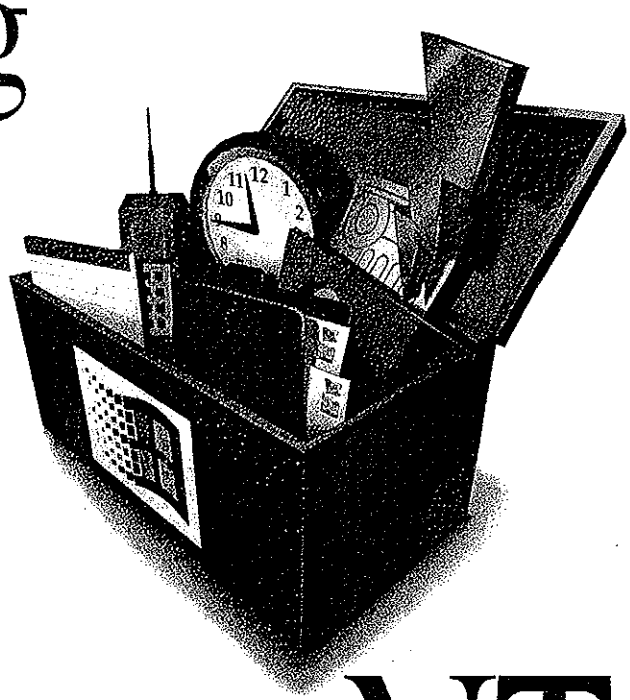
EXHIBIT 1008

Windows NT™ Networking Guide



The information you need to become an expert on Windows NT!

Microsoft® WINDOWS NT RESOURCE KIT

For Windows NT Workstation and Windows NT Server Version 3.5

Microsoft Press

Windows NT™ Networking Guide

Microsoft®

WINDOWS NT RESOURCE KIT

For Windows NT Workstation and Windows NT Server Version 3.5

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data
Windows NT networking guide : for Windows NT workstation and Windows NT server version 3.5 / by Microsoft Corporation.
p. cm.
Includes index.
ISBN 1-55615-656-1
1. Computer networks. 2. Microsoft Windows NT. I. Microsoft Corporation.
TK5105.5.M548 1995
005.7'13--dc20 94-45565
CIP

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QMQM 0 9 8 7 6 5

Distributed to the book trade in Canada by Macmillan of Canada, a division of Canada Publishing Corporation.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office. Or contact Microsoft Press International directly at fax number (206) 936-7329.

This book is dedicated to the system administrators who keep us all connected.
We hope this book makes your job easier.

Contributors to this book include the following:

Technical Writers:
Chris Dragich, Jeff Howard, Sharon Kay,
Doralee Moynihan, Annie Pearson, and Jim Purcell

Technical Consultants:
J. Allard, Pradeep Bahl, Sudheer Dhulipalla, James Gilroy, Tom Hazel, Steve Heaney,
Jan Keller, Leslie Link, James McDaniel, Kerry Schwartz, and Cliff Van Dyke

Technical Editor:
Sonia Marie Moore

Project Lead:
Peggy Etchevers

Indexer:
Jane Dow

Production Team:
Karye Cattrell, Yong Ok Chung, and Cathy Pfarr

Graphic Designer:
Sue Wyble

Graphic Artists:
Gwen Grey, Elizabeth Read, and Stephen Winard

Contents

PART V Appendixes

Figures and Tables

Figures

Tables

Introduction

Welcome to the *Microsoft Windows NT Resource Kit Volume 2: Windows NT Networking Guide.*

The *Windows NT Resource Kit* also includes the following volumes:

- *Volume 1: Windows NT Resource Guide*, which provides information to help administrators better understand how to install, manage, and integrate Windows NT™ in a network or multiuser environment.
- *Volume 3: Windows NT Messages*, which provides information on local and remote debugging and on interpreting error messages.
- *Volume 4: Optimizing Windows NT*, which provides a step-by-step approach to understanding all the basic performance management techniques.

The *Windows NT Networking Guide* is designed for people who are, or who want to become, expert users of Microsoft® Windows NT Workstation and Microsoft Windows NT Server networking features. The *Windows NT Networking Guide* presents detailed, easy-to-read technical information to help you better manage how Windows NT is used at your site. It contains specific networking information for system administrators who are responsible for installing, managing, and integrating Windows NT in both small and large networks.

The *Windows NT Networking Guide* is a technical supplement to the documentation included as part of the Windows NT product and does not replace that information as the source for learning how to use Windows NT networking features and utilities.

You should also use it in conjunction with the *Windows NT Resource Guide* since there are multiple cross-references between the two books. In addition, the tools for both books are contained on a single compact disc (CD) and in a single set of 3.5-inch floppy disks. (The CD is bound into the back cover of the *Windows NT Resource Guide,* and the floppy disks are available upon request from MS-Press.) See the "Introduction" section of the *Windows NT Resource Guide* for a partial list of the available tools. A complete list is available on the CD in the README.WRI file with instructions on how to use them in the RKTOOLS.HLP file.

