that is different than that of the user who performed the precompile will generate an error, and the application will not successfully execute.

4. Execution. From either the client workstation where the program was created or any client workstation with DB2 CAE installed, you can execute the application. Be aware that you may have to supply a database name, user Id or password that is different than those that are used to log on to the client workstation. Note that the utility, db2bfd.exe, is located in the subdirectory $(DB2PATH)misc. It has parameters that allow you to control the bind file.

5.6.2.2 Using Cursors in Embedded SQL

To allow an application to retrieve a set of rows, SQL uses a mechanism called a *cursor*. To help understand the concept of a cursor, assume that the database manager builds a result table to hold all the rows retrieved by executing a SELECT statement. A cursor makes rows from the result table available to an application, by identifying or pointing to a current row of this table. When a cursor is used, an application can retrieve each row sequentially from the result table until an end of data condition, that is, until the NOT FOUND condition, SQLCODE +100 (SQLSTATE 02000) is reached. The set of rows obtained as a result of executing the SELECT statement can consist of zero, one or more rows, depending on the number of rows that satisfy the search condition.

The steps involved in processing a cursor are as follows:

1. Specify the cursor using a DECLARE CURSOR statement. The DECLARE CURSOR statement associates a cursor with a prepared SQL statement. If the prepared SQL statement is a SELECT statement, a cursor is necessary to retrieve the rows from the result table.

2. Perform the query and build the result table using the OPEN CURSOR statement. The OPEN CURSOR statement initializes the cursor declared earlier to point before the first row of the result table is received. The USING clause specifies a host variable to replace the parameter marker in the prepared SQL statement. The data type and length of the host variable must be compatible with the associated column type and length.

3. Retrieve rows one at a time using the FETCH statement. The FETCH statement is used to move the column from the result table into the table_name host variable. The host variable is printed before the program loops back to fetch another row.

4. Process rows with the DELETE or UPDATE statements (if required).

5. Terminate the cursor using the CLOSE statement. The CLOSE statement closes the cursor and releases the resources associated with it.

An application can use several cursors concurrently. Each cursor requires its own set of DECLARE CURSOR, OPEN, CLOSE, and FETCH statements.

5.6.2.3 Dynamic Embedded SQL

The following table shows some of the statements for dynamic embedded SQL. Statements in C or C++ and COBOL are presented.

Table 14. Dynamic Embedded SQL Programming Examples

LANGUAGE	EXAMPLE SOURCE CODE DYNAMIC EMBEDDED SQL
C/C++	EXEC SQL BEGIN DECLARE SECTION; .. EXEC SQL END DECLARE SECTION; EXEC SQL CONNECT TO dbname ..; strcpy(prep_string, "SELECT tabname FROM syscat.tables" "WHERE tabschema = ?"); EXEC SQL PREPARE s1 FROM :prep_string; EXEC SQL OPEN c1 USING :host_var; EXEC SQL FETCH c1 INTO :table_name; EXEC SQL CLOSE c1 EXEC SQL COMMIT EXEC SQL CONNECT RESET
COBOL	EXEC SQL BEGIN DECLARE SECTION END-EXEC. host_var.. name.. EXEC SQL END DECLARE SECTION END-EXEC MOVE "SELECT TABNAME FROM SYSCAT.TABLES WHERE TABSCHEMA = ?"TO PREP-STRING. EXEC SQL PREPARE s1 FROM :PREP-STRING END-EXEC. EXEC SQL DECLARE c1 CURSOR FOR s1 END-EXEC. EXEC SQL OPEN c1 USING :host-var END-EXEC. EXEC SQL FETCH c1 INTO :name END-EXEC EXEC SQL CLOSE c1 END-EXEC.

In the dynamic SQL, the query is associated with a statement name assigned in a PREPARE statement. Any referenced host variables are represented by parameter markers. A DECLARE statement is associated with a dynamic SELECT.

5.6.2.4 Structure of an Embedded SQL Program Using Cursors

The following shows the steps when using cursors and embedded SQL within an application.

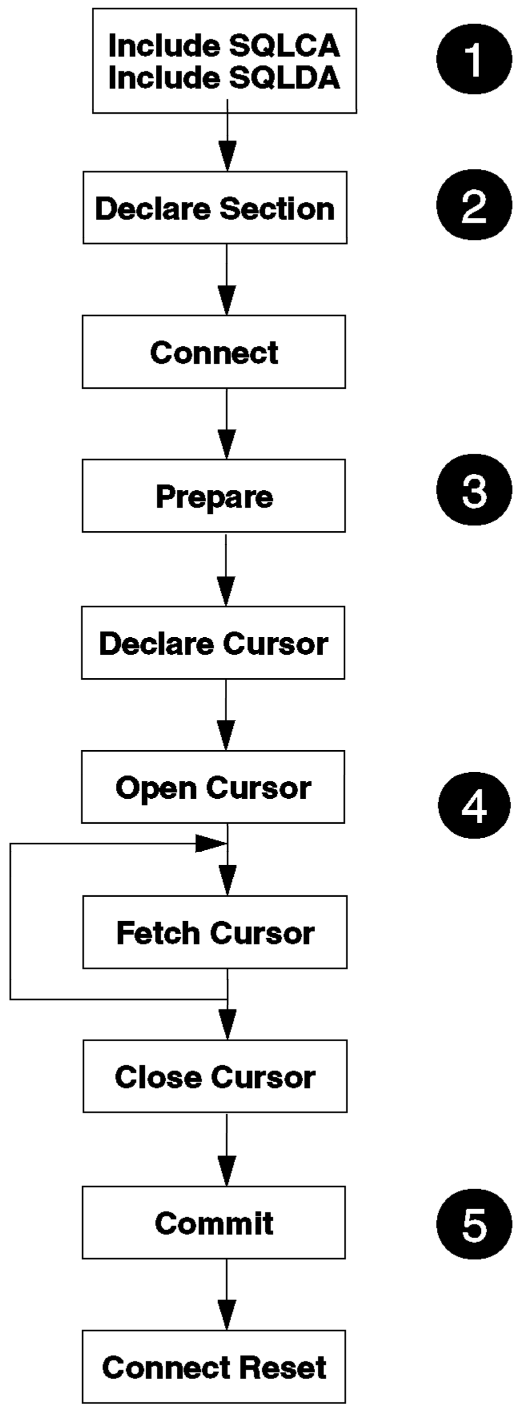


Figure 146. Structure of Dynamic Embedded SQL Program

The steps shown in Figure 146 are explained as follows:

1. Additional header files are needed to communicate with the database and to exchange data. The SQLCA is used for communication, and SQL Data Area (SQLDA) is used in data exchange with the application.

2. Declare host variables. This section includes declarations of three host variables:

   a. table_name is used to hold the data returned during the FETCH statement.

   b. st is used to hold the dynamic SQL statements in text form.

   c. parm_var supplies a data value to replace the parameter marker in st.

3. Prepare the statement. An SQL statement with one parameter marker (indicated by ?) is copied to the host variable. This host variable is passed to the PREPARE statement for validation. The PREPARE statement parses the SQL text and prepares an access section for the package in the same way that the precompiler or binder does, only it happens at run time instead of during preprocessing. Note that this is necessary if you run embedded dynamic SQL.

4. The following positions are the statements including the CURSOR statements if necessary.

5. Commit or rollback ends the transaction process depending on the logic of the program

If errors are detected, error code information with the variables is included in the header file SQLCA. The CHECKERR macro/function is an error checking utility which is external to the program that uses these variables. The location of this error checking utility depends upon the programming language used:

- C check_error is redefined as CHECKERR and is located in the util.c file.
- COBOL CHECKERR is an external program named checkerr.cbl

5.6.2.5 Sample of Dynamic Embedded SQL Using Cursors

The following program is a portion of program showing some of the keywords used with dynamic embedded SQL.

```
include<..>
EXEC SQL INCLUDE SQLCA;/* setup communication area

int main(int argc, char *argv[]) {

   EXEC SQL BEGIN DECLARE SECTION;
      char  table_name[19];
      char  st[80];
      char  parm_var[19];
      char userid[9];
      char passwd[19];
   EXEC SQL END DECLARE SECTION;

   EXEC SQL CONNECT TO sample USER :userid USING :passwd;

strcpy( st, "SELECT tabname FROM syscat.tables" "WHERE tabschema = ?" );

   EXEC SQL PREPARE s1 FROM :st;

   EXEC SQL DECLARE c1 CURSOR FOR s1;

   EXEC SQL OPEN c1 USING :parm_var;

   do {
       EXEC SQL FETCH c1 INTO :table_name;
       if (SQLCODE != 0) break;
      } while ( 1 );

   EXEC SQL CLOSE c1;

   EXEC SQL COMMIT;

   EXEC SQL CONNECT RESET;
```

```
}
/* end of program ******************:*/
```

5.6.2.6 Static Embedded SQL

When the syntax of embedded SQL statements is fully known at precompile time, the statements are referred to as static SQL. This is in contrast to dynamic SQL statements whose syntax is not known until run time.

The structure of an SQL statement must be completely specified in order for a statement to be considered static. For example, the names for the columns and tables referenced in a statement must be fully known at precompile time. The only information that can be specified at run time are values for any host variables referenced by the statement. However, host variable information, such as data types, must still be precompiled.

When a static SQL statement is prepared, an executable form of the statement is created and stored in the package in the database. The executable form can be constructed either at precompile time or at a later bind time. In either case, preparation occurs before run time. The authorization of the person binding the application is used, and optimization is based upon database statistics and configuration parameters that may not be current when the application runs. The prefix of the package is the user Id of the user that did the precompile.

Because the authorization of the person binding the application is used, the end-user does not require direct privileges to execute the statements in the package. For example, an application could allow a user to update parts of a table without granting update privilege on the entire table. This can be achieved by restricting the static SQL statements to allow updates only to certain columns or a range of values.

The following table shows some of the statements for static embedded SQL. Statements in C or C++ and COBOL are presented.

Table 15. Static Embedded SQL Programming Examples

LANGUAGE	EXAMPLE SOURCE CODE STATIC EMBEDDED SQL
C/C++	EXEC SQL BEGIN DECLARE SECTION; host_var.. pname.. EXEC SQL END DECLARE SECTION; EXEC SQL DECLARE cx CURSOR FOR SELECT NAME FROM STAFF WHERE JOB =:host_var; EXEC SQL OPEN cx ; EXEC SQL FETCH cx INTO :pname; EXEC SQL CLOSE cx; EXEC SQL COMMIT ;
COBOL	EXEC SQL BEGIN DECLARE SECTION END-EXEC. host_var.. pname.. EXEC SQL END DECLARE SECTION END-EXEC EXEC SQL DECLARE cx CURSOR FOR SELECT NAME FROM STAFF WHERE JOB =:host_var END-EXEC. EXEC SQL OPEN cx END-EXEC. EXEC SQL FETCH cx INTO :pname END-EXEC. EXEC SQL CLOSE cx END-EXEC. EXEC SQL COMMIT END-EXEC

Note that if SELECT statements are used in static embedded SQL within DB2, it is important that the user who binds the program has been granted the correct privileges. There are no group rights or public rights accepted.

Figure 147 on page 220 shows the steps when static embedded SQL within an application.

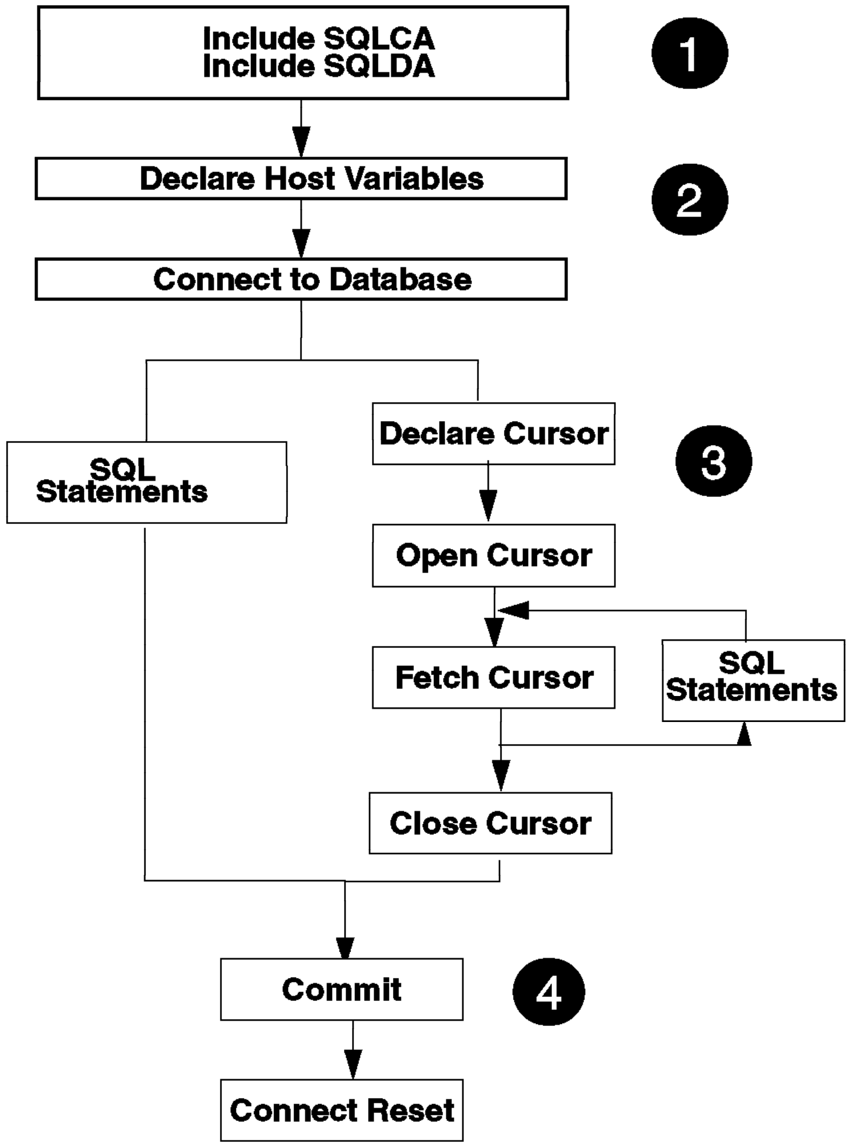


Figure 147. Structure of Static Embedded SQL Program

The structure of a program with embedded static SQL is similar to the embedded dynamic SQL without the prepare step:

1. Additional header files are needed to communicate with the database and exchange datas. SQLCA is used for communication and SQLDA for data exchange with the application.
2. Declare host variables. This section includes the declaration of three host variables:

   a. table_name is used to hold the data returned during the FETCH statement.

   b. statement st is used to hold the dynamic SQL statements in text form.

   c. parm_var supplies a data value to replace the parameter marker in st.

3. The following positions are the necessary statements including CURSOR statements. If only one row is returned, the SQL statement can be executed without a cursor.
4. COMMIT or ROLLBACK ends the transaction process depending on the logic of

the program.

If errors are detected, error code information with the variables are returned in the header file SQLCA. The CHECKERR macro/function is an error checking utility which is external to the program that uses these variables. The location of this error checking utility depends upon the programming language used:

- C check_error is redefined as CHECKERR and is located in the util.c file.
- COBOL CHECKERR is an external program named checkerr.cbl.

5.6.2.7 Static Embedded SQL Program

The following program shows a portion of a typcial static embedded SQL program without a cursor.

```
#include <..>

EXEC SQL INCLUDE SQLCA;

int main(int argc, char *argv[]) {

   EXEC SQL BEGIN DECLARE SECTION;
      char firstname[13];
      char userid[9];
      char passwd[19];
   EXEC SQL END DECLARE SECTION;

   EXEC SQL CONNECT TO sample USER :userid USING :passwd;

   EXEC SQL SELECT FIRSTNAME INTO :firstname
            FROM employee
            WHERE LASTNAME = 'JOHNSON';

   EXEC SQL CONNECT RESET;

   return 0;
}
/* end of program :*********************/
```

5.6.2.8 Executing the Sample Embedded SQL Programs

The environment in this section is the same as in 5.6.1.4, "Executing the CLI Sample Programs" on page 211. There are different types of samples delivered with the DB2 SDK product.The samples for embedded SQL in C are in the subdirectory $(DB2PATH)samplesc. If using a bind file there are three steps to perform; otherwise step 1 and step 2 will be done in one step.

- Precompile the source with the option BINDFILE. Use the following command from an NT DB2 Command Line Window connected to any database:

  `DB2 PREP <filename.SQC> BINDFILE`

  This will create a bind file in the same directory with the same filename and the extension.BND and a modified source file with the extension .C

- Bind the source to the database. To use the BIND statement, you should be connected to the database you will be using in your application.

  `DB2 BIND <filename.BND>`

  This will create a package <USERID.filename> in the database with the user Id of the person who did the prepare. There are some additional options such as BLOCKING with the extension ALL which can be used with the BIND statement. This option may provide some performance benefits when you are issuing only SELECT statements, rather than INSERT, UPDATE or DELETE.

You can also give execution privileges on the package to users or groups of users with the option GRANT group/user at this step. This is only meaningful in the case of static embedded SQL.

Compile and link the modified source file. The sample programs for DB2 CLI are in the subdirectory $(DB2PATH)samplescli. To use these sample programs, you should view the contents of the included MAKEFILE to add the following information:

- Change the entry regarding the compiler if using Visual C++ or VisualAge for C++. The entry should be set to COMPILER=IBM.
- Check the entry for datasource, userId and password.
  Use your user Id and password and the database name, for example, the SAMPLE database, as the datasource.

The following is an example of how to create a specific execution file to use with the sample CLI programs. Note that the following was executed from an NT DB2 command line window.

```
C:SQLLIBsamplescli>nmake static

IBM(R) Program Maintenance Utility for Windows(R)
Version 3.50.000 Feb 13 1996
Copyright (C) IBM Corporation 1988-1995
Copyright (C) Microsoft Corp. 1988-1991
All rights reserved.

        icc /c /Ti /W1 /I:C:\SQLLIB\include static.c
IBM(R) VisualAge(TM) for C++ for Windows(R), Version 3.5
- Licensed Materials - Program-Property of IBM
(C) Copyright IBM Corp. 1991, 1996 - All Rights Reserved.

        ilink /MAP /DEBUG /ST:64000 /PM:VIO -out:static.exe static.obj util.obj
    C:\SQLLIB\lib\db2api.lib

IBM(R) Linker for Windows(R), Version 02.00. r2_960215
Copyright (C) IBM Corporation 1988, 1996.
Copyright (C) Microsoft Corp. 1988-1989.
All rights reserved.
```

The following are the compiler options when programming in C:

```
icc /c /Ti /W1 /I:$(DB2PATH)include <filename.c>
```

The following are a description of the compiler options:

```
/c Perform compile only; no link
/Ti Generate debugger information
/W1 produce messages for severe errors and errors:no warnings
```

The linker command for C is:

```
ilink /MAP /DEBUG /ST:64000 /PM:VIO -out:<filename.exe> <filename.obj>
<additional file> $(DB@PATH)\lib\db2api.lib
```

The following explain the link options:

/MAP Generate a MAP file

/DEBUG Include debugging information

/ST:64000 Specify a stack size of at least 64000

/PM:VIO Enable the programm to run in a window or in a full screen

The number of additional files are dependent on the overall structure of the runtime (<filename.exe>). Note that when executing the sample programs, you should be in a NT DB2 Command Line window.

The process is also possible using a batch file with the name BLDVAEMB.BAT and using the parameters for filename, database, user Id, and password.

5.6.2.9 Deciding What Interface to Use

Which interface you select depends upon your application. DB2 CLI is ideally suited for query-based Graphical User Interface (GUI) applications that require portability.

You may want to use dynamic SQL when:

- You need all or part of the SQL statement to be generated during application execution.
- The objects referenced by the SQL statement do not exist at precompile time.
- You want the statement to always use the most optimal access path, based on current database statistics.
- You want to modify the compilation environment of the statement, that is, experiment with the special registers.

You want to use static SQL when:

- You don't know all the users of your program and their privileges against the tables used or you don't want to give them implicit privileges on that table.
- You know all parameters of the query at programming time.

The main differences between static and a dynamic SQL are:

- Static SQL is prepared at precompile time. Dynamic SQL is prepared at run time.
- Static SQL statements are persistent, meaning that the statements last as long as the package exists. Dynamic SQL statements are cached until they are either replaced or the connection is ended. If required, the dynamic SQL statements are recompiled implicitly by DB2 whenever a cached statement becomes invalid.

Note that combinations of the different types of SQL are possible within the same program. This means that you can call a stored procedure, including static embedded SQL, from within a CLI program. DUOW with two-phase commit using DB2 for Windows NT is only possible in a DB2 Common Server environment, not in a DRDA environment.

5.6.2.10 Summary

The following are further check points:

1. When building C or C++ sample programs, you must ensure that the INCLUDE environment variable contains %DB2PATH%\INCLUDE as the first directory.
2. When building COBOL sample programs, set the COBCPY environment variable to point to %DB2PATH%INCLUDEcobol_mf.
3. Ensure the LIB environment variable points to %DB2PATH%\lib as follows:

   ```
   set LIB=%DB2PATH%lib;%LIB%
   ```

4. Ensure that the DB2COMM environment variable is set at the remote database server.
5. Ensure that the security service has started at the server for SERVER authentication and at the client, depending on the level of authentication for CLIENT authentication. To start the security service, use the NET START DB2NTSECSERVER command.
6. Ensure that the instance is started. Whenever possible, use the service utility to start the instance.

Chapter 6. Performance and Event Monitoring in DB2 for NT

There are two monitoring utilities within Windows NT: the Performance Monitor and the Event Viewer. Both of these utilities allow you to monitor DB2 for NT on your workstation. There are also monitoring utilities provided with the DB2 for NT products. The DB2 Performance Monitor consists of the Snapshot Monitor and the Event Monitor.

This chapter focuses on how DB2 objects can be monitored using the Windows NT Performance Monitor and Event Viewer. We briefly discuss the monitoring utilities provided in DB2 for NT.

6.1 NT Performance Monitor

The Windows NT Performance Monitor (Perfmon) is a powerful tool for monitoring a Windows NT system. Using the utility, you monitor counters. Counters are attributes of objects within Windows NT. For example, LogicalDisk is an object. The number of available megabtyes (called Free Megabtyes in the utility) is an example of a counter within the LogicalDisk object.

The Performance Monitor is part of both NT Server and NT Workstation. Some of the features include the ability to:

1. Log minimum, maximum and average data for critical system values
2. Send alerts within a network based on triggered events
3. Provide a visual view of data as it is monitored
4. Export data logged over time to a variety of file types
5. Monitor machines over a network

The Performance Monitor is started from the Windows NT desktop by selecting Performance Monitor under Administrative Tools (Figure 148 on page 226).

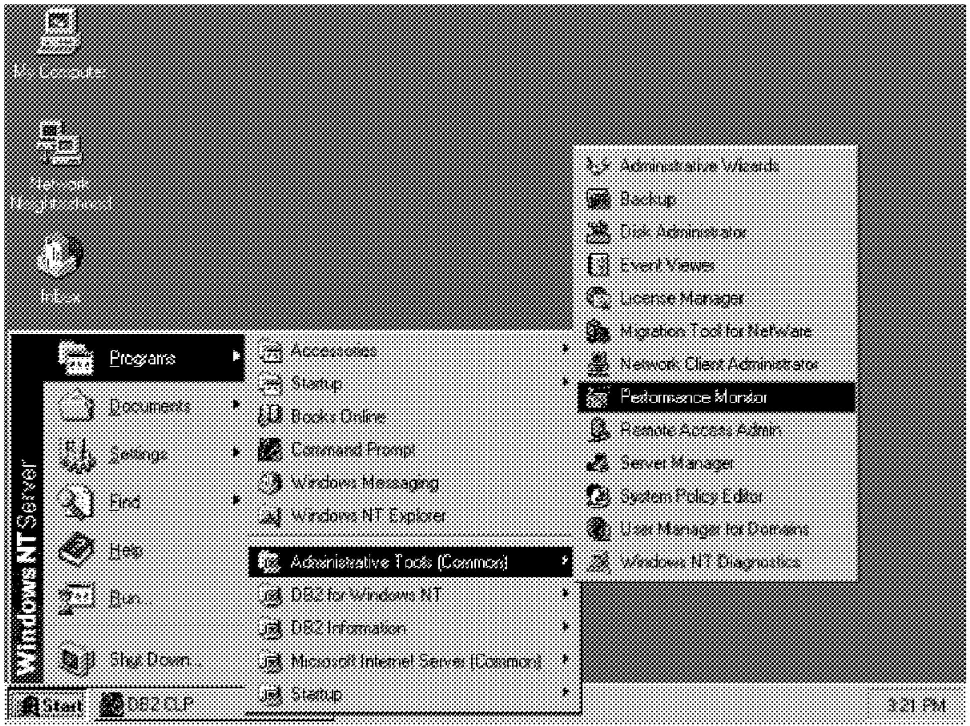


Figure 148. Starting Performance Monitor

6.1.1 Adding DB2 to NT Performance Monitor

After the DB2 product is installed, it must be explicitly added to the Windows NT Performance Monitor or Perfmon utility. There are three objects that are created for DB2 on Windows NT. They are:

- DB2 NT Database Manager
- DB2 NT Databases
- DB2 NT Applications

Databases and applications can be monitored for multiple instances. Here, an instance is defined in terms of a measurable NT Performance Monitor object. (When referring to an instance within DB2, a specific wording of DB2 instance will be used.) For example, a DB2 instance might contain two databases. Each of these databases then becomes an instance of the DB2 NT Databases object in the NT Performance Monitor.

To add the DB2 objects to the object list, execute the following command from a command line window:

```
DB2PERFI -I
```

To remove the objects that are created for DB2 in the NT Performance Monitor, issue the following command from a command line window:

```
DB2PERFI -U
```

If the objects have been successfully installed, a window similar to Figure 149 on page 227 will appear.

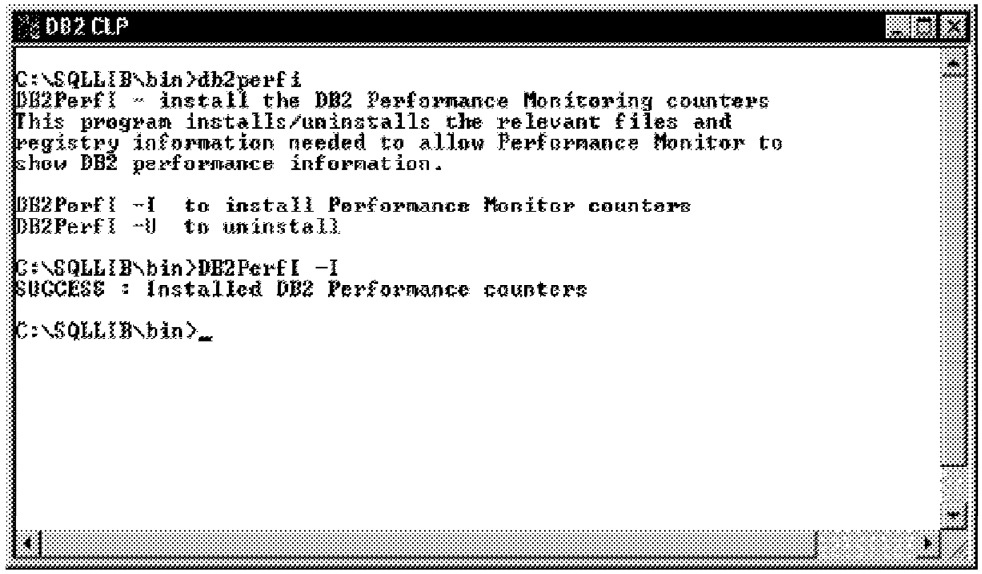


Figure 149. Installing DB2 for NT in Windows NT Performance Monitor

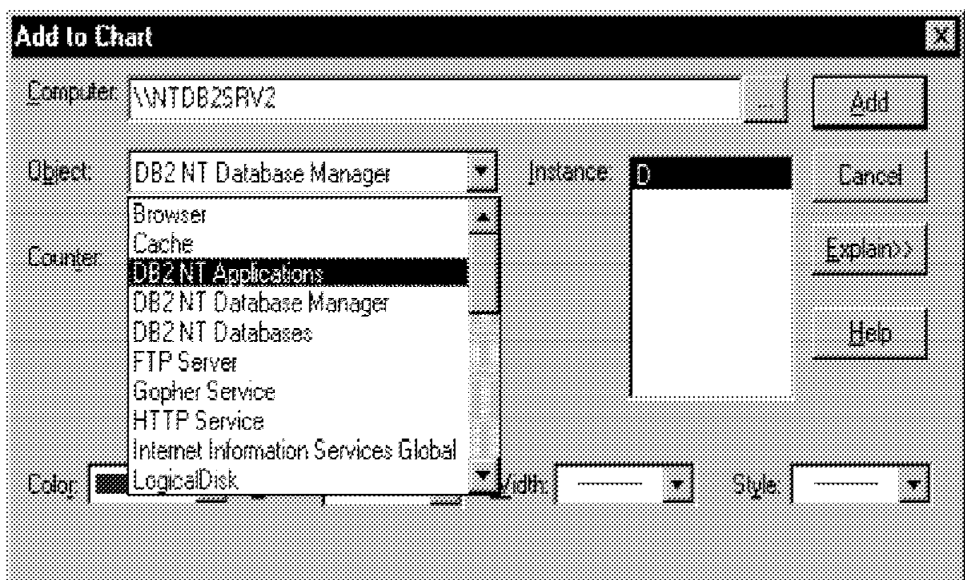


Figure 150. Adding DB2 NT Objects

Figure 150 shows that the three DB2 objects have been sucessfully added. The objects will be displayed in the object list when there is an active instance to monitor. For example the database manager will display after a DB2START command is executed. Databases will be displayed after a successful execution of either a DB2 ACTIVATE DATABASE or a DB2 CONNECT TO DATABASE command. Applications will also be displayed after a successful DB2 CONNECT TO DATABASE command is executed.

6.1.1.1 DB2 NT Database Manager Object

The DB2 NT Database Manager object provides general server information for a single Windows NT Server instance. The DB2 instance being monitored is referred to as the Object instance.

The following is a list of counters that can be monitored for a DB2 NT Database Manager object:

- Commited private memory
- Local active databases
- Local connections
- Local connections executing in the Database Manager
- Maximum number of agents registered
- Maximum number of agents waiting for a token
- Number of agents registered
- Number of agents waiting for a token
- Number of idle agents
- Piped sorts accepted
- Piped sorts requested
- Post threshold sorts
- Remote connections
- Remote connections executing in the Database Manager
- Sort heap currently allocated

Further detail for each counter can be obtained from the *Add to Chart* window. Press **Alt + E** or click on the **Explain>>** button to receive an explanation for each counter.

Performance data can be only be obtained from one DB2 instance at a time. The DB2 instance being monitored by the Windows NT Performance Monitor is determined by the DB2INSTANCE environment variable (PERFORI -I). If you have more than one instance running simultaneously and want to see performance information from several of these DB2 instances, it will be necessary to start a separate copy of Performance Monitor with DB2INSTANCE set to the relevant value for each DB2 instance you wish to monitor.

6.1.1.2 DB2 NT Database Object

The DB2 NT Database object provides information for a particular database. There is an instance shown for each currently active database.

The following is a list of counters that can be monitored for a DB2 NT Database object:

- Active sorts
- Applications currently connected
- Application currently executing in the database manager
- Binds/precompiles attempted
- Buffer pool asynchronous data reads
- Buffer pool asynchronous data writes
- Buffer pool asynchronous index writes
- Buffer pool asynchronous read time
- Buffer pool asynchronous write time
- Buffer pool asynchronous read requests
- Buffer pool data logical reads
- Buffer pool data physical reads
- Buffer pool data writes

- Buffer pool index logical reads
- Buffer pool index physical reads
- Buffer pool index writes
- Buffer pool log space cleaners triggered
- Buffer pool threshold cleaners triggered
- Buffer pool victim page cleaners triggered
- Catalog cache heap full
- Catalog cache inserts
- Catalog cache lookups
- Catalog cache overflows
- Commit statements attempted
- Connects since first database connect
- Current applications waiting on locks
- Data Definition Language (DDL) SQL statements
- Database files closed
- Database status
- Deadlocks detected
- Direct read requests
- Direct read time
- Direct reads from database
- Direct write requests
- Direct write time
- DIrect writes to database
- Dynamic SQL statements attempted
- Exclusive lock escalations
- Failed statement operations
- Internal automatic rebinds
- Internal commits
- Internal rollbacks
- Internal rollbacks due to a deadlock
- Internal rows deleted
- Internal rows inserted
- Internal rows updated
- Lock escalations
- Lock waits
- Locks held
- Log pages read
- Log pages written
- Maximum database heap allocated
- Maximum number of concurrent users
- Maximum secondary log space used
- Maximum total log space used
- Number of lock timeouts
- Package cache inserts
- Package cache lookups
- Rollback statements attempted
- Rows deleted
- Rows inserted
- Rows selected
- Rows updated
- Secondary logs allocated currently
- Select SQL statements executed
- Sort overflows
- Static SQL statements attempted
- Time waited on locks

- Total buffer pool physcial read time
- Total lock lisk memory in use
- Total sort heap allocated
- Total sort time
- Update/Insert/Delete SQL statements executed

6.1.1.3 DB2 NT Applications Object

The DB2 NT Applications object provides information for a particular active DB2 application. There is an instance shown for each currently active application.

The following is a list of the counters that can be monitored under the DB2 NT Applications object. Further detail for the counters can be obtained through the *Add to Chart* window and pressing **Alt + E** or by clicking on the **Explain>>** button.

- Accepted block cursor requests
- Agent ID holding lock
- Application codepage ID
- Application status
- Binds/precompiles attempted
- Buffer pool data logical reads
- Buffer pool data physical reads
- Buffer pool data writes
- Buffer pool index logical reads
- Buffer pool index physical reads
- Buffer pool index writes
- Catalog cache heap full
- Catalog cache inserts
- Catalog cache overflows
- Client communication protocol
- Client operating platform
- Client process ID
- Commit statements attempted
- Data Definition Langugage (DDL) SQL statement
- Database country code
- Deadlock detected
- Direct read requests
- Direct read time
- Direct reads from database
- Direct write requests
- Direct write time
- Direct writes to database
- DYnamic SQL statements attempted
- Exclusive lock escalations
- Failed statement operations
- Internal commits
- Internal rollbacks
- INternal rollbacks due to a deadlock
- Internal rows deleted
- Internal rows inserted
- Internal rows updated
- Lock escalations
- Lock mode
- Lock object name
- Lock object name type waited on

- Lock waits
- Locks held
- Number of lock timeouts
- Open local cursors
- Open local cursors with blocking
- Open remote cursors
- Open remote cursors with blocking
- Package cache inserts
- Package cache lookups
- Rejected block cursor requests
- Rollback statements attempted
- Rows deleted
- Rows inserted
- Rows selected
- Rows updated
- Rows written
- Section number
- Select SQL statements executed
- Sort overflows
- Statement length
- Statement operation
- Statement sorts
- Statement type
- Static SQL statements attempted
- Time waited on locks
- Total buffer pool physical read time
- Total buffer pool physical write time
- Total sort time
- Total sorts
- Total time unit or work waited on locks
- Unit of work completion status
- Unit of work log space used
- Update/Insert/Delete SQL statements executed

6.1.2 DB2 Performance Monitor Registry Keys

When a DB2 NT Performance Monitor object is installed (as described in 6.1.1, “Adding DB2 to NT Performance Monitor” on page 226), the following keys and values are added to the registry.

Subtree: HKEY_LOCAL_MACHINE
Key: Software
Subkey: IBMDB2ComponentsPerformance Monitor

SoftwareIBMDB2ComponentsPerformance Monitor (created with DB2 installation, see Figure 151 on page 232).

Table 16. List of Subkey Values for Performance Monitor Component of DB2

Name	Data Type	Value
Count	REG_DWORD	0x1

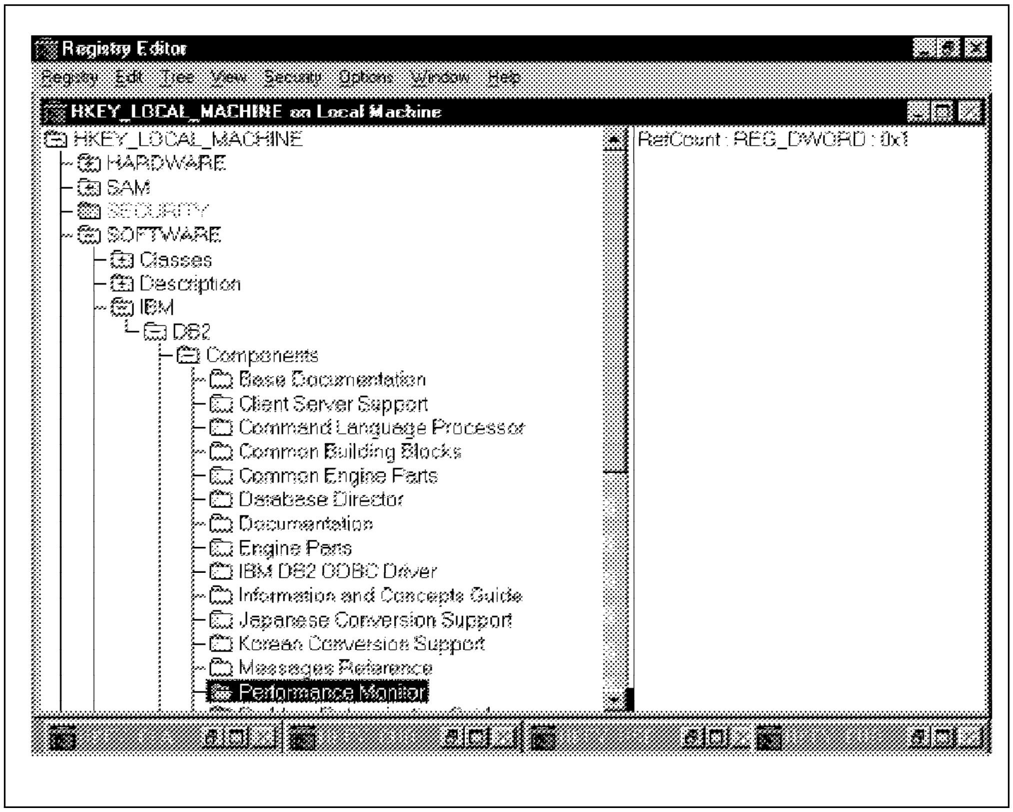


Figure 151. Performance Monitor Component of DB2 Registry Subkeys

SystemCurrentControlSetServicesDB2_NT_PerformancePerformance - see Figure 152 on page 233. The same subkeys, names and values appear under ControlSet001 and ControlSet002.

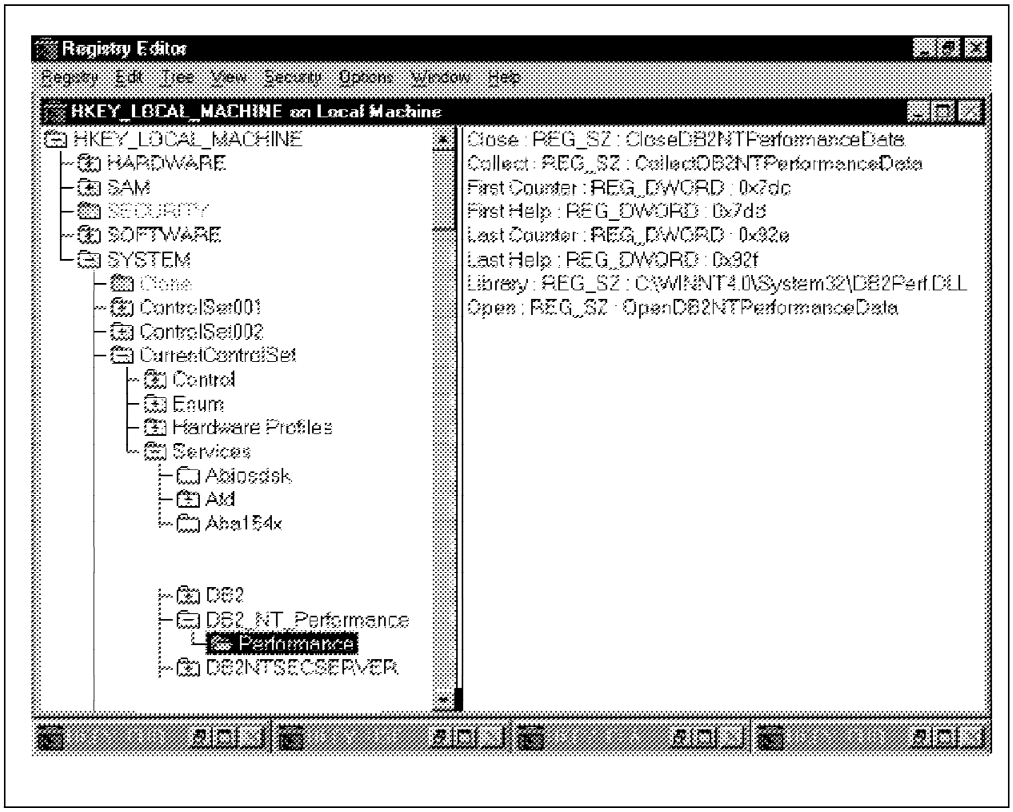


Figure 152. Registry Subkey Values for DB2 in NT Performance Monitor

Table 17. List of Subkey Values for DB2 in NT Performance Monitor

Name	Data Type	Value
Close	REG_SZ	CloseDB2NTPerformanceData
Collect	REG_SZ	CloseDB2NTPerformanceData
First Counter	REG_DWORD	0x7dc
First Help	REG_DWORD	0x7dd
Last Counter	REG_DWORD	0x92e
Last Help	REG_DWORD	0x92f
Library	REG_SZ	C:\WINNT4.0\System32\DB2Perf.DLL
Open	REG_SZ	OpenDB2NTPerformanceData

Table 17 is a list of the subkey values that can be monitored in DB2 using the NT Performance Monitor.

SystemCurrentControlSetEventLogApplicationDB2_NT_Performance The same subkeys, names and values appear under ControlSet001 and ControlSet002.

Table 18. List of Subkey Values for DB2 in NT Performance Monitor

Name	Data Type	Value
EventMessageFile	REG_EXPAND_SZ	C:\WINNT4.0\System32\DB2Perf.DLL
TypesSupported	REG_DWORD	0x7

6.1.3 Using the NT Performance Monitor

The Windows NT Performance Monitor has a number of modes in which data can be viewed. The mode options are the following:

- Chart
- Alert
- Log
- Report

Chart mode is used to graphically view performance data. It is also the mode required to export data to a delimited file. Alert mode is used to generate alerts when counter values go over or under threshold values. Log mode is for recording object data to a log for some future use. Report mode is for producing textual, formatted reports on performance data.

The range of items that can be monitored using Perfmon are grouped under their functional components, called objects. Examples of objects are Processor, Memory, Database and Database Manager. Within an object, additional items can be monitored. These items are called counters. For example, within the Database object, you will find counters such as rows selected, number of lock timeouts and static SQL statements attempted.

Some counters can have multiple instances. For example, a machine may have two disks, or there may be two databases within a DB2 instance. Section 6.1.1, "Adding DB2 to NT Performance Monitor" on page 226, discusses adding DB2 objects. Counters for those objects are also discussed. These counters are monitored in the same way as other Windows NT counters. This document only discusses monitoring the DB2 Objects.

6.1.3.1 Chart Mode

Chart mode is the default mode that appears when the NT (Performance Monitor) Perfmon is started. This is the mode that will let you graphically view data from counters you are monitoring. Chart mode will also allow you to export performance data to either a tab- or comma-delimited file. For example, information obtained from chart mode can be exported into a spreadsheet.

Data displayed on a chart can be obtained from either logged data or real-time data. Chart mode has two options for displaying the format of its data. Chart mode can either display output using a a line graph (the default) or a histogram. Figure 153 on page 235 is a line graph obtained from Chart mode.

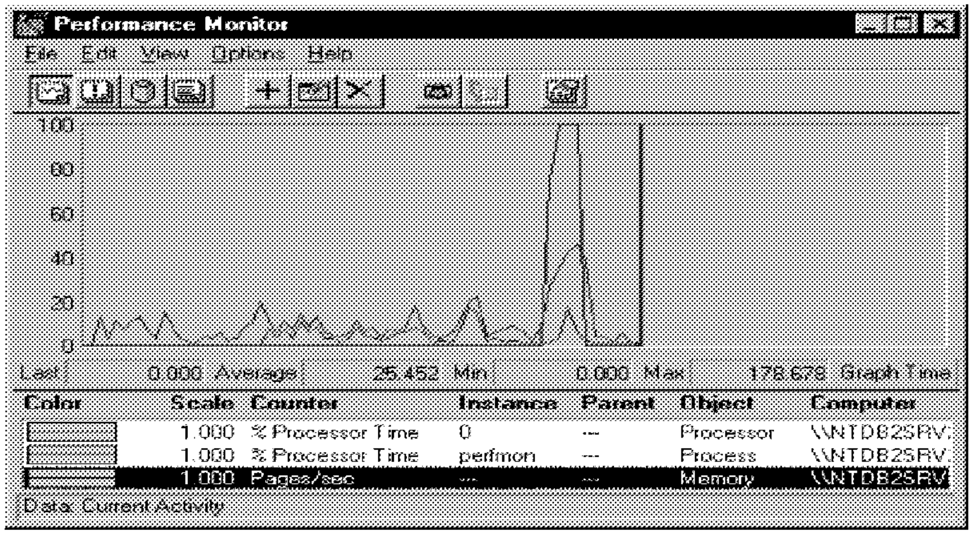


Figure 153. Chart Mode - Line Graph Display

Figure 153 is an example of Chart mode displaying data in a line graph. Notice here that Processor, Process and Memory are some of the objects being monitored.

To add or change a counter to be monitored within an object, you need to open the Add to Chart window. Click on **Edit** from the menu bar (Figure 153) and then select **Edit | Add to Chart**. Alternatively, you can click on the **+** (plus) sign icon that appears in the menu toolbar. The following window should appear:

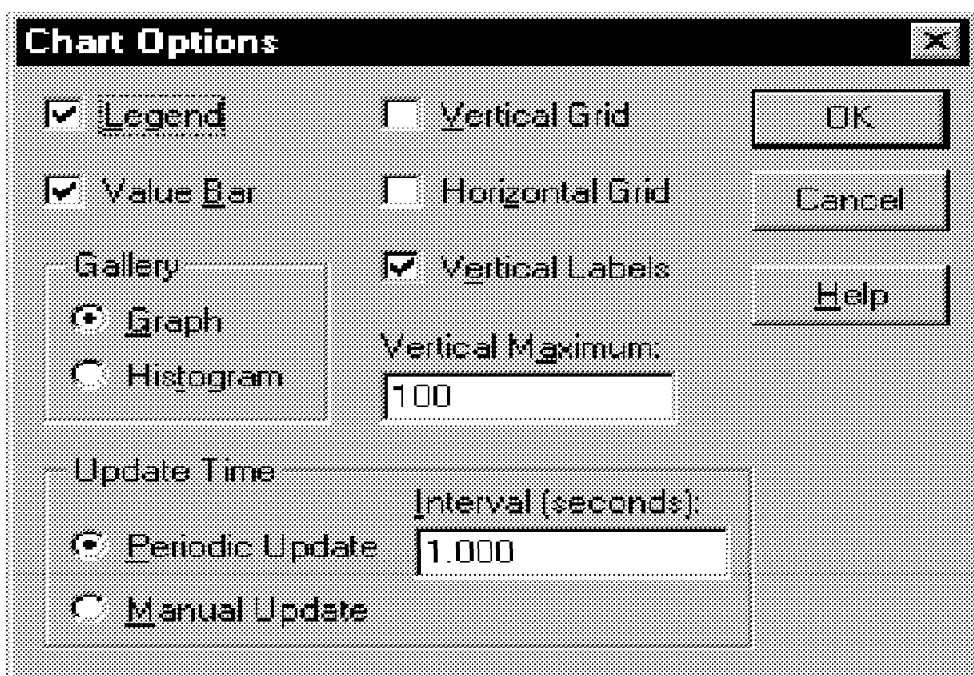


Figure 154. Chart Options Window

Figure 154 is an example of the Chart Options window. The Chart Options window is also where you set how often data is updated. You do have some control over the scale and appearance of the chart. To change the data source, select **Data From...** in the **Options** menu (Figure 153).

To change between line graph and histogram, select **Options** from the pull-down menu bar shown in Figure 153 on page 235. The same counters are displayed using a historgram as shown in the following window:

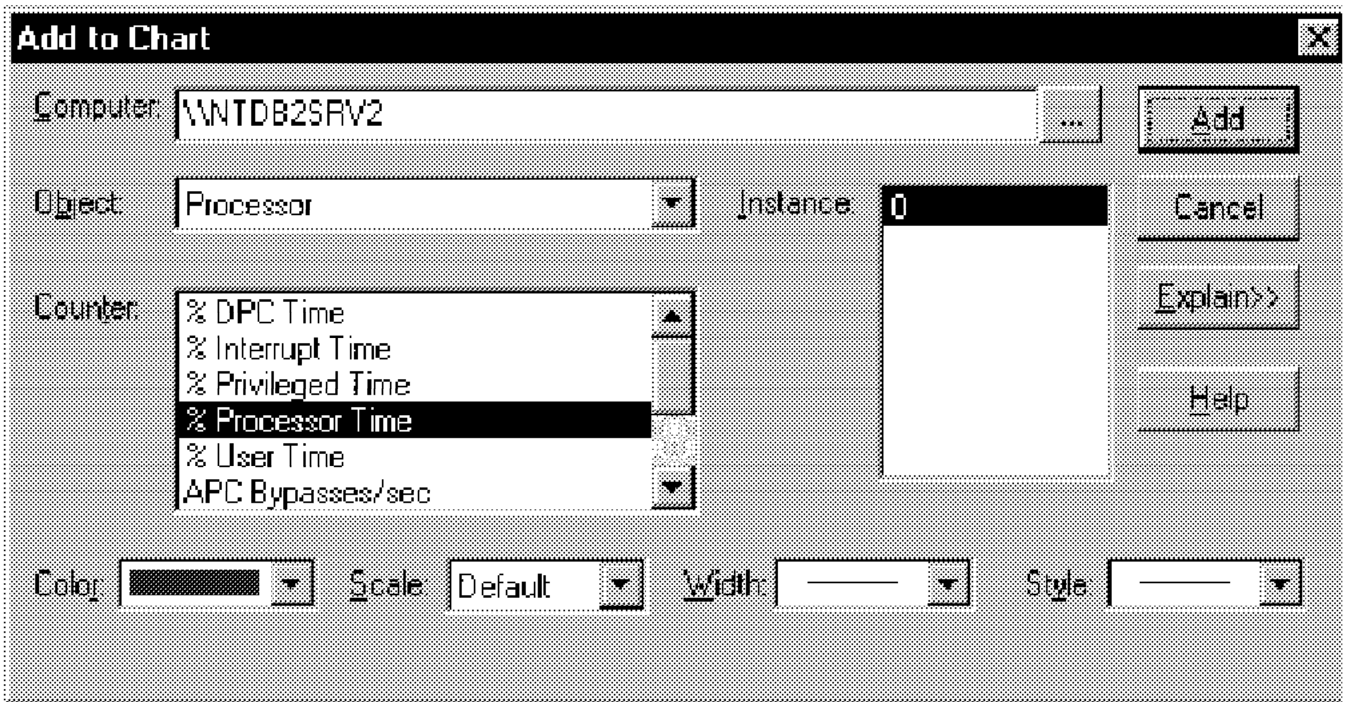


Figure 155. Add to Chart Window - Selecting Counters and Objects

As shown in Figure 155, objects can be selected by clicking on the drop-down menu. As different objects are selected, the counter will change. Select a counter and the instance if there is more than one. Click on **Add**.

Any number of counters can be monitored. However, the colors will start repeating after 16 counters. (There are then five line widths and four line styles to create variations.) Click on **Done** when you have completed your selection for counters to monitor.

Also notice that there is a Computer field in the Add to Chart window. Counters from other Windows NT machines can be monitored over a network. This is discussed further in Section 6.1.3.7, “Remote Access to DB2 Performance Monitor” on page 246.

To change the way a mode is displayed, select from the **File** menu or click on the appropriate toolbar icon as seen in Figure 154 on page 235. The Chart Options window will appear:

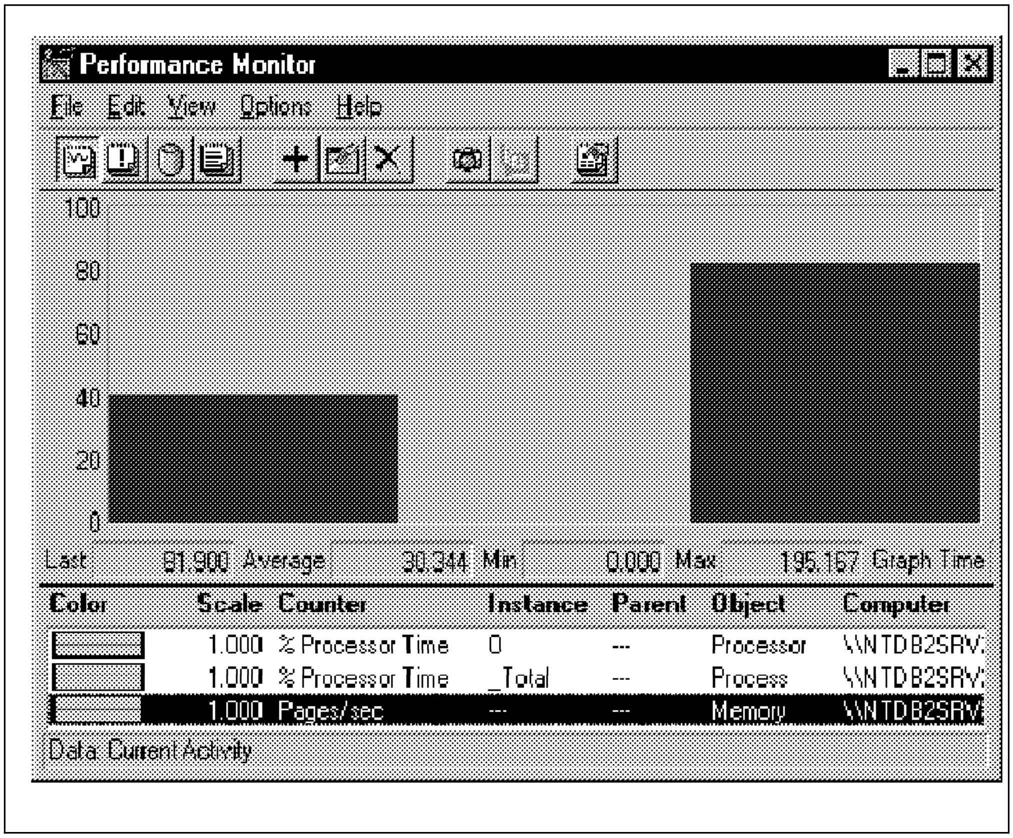


Figure 156. Chart Mode - Histogram Display

6.1.3.2 Alert Mode

The Windows NT Performance Monitor can generate alerts based on criteria you set for counters. A counter or even its object does not have to be monitored to generate an alert.

An alert can be an entry written in the system log, a message sent to a machine (for example, SLEEPY), or a user name (such as Grant).

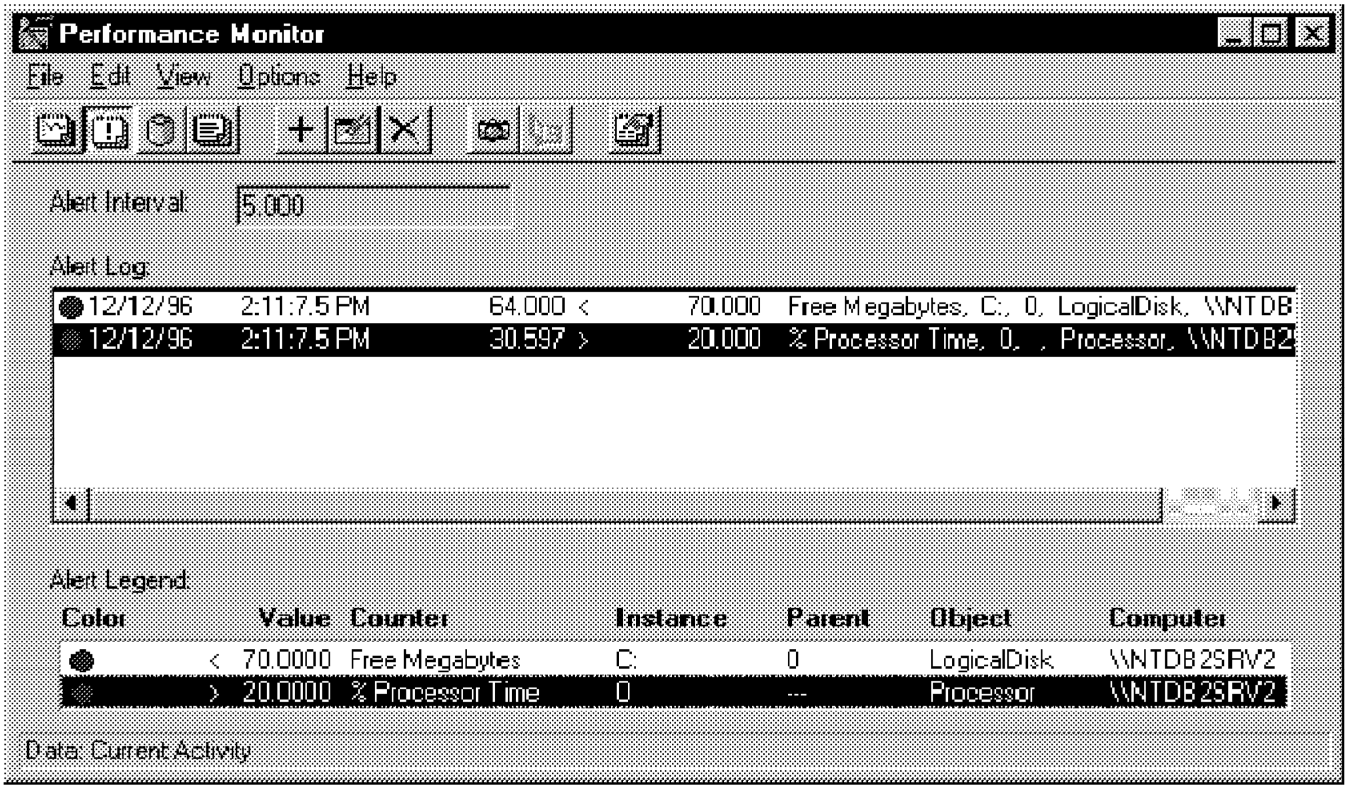


Figure 157. Alert Mode

The dialog box that appears is similar to that for Chart mode, but with two additional fields: Alert If and Run Program on Alert.

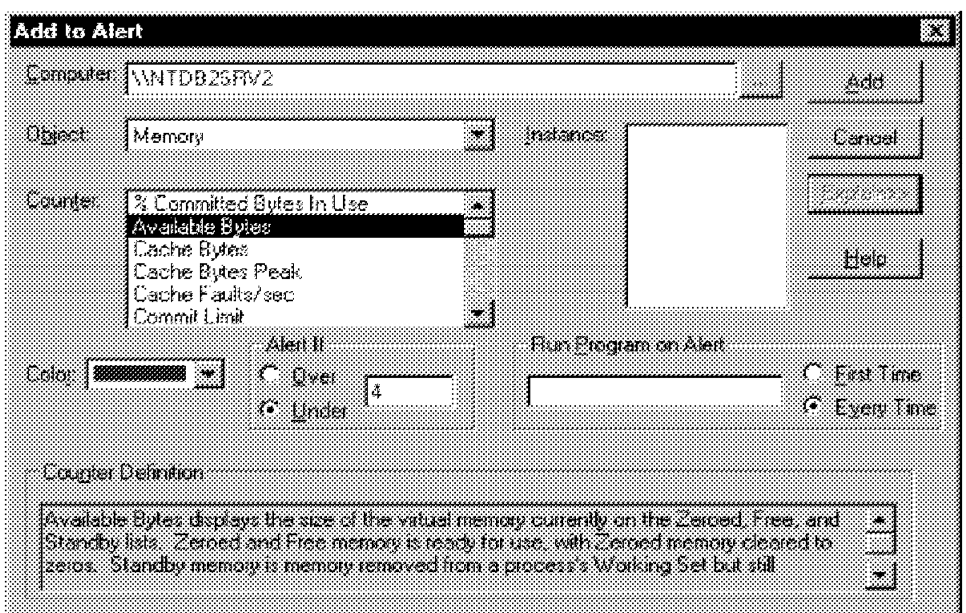


Figure 158. Add to Alert Window

To create an alert, select **Alert** from the View menu or click on the **Alert** icon (Figure 156 on page 237). Next select **Add to Alert** from the Edit menu. After selecting a counter from an object for a computer (see 6.1.3.1, “Chart Mode” on page 234 for help with this), you can then enter a value and click on the radio button that will trigger the alert. Make sure that you notice the counter unit being monitored as being one that for which you want to obtain values (for example bytes vs. megabytes).

If you want the alert to execute a program that might, for example, run a backup and then delete log files, enter the program name, with its path, in the Run Program on Alert field. Select whether you want that program to run just the first time the alert is triggered, or every time the alert is triggered.

When all the alerts you want have been added, click on **Done** and then open the Alert Options window (Figure 159) to set up the action or event that you want when the alert is triggered.

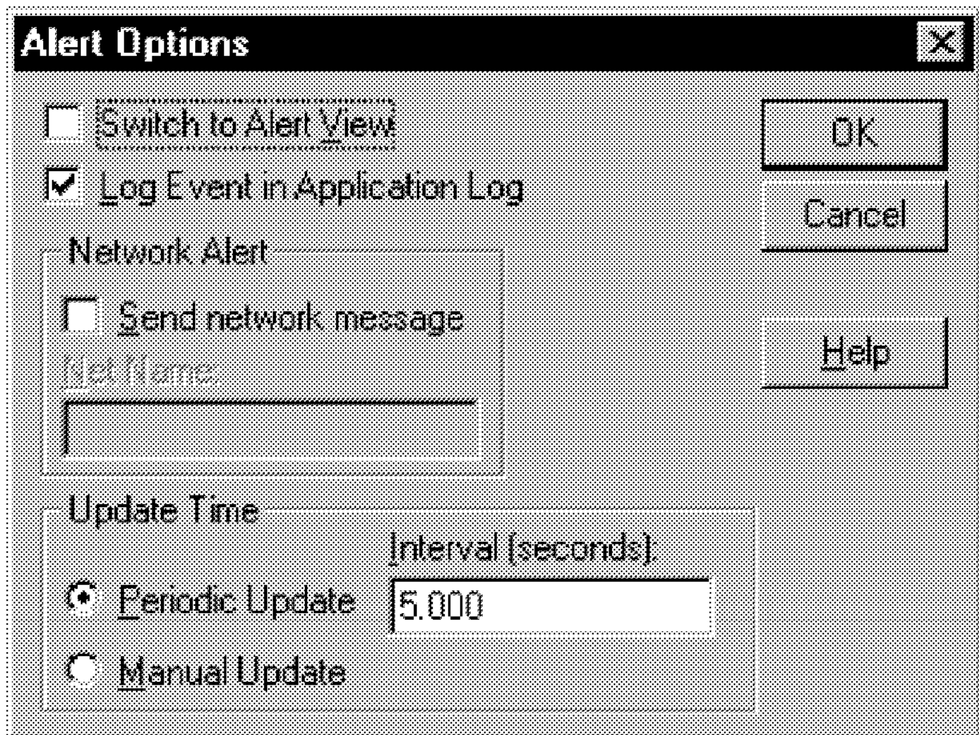


Figure 159. Alert Options Window

From the Alert Options window, you can do the following:

- Switch to Alert View when an alert is triggered. The Performance Monitor cannot be running as a minimized application for this to be effective.
- Log an event in the Application log. Refer to Section 6.2.4, "Application Log" on page 263, for more information.
- Generate a Network Alert. A message can be sent to any computer name or user name that the computer knows about. You can obtain a list of names by entering the following command:

  ```
  net name
  ```

  For help on adding a network name, use the following command:

  ```
  net name /?.
  ```

  Only one recipient can be specified for the network message. Alerts are NetBIOS messages. Therefore, the Messenger service and the NetBIOS interface must be active on both the alerting and receiving machines.