This introduction includes two kinds of information you can use to get started:

- The first section outlines the contents of this book, so that you can quickly find technical details about specific elements of Windows NT networking.
- The second section describes the conventions used to present information in this book.

About the Networking Guide

This guide includes the following chapters. Additional tables of contents are included in each part to help you quickly find the information you want.

Part I, About Windows NT Networking

Chapter 1, "Windows NT Networking Architecture," contains information for the support professional who may not have a local area network background. This chapter provides a technical discussion of networking concepts and discusses the networking components included with Windows NT.

Chapter 2, "Network Interoperability," describes how Windows NT works together with your existing Novell® networks, IBM® mainframe systems, and UNIX® systems.

Chapter 3, "Windows NT User Environments," explains the use of home directories and logon scripts in customizing the environment of individual users or related groups of users.

Chapter 4, "Network Security and Administration," describes how security is implemented for workgroups and domains under Windows NT, including local logon and pass-through validation for trusted domains and network browsing.

Chapter 5, "Windows NT Browser," explains how members of a Windows NT network can browse the resources of the network.

Part II, Using Windows NT Networking

Chapter 6, "Using NBF with Windows NT," describes NetBEUI Frame, the implementation of the NetBIOS Extended User Interface protocol under Windows NT, including how network traffic and sessions are managed.

Chapter 7, "Using DLC with Windows NT," presents details about the Data Link Control (DLC) protocol device driver in Windows NT that provides connectivity to IBM mainframes and to local area network printers attached directly to the network.

Chapter 8, "Client-Server Connectivity on Windows NT," discusses how MS-DOS®, Windows®, Windows NT, and OS/2® client workstations communicate with Windows NT databases, focusing on Microsoft SQL Server as an example of a distributed application.

Chapter 9, "Using Remote Access Service," explains the technical details of Windows NT RAS including security, interoperability, and scripting capabilities.

Part III, TCP/IP

Chapter 10, "Overview of Microsoft TCP/IP for Windows NT," describes the elements that make up Microsoft TCP/IP and provides an overview of how you can use Microsoft TCP/IP to support various networking solutions.

Chapter 11, "Installing and Configuring Microsoft TCP/IP and SNMP," describes the process for installing and configuring Microsoft TCP/IP, SNMP, and Remote Access Service (RAS) with TCP/IP on a computer running Windows NT.

Chapter 12, "Networking Concepts for TCP/IP," presents key TCP/IP networking concepts for networking administrators interested in a technical discussion of the elements that make up TCP/IP.

Chapter 13, "Installing and Configuring DHCP Servers," presents the procedures and strategies for setting up servers to support the Dynamic Host Configuration Protocol for Windows networks.

Chapter 14, "Installing and Configuring WINS Servers," presents the procedures and strategies for setting up Windows Internet Name Service servers.

Chapter 15, "Setting Up LMHOSTS," provides guidelines and tips for using LMHOSTS files for name resolution on networks.

Chapter 16, "Using the Microsoft FTP Server Service," describes how to install, configure, and administer the Microsoft FTP Server service.

Chapter 17, "Using Performance Monitor with TCP/IP Services," describes how to use the performance counters for TCP/IP, FTP Server service, DHCP servers, and WINS servers.

Chapter 18, "Internetwork Printing and TCP/IP," describes how to install TCP/IP printing and create TCP/IP printers on Windows NT computers with Microsoft TCP/IP.

Chapter 19, "Troubleshooting TCP/IP," describes how to troubleshoot IP connections and use the diagnostic utilities to get information that will help solve networking problems.

Part IV, Windows NT and the Internet

Chapter 20, "Using Windows NT on the Internet," describes typical scenarios for connecting a Window NT computer or network to the Internet and the logistical details involved in doing that.

Chapter 21, "Setting Up Internet Servers and Clients on Windows NT Computers," describes how to set up Internet servers and clients on a Windows NT computer.

Chapter 22, "Remote Access Service and the Internet," provides technical details about using RAS for Internet connections, including as an Internet Gateway Server and as a router to the Internet for small networks.

Part V, Appendixes

Appendix A, "TCP/IP Utilities Reference," describes the TCP/IP utilities and provides syntax and notes.

Appendix B, "MIB Object Types for Windows NT," describes the LAN Manager MIB II objects provided when you install SNMP with Windows NT.

Appendix C, "Windows Sockets Application," lists third-party vendors who have created software based on the Windows Sockets standard to provide utilities and applications that run in heterogeneous networks using TCP/IP. This appendix also lists Internet sources for public-domain software based on Windows Sockets.

Conventions in This Manual

This document assumes that you have read the Windows NT documentation set and that you are familiar with using menus, dialog boxes, and other features of the Windows operating system family of products. It also assumes that you have installed Windows NT on your system and that you are using a mouse. For keyboard equivalents to menu and mouse actions, see the Microsoft Windows NT online Help.

This document uses several conventions to help you identify information. The following table describes the typographical conventions used in the *Windows NT Networking Guide*.