- Change the Update Time. This sets how often the system should check for alerts or whether alert checking should only be done following a manual command. The default updated interval is five seconds.

6.1.3.3 Log Mode

Log mode allows you to record performance data to a log so that it can be reused at a later time. Data can only be recorded at the object level. If you want to record data for just one counter, you must log data for the whole object.

To start a log, select **Log** from the View menu or click on the **Log** icon. Next, select the objects to log by selecting **Add to Log...** from the Edit menu. The Add to Log box will appear, similar to Figure 160.

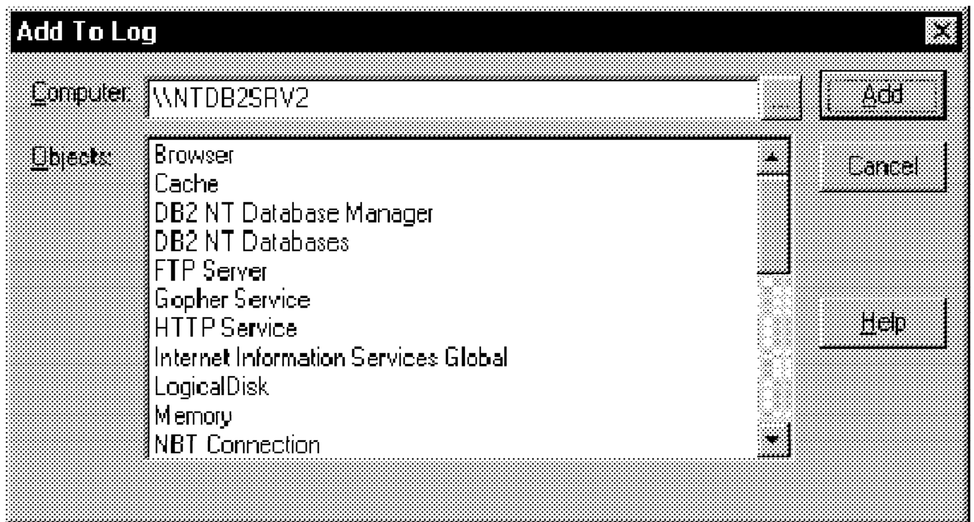


Figure 160. Add to Log Dialog Box

Select the computer that you want to monitor objects on from the Computer field. (See Section 6.1.3.7, "Remote Access to DB2 Performance Monitor" on page 246, for more detail). Then select the Object(s) that you want to log by clicking on them for each computer and clicking on **Add**. When you have selected all the objects you want, click on **Done**.

Logging Considerations

When selecting objects to log, remember the log file grows unbounded. You are logging data for all counters for each object you have selected. Also consider the network traffic you are going to generate if you are remotely monitoring another machine.

To start the logging process, select **Log** from the Options menu. A window similar to that in Figure 161 on page 241 will appear. Choose a file name (*.LOG) and location in which logged data will be written. Also set the logging interval. By default, the logging interval is 15 seconds. Consider the number of objects you are logging and the duration. When you have completed your selections, click on **Start Log**.

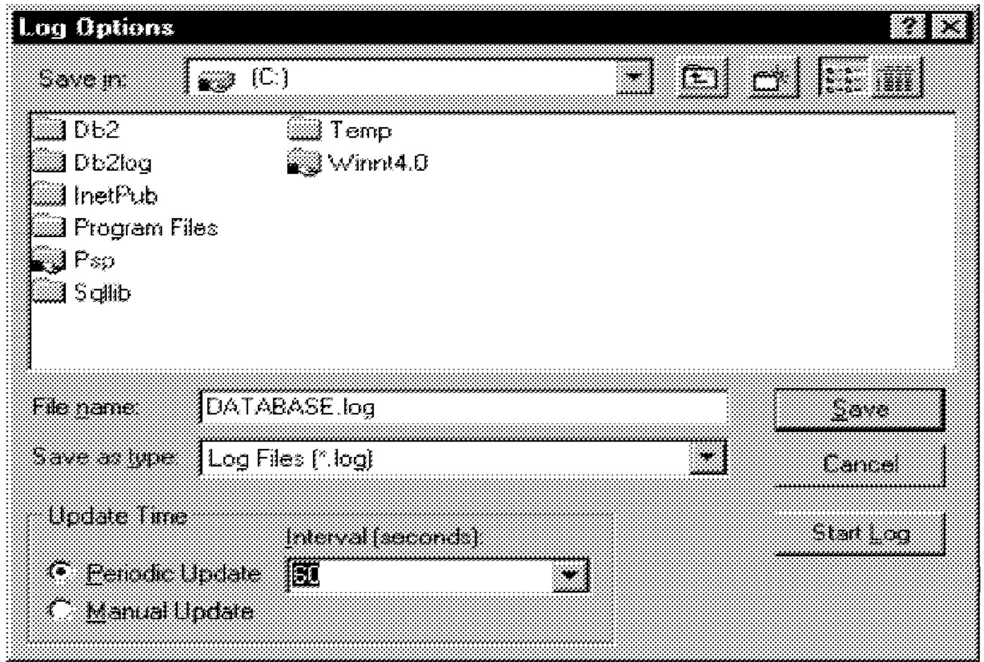


Figure 161. Log Options Dialog Box

The dialog box will disappear and logging will start. The main Perfmon window in Log mode (Figure 162 on page 242) shows the size of the file as it grows.

While data is being logged, commented bookmarks can be added by clicking on the bookmark icon. The book icon is similar to the one shown on the toolbar in Figure 162 on page 242. A dialog box will appear where text can be added. These bookmarks can be useful when data is used in other modes. For example, you can set the beginning or ending of a report period.

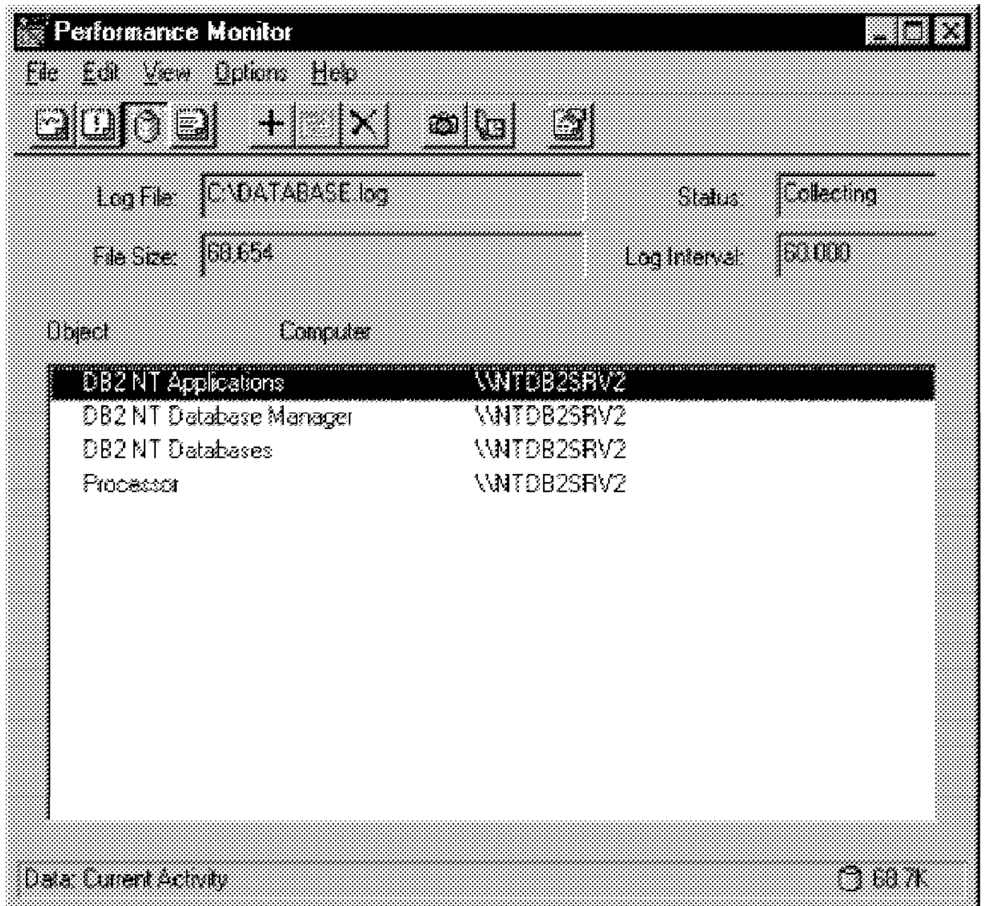


Figure 162. Performance Monitor Objects Logging

To stop logging, open the Log Options Dalog Box (Figure 161 on page 241). What was the Start Log button has now changed to Stop Log. Click on the **Stop Log** button.

The logged data can now be viewed through one of the other Perfmon modes by making it the data source in the relevant Data From option in the Options menu.

6.1.3.4 Report Mode

Reports similar to that in Figure 165 on page 244 can be produced from logged data. Reports are created from the Report mode of Performance Monitor. To enter this mode select **Report** from the View menu, or click on the **Report mode** icon.

Adding counters for reporting is similar to the procedure that has been described for the other Perfmon modes. You need to make sure that the logged data is the report source by selecting **Data** from under the Options menu. Next select **Add to Report** from the Edit menu and start selecting the counters you want. Note that only the objects which had data logged are available (Figure 163 on page 243). This is due to the fact that counters were added to obtain the logged information as selected in Figure 162.

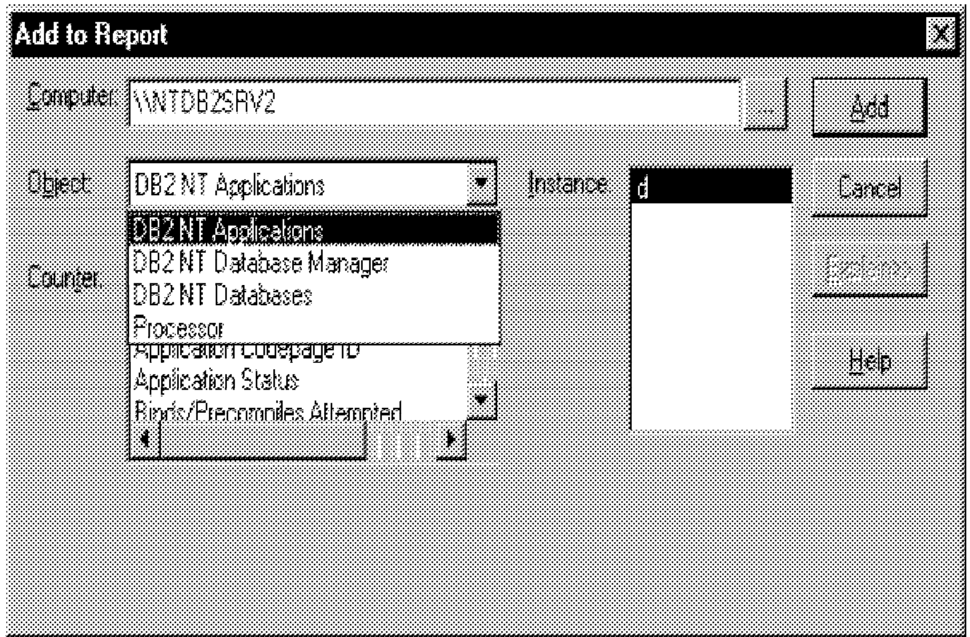


Figure 163. Add to Report Window

The period to which a report applies can be changed in the Time Window in the Edit menu (Figure 162 on page 242). The beginning and end time for a report can be set by either the sliding scale at the top of the window or from any bookmarks that were set in the log (see Figure 164).

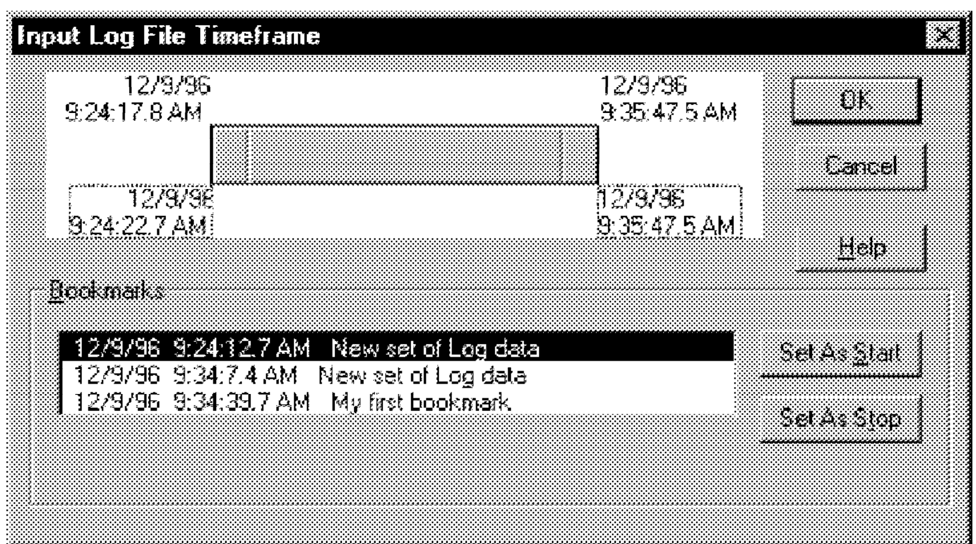


Figure 164. Editing the Time Scale of a Report

If you are going to produce a report using Perfmon, be careful to consider what counters represent cumulative data and what are discrete, or point-in-time data. Question the relevance of counters for these different data types in the same report.

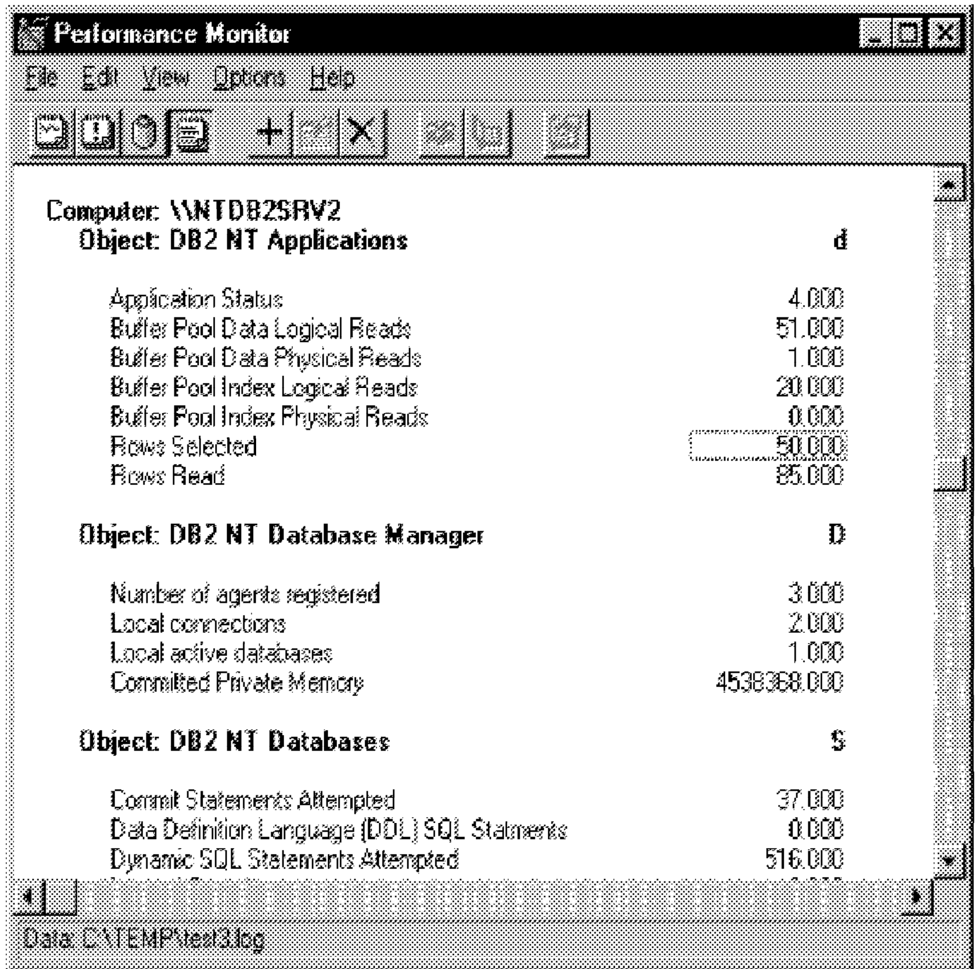


Figure 165. Example of a Report from Logged Data

Figure 165 is an example of the output obtained from the report of logged data.

6.1.3.5 Resetting DB2 Performance Counters

You must use the DB2PERFC.EXE program (located in the sqllibbin directory) to reset database performance values. By default, DB2PERFC.EXE resets performance values for all active databases. You can, however, specify an explicit list of databases to reset. The command can be issued from any command prompt. (Make sure that the sqllibbin directory is in the path for executables.) The syntax is:

```
DB2PERFC [Database [Database...]]
```

The default is all databases. The program resets the values for all programs which are currently accessing database performance information for the relevant DB2 server instance. (This is the instance defined in the environment variable DB2INSTANCE in the session in which DB2PERFC.EXE is executing.)

Invoking DB2PERFC also resets the values seen by anyone accessing DB2 performance information.

6.1.3.6 Saving Data to a File

Performance data can be viewed using the NT Performance Monitor. However, there may be occasions when you want to export this data to another application for statistical analysis or reporting, for example. The exporting of this saved data is a multi-step process. Log the data for the objects you want to export. Next, tell Windows NT to use that log as data. Then change to Chart mode, and finally **Export chart** to a tab- or comma-delimited (*.TSV or *.CSV) file.

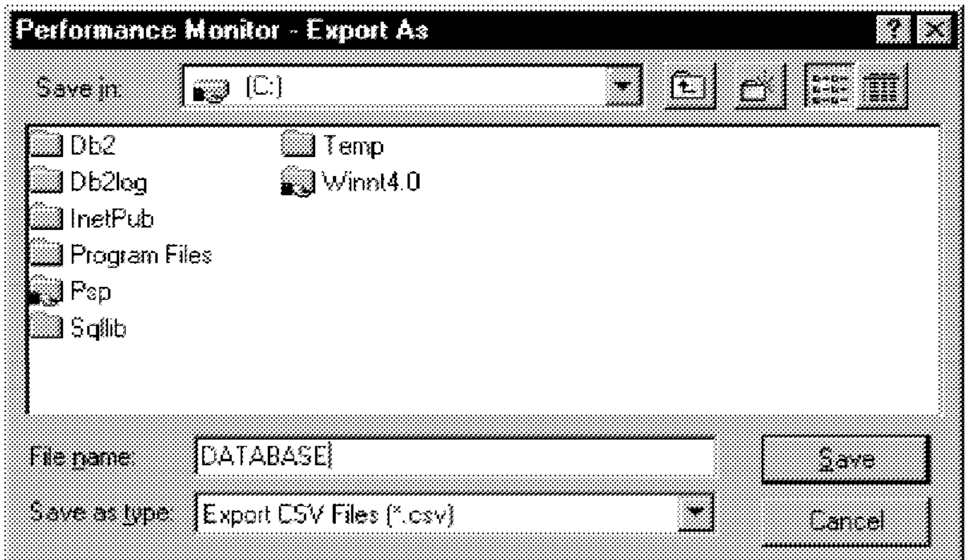


Figure 166. Exporting Logged Data from Chart Mode

The following example will illustrate the process of exporting logged data. Suppose you want to monitor and track applications currently connected to a database over a period of time. Applications currently connected is a counter in the DB2 NT Databases object. The procedure is as follows:

1. Select the object to log.
   a. Go to Log mode by selecting **Log** from the View menu or by clicking on the **Log** icon.
   b. Open the Add to Log window by selecting **Add to Log...** from the Edit menu or by clicking on the **Add to Log** icon.
   c. Select the computer name from the list in the computer field.
   d. Select the DB2 NT Databases object from the list of objects and click on **Add**. Choose the relevant database from the instance field if there is more than one database to choose from.
   e. Click **Add** and then **Done**.
2. Start the log.
   a. Select **Log** from the Options menu.
   b. In the window that appears, type in the name of the file that you want to write the logged data to, for example DATABASE.LOG). Also choose the directory in which you want to file to be located.
   c. Set the logging interval if you do not want the default of 15 seconds.
   d. Click on **Start Log**.

3. Stop the log (when you have enough data) and make the log the Perfmon data source:

    a. Select **Log** from the Options menu and click on **Stop Log**.

    b. Change to Chart mode by selecting **Chart** from the View menu or by clicking on the **Chart** icon.

    c. Select **Data** from from the Options menu.

    d. Choose the Log File radio button and enter the name of the log file (DATABASE.LOG).

    e. Click on **OK** to return to the Chart view of Perfmon.

4. Display data as a chart.

    a. Select **Add to Chart** from the Edit menu. Note that only the counters from the object(s) you have logged can be seen.

    b. Select the counter(s) from the available list by clicking on them and then click on **Add**. In this example, click on **Applications Currently Connected**, then **Add** and **Done**.

5. Export to a CSV file.

    a. Select **Export Chart** from the File menu. A new window will appear.

    b. Choose a file name and location for the file.

    c. In List files of type, choose **Comma Separated Variable** (CSV) from the list and then click on **OK**.

The data is now ready to be imported into another application that supports comma-delimited data, such as Microsoft Excel.

6.1.3.7 Remote Access to DB2 Performance Monitor

One of the features that the NT Performance Monitor provides is the ability to remotely access performance information on another Windows NT machine. In the Add to... dialog box, it is possible to select another computer to monitor by clicking on the Computer field. You are then shown a list of all the available performance objects on that computer.

In order to see Windows NT performance objects from another DB2 for Windows NT machine, you must register an administrator name and password with DB2. The reason for this is that when Perfmon is running locally and a call is made to request DB2 performance information, DB2 validates the user name that Perfmon is running under. The call succeeds only if the user is an administrator.

When a request is made for performance data from a remote computer, the DB2Perf.DLL is loaded in the system WinLogon process. Hence, when a request is made to DB2 for performance data, it appears that the request is coming from the user name of SYSTEM, a built-in special account used by the system, which WinLogon runs as. Since this user name is not an administrator, DB2 refuses this request. You must, therefore, explicitly connect to DB2 as a particular user name and password, rather than allowing it to do the implicit security check (which will fail).

You must use the DB2PERFR.EXE program to register or unregister the administrator user name and password that will be used. The information is held in a key in the security registry that only allows administrators and the

SYSTEM account to access it. Since it is encoded, there should be no concerns over the security of storing an administrator password in the registry.

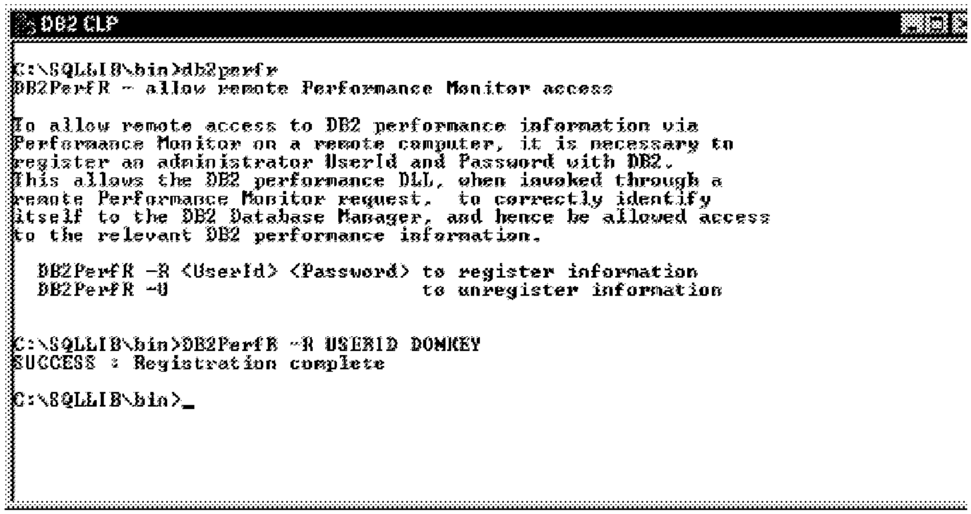


Figure 167. Registering User ID and Password for Remote Performance Monitoring

The following items should be noted when registering a user ID and password for remote monitoring:

1. Once a user name and password has been registered with DB2, even local instances of Performance Monitor will explicitly log on using it. This means that if the user name information registered with DB2 is incorrect, local Performance Monitors will not show DB2 performance information.
2. The user name and password combination must correspond to the user name and password values that are stored in the Windows NT Security (SAM) database. If the user name or password is changed in the Windows NT Security database, the user name and password combination that is used for remote performance monitoring must be reset.

Performance counters can be monitored for any Windows NT machine over a network. In fact, in some cases, it is better to remotely monitor a machine because NT Performance Monitor itself adds some overhead to the machine on which it is running.

The performance disadvantage to remote monitoring is the traffic generated over the network. Several servers can be monitored at the same time if desired.

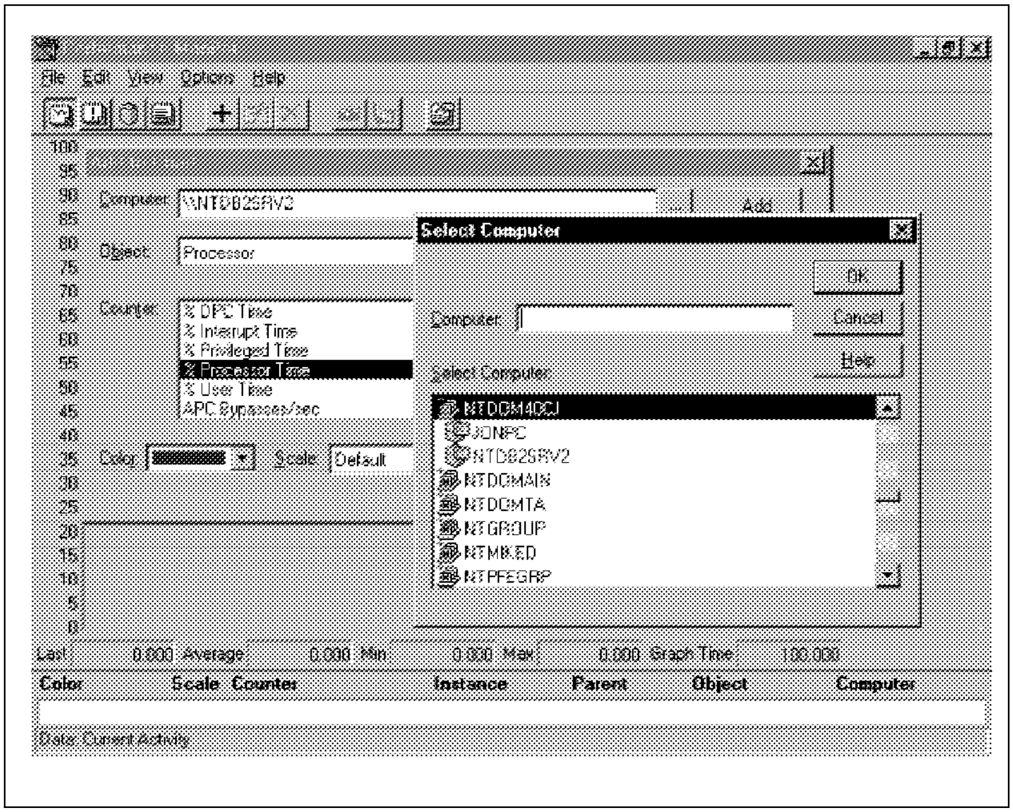


Figure 168. Monitoring Remotely Accessed Machines

6.1.4 Basic Performance Tuning

This section provides some tips and techniques to assist in monitoring DB2 in the Windows NT environment.

6.1.4.1 Tuning DB2 on Windows NT

Refer to *DB2 for Windows NT Installation and Operation Guide*, Appendix D ″DB2 Configuration Parameters,″ for both database and Database Manager configuration parameters.

6.1.4.2 Tuning Windows NT for Use with DB2

This section is not designed as an authoritive guide to tuning Windows NT, but will hopefully provide some guidleines about how you can enhance performance of a NT Server using DB2.

We will assume that the DB2 Server product has been installed on your Windows NT machine. One of the key recommendations is to increase the amount of RAM on the machine, if possible. One of the reasons for this is because of the way Windows NT does disk caching. NT often allocates half of the system's RAM to a disk cache.

The amount of RAM will depend on a number of things, for example the number of users, the volume and complexity of transactions, expected performance by users, and so on.

Another suggestion is to check the fragmentation on the disk in which pagefile.sys is located. If fragmentation is detected, run NTFS.

Multitasking for Windows NT can be configured on a server not to boost foreground applications currently executing. (In most cases, the system administrator will be the only used logged on to an application server). To accomplish the task of not boosting foreground applications, open System Properties in the Control Panel and click on the **Properties** tab (see Figure 169). Slide the pointer to **None** and exit the window.

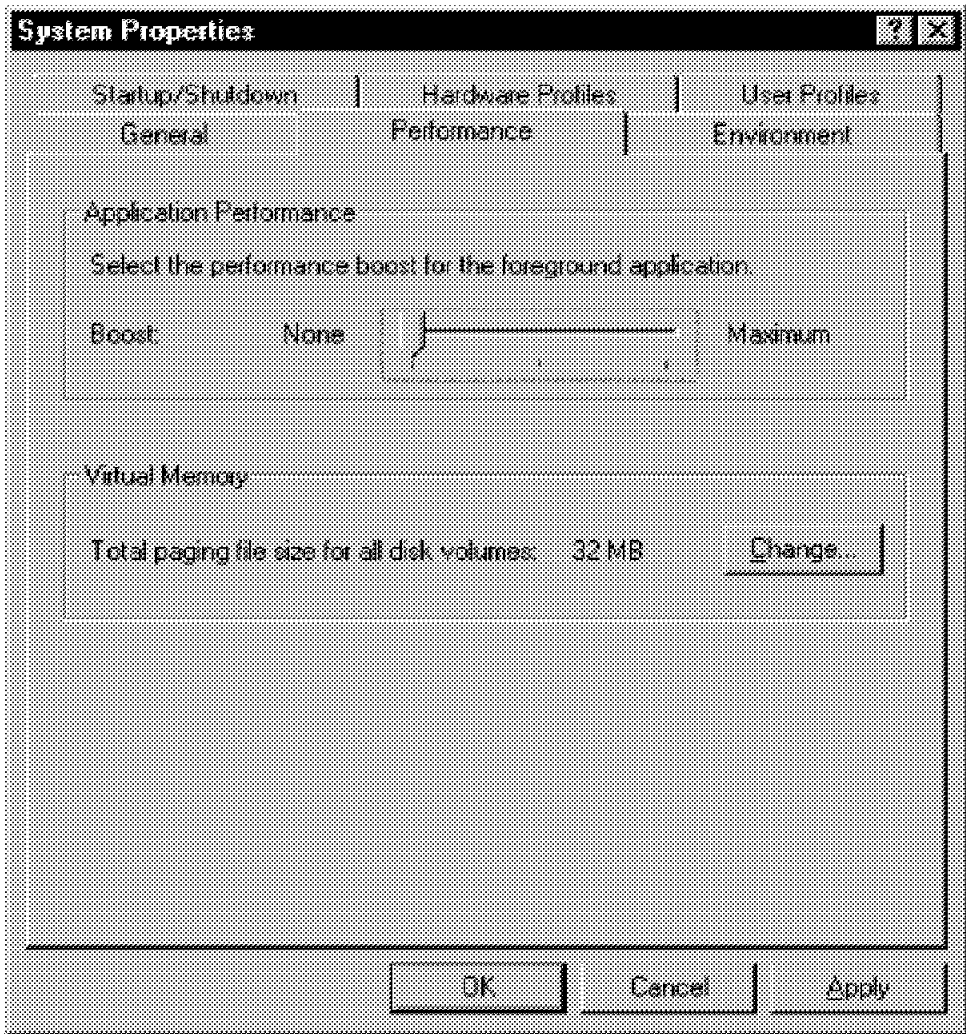


Figure 169. Changing the Foreground Application Boost

Balancing the application load among servers and removing unnecessary applications from your DB2 Server will also help overall performance of the system.

If you can improve the data transfer rate, processing time on the server should be reduced. This rate is controlled by the disk controller or host adaptor. As a guideline, consider 32-bit SCSI host adapters that support asynchronous input/output and create multiple stripe sets with multiple drives.

An additional area to perhaps influence system performance is the network. Locating network bottlenecks is a difficult task, but consider the following:

- Simplify the number of protocols on the network.

- Remove support for any unnecessary protocols from the NT Server.

The server priority can be increased. To change the priority, modify the Registry. In the HKEY_LOCAL_MACHINE, CurrentControlSet, in ServicesLanmanServerParameters, add a value entry ThreadPriority, type DWORD in the Data Type field and set it to 2 in the DWORD editor. By default, the value is 1. In general, the higher the number the greater the priority. An example is given in Figure 170:

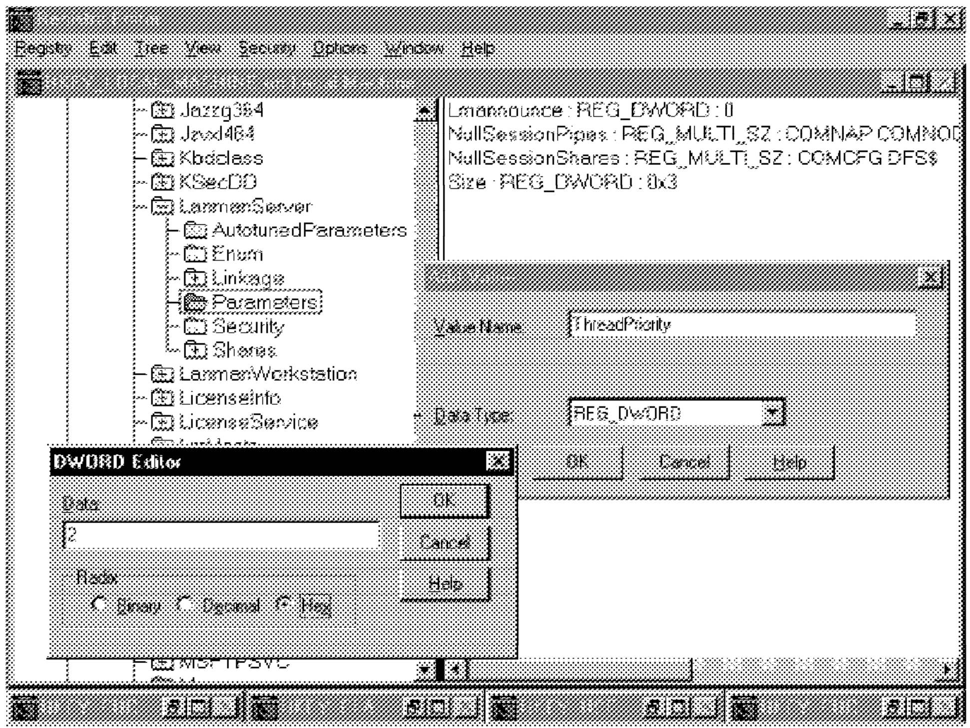


Figure 170. Increasing Server Priority in the Registry

6.1.5 Additional NT Performance Monitoring

There is another tool in Windows NT that can be used for server performance monitoring. The performance tool is located under Task Manager. This tool is shown in Figure 171 on page 251. To start, click the right mouse button on the taskbar at the bottom of the Windows NT desktop screen and select **Task Manager**. Then click on the **Performance** tab on the window that appears.

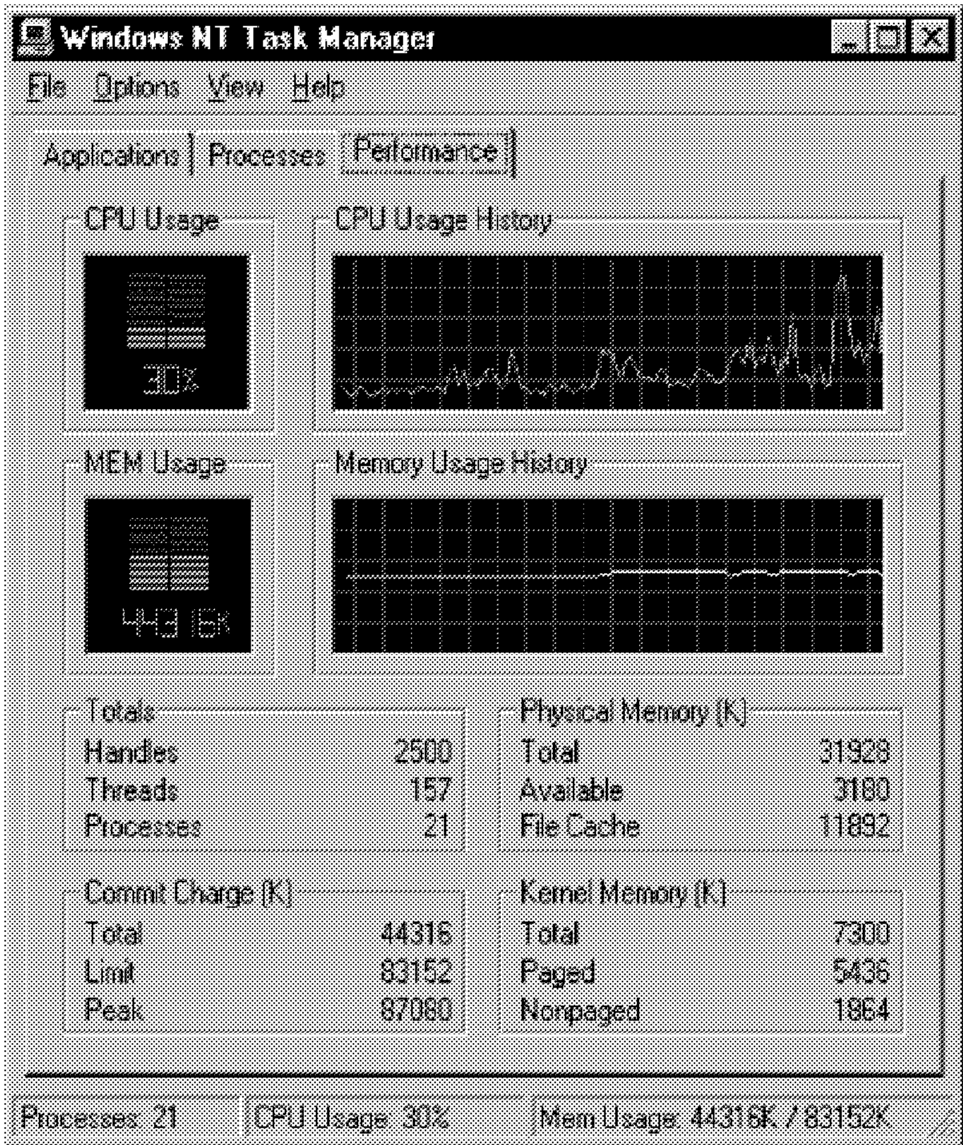


Figure 171. Task Manager Performance

The Task Manager is a real-time monitor with only a minimal history display. The display may be useful for a quick view of overall server performance. A summary of the data that is displayed is as follows:

- Current CPU Usage (%)
- CPU Usage History (Line graph)
- Current MEM Usage (%)
- Memory Usage History (Line Graph)
- Total threads, handles and processes
- Physical Memory (K)
- Commit Charge (K)
- Kernel Memory (K)

6.1.6 DB2 Performance Monitor

DB2 Performance Monitor is also provided with DB2 for Windows NT. It is accessed through Database Director and allows monitoring at both the database and database manager (instance) levels.

6.2 Windows NT Event Log

NT Server defines an event as any significant occurrence in the system or in an application that users should be aware of and perhaps be notified about. When an event occurs that NT Server decides is critical, such as a full disk on a server, a message will be sent to the screens of all workstations on the network. Where the event is less severe or meets the criteria of the audit policy, an entry will be made to one of three logs that Windows NT maintains.

These logs and an overview of their function are described in Table 19. The contents of the logs are viewed through an application called the Event Viewer. The Event Viewer can be found in the Administrative Tools folder (see Figure 148 on page 226).

Table 19. Windows NT Event Logs

Log Name	Log Contents
System	Events logged by the WIndows NT system components, such as the failure of a driver or other system component to load during startup.
Security	Security events (changes in security policy, attempts to log on or access a file or directory, etc.) based on the security and object auditing policies.
Application	Events logged by applications on the system. For example, it will record DB2 related errors.

The Event Viewer (see Figure 172 on page 253) will let you do to following:

- Sort, search and filter events
- Control log settings (for example, maximum log size or how long before events are deleted)
- Clear all entries to a log
- Archive and retrieve logs

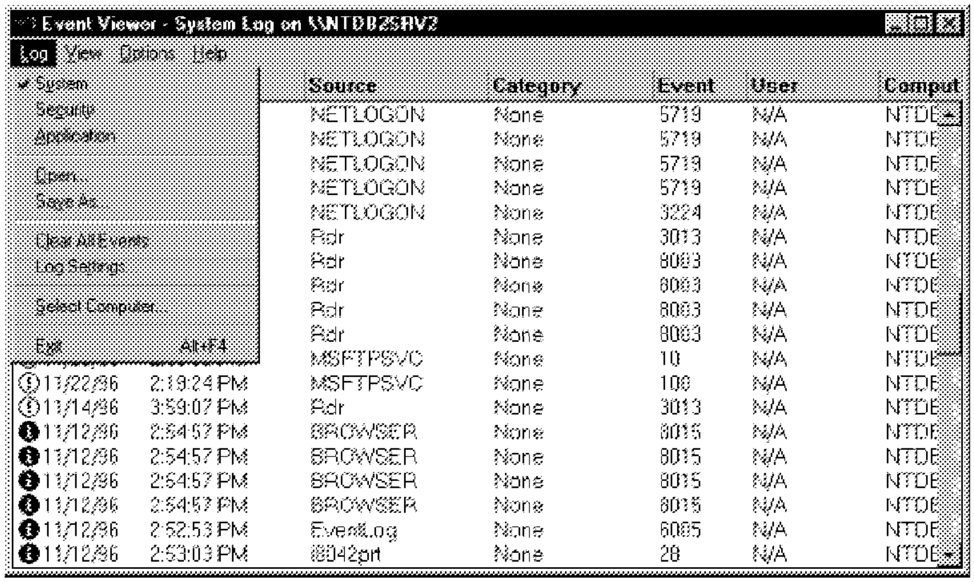


Figure 172. Log Menu of Event Viewer Default Screen

6.2.1 Examining Events in Event Viewer

To view events from any of the logs, open Event Viewer and select the log (System, Security, or Application) from the Log menu. By default, all events for that log are displayed in order of the last one updated or viewed. These options can be changed in the View menu.

If Filter Events is selected, a box similar to the one in Figure 173 on page 254 will appear. Here a range of options can be set, including the time range through which you want to view events, and which type of events to display. Other options are summarized in Table 20 on page 255. Setting a filter on events only changes the events that are displayed. It does not remove or erase events from the log itself.

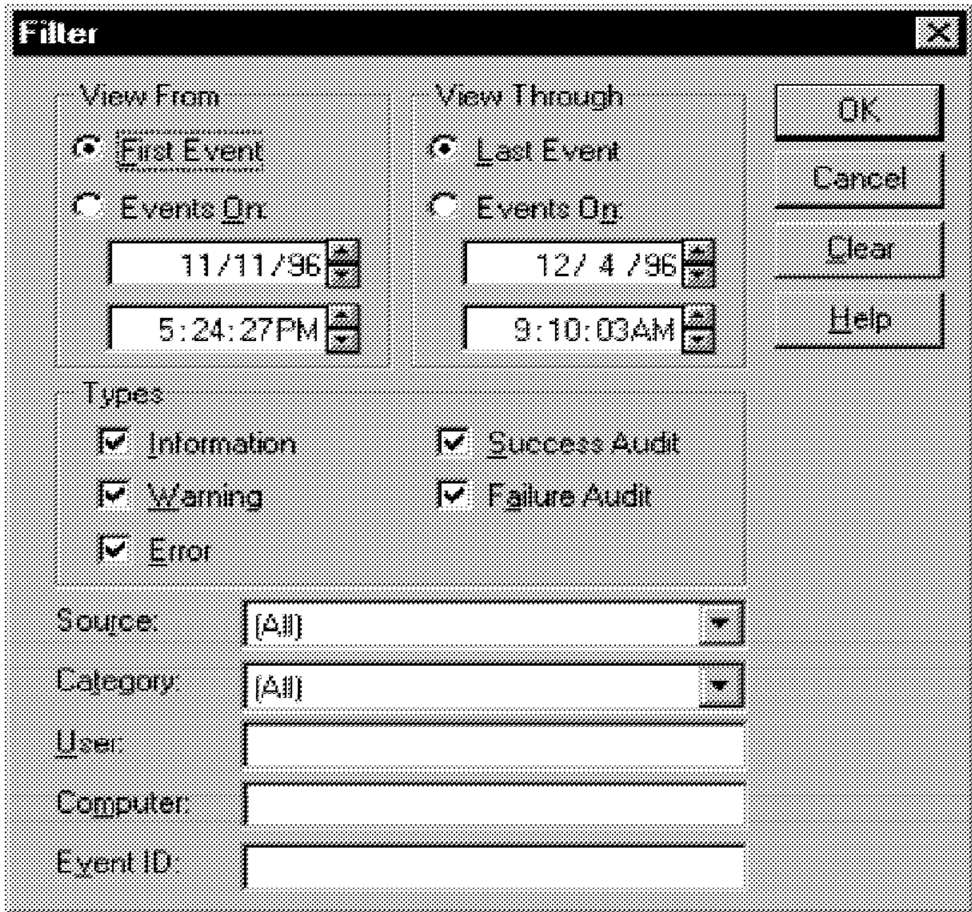


Figure 173. Filter Events Dialog Box

Table 20. Filter Options

Option	Filters
Category	All events of a particular classification. It is a drop-down menu. For example, categories for security events, including Policy Change, Object Access and Privilege Use.
Computer	Events that have occurred for a particular computer. The computer name is not case-sensitive.
Event ID	Events of a particular type in a category, identified by a specific event ID number. For example, event ID 535 (in the security log) identifies that a user's password has expired.
Source	Events logged by a specific source, such as an application, a system component or a device driver. Sources are different for each event log. Examples include UPS (system), Security Account Manager (security) and DrWatson (application).
Types	The types of events to be filtered. Options are informational, warning and error events, and successful and failed audit events. Any number of different types may be selected.
User	Events that occurred while a particular user was logged on and working (although not all events have user IDs associated with them). This field is not case-sensitive.
View From/Through	Allows dates and times to be specified over which events will be displayed. The default is to display from the first through the last event.

Each of the logs display a one line summary of each event with the following data (Figure 172 on page 253):

- Event type. This is indicated by an icon.
- Date and time the event occurred.
- Source of the event. This is the software that logged the event, which can be either an application or a system component.
- Category. Events are categorized by the event source. Not all events are categorized (displays as None category). Application events can be categorized as System Events or Administrative. Security events can be Logon/Logoff, Privilege Use, System Event, Policy Change, Account Management, Object Access, and Detailed Tracking.
- User name. The name of the user who was logged on at the time the event occurred.
- Computer name.

The size of log files can be controlled under Log Settings in the Log menu. For each of the event logs, a maximum size can be set. (The default size is 512 KB). Also, log wrapping can be set to overwrite as needed, not to overwrite files younger than a specified number of days, or not to overwrite events (in which case logs should be cleared manually). Figure 174 on page 256 shows the dialog box where log parameters are set.

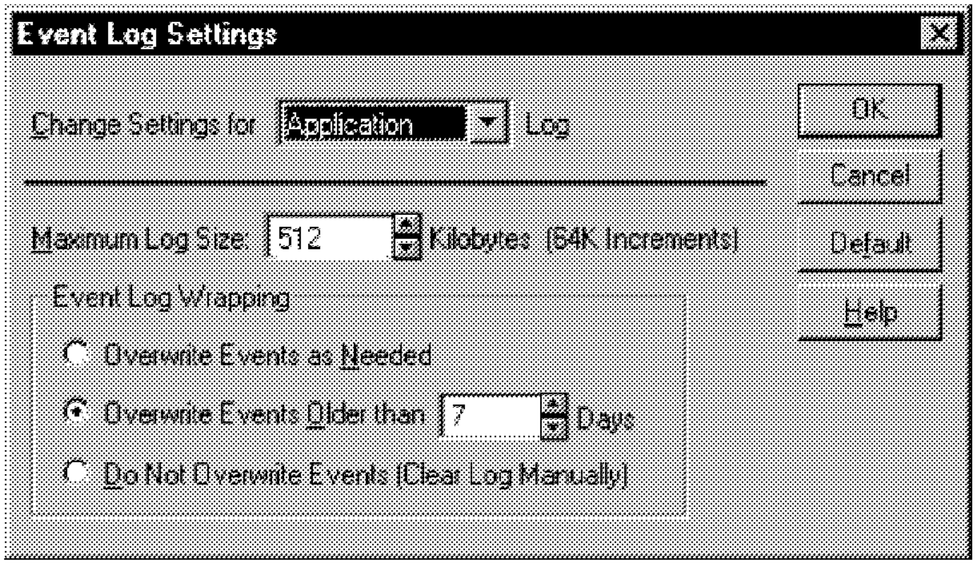


Figure 174. Log Settings Dialog Box

Note that if a log is full and the Event Log wrapping does not allow events to be overwritten, Windows NT will stop writing events to that log. You cannot resume logging simply by increasing the maximum log size. The log must be cleared, the settings changed and logging resumed from the beginning of that log.

Event Logs can be saved and viewed at a later time, if required. It may be a good idea to save logs before clearing them. Logs can be saved as event (*.EVT), text (*.TXT) or comma-delimited text (also *.TXT) files. The latter two can be used to import the data into another application such as a database or spreadsheet. You can use the comma-delimited format to print a log file (any binary data will be discarded).

There are some items to note about archiving and retrieving logs. They are summarized as follows:

- Archiving logs can be useful for problem determination. Often serious problems start out as minor ones that occur more frequently.
- Filtering events does not affect the archiving of logs. All events will be logged.
- Each of the System, Security and Application Logs need to be archived separately. All three cannot be archived at the same time.
- After archiving a log, the contentst of the log are not cleared automatically. The Event Log must be cleared manually.

To clear an Event Log, select **Clear All Events** from the Log menu. Two confirmation message windows will display, the first asking whether you want to save the log before clearing it and the second confirming you want to clear the log.

Logs can also be viewed, saved and cleared for other computers on the network. To see a log for another computer choose **Select Computer** from the Log menu. To minimize network traffic, especially over a slower WAN, you can activate Low Speed Connection, and not all machines will be displayed.

Further detail can be obtained on any specific event by double-clicking on the event or by highlighting it and pressing **Enter**. An Event Log can be updated while you have it open and are examining the events. This is done by selecting **Refresh** from the View menu or by pressing the **F5** key.

Examples of events in each log are given in the following sections.

6.2.2 System Log

The System Log is displayed the first time you start up the Event Viewer by default. The last log viewed will be the default display each time the Event Viewer is started.

Figure 175 gives an example of the System Log. Double-clicking on the highlighted line or selecting **Detail** from the View menu will display the window shown in Figure 176 on page 258. This is the System Log in detail. In this example, an elector for Master Browser has been forced on the network because an NT Server (the PDC in this case) has been started. The Master Browser is a machine that NT machines select to maintain the master browse list for an NT network (This is what is displayed when Network Neighborhood is opened from the desktop).

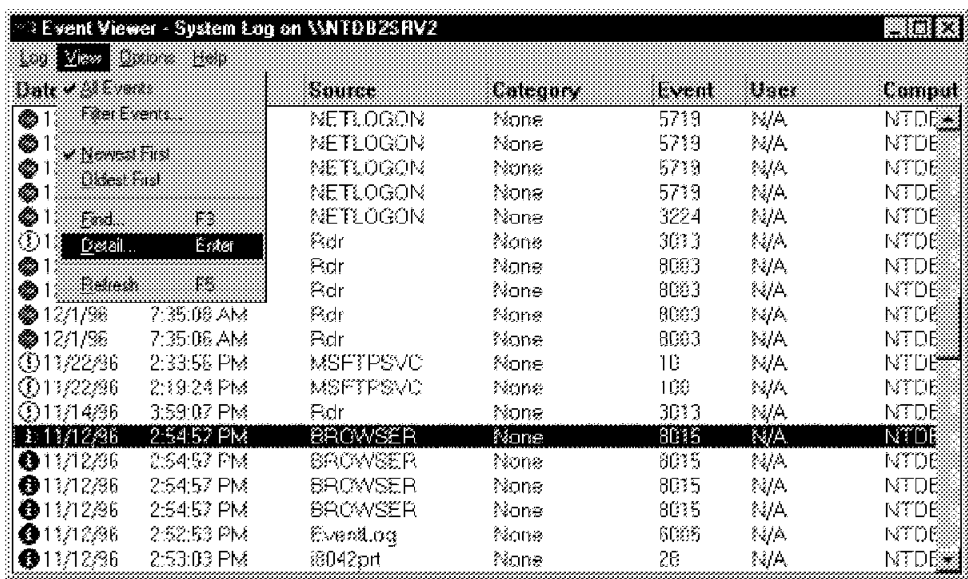


Figure 175. System Log

No specific permissions are required to view the System Log. You will have little influence on what events get logged. There are three types of events which will be displayed:

- Informational. Represented by the blue icon with a white i. This type of event represents the successful operation of a major service.
- Warning. Represented by a yellow icon with black !. This type of event may not necessarily have been significant, but may indicate future problems.
- Error. Represented by a red stop sign icon. This type of event will have resulted in a loss of data or function.

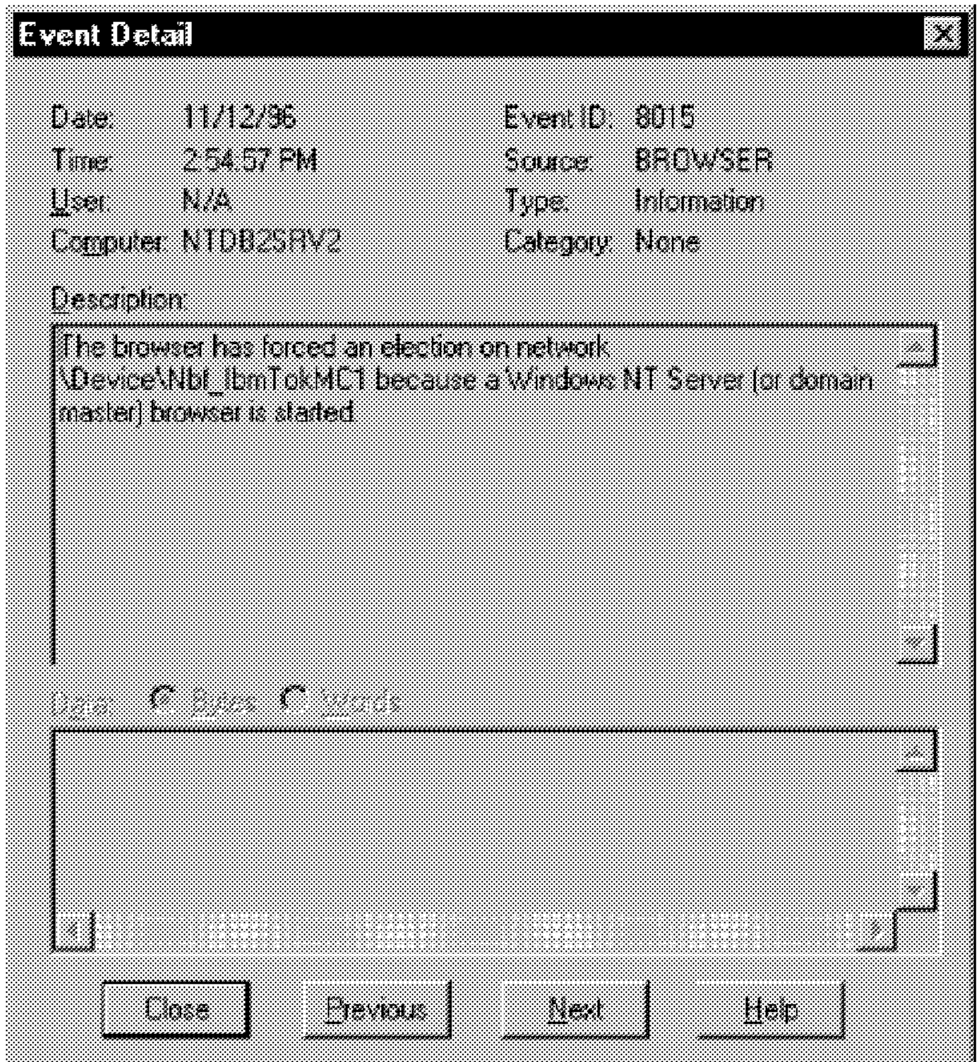


Figure 176. System Log Event Entry

Figure 177 on page 259 is an example of the type of output you might see from the System Log.

6.2.3 Security Log

An example of the Security Log is shown in Figure 177 on page 259. You must be Windows NT administrator to be able to view any entries in the Security Log. Entries with icons that resemble a key represent a successful audit event, whereas those with a padlock icon are failed audit events.

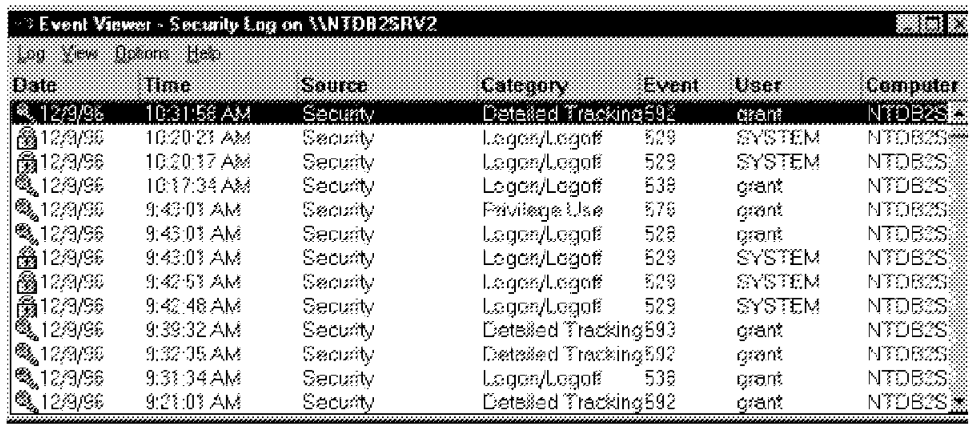


Figure 177. Security Log

Figure 178 on page 260 is an example of an entry in the Security Log. For a detailed understanding of each field, consult a detailed Windows NT reference manual. This example is the result of a user (Grant) successfully executing the `DB2START.EXE` command following the file audit policy set in Figure 180 on page 262.

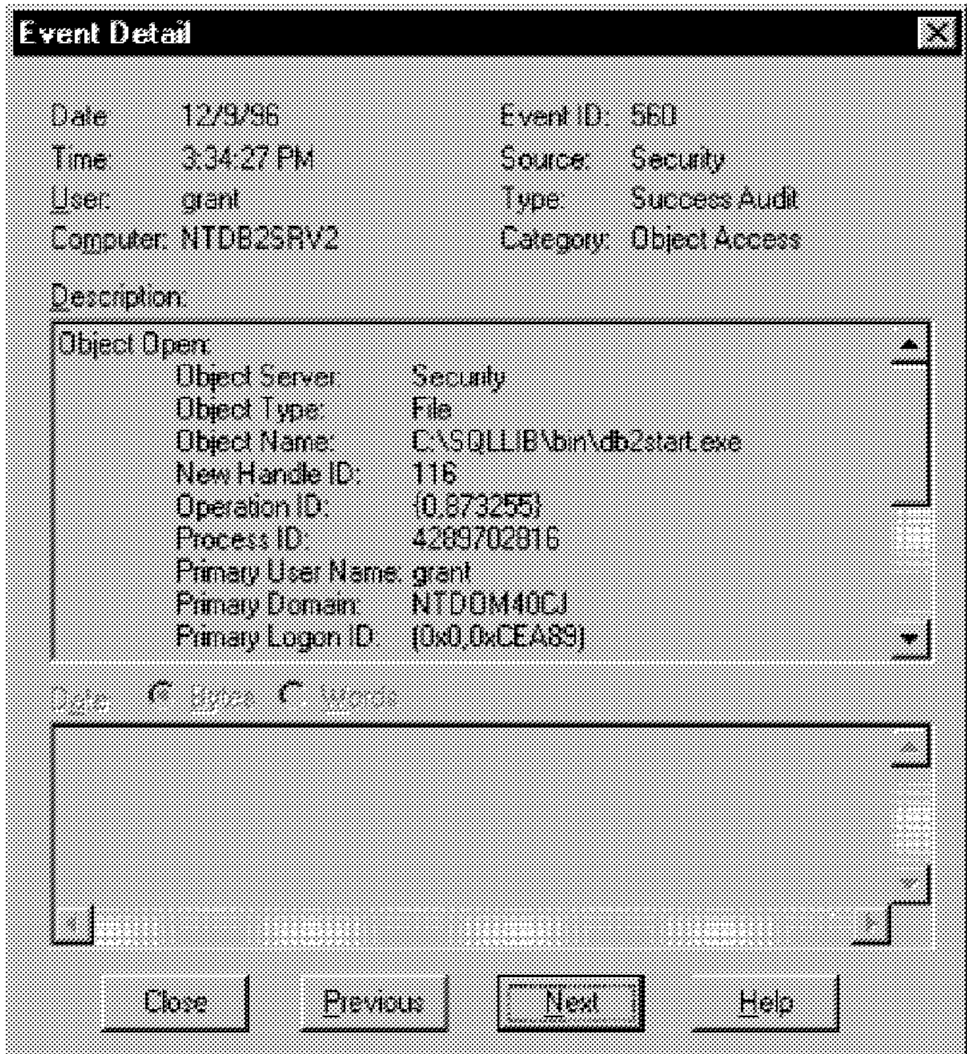


Figure 178. Security Log Entry

Events are written to the Security Log based on the audit policy of the system. In order to record, retrieve and store logs on an NT Server, the administrator must activate and configure event auditing. The following events can all be audited.

1. File and directory access. This is activated within Windows NT Explorer.
2. Printer access. This is activated within Print Manager.
3. Security. This is configured within User Manager for Domains.

File and directory access is activated and from Windows NT Explorer, and printer access from Print Manager. Security for both is configured in a similar way for all three objects. On any given object, click with the right-most mouse button. Select **Properties** from the menu that appears and then select the **Security** tab. A window similar to Figure 179 on page 261 will appear. (Note this represents the options for a file; the printer options are different).

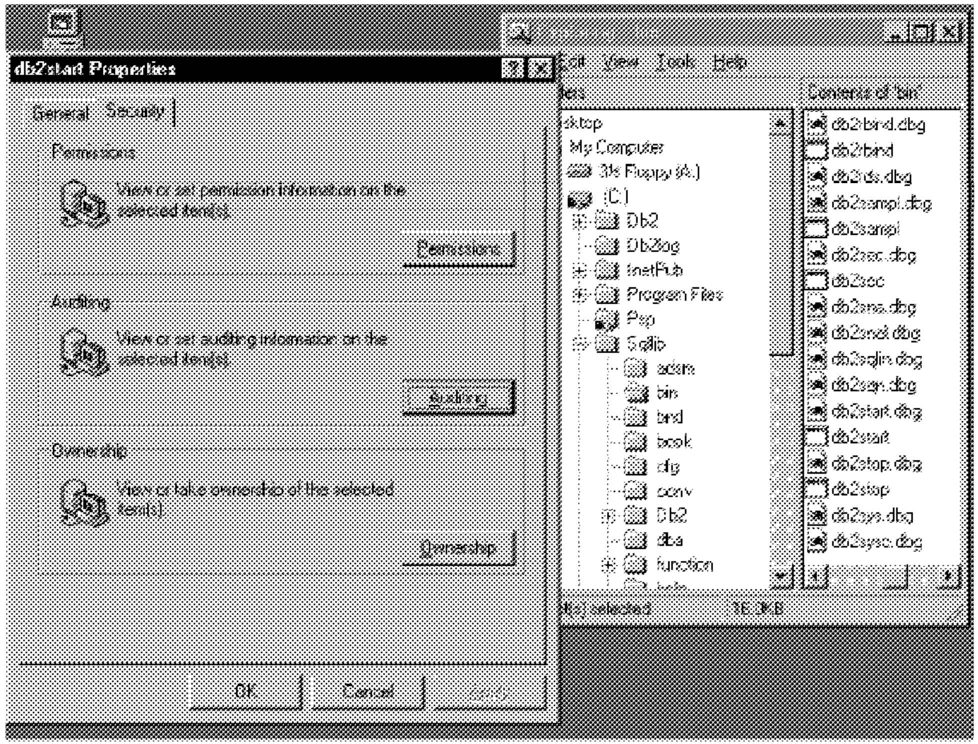


Figure 179. File Security Properties Window

Next, click on the **Audit** button, and a window similar to Figure 180 on page 262 will appear.