Convention	Used for
bold	MS-DOS–style command and utility names such as **copy** or **ping** and switches such as **/?** or **-h**. Also used for Registry value names, such as **IniFileMapping** and OS/2 application programming interfaces (APIs).
italic	Parameters for which you can supply specific values. For example, the Windows NT root directory appears in a path name as *systemroot*\SYSTEM32, where *systemroot* can be C:\WINNT35 or some other value.
ALL CAPITALS	Directory names, filenames, and acronyms. For example, DLC stands for Data Link Control; C:\PAGEFILE.SYS is a file in the boot sector.
`Monospace`	Sample text from batch and .INI files, Registry paths, and screen text in non-Windows–based applications.

Other conventions in this document include the following:

- "MS-DOS" refers to Microsoft MS-DOS version 3.3 or later.
- "Windows-based application" is used as a shorthand term to refer to an application that is designed to run with 16-bit Windows and does not run without Windows. All 16-bit and 32-bit Windows applications follow similar conventions for the arrangement of menus, dialog box styles, and keyboard and mouse use.

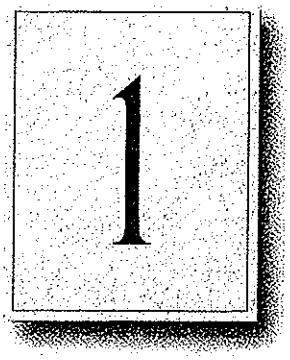
- "MS-DOS–based application" is used as a shorthand term to refer to an application that is designed to run with MS-DOS but not specifically with Windows or Windows NT and is not able to take full advantage of their graphical or memory management features.
- "Command prompt" refers to the command line where you type MS-DOS–style commands. Typically, you see characters such as C:\> to show the location of the command prompt on your screen. In Windows NT, you can double-click the MS-DOS Prompt icon in Program Manager to use the command prompt.
- An instruction to "type" any information means to press a key or a sequence of keys, and then press the ENTER key.
- Mouse instructions in this document, such as "Click the OK button" or "Drag an icon in File Manager," use the same meanings as the descriptions of mouse actions in the *Windows NT System Guide* and the Windows online tutorial.

PART I

About Windows NT Networking

CHAPTER 1

Windows NT Networking Architecture



Windows NT is a complete operating system with fully integrated networking capabilities. These capabilities differentiate Windows NT from other operating systems such as MS-DOS, OS/2, and UNIX for which network capabilities are installed separately from the core operating system.

Integrated networking support means that Windows NT offers these features:

- Support for both peer-to-peer and client-server networking. All Windows NT computers can act as both network clients and servers, sharing files and printers with other computers and exchanging messages over the network. Windows NT Server also includes features needed for full-scale servers, such as domain management tools.
- The ability to easily add networking software and hardware. The networking software integrated into Windows NT lets you easily add protocol drivers, network card drivers, and other network software. Windows NT includes four transport protocols—IPX/SPX (NWLink), TCP/IP, NBF (Windows NT NetBEUI), and DLC.
- Interoperability with existing networks. Windows NT systems can communicate using a variety of transport protocols and network adapters. It can also communicate over a variety of different vendors' networks.
- Support for distributed applications. Windows NT provides a transparent Remote Procedure Call (RPC) facility. It also supports NetBIOS, Sockets, and the Windows Network (WNet) APIs and named pipes and mailslots, for backward compatibility with LAN Manager installations and applications.
- Remote access to networks. Windows NT Remote Access Service (RAS) clients can dial into any PPP or SLIP server. Windows NT RAS servers support any remote clients using IPX, TCP/IP, or NetBEUI using PPP. For additional information about RAS, see Chapter 9, "Using Remote Access Service."
- Print and File sharing, and AppleTalk® routing for Macintosh® clients.

This chapter describes the Windows NT networking architecture and how it achieves each of these goals. For perspective, the next section provides a brief explanation of two industry-standard models for networking—the Open System Interconnection (OSI) reference model and the Institute of Electrical and Electronic Engineers (IEEE) 802 project model. The remainder of the chapter describes the Windows NT networking components as they relate to the OSI and IEEE models and as they relate to the overall Windows NT architecture.

Overview of Networking

In the early years of networking, several large companies, including IBM, Honeywell, and Digital Equipment Corporation (DEC™), each had its own standard for how computers could be connected together. These standards described the mechanisms necessary to move data from one computer to another. These early standards, however, were not entirely compatible. Networks adhering to IBM Systems Network Architecture (SNA) could not communicate directly with networks using DEC Digital Network Architecture (DNA), for example.

In later years, standards organizations, including the International Standards Organization (ISO) and the Institute of Electrical and Electronic Engineers (IEEE), developed models that became globally recognized and accepted as the standards for designing any computer network. Both models describe networking in terms of functional layers.

OSI Reference Model

ISO developed a model called the Open Systems Interconnection (OSI) reference model. It is used to describe the flow of data between the physical connection to the network and the end-user application. This model is the best known and most widely used model to describe networking environments.