Select the groups or users or any trusted domains to which the audit policy should apply by clicking on the **Add** button. Only user accounts local to the machine will be shown, along with global groups from any domains in the List Names From window. To see a list of members of a global group, click on the group and then click on **Members**. Any individual user can then be added to the auditing Name list.

The specific successful or failed events that you want to audit, and consequently display in the Security log, can then be selected. In the example shown in Figure 180 on page 262, information is logged about all users who attempt to execute the DB2START.EXE file, regardless of the permissions on executing the file.

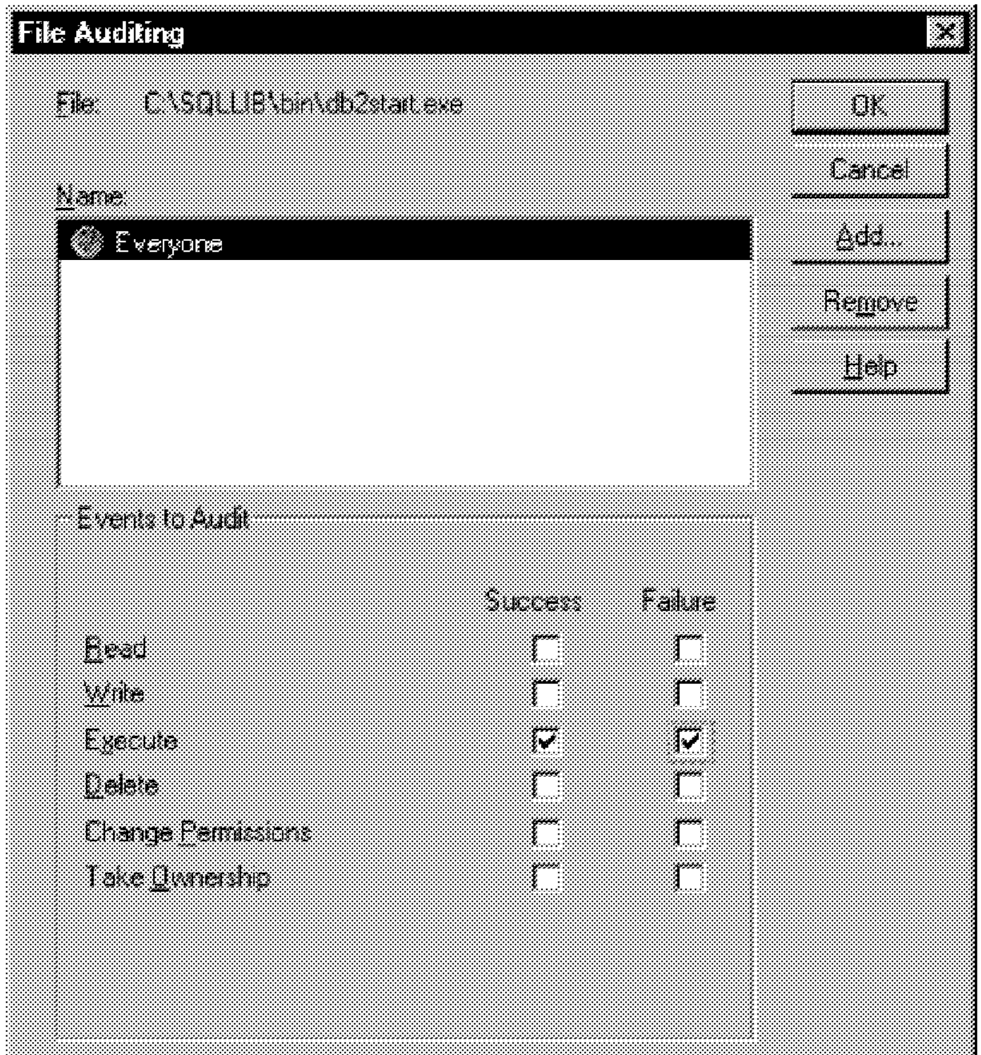


Figure 180. Setting File Audit Policy

Audit policy for Security is configured in User Manager for Domains (Figure 181 on page 263). Select **Audit** under the Policies menu.

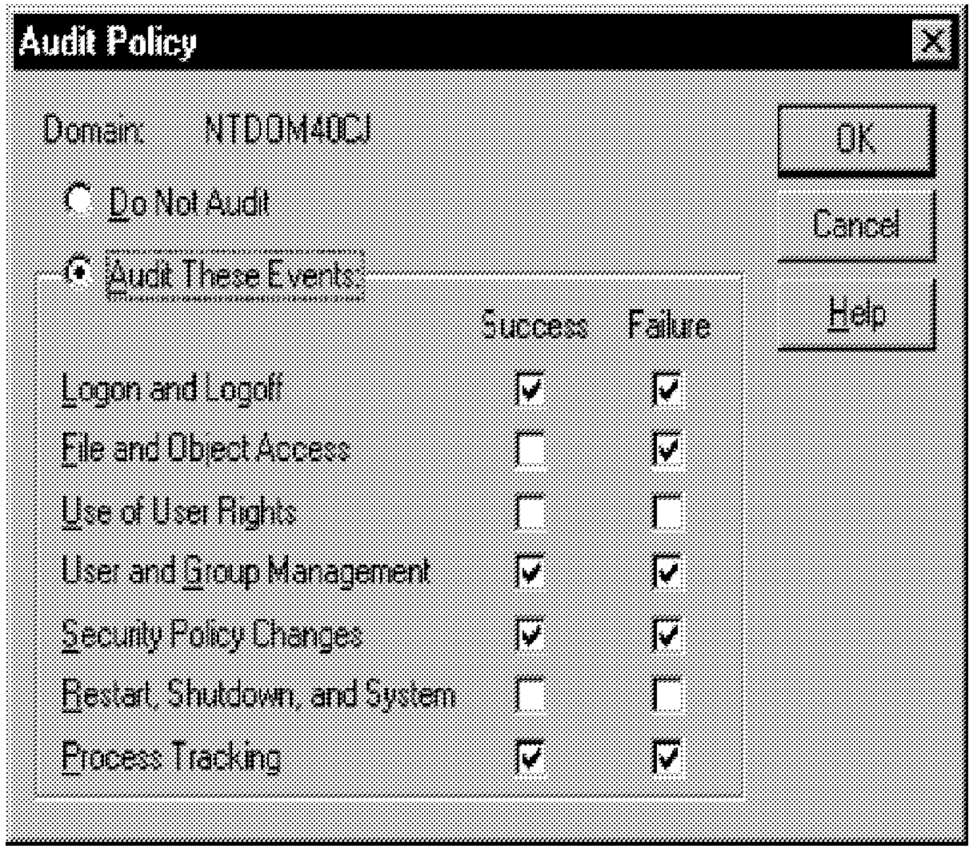


Figure 181. Setting Security Audit Policy

6.2.4 Application Log

All five event types can appear in the Event Log: information, warning, error, successful audit and failure audit.

Only applications that know about the NT Event Log and are coded to report to it will have events logged. DB2 for Windows NT is one such application. Remember that if an event is related to the audit policy of a DB2 file, that event will get logged to the Security Log.

Figure 182 on page 264 shows the Application Log. Here entries can be seen from two applications: DB2 and NT Performance monitor. Although there are only two event sources, there are a number of different events being logged, evidenced by the different event ID values.

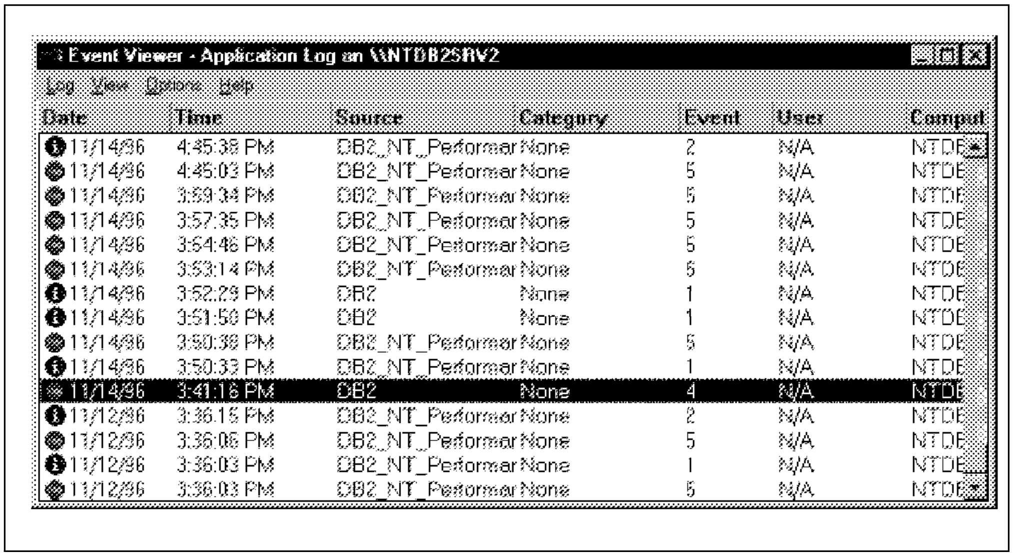


Figure 182. Application Event Log

Figure 183 shows an example of an application log entry.

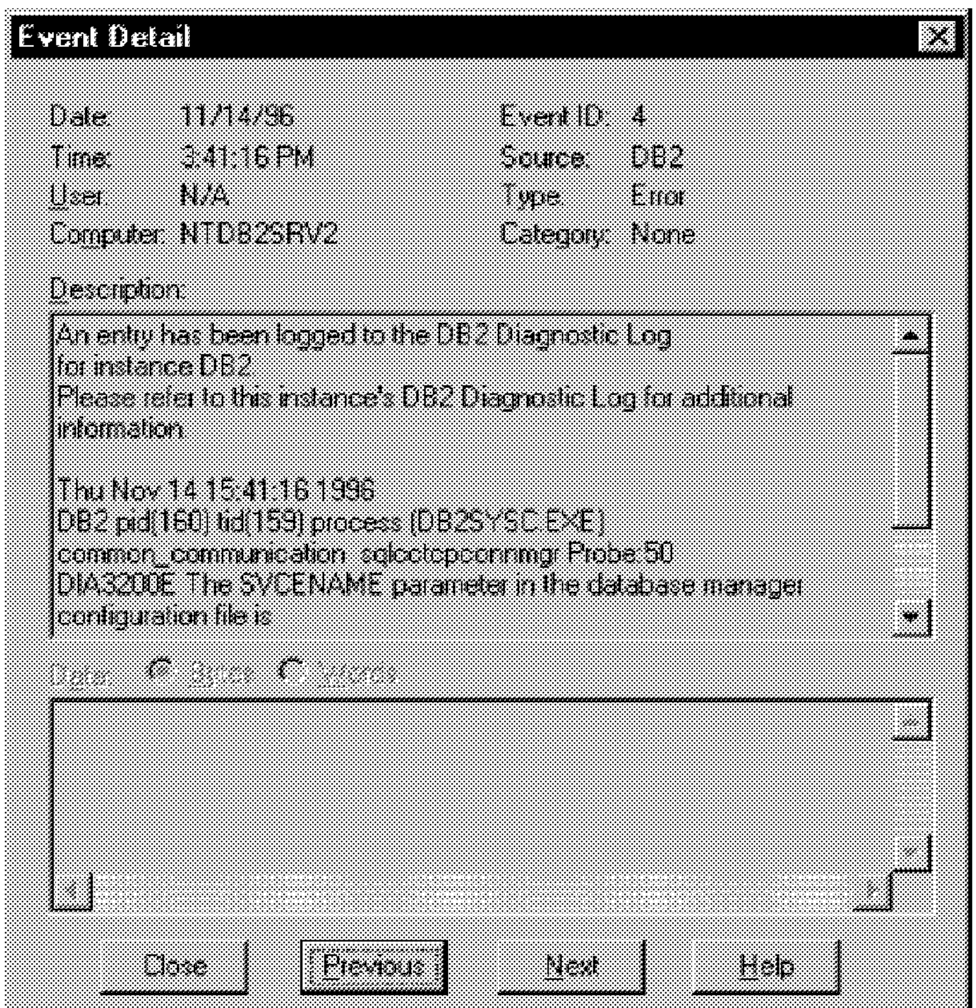


Figure 183. Application Log Event Entry

In this example, an error has occurred with DB2. An entry has been made in the DB2 diagnostic file, DB2DIAG.LOG. The specific detail relating to the error as written to the diagnostic file is also displayed. In this case the SVCENAME parameter has not been configured in the database manager configuration file with the name as it appears in the TCP/IP services file.

6.2.4.1 DB2 for NT Event Logging

Figure 183 on page 264 shows that DB2 entries in the Application Log will often point to more information in another file. In addition to the logging performed by Windows NT, DB2 for NT produced the following output files to assist with problem determination:

- DB2DIAG.LOG. This is the main diagnostic reporting file in DB2.
- *.TRP (trap files)
- *.DMP (dump files)

All of these files will be located in the path SQLLIB*instancename.*

For more information in this book on problem determination for DB2 for Windows NT, refer to Chapter 7, "Problem Determination" on page 267.

6.2.4.2 DB2 Event Monitor

In addition to the Windows NT event logging, discussed in Section 6.2.4.2, "DB2 Event Monitor," DB2 for Windows NT supports event monitoring of the DB2 Common Server products. You are able to monitor the following:

- Buffer pool
- Lock
- Sort
- Statement
- Table
- Unit of Work

Chapter 7. Problem Determination

This chapter provides background information about how to troubleshoot and resolve the most common errors that you may encounter when using DB2 for Windows NT Server. There are a number of different kinds of potential problems that you may encounter, and several techniques you can use in dealing with those problems. In order to make resolve problems as quickly as possible, the ability to determine the area where the problem is occurring is important, whether the problem is hardware- or software-generated.

This chapter is outlined as follows:

- Introduction to the topic of problem determination
- What help is available online
- Diagnostic tools available within DB2, such as DB2DIAG.LOG
- Using the Windows NT Event Log to assist in problem determination
- Performing a trace in DB2
- Performing an ODBC trace
- How to recover from errors occuring during installation

7.1 Introduction

There are two main techniques in the area of problem determination:

1. Use information that is created at the time the failure occurred, such as the information that is produced by with the *First Failure Data Capture* (FFDC) in the DB2 main error log file (DB2DIAG.LOG).
2. Try and re-create the problem and capture it using the DB2 trace facility. This would be done if IBM support personnel had been contacted and in order for them to diagnose the problem, a re-creation of the error situation is necessary.

FFDC produces diagnostic information in several different formats. Each of these formats make it easier to pinpoint and diagnose problems. The main formats for the diagnostic information are dumps and log files. Dumps provide a snapshot of what was in memory at the time the failure occurred. Log files can provide specific information to help determine what caused a failure.

Windows NT maintains three types of logs: the System Log, the Security Log, and the Application Log. These logs capture some of the FFDC information when a failure occurs. These logs can be examined using the Windows NT Event Viewer.

In addition to the Windows NT event logging facility, DB2-specific log information is also available:

- For every instance of DB2, there will be a First Failure Service Log with DB2-specific information to help diagnose failures. This is contained in a file called DB2DIAG.LOG (see 7.3, “DB2 Error Log (DB2DIAG.LOG)” on page 277).

- The SQL Communication Area (SQLCA) can provide valuable problem determination information. For more information on SQLCA, see 7.3.5, “SQL Communication Area Structure (SQLCA)” on page 281.

Note: For more detailed information about Problem Determination, refer to *Diagnostic Tips and Techniques for DB2 Common Server ,* SG24-4759-00.

7.1.1 Before You Contact IBM

Before contacting IBM support, you should use the diagnostic tools listed in this chapter to attempt to solve your problem or to determine when it is appropriate to contact IBM for assistance.

The information here is designed to help you respond to the following situations:

1. Unexpected messages or unexpected SQLCODEs
2. An abnormal termination (abend)
3. A hanging or looping situation
4. In case you are experiencing an unexpected SQLCODE or error message, perform the following steps:

    a. Note the SQL error code returned.

    b. Get the online message help text (for instance, `DB2 "?" SQLnnnn` -- where nnnn is the SQLCODE/message number).

    c. Examine the Windows NT Event Logs (particularly the Application and System Event Logs) to find out when and where the occurrence of your problem happened and if there are any other events related to your problem.

    d. Check to see that a DB2DIAG.LOG file has been generated.

    e. Search the contents of the DB2DIAG.LOG file for the SQL error code that you have received (using any editor).

    f. Examine the contents of the SQLCA structure (see 7.3.5, “SQL Communication Area Structure (SQLCA)” on page 281).

    g. If the problem can be reproduced, perform a DB2 trace using the `db2trc` command (see 7.5, “DB2 Trace Facility (DB2TRC)” on page 287).

    h. If the application is vendor-supplied, contact the vendor.

    i. If the problem remains, contact an IBM Service Representative.

5. In case you are experiencing an exception condition, perform the following steps:

    a. Note the executable module that reported the abend.

    b. Examine the instance directory for any .trp files in the $DIAGPATH directory.

    c. Examine the Windows NT Event Logs (particularly the Application and System Event Logs) to find out when and where the occurrence of your problem happened and if there are any other related events.

    d. If the problem can be reproduced, perform a DB2 trace using the `db2trc` command (see 7.5, “DB2 Trace Facility (DB2TRC)” on page 287).

    e. If the application is vendor-supplied, contact the vendor

f. If the problem remains, contact an IBM Service Representative. In case you notice that your application or the system is hanging, check the following:

g. Check the application for any erroneous looping and the operating system for any applications that are using too much CPU time.

h. Check the applications for possible contention on database ressources (lock escalation, inappropriate isolation level, catalog tables locked) which may result in hanging situations.

i. If the problem involves a client workstation, ensure the client configuration is correct.

j. Examine the instance directory for any .trp files in the $DIAGPATH directory (see 7.3.3, "Dump Files (.dmp)" on page 279).

k. If the problem can be reproduced, perform a DB2 trace using the `db2trc` command (see 7.5, "DB2 Trace Facility (DB2TRC)" on page 287).

l. If the application is vendor-supplied, contact the vendor.

m. If the problem remains, contact an IBM Service Representative.

7.2 Available Online Information

In many cases, the cause of a problem may be readily apparent, and you will be recover without assistance. In other cases, the user can recover using one of the following resources:

- Online help
- Messages Reference
- DB2 Technical Library

This section describes how to access the DB2 online help facility and interpret error messages. We also look at the help available on the WWW DB2 Technical Library.

7.2.1 Online Help

The DB2 online help facility should be the first tool to use whenever you encounter an error condition. It will provide you with a detailed explanation of your error message, and most of the time it will tell you what the cause is and which action to take to correct the error.

The DB2 Command Line Processor has online help that provides information about general topics, command syntax and error messages.

To access the DB2 Command Line Processor on Windows NT, you can use a DB2 icon:

1. Open the DB2 program group.
2. Double-click on the **Command Line Processor icon**. This places you in an interactive mode.
3. Or double-click on the DB2 Command Window. This will initialize a DB2 environment from which you can issue DB2 Command Line Processor commands.

Alternatively, from any Windows NT Command Prompt window, you can:

1. Enter: db2cmd
2. Use the DB2 Command Window to issue DB2 Command Line Processor commands.
3. Enter a Command Line Processor command such as:
4. db2 connect to sample

The DB2CMD Command

db2cmd.exe calls a batch program, db2env.bat, (which should be located in your SQLLIBBIN directory) and passes to it the pid (process Id) of the background process that has been created. db2env.bat sets the environment variable DB2CLP to be this pid.

General Help provides a list of the commands that are available. To get General Help in the Command Line Processor, type ? from interactive input mode.

Syntax Help provides help text for the Command Line Processor command syntax. To get Syntax Help in the Command Line Processor, type ? <phrase> from interactive input mode, where <phrase> is a Command Line Processor command.

SQLSTATE Help provides help text for SQL states and class codes. To get SQLSTATE help from Command Line Processor interactive mode, issue the commands:

```
? <sqlstate>
```

or

```
? <class-code>
```

where <sqlstate> is a valid five digit SQL state and <class-code> is a valid two-digit class code. For example, ? 08003 and ? 08

Figure 184. SQLSTATE Help from the DB2 Command Line Processor

Command Help provides help information on specific commands. To get help on a command, type ? command where <command> can be the first few keywords. For example,

```
? catalog database
```

displays help for the catalog database command, whereas

```
? catalog
```

displays help for all the catalog commands.

Message help is available for DB2 messages, describing the cause of the message and any action that should be taken to solve the problems. To get Message Help from the Command Line Processor interactive mode, issue the command:

```
? <message number>
```

where <message number> is a valid command line processor or SQL message number. For example, ? SQL30061N provides help on the SQL30061N message.

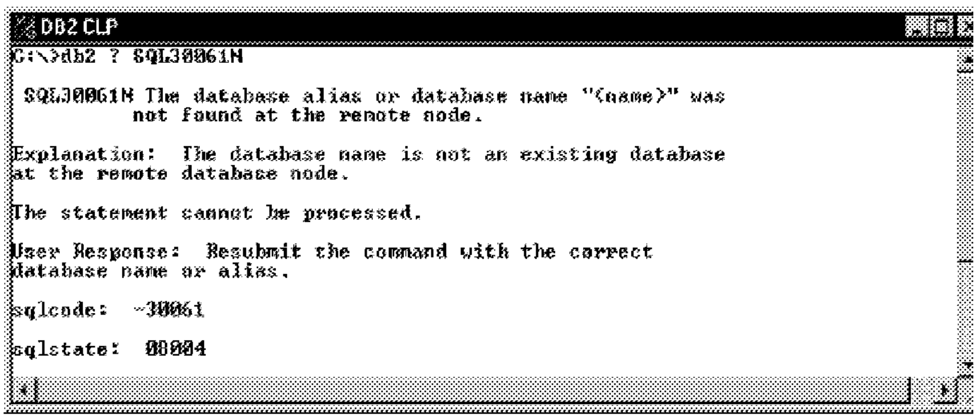


Figure 185. Message Help From DB2 Command Line Processor

If a message is larger than the scrollable section of the screen you are using, it can be sent to a temporary file. For example:

```
update command options using z on output.msg
? SQL0008N
update command options using z off
```

or

```
db2 ? SQL0008N > output.msg
```

Alternatively, you can display the output of the ? command one screen at a time by by redirecting the output using the more command as in the following example:

```
db2 ? SQL0008N | more
```

7.2.2 Reference Messages

Depending on the prefix of the message identifier, you can identify the main DB2 component from which the message is originated.

The following DB2 messages are accessible from the DB2 Command Line Processor:

Table 21. Messages Available from the Command Line Processor

PREFIX	DESCRIPTION
CLI	Call Level Interface messages
DBA	Error messages generated by the Database Director and the Database Administration utility
DBI	Error messages generated by installation and configuration
DB2	Error messages generated by the Command Line Processor
SQL	Error messages generated by the database manager when an error condition has been detected

Message identifiers consist of a three character message prefix (CLI, DBA, DBI, DB2, SPM, or SQL), followed by a four or five digit message number. The single digit letter at the end describing the severity of the error message is optional.

To access help on these error messages, enter the following at the from any DB2 Command window (see 7.2.1, “Online Help” on page 269):

```
db2 ? XXXnnnnn
```

where XXX represents the message prefix and where nnnnn represents the message number.

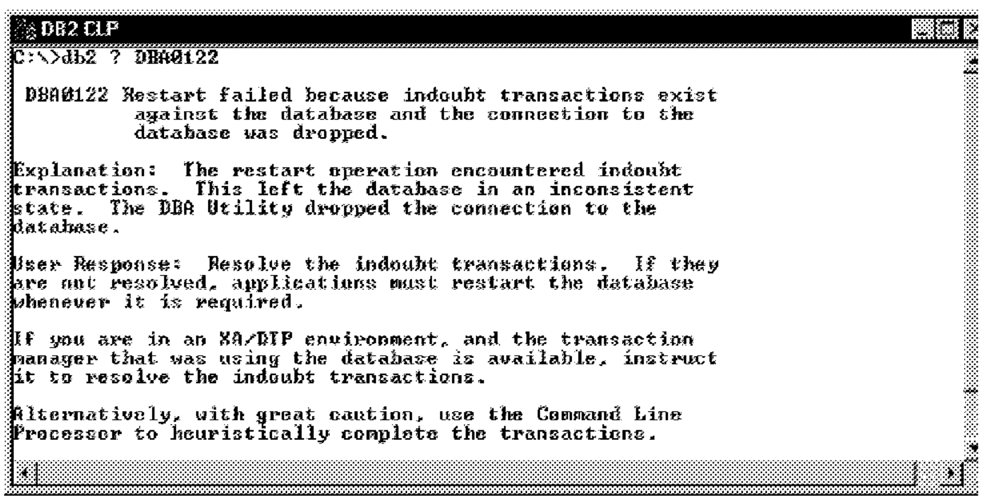


Figure 186. Help on Error Messages in the Command Line Processor

Note that the message identifier accepted as a parameter of the db2 command is not case sensitive, and the terminating letter is not required.

If the message text is too long for your screen, issue the following command from the NT Command Prompt:

```
db2 ? XXXnnnnn | more
```

Help can also invoked in the interactive input mode. To enter the interactive input mode, enter the following at any DB2 Command window:

```
db2
```

Once in the interactive input mode, you can enter any ? command, as in the following example:

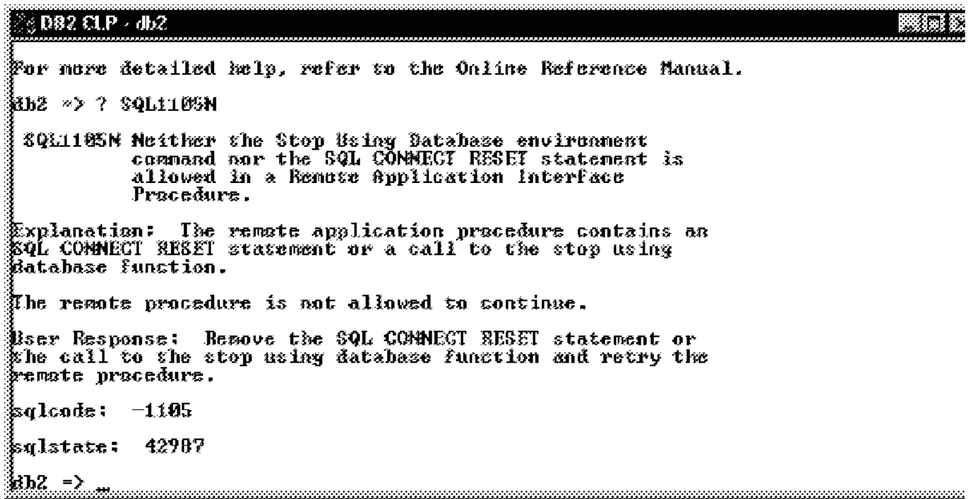


Figure 187. Help Invoked in Interactive Input Mode

7.2.3 DB2 Technical Library on the Web

The DB2 Product and Service Technical Library lets you access information such as README files, the basic core set of books, and technical notes on fixes and Frequently Asked Questions. This can be very useful in problem determination since it allows you to search for similar problems or errors that have been encountered before.

7.2.3.1 What You Can Access

You can access information in many ways:

- **Library Search** lets you find documents containing keywords and will let you limit your search to one or more types of documents. (For help on choosing the keywords that will work best for you, click on the question mark provided on the search page.)
- **Users′ Picks** shows those documents that answer the questions that are most frequently asked by DB2 users.
- **Recent Additions** lists the newest documents added to the library.
- **DB2 Publications** lets you link to the core set of DB2 books.
- **Debugging Problems** gives some tips on how solving problems to make DB2 work for you.

Figure 188. Web Page of DB2 Technical Library

7.2.3.2 Debugging DB2 Problems

When debugging DB2 problems using the DB2 Product and Service Technical Library, there are a few basic principles to keep in mind when structuring your search. By listing them here, we hope that they will help you use the Library more effectively.

1. When specifying your search terms, you should always attempt to use keywords that reflect the symptoms of the problem that you are encountering. Since the number of possible causes of a problem can be infinite, it is better to search on the symptoms because there are a finite number of symptoms. This provides the most effective way of narrowing your search to the information you need.

   Possible problem symptoms include:

   - Return codes and messages
   - Hang situations
   - Abends, access violations, and segmentation faults
   - Incorrect output
   - Poor performance

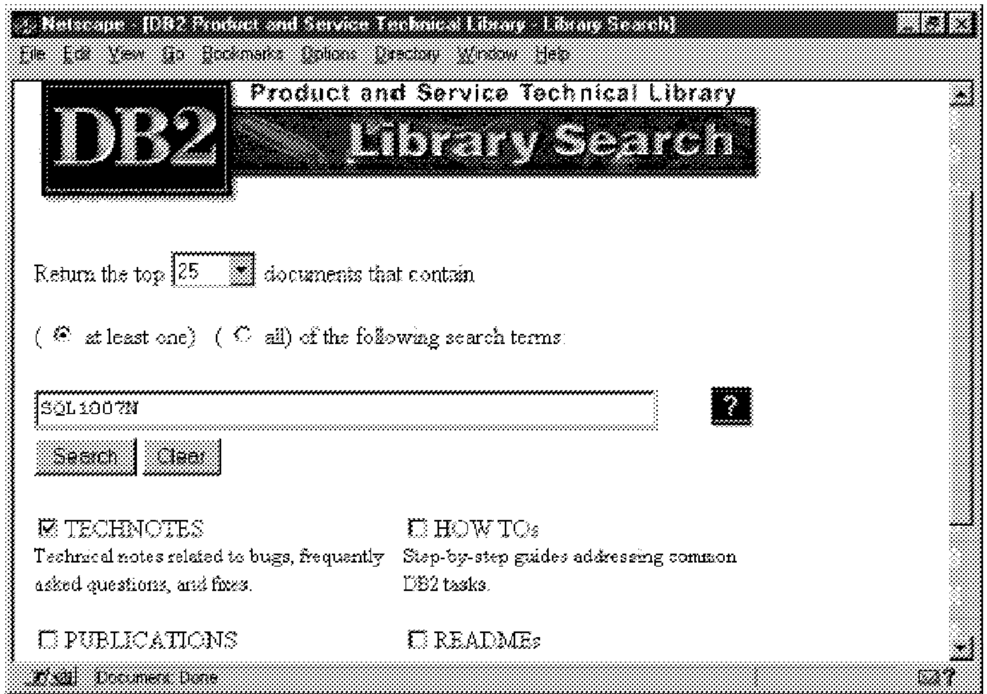


Figure 189. DB2 Library Search Facility

a. Return codes are by far the best search itmes to use in specifying your search. Return codes will quickly identify your particular problem situation.

   When using return codes as search criteria, it is important to spell the return code exactly as it appears. For example, an SQL0805N error message *should not* be spelled as SQL0805 or SQL805N.

b. If your searches have been yielding a small number of documents, it may be possible that your search string is too specific. You may wish to use wild-card characters to increase the yield of your searches, allowing for variations in style across documents.

   For example, to search for documents related to an SQL2003C message, you may wish to search using the following terms:

   ```
   %2003%
   ```

c. Problems involving poor performance and incorrect output can be much harder to search on due to the wide variation of operating environments and data among users. While many attempts have been made to use the words 'poor performance' in all documents related to such problems, you might also search using keywords such as 'slow'.

   It is also useful to use other keywords to indicate what task you are attempting to perform or which application program you are using.

   For example, if getting poor performance backing up your DB2 for AIX database to a tape drive, you should use the following search terms:

   ```
   performance and backup and tape
   ```

d. Many return codes and messages have sub-codes or additional reason codes that further identify the problem. If searching on the primary return code yields too many documents, you should use the secondary return code to further restrict the set of documents returned.

For example, if you encounter an SQL0902C error with reason code 59, you should search using the following terms:

```
SQL0902C and 59
```

e. It is not necessary to describe your complete operating environment when constructing your search. In case you have chosen to search for documents containing all of your search terms, this can have the effect of over-specifying your search. For the most part, searching based on a concise set of symptoms, such as SQL codes, should return a useful and compact result set.