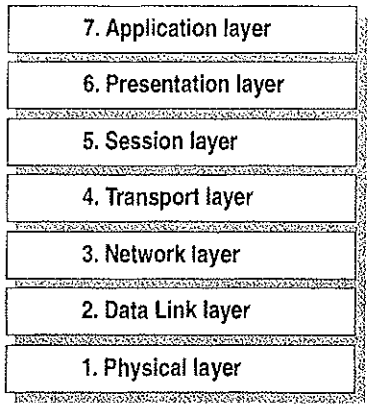


Figure 1.1 Open Systems Interconnection (OSI) Reference Model

As shown in Figure 1.1, the OSI layers are numbered from bottom to top. The most basic functions, such as putting data bits onto the network cable, are on the bottom, while functions attending to the details of applications are at the top.

In the OSI model, the purpose of each layer is to provide services to the next higher layer, shielding the higher layer from the details of how the services are actually implemented. The layers are abstracted in such a way that each layer believes it is communicating with the same layer on the other computer. In reality, each layer communicates only with adjacent layers on one computer. That is, for information to pass from Layer 5 on Computer A to Layer 5 on Computer B, it actually follows the route illustrated by Figure 1.2.

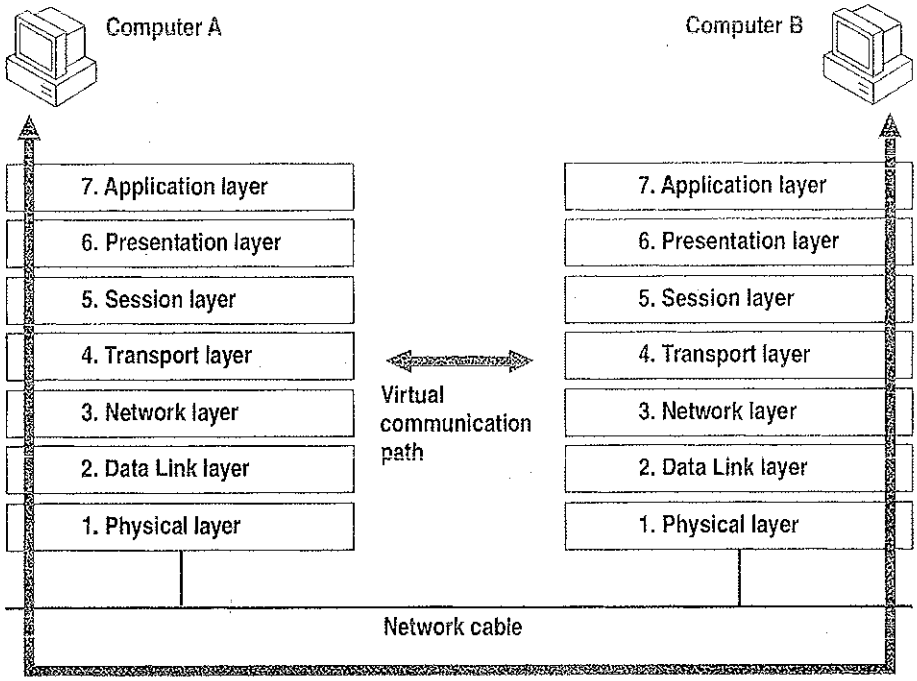


Figure 1.2 Communication Between OSI Layers

The following list describes the purpose of each of the seven layers of the OSI model and identifies services that they provide to adjacent layers.

1. The Physical Layer addresses the transmission of the unstructured raw bit stream over a physical medium (that is, the networking cable). The Physical Layer relates the electrical/optical, mechanical, and functional interfaces to the cable. The Physical Layer also carries the signals that transmit data generated by all the higher layers.

   This layer defines how the cable is attached to the network adapter card. For example, it defines how many pins the connector has and what each pin is used for. It describes the topology used to connect computers together (Token Ring, Ethernet, or some other). It also defines which transmission technique will be used to send data over the network cable.

2. The Data Link Layer packages raw bits from the Physical Layer into *data frames*, which are logical, structured packets in which data can be placed. The exact format of the frame used by the network depends on the topology. That is, a Token Ring network data frame is laid out differently than an Ethernet frame. The Data Link Layer is responsible for providing the error-free transfer of these frames from one computer to another through the Physical Layer. This allows the Network Layer to assume virtually error-free transmission over the network connection. Frames contain source and destination addresses so that the sending and receiving computers can recognize and retrieve their own frames on the network.
3. The Network Layer is responsible for addressing messages and translating logical addresses and names into physical addresses. This layer also determines the route from the source to the destination computer. It determines which path the data should take based on network conditions, priority of service, and other factors. It also manages traffic problems on the network, such as switching, routing, and controlling the congestion of data packets.

   The Network Layer bundles small data frames together for transmission across the network. It also restructures large frames into smaller packets. On the receiving end, the Network Layer reassembles the data packets into their original frame structure.
4. The Transport Layer takes care of error recognition and recovery. It also ensures reliable delivery of host messages originating at the Application Layer. Similar to how the Network Layer handles data frames, this layer repackages messages —dividing long messages into several packets and collecting small messages together in one packet—to provide for their efficient transmission over the network. At the receiving end, the Transport Layer unpacks the messages, reassembles the original messages, and sends an acknowledgment of receipt.
5. The Session Layer allows two applications on different computers to establish, use, and end a connection called a *session*. This layer performs name recognition and the functions needed to allow two applications to communicate over the network, such as security functions.

   The Session Layer provides synchronization between user tasks by placing checkpoints in the data stream. This way, if the network fails, only the data after the last checkpoint has to be retransmitted. This layer also implements dialog control between communicating processes, regulating which side transmits, when, for how long, and so on.

6. The Presentation Layer determines the form used to exchange data between networked computers. It can be called the network's translator. At the sending computer, this layer translates data from a format received from the Application Layer into a commonly recognized, intermediary format. At the receiving end, this layer translates the intermediary format into a format useful to that computer's Application Layer.

   The Presentation Layer also manages network security issues by providing services such as data encryption. It also provides rules for data transfer and provides data compression to reduce the number of bits that need to be transmitted.

7. The Application Layer serves as the window for application processes to access network services. This layer represents the services that directly support the user applications such as software for file transfers, database access, and electronic mail.

IEEE 802 Model

Another networking model developed by the IEEE further defines sublayers of the Data Link Layer. The IEEE 802 project (named for the year and month it began—February 1980) defines the *Media Access Control* (MAC) and the *Logical Link Control* (LLC) sublayers.

As Figure 1.3 shows, the Media Access Control sublayer is the lower of the two sublayers, providing shared access for the computers' network adapter cards to the Physical Layer. The MAC Layer communicates directly with the network adapter card and is responsible for delivering error-free data between two computers on the network.

The Logical Link Control sublayer, the upper sublayer, manages data link communication and defines the use of logical interface points [called Service Access Points (SAPs)] that other computers can reference and use to transfer information from the LLC sublayer to the upper OSI layers. Two protocols running on the same computer would use separate SAPs.

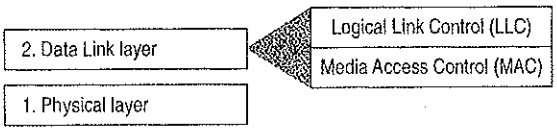


Figure 1.3 Logical Link Control and Media Access Control Sublayers

Project 802 resulted in a number of documents, including three key standards for network topologies:

- 802.3 defines standards for bus networks, such as Ethernet, that use a mechanism called Carrier Sense Multiple Access with Collision Detection (CSMA/CD).
- 802.4 defines standards for token-passing bus networks. (The ArcNet® architecture is similar to this standard in many ways.)
- 802.5 defines standards for Token-Ring networks.

IEEE defined functionality for the LLC Layer in standard 802.2 and defined functionality for the MAC and Physical Layers in standards 802.3, 802.4, and 802.5.

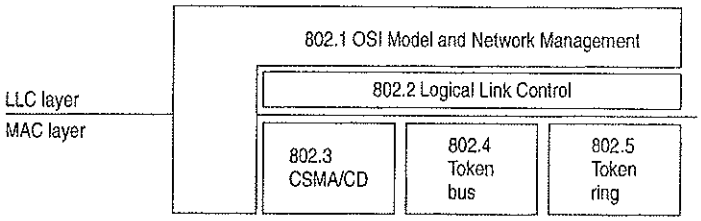


Figure 1.4 Project 802 Standards as Related to LLC and MAC Layers

This chapter describes the layered components of the Windows NT networking architecture, beginning with an overall description of that architecture.

Windows NT Networking Model

As with other architecture components of Windows NT, the networking architecture is built of layers. This helps provide expandability by allowing other functions and services to be added. Figure 1.5 shows all of the components that make up the Windows NT networking model.

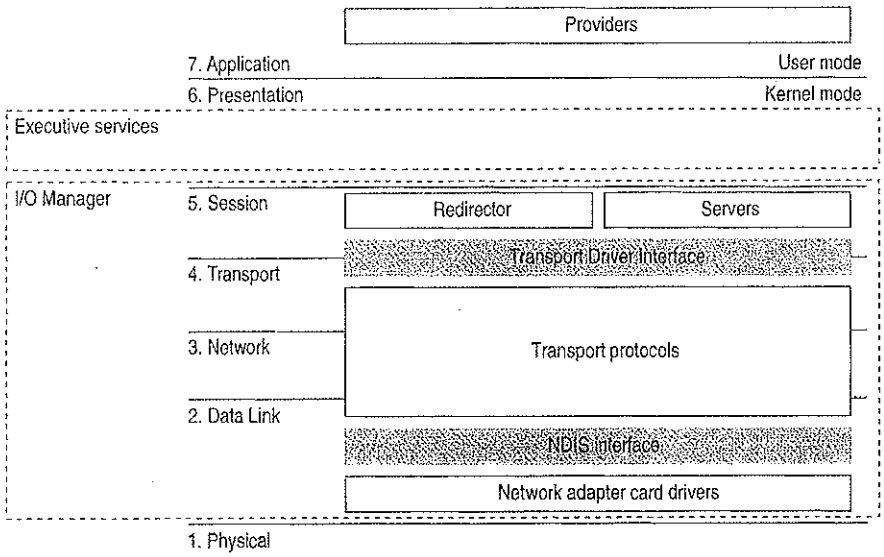


Figure 1.5 Windows NT Networking Model

Each of the Windows NT networking layers performs these functions.