   For example, when getting an SQL1403N error attempting to connect to an MVS database from an NT client, you should initially use only the following search term:

   ```
   sql1403n
   ```

   Searching on ′sql1403n and mvs and NT′ may exclude a document from being included that describes a similar problem encountered on a DOS client where the cause was the same.

f. Because the Library uses the underscore (_) character as a wild-card character, you should search the database for terms containing underscores in the following manner.

   For example, when searching on the DB2 Text Extender return code `RC_SE_EMPTY_INDEX`, you should use the following search terms:

   ```
   ″rc se empty index″
   ```

g. When searching on problems related to DB2 Text Extender or DB2 WWW and not using return codes specific to these products, it may be useful to include the name of the product as a search argument.

   For example, to look for documentation related to the DB2 WWW macro report section, you should use the following search:

   ```
   www and report and section
   ```

   (**Note**: The word ′db2′ was excluded from the list of search terms as it is contained in nearly every document in the Library and does not further restrict the set of documents returned.)

   Figure 190 on page 277 shows a technical note related to a problem. The problem was searched for using the keyword `SQL1007N`. A diagram of the Symptom, Possible cause, and Action to solve the problem are presented.

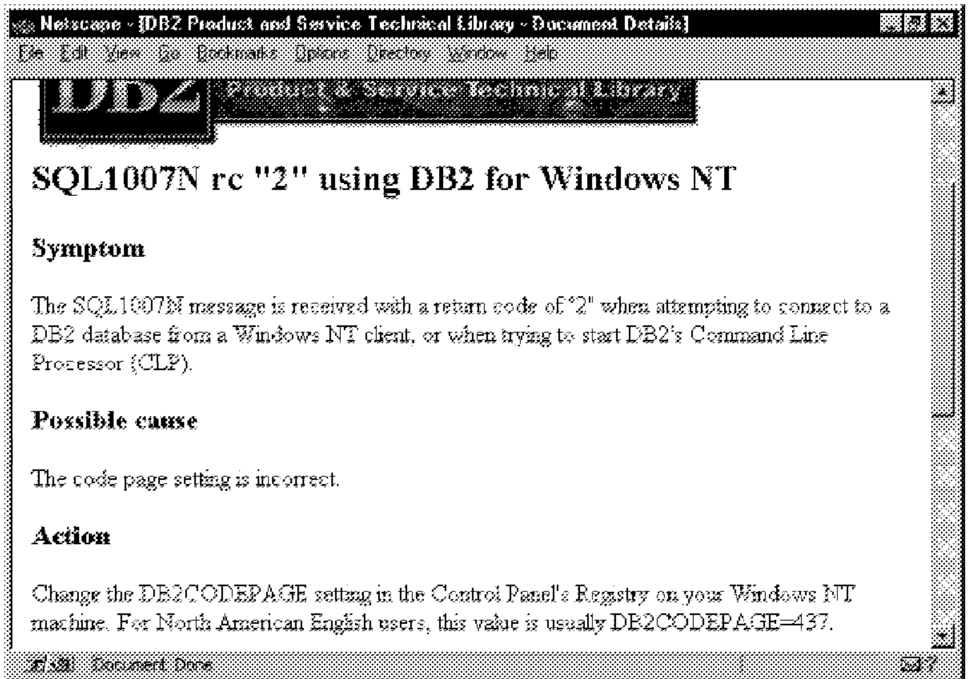


Figure 190. Example of a Technical Note Related to a Problem

7.3 DB2 Error Log (DB2DIAG.LOG)

This section describes one of the most important diagnostic tools available, the DB2DIAG.LOG file. This file is provided by DB2 and should be one of the first places that you investigate when a problem has been detected. Basically, the DB2DIAG.LOG is a file that logs errors recorded by the DB2 Database Manager. When DB2 errors occur at the Database Server, the detecting component within the DB2 product will attempt to log information that will help a database administrator fix the problem or provide enough information for a DB2 service analyst to fix the problem.

This log is written in an easy-to-read format. Its location is specified by the database manager configuration parameter DIAGPATH. If this parameter is null (default value) DB2DIAG.LOG and all the dump files will be created in:

```
x:SQLLIB$DB2INSTANCE
```

where x: is the drive referenced in the DB2PATH environment variable.

Information is appended to the end of the file as errors or events occur. The DB2DIAG.LOG file will not be truncated. If its size becomes too large, erase the file and a new file will be created as needed.

7.3.1 Setting the DIAGLEVEL Configuration Parameter

You can control the amount of information and type of errors that are recorded in the DB2DIAG.LOG file by setting the database manager configuration parameter `DIAGLEVEL` to different values:

- 0 - NO diagnostic data captured
- 1 - SEVERE errors only
- 2 - SEVERE and NON-SEVERE errors

- 3 - SEVERE, NON-SEVERE, and WARNING errors (Default)
- 4 - SEVERE, NON-SEVERE, WARNING, and INFORMATION errors

The default setting for DIAGLEVEL is 3. A DIAGLEVEL of 4 provides the capturing of the maximum amount of data that can be used in problem determination. But since this DIAGLEVEL records the most information, more disk space will be used. This can be undesirable, especially if there is no problem, but informational or warnings are captured.

A DIAGLEVEL of 3 is sufficient most of the time. In case you receive an error and the information provided is not detailed enough, increase DIAGLEVEL to 4 and re-create the problem. If the error is intermittently reported, set DIAGLEVEL to 4 until you are sure the error has been recorded. A DIAGLEVEL of 4 is recommended during the installation and setup of DB2 or during times of errors.

If you begin the process of error investigation yourself, you will find that it is useful to extract only the error situation to reduce the amount of data that is being recorded on a daily basis. To reduce the amount of data, you can rename the DB2DIAG.LOG to a file that includes the timestamp. DB2 will automatically create a new DB2DIAG.LOG file.

7.3.2 DB2DIAG.LOG File Entry Format

Let's look at an example of a DB2DIAG.LOG file to understand its format and the type of information recorded. For our example, our DIAGLEVEL is set to 4. We are using the SAMPLE database; the SMS tablespace is USERSPACE1, and the table is department. Here, we have purposely manipulated the container (file) that stores the index for the department table.

db2 connect to SAMPLE

db2 select * from departmentNext, we issued the following commands:

```
db2 connect to SAMPLE
db2 select * from department
```

This produces the following DB2DIAG.LOG. (Only a portion of the file is shown here.)

```
Mon Dec 17 23:30:44 1996 (1)
DB2 (2)pid(120) (3)tid(62) (4)process (DB2SYSC.EXE (5)
buffer_pool_services (7) sqlbStartPools (8) Probe:0 (9) Database (SAMPLE (6)

Starting the database.
```

The general information of each DB2DIAG.LOG entry includes (The numbers in brackets () are explained.):

- (1) Date of the error or information message
- (2) Instance name - Here it is DB2.
- (3) Process id - This is identified by pid(163).
- (4) Thread id - This is identified by tid(49).
- (5) Process - The process that is being executed here is a DB2SYSC.EXE.
- (6) Database name - The database is SAMPLE.

- (7) Component identifiers - This is the component that has detected and recorded the error. In our example, it was buffer pool services.
- (8) Function name - This is the function where the error occurred.
- (9) Position Id (within the function) - In our example, this was indicated by Probe:0.

In our example, the first entry to DB2DIAG.LOG was made while a connection to database was done. If connections to the database are issued from remote clients, you will see the start of the communication protocol that is defined in the DB2COMM variable.This is done when the instance is started.

This second example is taken from the same DB2DIAG.LOG in the previous section. Here, we will look at more information, the internal return code:

```
Tue Dec 17 23:38:43 1996
DB2 pid(120) tid(62) process (DB2SYSC.EXE)
buffer_pool_services  sqlbrdpg Probe:1141 Database (SAMPLE)
DIA3700C A bad page was encounterred.

ZRC=01E1FFFF
```

You always have to check if the hex internal return code you received is byte-reversed. It is not necessarily byte-reversed all the time. A hex value beginning with FFFF is in a valid format. If your value does not begin with hex FFFF then it is byte-reversed, and you have to convert it into a valid format.

A conversion of a byte reversed value 01 E1 FF FF is performed by going backwards beginning with FF FF E1 01.

An internal return code (reported by DB2 directly or converted due to being byte-reversed) can be further processed by a conversion into decimal to see if the result is a valid SQL code.

If there is a valid SQL code, look up the SQL code in the *DB2 Message Guide* or with the following from the command line:

```
db2 ? SQLXXXX
```

where XXXX is the number received with the SQL error code.

In this example, hex FFFF E101 converts to -7935, which is not a valid SQL code. If the return code does not convert to an valid SQL code, then try to look up the return code E101 in the file internal return code file:

```
/*  -980  Bad Page */
/*         E101
```

7.3.3 Dump Files (.dmp)

Dump files are created when an error occurs for which there is additional information, such as internal control blocks, that would be useful in diagnosing a problem. When a dump file is created or appended to, a reference is made in the DB2DIAG.LOG file. This information is also described in the *Problem Determination Guide.*

The following section shows some examples of the data that gets placed in dump files. These files will have an file name made up of the pid (process Id) concatenated to the tid (thread Id) and a *.dmp extension. The data provided in the *.dmp file depends on the error that has occurred.

The DB2 engine tries to dump all available data of an application (process-Id) that is found at the moment the error occurred. This file can only be examined by using a hex editor and then by examining the source code. This process can only be done by an analyst in the DB2 support team. To figure out which bit contains information, you need to determine the errors, and you need to know internal structures.

However, some of the information that is placed in the dump files also gets written in ASCII format to the DB2DIAG.LOG file. We will go through an example of what you might find in the DB2DIAG.LOG file as it relates to information that is also placed in a *.dmp file.

7.3.4 DB2DIAG.LOG and *.dmp Files

This section illustrates how information from the *.dmp files is placed in the DB2DIAG.LOG file. We look at five examples.

7.3.4.1 Finding the Name of *.dmp File

If it is necessary to establish if something is wrong in the following entry, the IBM Support team would check the function in the source code. The following portion of the DB2DIAG.LOG shows what the dump looks like. The structure in the function sqlbrdpg at Probe Id:74 gives more information to the DB2 analyst about each bit included in this dump. Not all data is meaningful in relationship to the error. Notice that the location of the dump file is documented in the DB2DIAG.LOG as you can see in the following example. According to the pid (process Id) and the tid (thread Id), the actual file name is 12062.dmp.

```
Mon Dec 17 23:38:44 1996
DB2 pid(120) tid(62) process (DB2SYSC.EXE)
buffer_pool_services  sqlbrdpg Probe:74 Database (SAMPLE)

DiagData
0000 0000 0000 0000 0000 0000 0000 0000      ................
0000 0000 0000 0000 0000 0000 0000 0000      ................
0000 0000                                    ....

Dump File : C:\SQLLIB\DB2\12062.dmp Data : SQLB_PAGE
```

7.3.4.2 Reading Information Written to a *.dmp File

This example shows the information that is placed in the *.dmp file. Here we can see in the DB2DIAG.LOG file, that the statement where the error was found was during the execution of SQL SELECT statement on the DEPARTMENT table.

```
Data Title :section stmt pid(120) tid(62)
7365 6c65 6374 202a 2066 726f 6d20 6465      select * from de
7061 7274 6d65 6e74                          partment

Dump File : C:\SQLLIB\DB2\12062.dmp Data : section stmt
```

7.3.4.3 Restore Command Issued with Invalid Path

This portion of this particular error in DB2DIAG.LOG indicates that a RESTORE command with an invalid path or timestamp for the backup image has been issued.

```
Wed Dec 18 16:35:50 1996
DB2 pid(93) tid(164) process (DB2SYS.EXE)
database_utilities  sqluMCInitRestoreMC  Probe:360

Media Controller -- sqluInitFileDevice() | device initialization warning

Wed Dec 18 16:35:51 1996
DB2 pid(93) tid(150) process (DB2SYS.EXE)
database_utilities  sqludrsa  Probe:125 Database (GEDRESULT)

DiagData
12f6 ffff

Wed Dec 18 16:35:51 1996
DB2 pid(93) tid(150) process (DB@SYS.EXE)
oss_2               sqloTermWaitPost  Probe:20 Database (GEDRESULT)

0600 0000

Wed Dec 18 16:35:51 1996
DB2 pid(93) tid(150) process (DB@SYS.EXE)
database_utilities  sqludrsa  Probe:0 Database (GEDRESULT)

Restore terminated.
```

7.3.5 SQL Communication Area Structure (SQLCA)

For severe errors, the SQLCA structure and its contents will be dumped into the DB2DIAG.LOG file as well as a *.dmp file. The *.dmp file will also contain additional information specific to the error.

```
Mon Dec 17 23:38:51 1996
DB2 pid(120) tid(62) process (DB2SYSC.EXE)
relation_data_serv    sqlrerlg Probe:17 Database (SAMPLE)
DIA9999E A internal return code occurred. Report the following : "0xFFFFE101". (1)

Data Title :SQLCA pid(120) tid(62)
 sqlcaid : SQLCA     sqlcabc: 136 (2)sqlcode: -980   sqlerrml: 0
(3)sqlerrmc:
 sqlerrp : sqlrita
(4)sqlerrd : (1) 0xFFFFE101      (2) 0x00000000      (3) 0x00000000
           (4) 0x00000000      (5) 0x00000000      (6) 0x00000000
 sqlwarn : (1)      (2)      (3)      (4)        (5)        (6)
           (7)      (8)      (9)      (10)       (11)
 sqlstate:

Dump File : C:\SQLLIB\DB2\12062.dmp (5)Data : SQLCA
Dump File : C:\SQLLIB\DB2\12062.dmp (5)Data : DB2RA
Dump File : C:\SQLLIB\DB2\12062.dmp (5)Data : ACB
```

We will explain the numbers that are found in the portion of the DB2DIAG.LOG:

- (1) This is an internal return code that was reported by DB2. The SQL error code that the user receives is SQL1224N. Unfortunately, this error code is used for many different errors. It can be very difficult to determine the error based on the return code of SQL1224N.

- (2) This indicates the SQL error code that was found. If there were reason codes associated with it, then you would find it displayed in the (3) `sqlerrmc` field.
- (4) Sometimes there have been several errors leading to the final SQL error code. These errors would be shown in sequence in the `sqlerrd` area.
- (5) This indicates that the dump file; here it is 12062.dmp, contains the contents of the SQLCA area.

7.3.6 Trap Files (.trp)

Under severe error conditions, the DB2 system issues an exception to itself. All exceptions initiated by DB2 are reported in a *.trp file. The meaning of an exception is specific to the operating system, Windows NT in our case.

Besides the exception, there will be a stack trace back (this is a list of the functions that have been called by the process) showing the last steps of the function flow at the time the exception occurs. Each *.trp file is reporting a single exception that was issued by the DB2 engine to stop or interrupt a process. The DB2 support team wants to know which exception was issued and the last steps that were performed.

In the following example, you will see that the last functions were the following: sqloDumpEDU, sqlbLogErr, sqlbrdpg, sqlbReadPage, and sqlbfix. Depending of the severity of the error, the database manager may be terminated.

In the following example, DB2 tried to read a page from the buffer. Errors have been written to the DB2DIAG.LOG, and the database connection was terminated.

```
Exception C0010002 Occurred
Exception Address  =  17DA8658        sqloDumpEDU

Registers dump:
---------------

GS  : 0000     FS  : 150B     ES  : 0053     DS  : 0053
EDI : 0DAF4DD0 ESI : 0040E1AC EAX : 00000000 EBX : 12DE95D4
ECX : 00000004 EDX : 0040DE68
EBP : 0040DEA0 EIP : 17DA8658 EFLG: 00082202 ESP : 0040DE58
CS  : 17DD005B     SS  : 400053

Instruction dump:
-----------------

 17DA8658 : 83 C4 04 89 45 F8 0B C0 0F 84 95 00 00 00 50 B9
 17DA8668 : 05 00 00 00 83 EC 0C BA 5F 00 00 00 B8 4A 00 00
 17DA8678 : 00 E8 8A 21 02 00 83 C4 10 89 45 F8 E9 72 00 00
 17DA8688 : 00 8D 40 00 8B 5D 08 8B C3 E8 56 8B 01 00 0B C0
 17DA8698 : 0F 84 5D 00 00 00 8D 4B FF 53 8B D1 BB 01 00 00

Stack calling chain:
--------------------
16923CBF 00000007 00000000   sqldLoadTCB__FP7sqleacbP8SQLD_TCBi
16923765 0DAF4DD0 005F6340   sqldFixTCB__FP7sqleacbiN32PcT2T6T2PP8SQLD_TCB

DLL Addresses:
--------------

Code Address range for DB2ABIND is 16830000 .. 16842724
Code Address range for DB2APP   is 16D70000 .. 16DF2760
...

Date/Time: 1996-11-27-16.58.33

Thread slot 121 , Id 7 , priority 200
Stack Base   : 003D0000 ;Stack Top    : 00410000
Process Id   : 130
Process name : C:\SQLLIB\BIN\DB2SYSC.EXE

/*----- Stack Bottom ---*/
 0040DE58 :0040DE68 00000007 00000E83 00000000
 0040DE68 :C0010002 00000000 00000000 17DA8658
 0040DE78 :00000000 00000007 00000000 00000000
```

Additional activities, like marking the database or the affected tablespace as ″bad″ are performed to avoid inconsistency problems. The database or tablespace will remain in this status as long as repair actions are required or if restart of the database is able to correct the errors.

7.4 Windows NT Event Log

Part of the DB2 First Failure Support (FFST) is implemented using the Windows NT Event Log facility. It provides failure analysis support functions, such as error detection and message logging during the operation of certain DB2 components.

Unlike the DB2DIAG.LOG file which may happen to get pruned more often (due to possible disk space problems), the NT Windows Event Log allows you review a history of incidents (which may have been overlooked) over a fairly large period of time. This can allow the user to trace the cause or origin of a problem that occurred in the past.

Among the three Event Logs that Windows NT maintains, the Application Log is the most critical to when investigating problems with DB2 for Windows NT. DB2 for Windows NT is automatically configured to use the Windows NT Application Log.

7.4.1 Viewing Event Logs

To view events from any of the logs, open Event Viewer and select the log (System, Security, or Application) from the Log menu. By default, all events for that log are displayed with the most recent event appearing first. These options can be changed in the View menu along with a range of other options. You may change the time range through which you want to view events and the type of events to display. Figure 192 on page 286 shows the Application Event Log. Here entries can be seen from DB2. There are a number of different events being logged.

If you want to see more details about an occurrence of a specific event, then double-click on the event you want to see, and the Event Detail dialog box for that event will popup. A set of data pertinent to the event can be viewed, for example:

- The date and time the event occurred
- The source of the event (typically the software that logged the event)

In our example, an error has occurred with DB2. An entry has been made in the DB2 diagnostic file DB2DIAG.LOG. The specific detail relating to the error as written to the diagnostic file is also displayed in the Event Detail dialog box. In this case a page of the SAMPLE database is either missing or has been found to contain bad or corrupted information. A detailed analysis of the DB2DIAG.LOG file will allow to locate the failed object that caused the error.

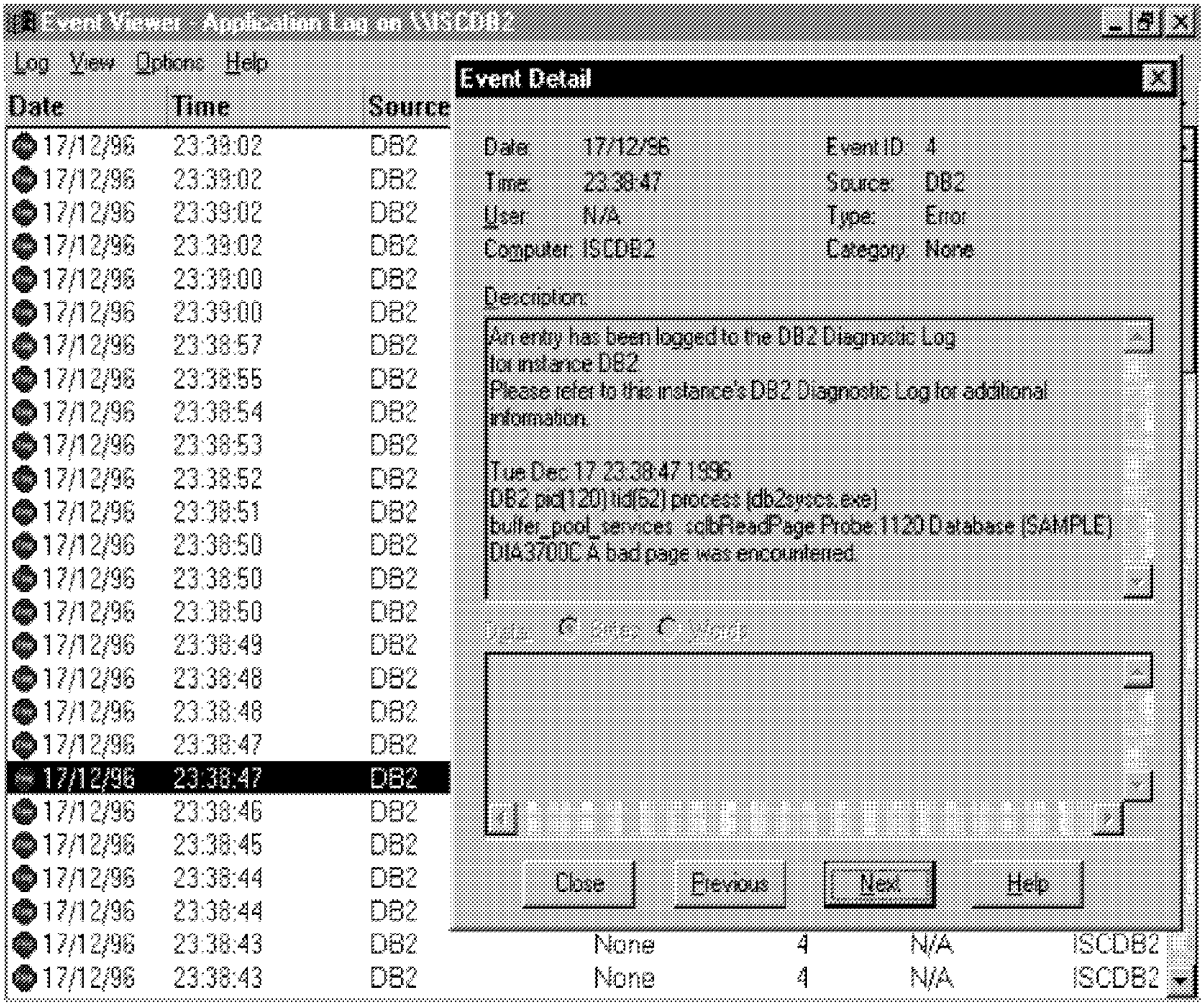


Figure 191. Viewing Windows NT Application Event Log

7.4.2 Recording DB2 Events

Five event types can appear in the event log, namely: information, warning, error, successful audit and failure audit. Figure 192 on page 286 shows the details of an Event Log entry, and which type of information has been logged when the DB2 server was started.

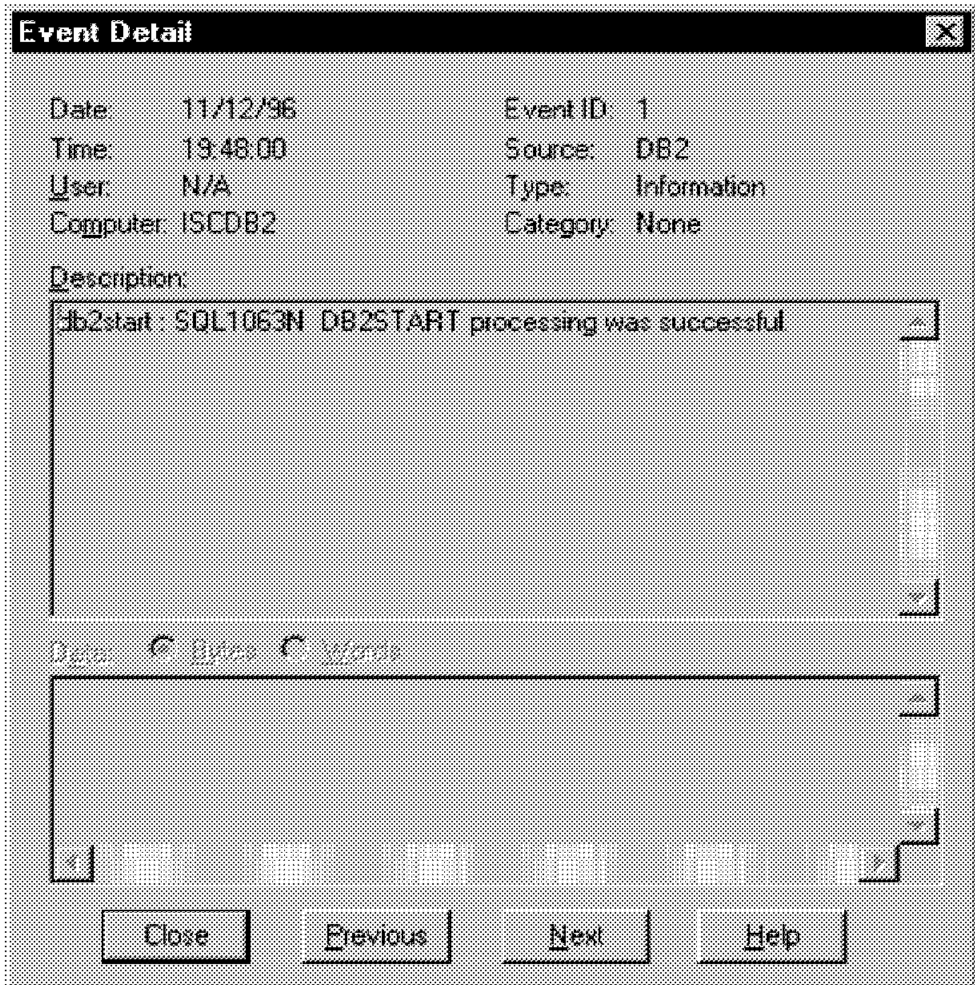


Figure 192. Windows NT Event Detail Dialog Box

7.4.3 Archive Windows NT Event Logs (.evt)

You may want to keep the Windows NT Event Logs for a certain period of time in order to be able to trace back the origin or cause of an error or a problem. This can be very useful at the stage when you start running your applications. In order to save disk space, as well as in case of Windows NT Event Log full-condition error (in which case the error occurrences will stop being recorded in the Windows NT Event Logs), you may want to archive your current Event Log contents by Event Log type such as System, Security, Application and retrieve them at a later stage. To do so, use the ″Save As″ option of the Log window in the Event Viewer as shown in Figure 193 on page 287. The log will be archived as a *.evt file.

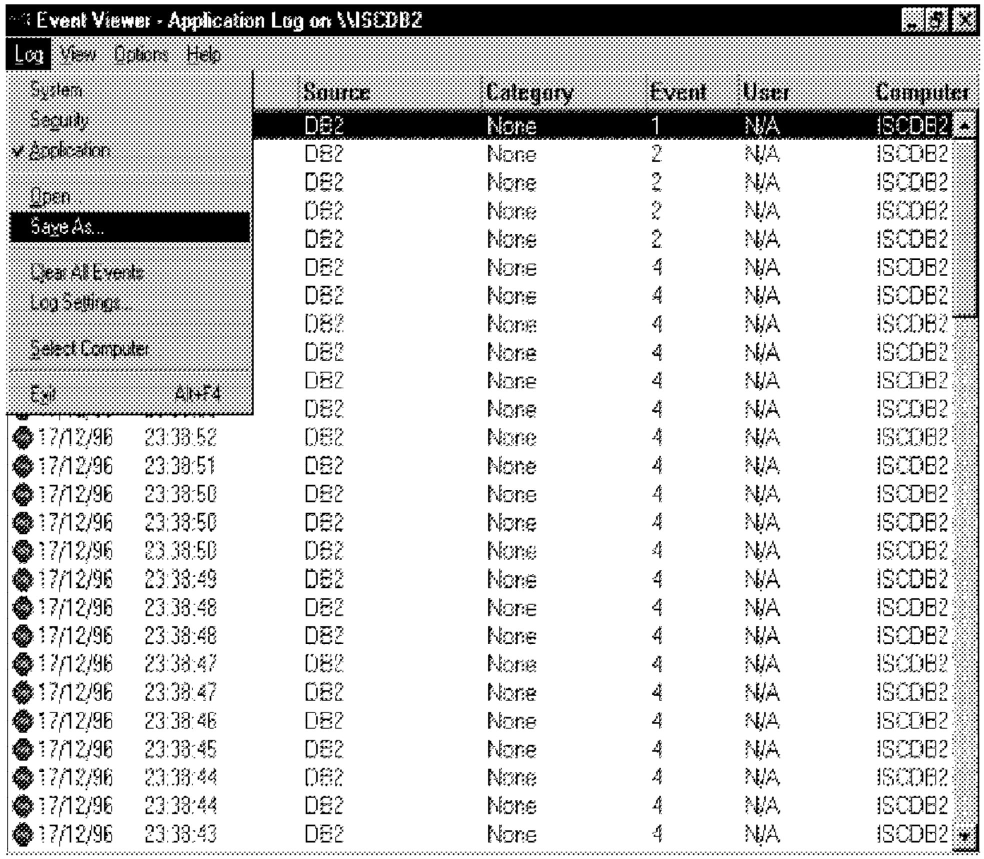


Figure 193. Save As Option of the Log Window in the Event Viewer

To retrieve a Windows NT Event Log (previously archived), use the Open option of the Log window in the Event Viewer.

7.5 DB2 Trace Facility (DB2TRC)

DB2 trace information is either in memory or on disk. If trace collection was activated, trace information is recorded in a chronological order. Each entry in the trace file is called a *trace point* and is sequential.

The amount of information gathered from the trace grows rapidly. A goal of performing the DB2 trace is to capture only the error situation. Any other activities, such as starting the database manager instance (NET START <db2instance>) or connecting to a database (DB2 CONNECT) that does not reproduce the error situation should be avoided. The goal in capturing trace information is to reproduce the **smallest, re-creatable** scenario and capture it for further analysis.

You should consider that the process of performaing a trace will have a global effect on the behavior of the DB2 instance. How much degradation you experience is dependent on the type of problem and how much resources are being used to gather the trace information.

To perform a trace, you must use the `db2trc` command. We will examine some of the ways in which you gather trace information.

7.5.1 The db2trc Command

The general syntax of the db2trc command is:

db2trc <subcommand> <options>

The complete syntax of the db2trc command can be found when the command is issued with -u option. The following output shows the subcommands of the db2trc command:

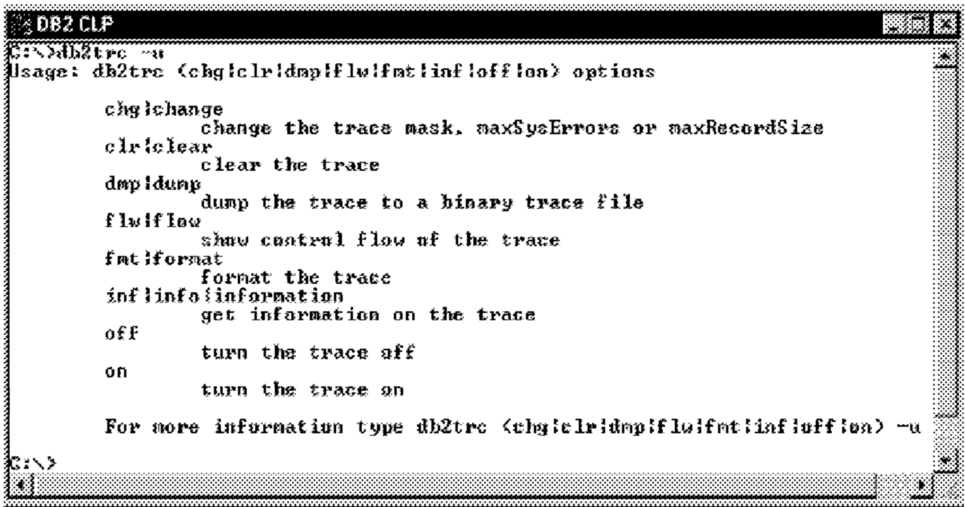


Figure 194. Syntax of the db2trc Command

More detail for each of the different subcommands can also be found by issuing a subcommand and using the -u option. Let's have a closer look at the syntax of the on subcommand of the db2trc command.

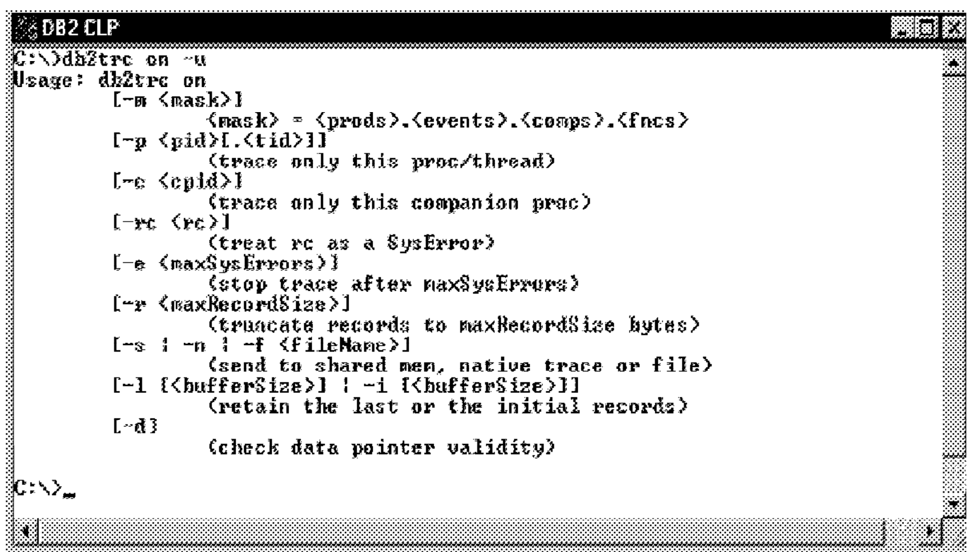


Figure 195. Subcommand Using the -u Option

Collecting information for a DB2 trace starts when the on subcommand is used . This will start the trace and direct the collected trace entries to a specified destination. By default, trace entries are collected into *shared memory.* However, they can also be directed to other destinations, to a disk file, for example. Trace collection ends when the `off` subcommand of the `db2trc` command is issued. We will discuss different destinations for collecting trace information.

7.5.1.1 Capturing Trace Information into Memory

The `db2trc` command can be used to capture a DB2 trace file to shared memory. For example:

```
db2trc on -e -1 -l 4000000
```

The command parameters used in this example are explained further:

- `-e` Specifies the maximum number of system errors allowed before trace collection is terminated.
- `-1` Indicates that all system errors should be traced.
- `-l` (note that it is a lowercase l, not the number one) Specifies the size of the capture trace buffer in memory. It is expressed in the number of bytes (recommended size to use is 4 MB).

This will start the trace facilities. The very next step should be the re-creation of the problem for which the trace is being collected. After you are convinced that the problem has been successfully re-created, the information that is currently in memory needs to be placed in a file so that is can be analyzed by a DB2 support analyst. For example, we will place the information found in memory in a file called tracefile.dmp, by using the `db2trc` command:

```
db2trc dump tracefile.dmp
```

It is very important that as soon as collection of the trace is completed and the information that is contained in memory is sent to a file, you stop the trace facility with the following command:

```
db2trc off
```

7.5.1.2 Capturing Trace Information to a Disk File

The `db2trc` command can be used to immediately capture the contents of information found in memory to a file:

```
db2trc on -e -1 -l 4000000 -f tracefile.dmp
```

The command parameters used in this example are the same as those used in the previous section with the addition of:

- `-f` Specifies the file name and extension of the file to which trace should be written.

The option of collecting data directly to a file is necessary if the re-creation of the problem is causing the DB2 engine (DB2 database manager) to abort. If DB2 has terminated abruptly, it may not be possible to capture the information contained in shared memory in two separate steps.

You must also stop the trace facility:

```
db2trc off
```

The difference between performing a trace in memory and then dumping its contents to a file and specifying the placement of that information when the trace is activated is performance. A trace that is activated with the `-f` option will cause the system to be significantly slower than the trace that is performed in memory.

The -f option of db2trace

The -f option of the db2trc utility is recommended when the error results in system hang situations such that you can no longer dump the trace file manually.

7.5.1.3 Using the info Subcommand

The info subcommand can be executed while trace capture is ongoing. This will provide you with additional information. The info subcommand of the db2trc command can also be executed with a binary trace file as its input operand. In this case, it will provide you with additional information about environment and settings that were present when the collection of the of trace was activated.

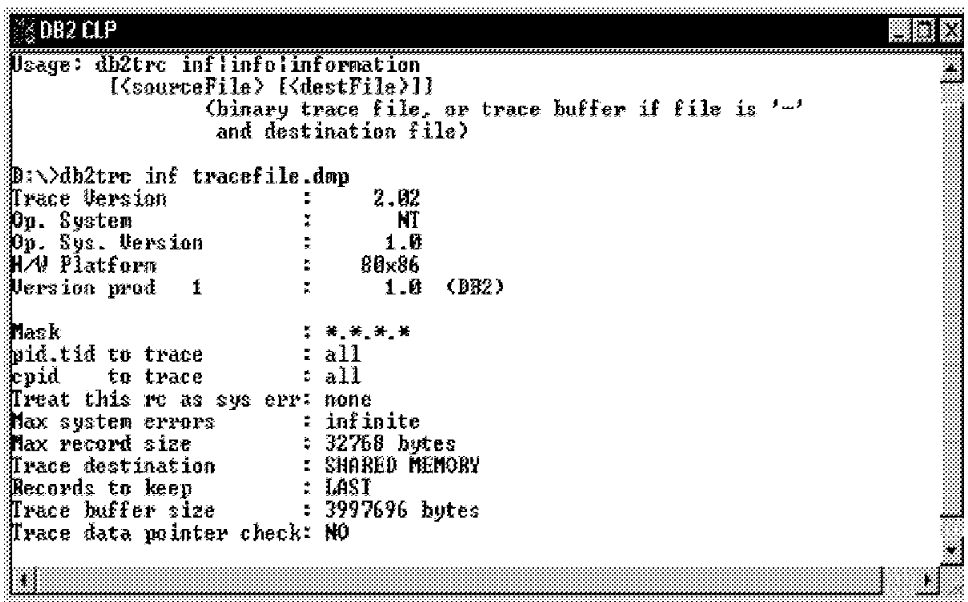


Figure 196. info Subcommand Using Trace File as Input

7.5.2 Starting a DB2 Trace

In DB2 for Windows NT, a trace can be initiated either from the DB2 Command Window in the DB2 for Windows NT group of the Programs menu, or from any Windows NT Command Prompt.

7.5.3 DB2 Trace Entry Format

The file that you specify for recording the trace information in DB2 is a binary file. In order to analyze this trace file, you have to process it into a readable format. You can extract two different sets of information from the same binary trace file depending on the way in which you process the binary trace file:

- *flow* - Represents the flow of control and an overview of invoked functions and return codes (or error codes), and is broken down by process/thread.
- *format* - Represents each trace entry in formatted form as it happened.

The following example is a fragment of the flow and corresponding formatted entries from the same trace file showing some of the trace entries. This example is taken from DB2 for Common Server because each trace entry starts with DB2:

```
Flow:

1435     |sqlodelq   fnc_entry    ...
1436     |sqlodelq   fnc_data     ...
1437     | |sqlofmblk  cei_entry    ...
1438     | |sqlofmblk  cei_data     ...
1439     | |sqlofmblk  cei_retcode  0
1440     |sqlodelq   fnc_retcode  0

...

Format:

1435    DB2 fnc_entry     oper_system_services sqlodelq (1.30.15.196) ---------(1)
        pid 49; tid 1; cpid 0; time 197028; trace_point 0
        called_from 17DF6733

1436    DB2 fnc_data      oper_system_services sqlodelq (1.35.15.196) ---------(2)
        pid 49; tid 1; cpid 0; time 197028; trace_point 1
        0600 0000 0100 0100 50c4 3c0e          ........P.<.

1437    DB2 cei_entry     oper_system_services sqlofmblk (1.20.15.62) ---------(1)
        pid 49; tid 1; cpid 0; time 197028; trace_point 0
        called_from 17DF66EB

1438    DB2 cei_data      oper_system_services sqlofmblk (1.25.15.62) ---------(2)
        pid 49; tid 1; cpid 0; time 197028; trace_point 1
        c400 3c0e 50c4 3c0e                    ..<.P.<.

1439    DB2 cei_retcode   oper_system_services sqlofmblk (1.23.15.62) ---------(3)
        pid 49; tid 1; cpid 0; time 197028; trace_point 254
        return_code = 000000 = 0

1440    DB2 fnc_retcode   oper_system_services sqlodelq (1.33.15.196) ---------(4)
        pid 49; tid 1; cpid 0; time 197028; trace_point 254
        return_code = 000000 = 0

...

715     DB2 fnc_errcode   base_sys_utilities   sqlepvrm (1.7.5.235) ----------(4)
        pid 142; tid 1; cpid 0; time 14662778; trace_point 254
        return_code = 0xfffffc00 = -1024
```

There are several types of trace entries shown in this example:

- Function Entry (1) - Note that in trace file, both fnc_entry and cei_entry actually denote a function entry. The difference between them is subtle.
- Data Trace Entry (2) - This is where variable values are recorded at different trace points within a function and dumped to a trace file. The values are represented as hex digits and ASCII characters.
- Function Exit Trace Entry (3)- This is where the function return code is recorded;
- Error Trace Entry (4)- This is where any additional information relevant to the error condition is recorded and dumped to the trace file.

Other trace entry types include API entries such as api_entry and api_exit, and performance-related entries such as prf_entry and prf_data.

7.5.4 Trace Entry Format

Each trace entry contains diagnostic information. Let's look at a sample formatted trace entry:

```
(1)339 (2)DB2 (3)cei_errcode (4)oper_system_services (5)sqloopenp
(6) (1.6.15.140)
(7)pid 44; (8)tid 11; (9)cpid 44;(10)time 500247;(11)trace_point 254
       return_code = (12)0xffffe60a = -6646 = SQLO_FNEX
```

- (1) Trace Entry number.
- (2) Version Indicator - DBM for Version 1.x; DB2 for Version 2.x.
- (3) Type of trace entry.
- (4) Component to which function being executed belongs
- (5) Function being executed.
- (6) Mask identifying (<Product>.<Event>.<Component>.<Function>) - This can be used to filter the trace collection, but only for specific Events, Components or Functions. Mask settings should be used as advised by a DB2 support analyst.
- (7) Process Id (pid) of the process under which the function was executed.
- (8) Thread Id (tid) of the thread executing the function.
- (9) Companion Process Id (pid) - This is the process Id of the process whose child is the process being traced. Frequently the companion process is a DB2 System Controller (DB2SYSC.EXE) process).
- (10) A time place holder. A parameter must be set to gather real time information. This slows down the system drastically.
- (11)Trace point number - This entry further identifies the control flow within a function. The DB2 support analyst having access to function source code is able to follow control flow. The trace point number also enables him/her to properly interpret data that was dumped, if any was associated with this trace point number.
- (12) The return code is represented as a 32-bit hex number and a signed decimal number. For some of the internal return codes, a symbolic name associated with this return code is also shown. Frequently, this symbolic name makes the identification of the problem much easier. In this example, SQLO_FNEX means File Not EXists.

7.5.5 Verification of Trace

The collecting and capturing of trace information does not imply that the trace information can be helpful in diagnosing the error. There are several reasons where the error condidtion was not captured in the trace information. We discuss those reasons in this section.

7.5.5.1 Trace Information Not Captured

The following describes situations that result in trace information not being captured.

- The trace buffer size as specified was not large enough to hold a complete set of trace events. This situation is frequently called ″trace wrapped″. Although a more correct explanation would be that according to the trace capture request, only the requested last (or first) *nnnn* records were retained in the trace buffer.
- The traced scenario did not result in re-creating the error situation.
- The error situation was re-created, but the assumption as to where the problem occurred was incorrect. For example, the trace was collected at a client workstation while the actual error occurred on server or vice versa.

To verify that the trace contains complete set of trace events which has not been overwritten, it is sufficient to format the binary trace file showing the flow control and optionally sending the formatted output to a null device. We are interested here only in the verification that the trace buffer has not ″wrapped″. An example of the command to perform this task and its output follows:

```
C:>db2trc flw example.trc nul

Trace wrapped : YES ------------------------------------------(1)
Size of trace : 65510 bytes
Records in trace: 1265
Records formatted : 828 (pid: 72; tid: 11)
Records formatted : 215 (pid: 122; tid: 1)
Records formatted : 148 (pid: 133; tid: 1)
Records formatted : 36 (pid: 72; tid: 9)
Records formatted : 4 (pid: 72; tid: 13)
Records formatted : 34 (pid: 134; tid: 1)
```

The trace was wrapped in this case (1). Therefore, it would be advisable to repeat capturing the trace, specifying a larger buffer before proceeding with any further verification of the trace. However, you need to take care that the trace data does not get too large. For example, a size of 40 MB cannot be easily trasmitted through an ftp. Moreover, investigations within a file that is, say 40 MB, for example, are almost impossible since the file becomes larger after it is formatted.

7.5.5.2 Capturing a Trace (Not Wrapped)

The recommended method to capture a trace that has not wrapped should be a goal, but may not be possible for the user to accomplish. Consider as an example a scenario of DB2 reporting an error while accessing one user table. Unfortunately, this particular table contains over 1,100,000 rows. If, as a simple recreation scenario, we used a simple SQL statement to select everything from this table, our chances to define a trace buffer large enough to capture all information from this table in a complete, not overwritten trace file are very small. We will need to use more subtle methods to create a recreatable scenario in such a case. Remember, our goal is to create the *smallest repeatable re-creatable scenario* and capture trace information from it. We also need to be assured that the problem we are trying to solve actually occurred and was captured.

7.5.5.3 Verifying the Recording of the Error

The obvious and best sign that the problem being traced has indeed occurred and was captured in the trace is the occurrence of the error code being reported in the trace. Unfortunately, error codes reported to the user are final error codes. In the formatted trace, you are most likely to find internal error codes reported by different functions called by the DB2 engine.

We have to search the formatted trace for either an internal code, which we are not likely to know as several internal return codes may be mapped to the same external return code, or for the hex interpretation of the external return code. The trace will contain a dump of the SQLCA which contains the external return code.

7.6 ODBC Trace

The DB2 Call Level Interface is an application programming interface for database access that uses function calls to invoke dynamic SQL statements. The DB2 CLI provides a set of functions that includes the core and a subset of the extended functions of Microsoft's Open Database Connectivity interface (ODBC).

7.6.1 Enabling ODBC Trace (DB2CLI.INI)

A DB2 CLI trace is activated by modifying the DB2CLI.INI file (C:SQLLIBDB2CLI.INI) and including the following lines:

```
Trace=1
TraceFileName=<path and file name of the trace file>
TraceFlush=1
```

These lines should be inserted in the [COMMON] section of the DB2CLI.INI file. The ODBC trace will then be active for all databases accessible through ODBC.

Note that ODBC trace is activated and deactivated by setting the value of the TRACE parameter to 1 and 0 respectively.

We are including an example of ODBC trace from an unsuccessful attempt to connect to a DB2 source database from an MSQuery application.

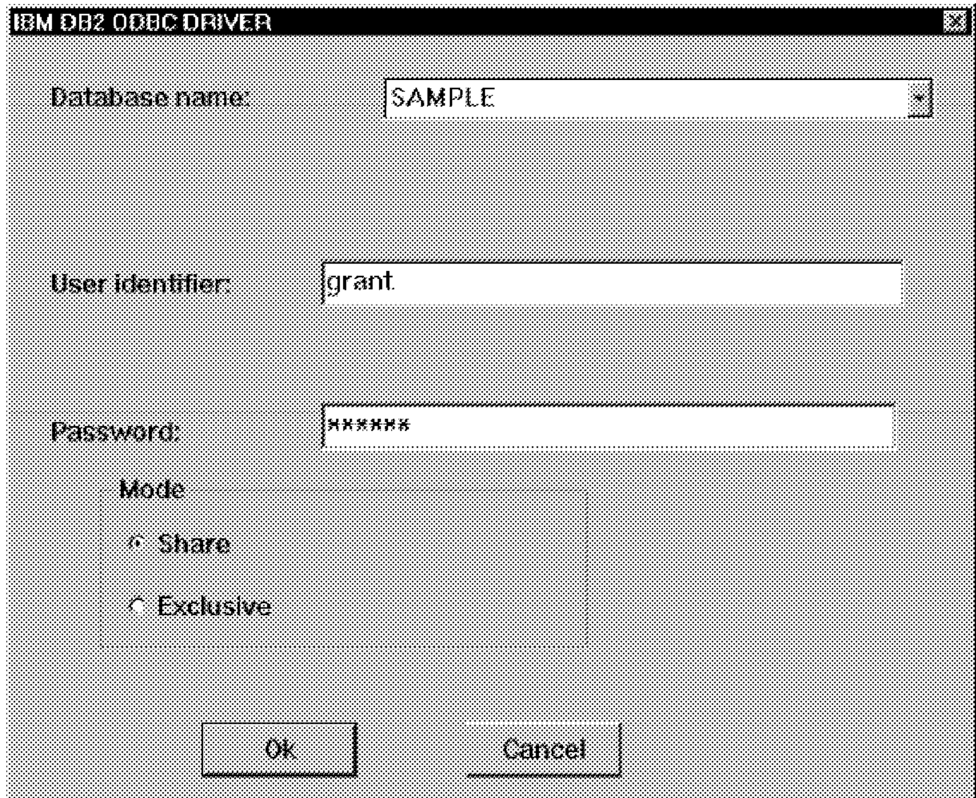


Figure 197. Unsuccessful Connection Attempt to DB2 Source Database

We diagnose this problem immediately from the SQL1403N error message which indicates that the username and/or password supplied is incorrect, but let's examine the ODBC trace from this example to find more information.

```
SQLAllocEnv( phEnv=&73068 ) -----(1)

SQLAllocEnv( phEnv=1 )
    ---> SQL_SUCCESS   Time elapsed - +0.000000E+000 seconds -----(2)

SQLAllocConnect( hEnv=1, phDbc=&73064 )
  ----> Time elapsed - +1.000000E-002 seconds
SQLAllocConnect( phDbc=1 )
    ---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds

SQLGetInfo( hDbc=1, fInfoType=SQL_DRIVER_ODBC_VER, rgbInfoValue=&3ed58, cb
InfoValueMax=6, pcbInfoValue=&3ed56 )
  ----> Time elapsed - +0.000000E+000 seconds
SQLGetInfo( rgbInfoValue="02.10", pcbInfoValue=5 )
    ---> SQL_SUCCESS   Time elapsed - +2.000000E-002 seconds

SQLDriverConnect( hDbc=1, hwnd=10027236, szConnStrIn="DSN=SAMPLE;UID=grant;
DBQ=",
cbConnStrIn=-3, szConnStrOut=&3f12c, cbConnStrOutMax=256,
pcbConnStrOut=&3f3f6, fDriverCompletion=SQL_DRIVER_COMPLETE_REQUIRED )
  ----> Time elapsed - +0.000000E+000 seconds.-------(3)
SQLDriverConnect( )
    ---> SQL_ERROR   Time elapsed - +5.471800E+001 seconds. ------(4)

SQLError( hEnv=0, hDbc=1, hStmt=0, pszSqlState=&3ec5c, pfNativeError=&3ec58
pszErrorMsg=&3ec64, cbErrorMsgMax=511, pcbErrorMsg=&3ec56 )
  ----> Time elapsed - +0.000000E+000 seconds.------(5)

SQLError( pszSqlState="08004", pfNativeError=-1403, pszErrorMsg="[IBM]
[CLI Driver]
SQL1403N
The username and/or password supplied is incorrect.  SQLSTATE=08004
", pcbErrorMsg=97 )
    ---> SQL_SUCCESS   Time elapsed - +3.000000E-002 seconds --------(6)
```

1. Statement SQLAllocEnv is invoked with an address of a variable which, upon completion, will contain a handle to an environment;
2. Upon completion, SQLAllocEnv returned a handle to an environment (handle 1). This statement completed successfully. The return code is SQL_SUCCESS.
3. Statement SQLDriverConnect is invoked with a previously acquired connection handle and other parameters describing the data source name (SAMPLE is an alias of a database we are trying connect to; grant is the identification of the user, and no authorization data is provided. Note the Null Pointer instead of an authorization string.)

   Here we have additional information to diagnose this problem. The password was not supplied. An SQL error message of SQL1403N was received. It indicates that the username and/or password is incorrect. From the ODBC trace, we found that user did not specify a password at all during the connection attempt to the database.
4. Upon completion, SQLDriverConnect returned SQL_ERROR. This indicates a non-zero return code, thus indicating a failure of this function.
5. When SQL_ERROR is returned as a value of the return code, it is the responsibility of the application developer to invoke the SQLError function to learn more about the error reported. Up to this moment, we only know that the last function did not complete successfully. In this case, SQLError is invoked with the addresses of the variables which will contain further description(s) of the error.

6. Upon completion, an SQLError returned value of the SqlState=08004, SQLCODE (-1403), as well as a full description of this error. Note that SQL_SUCCESS is returned as the return code of the SQLError function. It only means that the SQLError function completed succesfully!

Failure was caused by not supplying an authorization string (password was not specified).

This was a simplified example which could possibly be diagnosed from the returned SQLCODE alone if the application was written to check the return code after each function call and interpret returned values correctly.

7.6.2 Other ODBC Diagnostic Parameters

This section lists the set of optional keywords that you can specify in your db2cli.ini file which would be also helpful in verifying that application catches all error conditions and deals with them correctly.

The default value is underlined for the keyword (if it exists).

- **APPENDAPINAME** = 0 | 1

  Controls whether the DB2 CLI function (API) that generated an error appends that function′s name to an error message retrieved using SQLError(). Enclose the function name in curly braces {}.

  - 0 = Do not display the DB2 CLI function name (default).
  - 1 = Display the DB2 CLI function name.

  Example:

  ″[IBM][CLI Driver]″ CLIxxxx: < text >
  ″SQLSTATE=XXXXX {SQLGetData}″

  You can use APPENDAPNAME in the DB2CLI.INI file or in the ODBC.INI (x:WINNT35ODBC.INI or x:WINNT41ODBC.INI) file. This keyword is useful only for debugging.

- **POPUPMESSAGE** = 0 | 1

  Controls whether a message box is displayed when DB2 CLI generates an error that can be retrieved using SQLError(). This can be useful when debugging applications that do not report messages to users.

  - 0 = Do not display message box (default).
  - 1 = Display message box.

  You can use POPUPMESSAGE in the DB2CLI.INI file or in the ODBC.INI file. See also SQLSTATEFILTER and APPENDAPINAME.

- **SQLSTATEFILTER** = ″′S1C00′,′XXXXX′,...″

  Use with POPUPMESSAGE to prevent DB2 CLI from displaying errors associated with these states. You can use SQLSTATEFILTER in the DB2CLI.INI file or in the ODBC.INI file.

An ODBC trace (DB2 CLI Trace) is helpful when diagnosing ODBC-specific problems or when we suspect that something strange is happening with the ODBC driver rather than in the DB2 client or server software.

7.6.3 ODBC Trace Example

The following example shows the need for an ODBC trace to be performed. An application developer has reported that while running an application, unexpected results occurred. As the database administrator, you might first want to set the DIAGLEVEL to 4 (unless this is the current setting.) You will need to stop and restart the instance and re-execute the application so that you can view the contents of the DB2DIAG.LOG file for more information.

Unfortunately (for this example), the DB2DIAG.LOG file has not recorded any information that can be used to diagnose the application problem. As a first step, you might consider re-executing the application against a non-production database, such as SAMPLE. First, modify the application so that it only contains the function that is experiencing problems. Then execute the modified application against the SAMPLE database.

The smallest re-creatable scenario is an SQL SELECT statement from the ORG table in the SAMPLE database. The following shows all the rows contained in the ORG table:

Figure 198. SQL SELECT Statement from the ORG Table

When the ODBC application executes and tries to select all the rows from the ORG table, different results are obtained.

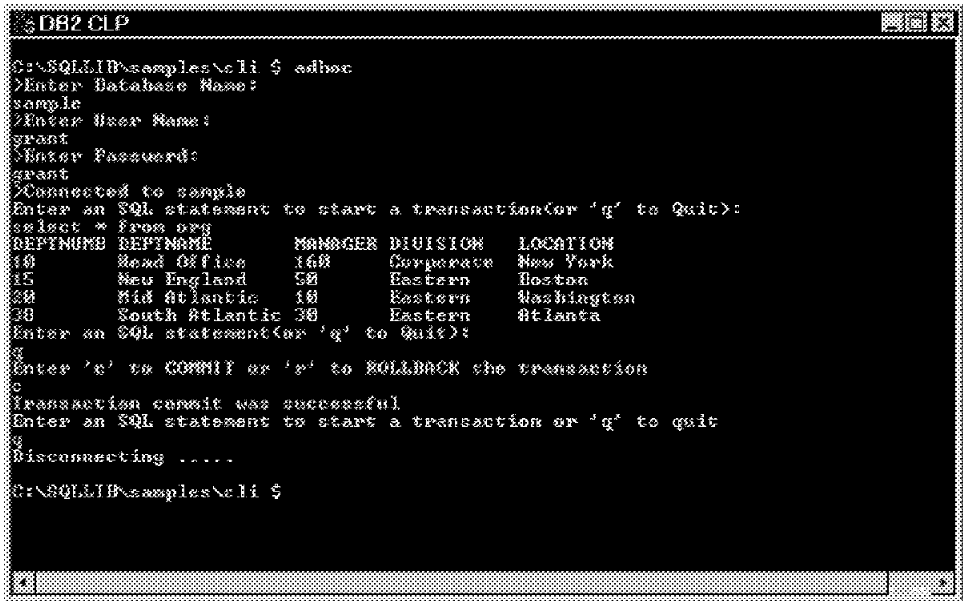


Figure 199. SQL SELECT Statement from the ODBC Application

When issuing the same SQL SELECT statement from the ODBC application, not all rows are returned, and there is no error message received.

Upon examining the contents of a formatted trace performed on the server, it was discovered that all rows were fetched from the ORG table and passed to the communication component of the DB2.

The next step is to perform the trace from the client workstation. This yielded the same results. All of the rows were selected from the application from the communication component of the DB2 Client Application Enabler (CAE).

The next option is to perform an ODBC trace. The following is a portion of the output from the ODBC trace:

:

```
SQLAllocEnv( phEnv=&3ff98 )
SQLAllocEnv( phEnv=1 )---> SQL_SUCCESS   Time elapsed - +4.000000E-002 seconds
SQLAllocConnect( hEnv=1, phDbc=&3ff94 )----> Time elapsed - +0.000000E+000 seconds
SQLAllocConnect( phDbc=1 )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLSetConnectOption( hDbc=1, fOption=SQL_AUTOCOMMIT, vParam=0 )
----> Time elapsed - +5.000000E-002 seconds
SQLSetConnectOption( )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLConnect( hDbc=1, szDSN="SAMPLE", cbDSN=-3, szUID="grant", cbUID=-3, szAuthStr="*****",
cbAuthStr=-3 )----> Time elapsed - +0.000000E+000 seconds
SQLConnect( )---> SQL_SUCCESS   Time elapsed - +1.372000E+000 seconds
SQLAllocStmt( hDbc=1, phStmt=&3fe6c )----> Time elapsed - +4.822900E+001 seconds
SQLAllocStmt( phStmt=1 )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLSetStmtOption( hStmt=1, fOption=SQL_MAX_ROWS, vParam=3-----------------------------(1)
----> Time elapsed - +0.000000E+000 seconds
SQLSetStmtOption( )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLExecDirect( hStmt=1, pszSqlStr="select deptname, location from org", cbSqlStr=-3 )
----> Time elapsed - +0.000000E+000 seconds
SQLExecDirect( )---> SQL_SUCCESS   Time elapsed - +2.610000E-001 seconds
SQLNumResultCols( hStmt=1, pcCol=&3fe6a )----> Time elapsed - +0.000000E+000 seconds
SQLNumResultCols( pcCol=2 )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLDescribeCol( hStmt=1, iCol=1, pszColName=&3fe30, cbColNameMax=32, pcbColName=&3fe2c,
pfSQLType=&3fe2e, pcbColDef=&3fc94, pibScale=&3fc92, pfNullable=NULL )
----> Time elapsed - +0.000000E+000 seconds
SQLDescribeCol( pszColName="DEPTNAME", pcbColName=8, pfSQLType=SQL_VARCHAR,
pcbColDef=14,pibScale=0)---> SQL_SUCCESS   Time elapsed - +7.000000E-002 seconds
SQLColAttributes( hStmt=1, iCol=1, fDescType=SQL_COLUMN_DISPLAY_SIZE, rgbDesc=NULL,
cbDescMax=0,pcbDesc=NULL, pfDesc=&3f864 )----> Time elapsed - +1.000000E-002 seconds
SQLColAttributes( pfDesc=14 )---> SQL_SUCCESS   Time elapsed - +2.000000E-002 seconds
```

```
SQLBindCol( hStmt=1, iCol=1, fCType=SQL_C_CHAR, rgbValue=&e53e50, cbValueMax=15,
pcbValue=&3fb00 )
----> Time elapsed - +0.000000E+000 seconds
SQLBindCol( )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLDescribeCol( hStmt=1, iCol=2, pszColName=&3fe30, cbColNameMax=32, pcbColName=&3fe2c,
pfSQLType=&3fe2e, pcbColDef=&3fc94, pibScale=&3fc92, pfNullable=NULL )
----> Time elapsed - +1.000000E-002 seconds
SQLDescribeCol( pszColName="LOCATION", pcbColName=8, pfSQLType=SQL_VARCHAR, pcbColDef=13,
pibScale=0 )---> SQL_SUCCESS   Time elapsed - +2.000000E-002 seconds
SQLColAttributes( hStmt=1, iCol=2, fDescType=SQL_COLUMN_DISPLAY_SIZE, rgbDesc=NULL,
cbDescMax=0, pcbDesc=NULL, pfDesc=&3f864 )----> Time elapsed - +0.000000E+000 seconds
SQLColAttributes( pfDesc=13 )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLBindCol( hStmt=1, iCol=2, fCType=SQL_C_CHAR, rgbValue=&e53e70, cbValueMax=14,
pcbValue=&3fb04 )
  ----> Time elapsed - +0.000000E+000 seconds
SQLBindCol( )---> SQL_SUCCESS   Time elapsed - +2.000000E-002 seconds
SQLFetch( hStmt=1 )----> Time elapsed - +0.000000E+000 seconds
SQLFetch( )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
( iCol=1, fCType=SQL_C_CHAR, rgbValue="Head Office", pcbValue=11 )
( iCol=2, fCType=SQL_C_CHAR, rgbValue="New York", pcbValue=8 )
SQLFetch( hStmt=1 )----> Time elapsed - +2.000000E-002 seconds
SQLFetch( )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
( iCol=1, fCType=SQL_C_CHAR, rgbValue="New England", pcbValue=11 )
( iCol=2, fCType=SQL_C_CHAR, rgbValue="Boston", pcbValue=6 )
SQLFetch( hStmt=1 )----> Time elapsed - +2.000000E-002 seconds
SQLFetch( )---> SQL_SUCCESS   Time elapsed - +0.000000E+000 seconds
( iCol=1, fCType=SQL_C_CHAR, rgbValue="Mid Atlantic", pcbValue=12 )
( iCol=2, fCType=SQL_C_CHAR, rgbValue="Washington", pcbValue=10 )
SQLFetch( hStmt=1 )----> Time elapsed - +3.000000E-002 seconds
SQLFetch( )---> SQL_SUCCESS   Time elapsed - +0.000000E+000 seconds
( iCol=1, fCType=SQL_C_CHAR, rgbValue="South Atlantic", pcbValue=14 )
( iCol=2, fCType=SQL_C_CHAR, rgbValue="Atlanta", pcbValue=7 )
SQLFetch( hStmt=1 )----> Time elapsed - +1.999999E-002 seconds
SQLFetch( )---> SQL_NO_DATA_FOUND   Time elapsed - +1.000000E-002 seconds----------------(2)
SQLFreeStmt( hStmt=1, fOption=SQL_DROP )----> Time elapsed - +0.000000E+000 seconds
SQLFreeStmt( )---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds
SQLDisconnect( hDbc=1 )
  ----> Time elapsed - +1.883000E+000 seconds
SQLDisconnect( )
    ---> SQL_SUCCESS   Time elapsed - +1.000000E-002 seconds

SQLFreeConnect( hDbc=1 )
  ----> Time elapsed - +1.000000E-002 seconds
SQLFreeConnect( )
    ---> SQL_SUCCESS   Time elapsed - +0.000000E+000 seconds

SQLFreeEnv( hEnv=1 )
  ----> Time elapsed - +0.000000E+000 seconds
SQLFreeEnv( )
    ---> SQL_SUCCESS   Time elapsed - +0.000000E+000 seconds
```

The following remarks explain some of the detail from the ODBC trace information:

1. Reviewing the ODBC trace, we found the function SQLSetStmtOption was invoked with a parameter limiting the number of output rows. Setting the parameter SQL_MAX_ROWS to a value = 3 causes an end-of-data signal when the number of rows fetched is greater than SQL_MAX_ROWS. In our example, the fourth row would generate an error.