The Windows NT networking model begins at the MAC sublayer where *network adapter card drivers* reside. These drivers link Windows NT to the network via corresponding network adapter cards. Windows NT includes RAS to allow network access to computers for people who work at home or on the road. For more information, see "Remote Access for Windows NT Clients," later in this chapter.

The network model includes two important interfaces—the *NDIS 3.0 Interface* and the *Transport Driver Interface* (TDI). These interfaces isolate one layer from the next by allowing an adjacent component to be written to a single standard rather than many. For example, a network adapter card driver (below the NDIS interface) does not need to include blocks of code specifically written for each transport protocol it uses. Instead, the driver is written to the NDIS interface, which solicits services from the appropriate NDIS-conformant transport protocol(s). These interfaces are included in the Windows NT networking model to allow for portable, interchangeable modules.

Between the two interfaces are *transport protocols*, which act as data organizers for the network. A transport protocol defines how data should be presented to the next receiving layer and packages the data accordingly. It passes data to the network adapter card driver through the NDIS Interface and to the redirector through the TDI.

Above the TDI are *redirectors*, which "redirect" local requests for network resources to the network.

For interconnectivity with other vendors' networks, Windows NT allows multiple redirectors. For each redirector, the Windows NT computer must also have a corresponding *provider* DLL (supplied by the network vendor). A Multiple Provider Router determines the appropriate provider and then routes the application request via the provider to the corresponding redirector.

The rest of this chapter describes these Windows NT networking components in detail.

NDIS-Compatible Network Adapter Card Drivers

Until the late 1980s, many of the implementations of transport protocols were tied to a proprietary implementation of a MAC-Layer interface defining how the protocol would converse with the network adapter card. This made it difficult for network adapter card vendors to support the different network operating systems available on the market. Each network adapter card vendor had to create proprietary interface drivers to support a variety of protocol implementations for use with several network operating system environments.

In 1989, Microsoft and 3Com jointly developed a standard defining an interface for communication between the MAC Layer and protocol drivers higher in the OSI model. This standard is known as the Network Device Interface Specification (NDIS). NDIS allows for a flexible environment of data exchange. It defines the software interface—called the NDIS interface—used by transport protocols to communicate with the network adapter card driver.

The flexibility of NDIS comes from the standardized implementation used by the network industry. Any NDIS-conformant protocol can pass data to any NDIS-conformant network adapter card driver, and vice versa. A process called *binding* is used to establish the initial communication channel between the protocol driver and the network adapter card driver.

Windows NT currently supports device drivers and transport protocols written to NDIS version 3.0.

NDIS allows multiple network adapter cards on a single computer. Each network adapter card can support multiple transport protocols. The advantage of supporting multiple protocol drivers on a single network card is that Windows NT computers can have simultaneous access to different types of network servers, each using a different transport protocol. For example, a computer can have access to both a Windows NT Server using NBF (the Windows NT implementation of NetBEUI) and a UNIX server via TCP/IP simultaneously.

Unlike previous NDIS implementations, Windows NT does not need a protocol manager module to link the various components at each layer together. Instead, Windows NT uses the information in the Registry (described in Chapter 10, "Overview of the Windows NT Registry" of the *Windows NT Resource Guide*) and a small piece of code called the *NDIS wrapper* that surrounds the network adapter card driver.

Transport Protocols

Sandwiched between the NDIS interface and the TDI are transport protocol device drivers. These drivers communicate with a network adapter card via a NDIS-compliant device driver.

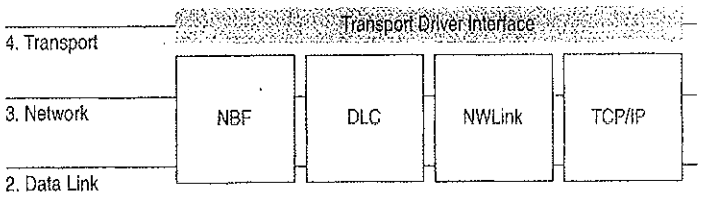


Figure 1.6 Transport Protocols

Windows NT includes these transports:

- NBF is a transport protocol derived from NetBEUI and provides compatibility with existing LAN Manager, LAN Server, and MS-Net installations. (For more information, see Chapter 6, "Using NBF with Windows NT.")
- TCP/IP is a popular routable protocol for wide-area networks.
- NWLink is an NDIS-compliant version of Internetwork Packet Exchange (IPX/SPX) compatible protocol. It can be used to establish connections between Windows NT computers and either MS-DOS, OS/2, Windows, or other Windows NT computers via RPC, Sockets, or Novell NetBIOS.
- Microsoft Data Link Control (DLC) provides an interface for access to mainframes and network attached printers. (For more information, see Chapter 7, "Using DLC with Windows NT.")

- AppleTalk supports Services for Macintosh in Windows NT Server. Developers using Windows NT Workstation can also install the AppleTalk protocol, as needed, when developing AppleTalk-compliant programs.

Transport Protocols and Streams

Windows NT supports Streams-compliant protocols provided by third parties. These protocols use Streams as an intermediary between the protocol and next interface layer (NDIS on the bottom and TDI on top). Calls to the transport protocol driver must first go through the upper layer of the Streams device driver to the protocol, then back through the lower layer of Streams to the NDIS device driver.

Using Streams makes it easier for developers to port other protocol stacks to Windows NT. It also encourages protocol stacks to be organized in a modular, stackable style, which is in keeping with the original OSI model.

Transport Driver Interface

The Windows NT networking model was designed to provide a platform on which other vendors can develop distributed applications. The NDIS boundary helps to do this by providing a unified interface at a significant break point in the model. At another significant breakpoint, namely the Session Layer of the OSI model, Windows NT includes another boundary layer. The TDI provides a common interface for networking components that communicate at the Session Layer. These boundaries allow software components above and below a level to be mixed and matched without reprogramming.

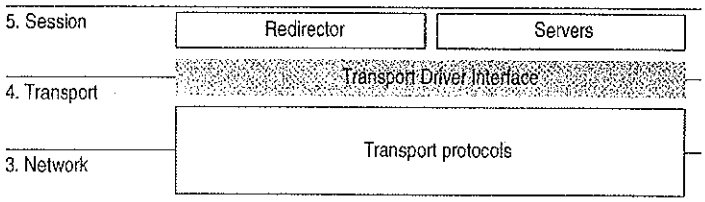


Figure 1.7 The Transport Driver Interface

The TDI is not a single program but a protocol specification to which the upper bounds of transport protocol device drivers are written. (Windows NT also includes a TDI driver that handles IRQ packet traffic from multiple TDI providers.) At this layer, networking software provides a virtual connection between the local redirector and each local or remote destination with which the redirector communicates. Similar connections are made between the server and the sources of the requests it receives.

Windows NT Workstations and Servers

Above all, the goal of a network is to share resources in one location on the network and to use them from another location on the network. On a network, computers can be organized in one of two ways:

- On networks using a classic *client-server model,* dedicated servers share resources and client workstations can access those resources.
- On networks using the *peer-to-peer networking model* (also called workgroup computing), each computer can act as both client workstation and server. Computers running

Windows NT allows you to configure your network using either or both of these models. Windows NT Workstation can use the peer-to-peer model with as many as ten users simultaneously connected to each workstation.

In the Windows NT architecture, two software components—called the server and the redirector—provide server and workstation functionality. Both of these components reside above the TDI and are implemented as file system drivers.

Being implemented as file system drivers has several benefits. Applications can call a single API (namely, Windows NT I/O functions) to access files on local and remote computers. From the I/O Manager's perspective, there is no difference between accessing files stored on a remote networked computer and accessing those stored locally on a hard disk. The redirector and server can directly call other drivers and other kernel-mode components such as the Cache Manager, thus optimizing performance. Each can be loaded and unloaded dynamically. In addition, the Windows NT redirector can coexist with other redirectors (discussed more fully in the section called "Interoperating with Other Networks," later in this chapter).

Windows NT Redirector

The redirector is the component through which one computer gains access to another computer. The Windows NT redirector allows connection to other Windows NT computers as well as to LAN Manager, LAN Server, and MS-Net servers. This redirector communicates to the protocol stacks to which it is bound via the TDI. Because network connections are not entirely reliable, it is up to the redirector to reestablish connections when they go down.

As illustrated by Figure 1.8, when a process on a Windows NT workstation tries to open a file on a remote computer, these steps occur:

1. The process calls the I/O Manager, asking for the file to be opened.
2. The I/O Manager recognizes that the request is for a file on a remote computer, so it passes it to the redirector file system driver.
3. The redirector passes the request to lower-level network drivers, which transmit it to the remote server for processing.