2. The fourth call to the SQLFetch() function returned an SQL_NO_DATA_FOUND return code. From the analysis performed in DB2 on both the client and server, we determined that DB2 returned all eight rows of data. This was the reason for performing the trace within ODBC.

7.7 Handling Installation Errors

This section discusses some of the common errors you may encounter during the installation of DB2 for Windows NT. First we look at the registry keys that are being added or updated by the installtion process and how they get reflected in the system environment variables. It is important for problem determination that you understand which registry keys and environment variables get created so you can make sure they contain the right values, and if not, be able to correct them.

The system keeps a log (C:DB2LOGDB2.LOG) that tracks installation and uninstall activities and records information on any errors encountered during product installation or uninstall. The log is stored in the DB2LOG directory. If this directory exists, the Setup program uses it. If not, the Setup program creates the directory on the drive containing the Windows directory. If space constraints or other errors prevent the log from being opened or written to, the installation or uninstall continues.

Note that during the execution of the Setup program, you should make sure the C:DB2LOGDB2.LOG file exists and is being updated. Once the Setup program has completed, it is important that you review the contents of the C:DB2LOGDB2.LOG file for any possible installation error messages.

7.7.1 Registry Updates

Windows NT maintains information about the configuration of your machine, and the settings of your applications in a hierarchical database of settings known as the *registry*.

7.7.1.1 Updating Registry Keys for DB2

The DB2 Setup program updates the Windows NT registry as follows:

1. Adds or updates environment variables in the following subkey:

```
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetControlSessionManagerEnvironment
```

For the Environment subkey, the following value entries are added:

Table 22. Registry Key Updates for DB2

Name	Data Type	Value for DB2 Server product
BOOKSHELF	REG_SZ	C:\SQLLIB\BOOK
DB2INSTANCE	REG_SZ	DB2
DB2PATH	REG_SZ	C:\SQLLIB
HELP	REG_SZ	C:\SQLLIB\HELP
Include	REG_SZ	C:\SQLLIB\INCLUDE
Lib	REG_SZ	C:\SQLLIB\LIB
Title	REG_SZ	DB2 Server
Version	REG_DWORD	0x2

See 7.7.3, “System Environment Variables” on page 308 for a description of the environment variables that are added or updated.

1. Adds or updates per-application paths in the following key:

```
HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionAppsPaths
```

1. Keeps track of shared DLLs on the system by updating a usage counter in the registry under the following key:

```
HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionSharedDLLs
```

1. By updating the following key, adds the DB2 uninstall program to the list of applications that can be uninstalled:

```
HKEY_LOCAL_MACHINESOFTWAREMicrosoftWindowsCurrentVersionUninstall
```

1. Adds or updates the following keys or subkeys:

```
HKEY_LOCAL_MACHINESOFTWAREIBM
   \DB2 (see Note 1)
      \Components (see Note 2)
         \Client Server Support
         \Command Language Processor
         \Common Building Blocks
         \Common Engine Parts
         \Database Director
         \Engine Parts
         \IBM DB2 ODBC Driver
         \<Language> Conversion Support
         \Performance Monitor
         \SDK Sample Programs
         \Server Parts
         \Visual Explain
      \DB2 Server\CurrentVersion (see Note 3)
         \Components (see Note 4)
      \DB2 Single-User\CurrentVersion (see Note 3)
         \Components (see Note 4)
      \DB2 Software Developer’s Kit\CurrentVersion (see Note 3)
         \Components (see Note 4)
      \DDCS Multi-User Gateway\CurrentVersion (see Note 3)
         \Components (see Note 4)
```

The following notes explain the addition or updates of the keys and subkeys just listed:

1. The value name, DB2 Path Name, for the DB2 key is set to the path in which the DB2 products are installed:

   ```
   DB2 Path Name : REG_SZ : C:SQLLIB
   ```

2. The list and number of subkeys for Components will vary depending on the DB2 products that are installed on your machine. Each subkey of the Components key has a value name, RefCount, set to the number of times that the corresponding component has been installed. For example, the Database Director subkey will have the following value entry:

   ```
   RefCount : REG_DWORD : 0x1
   ```

3. The following value entries are added to the CurrentVersion key for each product installed:

Table 23. Table of Added Items to CurrentVersion Key for DB2 Products

Name	Data Type	Value for DB2 Server product
Description	REG_SZ	DB2 Server
Install Date	REG_SZ	05/31/96
Last Modified	REG_SZ	05/31/96
Modification	REG_DWORD	0x1
Release	REG_DWORD	0x1
Service Level	REG_SZ	
Title	REG_SZ	DB2 Server
Version	REG_DWORD	0x2

4. The following value entries are added to the Components key for each product installed:

Table 24. Table of Added Items to Components Key for DB2 Products

Name	Data Type
Client Server Support	REG_SZ
Command Language Processor	REG_SZ
Common Building Blocks	REG_SZ
Common Engine Parts	REG_SZ
Engine Parts	REG_SZ
Server Parts	REG_SZ

7.7.1.2 Viewing the Contents of the Registry Keys

The contents of the registry keys can be viewed by using the Registry Editor. In general, do not use the registry to modify a value that can be modified otherwise. Most of the registry key values are set by programs and should not be modified manually, except in some specific cases.

To visualize the values of your registry keys, you can run the Registry Editor as follows:

1. From the Start menu, choose **Run**.
2. On the command line, type `REGEDT32` and press **Enter**.
3. Click on **Window** and choose **HKEY_LOCAL_MACHINE.** Maximize that window and you see a local machine dialog box.

7.7.1.3 Saving the Contents of a Registry Subtree

You can place the contents of an entire registry subtree into a file. This may be useful if you have to supply the IBM support staff with your current environment settings.

1. From the Register Editor, click on **Registry** and select the **Save Subtree As** option as shown in the Figure 201 on page 305. In this example, we are asking for the contents of the following subtree to be saved in a file:

   `HKEY_LOCAL_MACHINESOFTWAREIBMDB2DB2 ServerCurrentVersion`

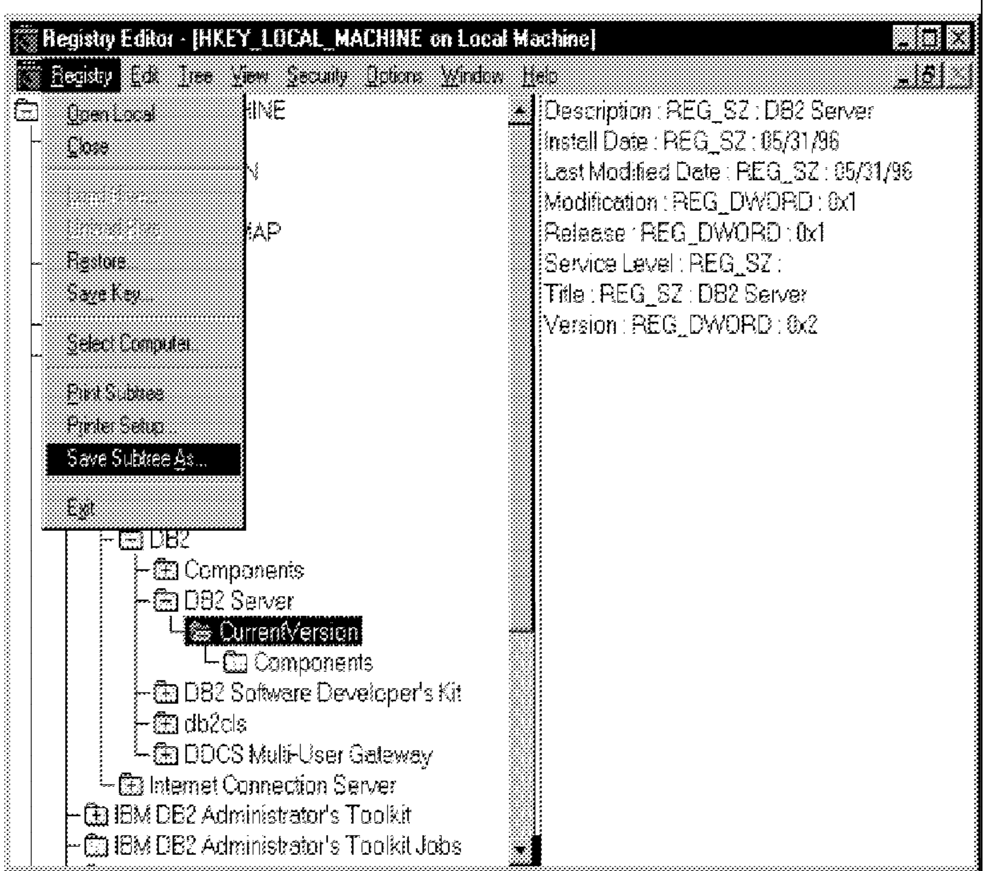


Figure 200. Placing Contents of Registry Subtree into a File

2. In the File name line of the Save As dialog box, type the name of the file into which you want to write the contents of the registry subtree and press **Enter**. In our example, we are saving the contents of the subtree in a file named REGISTRY.CRV as shown in Figure 201 on page 305

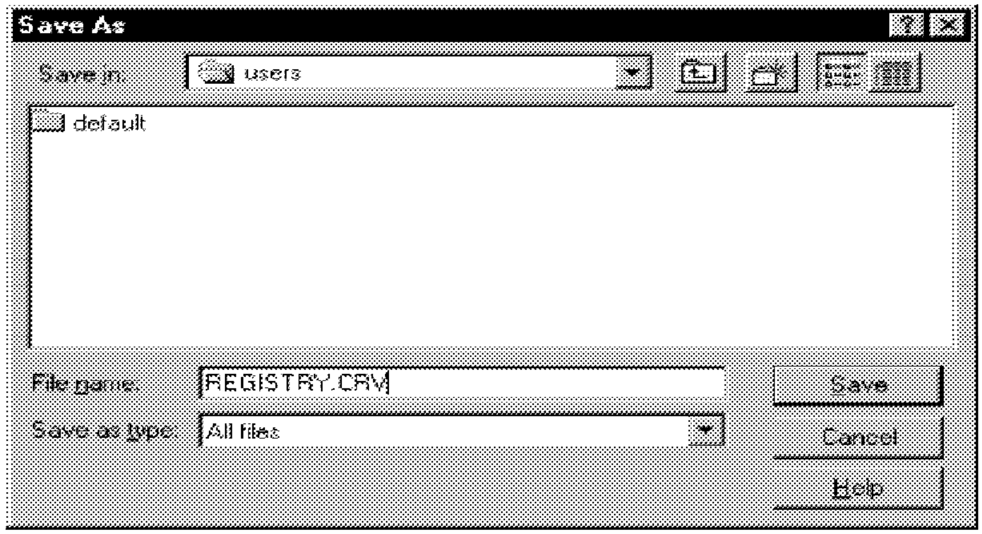


Figure 201. Saving the Contents of Registry Subtree in REGISTRY.CRV File

3. The contents of the file obtained for the subkey will be similar to the following where each entry describes an entry value for the different subkeys as shown in Figure 202 on page 306.

```
Key Name:          SOFTWAREIBMDB2DB2 ServerCurrentVersion
Class Name:        REG_SZ
Last Write Time:   31/05/96 - 17:19
Value 0
  Name:            Description
  Type:            REG_SZ
  Data:            DB2 Server
Value 1
  Name:            Install Date
  Type:            REG_SZ
  Data:            05/31/96
Value 2
  Name:            Last Modified Date
  Type:            REG_SZ
  Data:            05/31/96
Value 3
  Name:            Modification
  Type:            REG_DWORD
  Data:            0x1
Value 4
  Name:            Release
  Type:            REG_DWORD
  Data:            0x1
Value 5
  Name:            Service Level
  Type:            REG_SZ
  Data:
Value 6
  Name:            Title
  Type:            REG_SZ
  Data:            DB2 Server
Value 7
  Name:            Version
  Type:            REG_DWORD
  Data:            0x2

Key Name:          SOFTWARE\IBM\DB2\DB2 Server\CurrentVersion\Components
Class Name:        REG_SZ
Last Write Time:   31/05/96 - 17:19
Value 0
  Name:            Client Server Support
  Type:            REG_SZ
  Data:
Value 1
  Name:            Command Language Processor
  Type:            REG_SZ
  Data:
Value 2
  Name:            Common Building Blocks
  Type:            REG_SZ
  Data:

Value 3
  Name:            Common Engine Parts
  Type:            REG_SZ
  Data:

Value 4
  Name:            Engine Parts
  Type:            REG_SZ
  Data:

Value 5
  Name:            Server Parts
  Type:            REG_SZ
  Data:
```

Figure 202. Format of the File (REGISTRY.CRV) of Contents Registry Subtree

7.7.2 Running DB2 as Windows NT Service

DB2 for Windows NT provides the DB2START and DB2STOP commands to start and stop, respectively, a DB2 instance as a user program. This means that the DB2 instance will be active as long as the user is logged on.

Alternatively, you can start a DB2 instance as a Windows NT service by using NET START <instance_name>**.** In the same way, you can stop the DB2 instance service by issuing the command NET STOP <instance_name>.

DB2 as a NT Service

Note that using DB2START and DB2STOP to start and stop an instance is only practical in a test or development environment. A production environment must have DB2 for Windows NT start as a Windows NT service because a user program terminates when that user logs off. If you want the DB2 instance to remain active as a daemon process, you must start DB2 as a service. Moreover, log entries related to starting/stopping the DB2 Server will be logged in the Windows NT Event Log only if DB2 has been started as an Windows NT service.

DB2ICRT creates a registry subkey for each DB2 instance. For example, the DB2 instance has been created under the name of MYDB2INST. Therefore the following subkey has been added to the registry:

HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesMYDB2INST

For the MYDB2INST subkey, the following value entries are added:

Table 25. List of Value Entries for DB2 Subkey per DB2 Instance

Name	Data Type	Value for DB2 Server product
DisplayName	REG_SZ	DB2 - MYDB2INST
ImagePath	REG_EXPAND_SZ	C:\SQLLIB\bin\db2syscs.exe

That registry subkey tells the Service Manager to start a program called DB2SYSCS.EXE that will pass as an argument the name of the instance it is starting. The only messages that DB2 accepts are STOP and INTERROGATE messages. DB2 does not accept PAUSE or RESUME messages.

7.7.2.1 Starting a DB2 Instance Service

You can start a DB2 instance as a Windows NT service in one of two ways:

1. From the Services dialog in the Control Panel, find the entry for the instance you want and press the **Start** button.

   DB2 creates the entry for the instance so that manual starting is required. You can change the entry so that the instance is started automatically at system startup.

2. Use the command NET START <instance_name> from any command window.

Note: When you start the database manager, you must also start the DB2 Security Service. See 7.7.2.3, "Starting a DB2 Security Server" on page 308 for more information.

7.7.2.2 Stopping a DB2 Instance Service

You can stop a DB2 instance in one of three ways:

1. From the services dialog in the Control Panel, find the entry for the instance you want and press the **Stop** button.
2. Use the command NET STOP <instance_name> from any command window.
3. Issue DB2STOP from a window where the DB2INSTANCE environment variable is the same as the instance you want to stop. All START and STOP completion messages are logged to the Windows NT Application Log. You can use the Windows NT Event Viewer to look at these messages (See 6.2.1, "Examining Events in Event Viewer" on page 253).

7.7.2.3 Starting a DB2 Security Server

The APIs used to validate a user name and password are restricted APIs. DB2 for Windows NT provides a Windows NT Service called db2sec.exe. This program needs to be started on any machine (client or server) on which a user name or password must be validated. For cient machines, this is necessary only if any connected instance requires Client Authentication. During installation of a DB2 product, this service is registered with Windows NT. It is removed during the uninstall process.

By default, you must start this service manually at system startup. The service is known as DB2 Security Server. You can start it manually by selecting the **Services icon** from the Control Panel, or you can enter the following command on the command line:

```
NET START DB2NTSECSERVER
```

You can also stop it manually by selecting the **Services icon**, or you can enter the following command on the command line:

```
NET STOP DB2NTSECSERVER
```

If you want it to start automatically at system startup, select the **Services icon** from the Control Panel to change the service startup options.

7.7.3 System Environment Variables

The environment variables control your database environment. For example, the DB2INSTANCE variable indicates which DB2 instance you are currently using.

During the installation of the first DB2 product, the installation program adds or updates the following System Environment Variables:

- **DB2PATH**: The value for DB2PATH is set to the install path.
- **PATH**: The $DB2PATHBIN and $DB2PATHFUNCTION are added to the existing path.
- **INCLUDE**: The $DB2PATHINCLUDE is added to the INCLUDE variable.
- **LIB**: The $DB2PATHLIB is added to the LIB variable.
- **DB2INSTANCE**: The value for DB2INSTANCE is set to DB2.
- **BOOKSHELF**: The value for BOOKSHELF is set to $DB2PATHBOOK.
- **HELP**: The value for HELP is set to $DB2PATHHELP.

7.7.3.1 Viewing the Setting of a Variable

You can determine the setting of environment variables by using the set command. For example, to check the value of the DB2COMM environment variable, enter:

```
set DB2COMM
```

or:

```
echo %DB2COMM%
```

7.7.3.2 Changing the Value of a Variable

You can set environment variables as needed, or alter the values set by setup as follows:

1. Double-click on the **Control Panel icon.**
2. Double-click on the **System icon**.
3. In the System Control Panel, in the Environment tab, do the following:
   a. If the environment variable does not exist, do the following to add it:
      - Select any environment variable.
      - Change the name in the Variable box to environment variable to be added. (For example: DB2COMM)
      - Change the value in the Value box as required. (For example: **APPC,TCPIP**)
      - Select **Set**.

   b. If an environment variable already exists, do the following to change for a new value:
      - Select the environment variable to be updated. (For example: **DB2COMM**)
      - Change the value in the Value box as required. (For example: **APPC,TCPIP**)
      - Select Set.

You can also update environment variables by using a command prompt (using the set command) or by running command files that you create for this purpose. Note that in this case, the new setting of the environment variable will only remain as long as your Windows session is active. For example:

```
set db2instance=inst2
```

This command makes inst2 the active instance for the Windows session that issues it.

```
set db2comm=tcpip
```

This command makes tcpip active for the Windows session that issues it.

7.7.4 Common Installation Errors

If an error occurs while you are running the Setup program, you should first read the error message text to find the probable cause and possible solution for the problem. You will find additional information in the C:DB2LOGDB2.LOG file.

We strongly advise you to set the DIAGLEVEL configuration parameter to 4 whenever you are performing some initial DB2 settings or when you are installing any additional DB2 software. This way, you will be able to troubleshoot any installation problems should any incident occur.

The Setup program copies some files from the install media to your install directory and then tries to copy them to the Windows NT directory locations. Nothing else should be running on the Windows NT machine when you do the DB2 for NT installation because files may be open and locked by other processes, thereby preventing the Setup program from updating them.

On Windows NT, services that are running can also lock shared DLLs and cause some installation errors. Try stopping all services that are running, and ensure once again that no other applications are running (including the Control Panel). In particular, we have found that the OLE32 service locks up the ODBC DLLs.

The following table lists installation errors that you may encounter and the action you should take to resolve the problem.

Table 26. DB2 for NT Common Installation Errors

ERROR	ACTION
Insufficient authority to run the Setup program or the user name is not valid.	1. Logon with a user name that has administrator authority on the local machine and follow the naming rules described in Naming Rules. 2. Reinstall the product.
Windows NT program groups or short-cuts cannot be created.	To create program groups or shortcuts: 1. From the task bar, select **START**, then **SETTINGS**, then **TASK BAR**. The TASKBAR PROPERTIES window appears. 2. Select the **START MENU PROGRAMS** tab. 3. Select **ADD**. 4. In the COMMAND LINE field, type the path to the program's executable file. 5. Select **NEXT**. 6. Select **NEXT**. 7. Select an existing folder or create a new folder. 8. Select **NEXT**, then **FINISH**, then **OK**. You do not need to re-install the product.
Windows NT program groups or items cannot be created.	To create a new program group, do the following: 1. Create a common program group: a. Go to the PROGRAM MANAGER FILE item. b. Select **NEW.** c. Select **COMMON PROGRAM GROUP**. d. Select **OK**. e. Specify the properties of the group to be made. 2. Create a program item in that group: a. Go to the PROGRAM MANAGER FILE item. b. Select **NEW**. c. Select **PROGRAM ITEM**. d. Specify the properties of the new item to be made. You do not need to reinstall the product.
DB2 Security Service cannot be registered.	To register the DB2 Security Service, run `DB2REGSC`, specifying "I" for install, as follows: `$DB2PATH\bin\db2regsc I` You do not need to reinstall the product.

ERROR	ACTION
Environment variables cannot be added or updated.	In most cases, the Setup program will try to recover from this error and continue processing. The error message indicates if the install program continues or terminates. 1. If it continues, you may need to correct some values. Correct the environment variables using the System program group within the PROGRAM MANAGER CONTROL PANEL. You do not need to reinstall the product. 2. If it does not continue, you need to correct the cause of the error before reinstalling.
Process environment variables may not be set.	This error may cause other install steps to fail. The error message gives the name of the environment variable that needs to be updated. If the Setup program gives errors, update the process environment variable using the `SET` command. Then uninstall and reinstall the product.
Default instance cannot be created, or the service for default instance cannot be registered.	If the default instance, "DB2", could not be created, run `DB2ICRT` to create it, as follows: `$DB2PATH\bin\db2icrt DB2` You do not need to reinstall the product.
Key cannot be added to the registry.	In most cases, the Setup program will try to recover from this error and continue processing. The error message includes the following information: 1. The key, subkey, and value names. 2. The operation that was executing at the time of error, for example, `OPEN`, `QUERY`, or `SET`. 3. The return value from the operation. 4. Whether the installation process continues or terminates. If the Setup program continues after an error, you do not need to reinstall, but you may need to correct the software key in the registry. If it does not continue, you may need to correct the cause of the error before reinstalling.
Microsoft Visual C++ run-time library, MSVCRT20.DLL, cannot be copied to the Windows system directory **(see Note 1)**.	1. Copy the Microsoft Visual C++ Version 2.1 run-time library, MSVCRT20.DLL, from the first diskette or the "DISK1" directory within the path you are installing from, to the Windows system directory that is accessible by the DB2 Setup program. 2. Reinstall the product.
The ODBC Driver Manager, ODBC32.DLL, cannot be copied to the Windows system directory **(see Note 2)**	1. Copy the ODBC32.DLL, from the first diskette or the "DISK1" directory within the path you are installing from, to the Windows system directory that is accessible by the DB2 Setup program. 2. Reinstall the product.

ERROR	ACTION
The HELP file cannot be loaded. The environment variables are set properly in NT (System properties, Environment). However, `%help%` does not seem to pick up the value set in SystemProperties/Environment. The other variables, like `DB2INSTANCE`, and so forth are okay.	DB2 sets the system environment variables during install. However, on NT, the user environment variables take precedence. Check the following: 1. The user environment variable `HELP` is set to a value and does not also pick up the system settings. For instance, the user variable `help` should be set to something like `HELP=%help%;<path>` and not `HELP=<path>`.
After installing DB2/NT and rebooting the system, when trying to connect from a remote Win95 client, receive an error message indicating wrong user/password (using same as used on server).	Check the following: 1. The DB2/NT Security Server is running. for example, type `net start db2ntsecserver` from a Command Prompt. 2. The password is entered in the correct case. NT passwords are case-sensitive. 3. The password used has not expired. 4. The password does not have to be changed on first logon.
Each time you try to start a CLP window from the shortcut, you get an error message stating the environment variable is not set or something like that.	DB2CMD.EXE calls a batch program db2env.bat (which should be located in your sqllib\bin directory) and passes to it the PID of the background process that has been created. Then db2env.bat sets the environment variable `DB2CLP` to be this PID. Check the following: 1. The `DB2CMD` is working correctly from the command line. 2. If so, check the properties of the Shortcut. The working directory should be the %db2path%\bin directory, and you should ensure that db2env.bat is in that director.

Note:

1. The DB2 CD-ROM contains the Microsoft Visual C + + Version 2.1 run-time library, msvcrt20.dll. This library is located within the DISK1 directory in uncompressed format. If you have the same or an earlier version of this library on the installed disk, the Setup program copies this run-time library to the Windows system directory. If the copy fails, an informational message is displayed, and the likely reason is because the msvcrt20.dll library is being used by another process and cannot be overwritten.
2. If you have the same or an earlier version of this library on the installed disk, the Setup program copies this run-time library to the Windows system directory. If the copy fails, an informational message is displayed, and the likely reason is because the odbc32.dll library is being used by another process and cannot be overwritten. You can use Windows NT Explorer to check the version level information for the DLL in question and compare it with the one located in the c:sqllibbin directory.

7.8 Information to Collect for IBM Support

Once you have determined that you have to call IBM for support, you should gather and have ready some necessary information that will help IBM to investigate further your problem.

7.8.1 What to Collect

The necessary information you gather may be staged in two phases:

1. An initial set of files and information that will allow the IBM Staff to start investigating your problem, and hopefully it will be sufficient to find a fix to your problem.

   - A detailed description of the problem situation, which will include all the SQL codes and any Reason codes (if applicable). For example, the SQL error code -903 has several possible Reason codes as shown below:

   > **SQL0903N** COMMIT statement failed, transaction rolled back. Reason code: "<reason-code>"
   > **Cause:** One or more of the servers participating in the current unit of work was unable to prepare the database to be committed. The COMMIT statement has failed and the transaction has been rolled back.
   > **Possible reason codes are:**
   > 01 - A connection to one of the databases participating in the unit of work was lost.
   > 02 - One of the databases participating in the unit of work was accessed, but unable to prepare to commit.
   > **Action:** If a connection to a database was lost, reestablish the connection. If the failure was not connection related, reference the error diagnostic logs on the remote system to determine the nature of the failure and what action might be required. Rerun the application.
   > **sqlcode:** -903
   > **sqlstate:** 40504

   - The **DB2DIAG.LOG** file
   - All the dump files (.dmp) if any
   - All the trap files (.trp) if any
   - The Database Manager configuration information
   - The database configuration information (if applicable)
   - The DB2 Service Level you have installed on your machine. This can be obtained by looking at the Service Level value entry of the following registry subkey:

   `HKEY_LOCAL_MACHINESOFTWAREIBMDB2DB2 ServerCurrentVersion`

2. An additional set of files and information which might be necessary in case the first set of information you have provided IBM is not sufficient to fix your problem. This set includes:

   - The Windows NT Application Event Log(either in a binary format file (.evt) or in a text format file or in a comma-delimited text format file (.txt)
   - The database directory information.
   - The node directory information (if applicable.)
   - The DCS directory information, if DDCS is installed on this workstation.
   - The Nodelock file. This file contains information about the DB2 license of this workstation.
   - Network Configuration information about your workstation. You can find this information by going to the Control Panel and clicking on the **Networks icon.**

     `HKEY_LOCAL_MACHINEControlSet001ServicesTcpipParameters`

   - The Services file. This file provides TCP/IP connection information.

- A dump of the Windows NT Registry subkey, HKEY_LOCAL_MACHINE, for your machine.

7.8.2 History of Reported Problems

Technical support requires a certain amount of record keeping. In particular, you should establish and maintain problem-related records.

IBM recommends that you maintain a history of reported problems and actions taken to resolve or identify them. This information can help you isolate some problems and anticipate or avoid others.

In particular, you should:

1. Record the information you gather related to the problem.

   The information includes:

   - The problem symptoms
   - The problem number (or Problem Management Record (PMR) number)
   - The problem description
   - Configuration of the hardware and software
   - The latest Program Temporary Fix (PTF) or service level installed on each user′s system
   - The message displayed and the results of the corrective action
   - Should you be directed to contact IBM to resolve your problem, the information helps IBM respond to you in a prompt and effective manner. Information should be sent to IBM for analysis only upon direction from the IBM Support Center. Additional information is provided by Contacting IBM for Support and Maintenance.

2. Record the resolution of problems for later reference.

Appendix A. Special Notices

This publication is intended to help database administrators manage a DB2 environment running on Windows NT. The information in this publication is not intended as the specification of any programming interfaces that are provided by DB2 for Windows NT. See the PUBLICATIONS section of the IBM Programming Announcement for DB2 for Windows NT for more information about what publications are considered to be product documentation.

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

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

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

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

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

ADSTAR®	AIX®
AS/400®	AT®
BookManager®	DATABASE 2
DB2®	Distributed Relational Database Architecture
DRDA®	FFST
First Failure Support Technology	IBM®
OS/2®	OS/400®
PowerPC®	VisualAge®
VTAM®	

The following terms are trademarks of other companies:

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Pentium, MMX ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

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

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

Appendix B. Related Publications

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

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 319.

- *DB2 Planning Guide for Database Administrators, SG24-2523*
- *Diagnostic Tips and Techniques for DB2 Common Server, SG24-4759*
- *DB2 Certification Guide for Common Servers, SC09-2465-00 or ISBN 0-13-727413-0*

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

B.3 Other Publications

These publications are also relevant as further information sources:

- *DB2 for NT V2.1 Planning Guide,*S33H-0314
- *DB2 Command Reference for Common Server V2*, S20H-4645
- *DB2 SQL Reference for Common Server V2*, S20H-4665
- *DB2 API Reference for Common Server V2*, S20H-4984
- *DB2 Administration Guide for Common Server V2,* S20H-4580

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

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

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

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

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

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

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

  For a list of product area specialists in the ITSO:

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

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

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

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

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

How Customers Can Get ITSO Redbooks

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

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

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

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

AIX	Advanced Interactive Executive
ADSM	ADSTAR Distributed Storage Manager
API	Application Programming Interface
APPC	Advanced Program to Program Communications
AR	Application Requestor
AS	Application Server
ASCII	American National Standard Code for Information Interchange
BDC	Backup Domain Controller
BLOB	Binary Large Object
CAE	Client Application Enabler
CLI	Call Level Interface
CLOB	Character Large Object
CLP	Command Line Processor
CSV	Comma Separated Variable
DB2	IBM Database 2 for Common Server
DBCLOB	Double-Byte Character Large Object
DDCS	Distributed Data Connection Services
DMS	Database Managed Space
DRDA	Distributed Relational Database Architecture
DUOW	Distributed Unit of Work
FFDC	First Failure Data Capture
IBM	International Business Machines Corporation
IPX/SPX	Internetwork Packet Exchange/Sequenced Packet Exchange
ITSO	International Technical Support Organization
LAN	Local Area Network
LAPS	LAN Adapter Protocol Support
LOB	Large Object
MPTS	Multi-Protocol Transport Services
MVS	Multiple Virtual Storage
NCP	Network Control Program
NetBEUI	NetBIOS Extended User Interface
NetBIOS	Network Basic Input/Output System
ODBC	Open Database Connectivity
PDC	Primary Domain Controller
PROFS	Professional Office System
RDBMS	Relational Database Management System
SAG	SQL Access Group
SAM	Security Access Manager
SCO	Santa Cruz Operations
SDK	Software Developer's Kit
SINIX	Siemens Nixdorf
SQL	Structured Query Language
SQLCA	SQL Communications Area
SQLDA	SQL Database Area
SMS	System Managed Space
SNA	Systems Network Architecture
TCP/IP	Transmission Control Protocol/Internet Protocol
UNIX	An Operating System Developed by Bell Labs
VM	Virtual Machine
VSE	Virtual Storage Extended
VTAM	Virtual Telecommunications Access Method
WAN	Wide Area Network

Index

ITSO Redbook Evaluation

DB2 Meets Windows NT
SG24-4893-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____________

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:
____

____

____

____

What other redbooks would you like to see published?
____

____

____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**
____

____

____

____

____

Printed in U.S.A.

SG24-4893-00