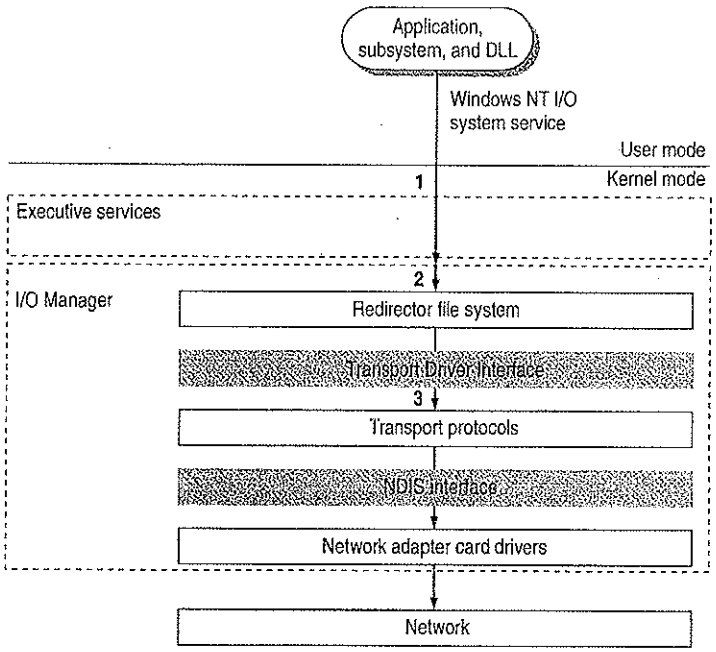


Figure 1.8 Client-Side Processing Using the Redirector

Windows NT Server

The server component entertains the connections requested by client-side redirectors and provides them with access to the resources they request. When a Windows NT server receives a request from a remote workstation to read a file on the server, these steps occur (as shown in Figure 1.9):

1. The low-level network drivers receive the request and pass it to the server driver.
2. The server passes a file-read request to the appropriate local file system driver.
3. The local file system driver calls a lower-level disk driver to access the file.
4. The data is passed back to the local file system driver.
5. The local file system driver passes the data back to the server.
6. The server passes the data to the lower-level network drivers for transmission back to the client computer.

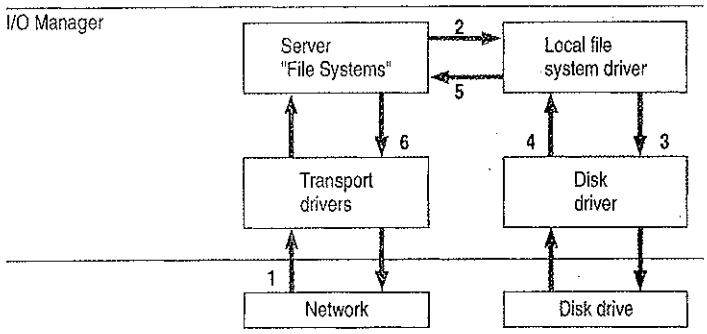


Figure 1.9 Server-Side Processing Using the Server

Interoperating with Other Networks

As mentioned before, the Windows NT redirector allows connections to LAN Manager, LAN Server, and MS-Net servers. It can also coexist with redirectors for other networks, such as Novell NetWare® and Banyan® VINES®.

While Windows NT includes integrated networking, its open design provides for transparent access to other networks. For example, a Windows NT user can concurrently access files stored on Windows NT and NetWare servers.

For details about interoperating with other networks, see Chapter 2, "Network Interoperability."

Providers and the Provider Interface Layer

For each additional type of network (NetWare, VINES, or some other), you must install a component called a *provider*. The provider is the component that allows a Windows NT computer to communicate with the network. Windows NT includes a provider for the Windows NT network. It also includes the Client Service for NetWare with Windows NT Workstation and the Gateway Service for NetWare with Windows NT Server, with which a Windows NT computer can connect as a client to a NetWare network. Other provider DLLs are supplied by the appropriate network vendors.

From the application viewpoint, there are two sets of commands that can cause network traffic—uniform naming convention (UNC) commands and WNet commands.

UNC is a method of identifying a shared resource on a network. UNC names start with two backslashes followed by the server name. All other fields in the name are separated by a single backslash. Although it's enough to simply specify the servername to list a server's shared resources, a full UNC name is in this form:

```
\\server\share\subdirectory\filename
```

WNet is part of the Win32® API and is specifically designed to allow applications on Windows NT workstations to connect to multiple networks, browse the resources of computers on those networks, and transfer data between computers of various networks. File Manager, for example, uses the WNet interface to provide its network browsing and connection facilities.

As shown in Figure 1.10, the provider layer spans the line between kernel and user modes to manage commands that may cause network traffic. The provider layer also includes two components to route UNC and WNet requests to the appropriate provider:

- The Multiple UNC Provider (MUP) receives UNC commands and locates the redirector that can make a connection to the UNC name.
- The Multiple Provider Router (MPR) receives WNet commands and passes the request to each redirector in turn until one is found that can satisfy the request.

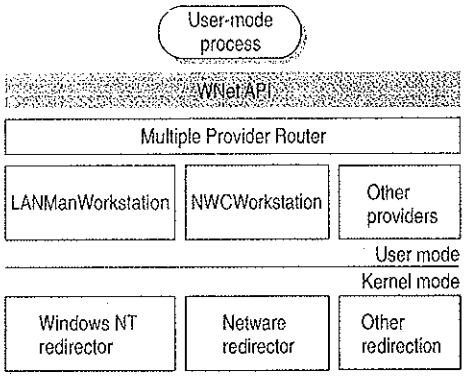


Figure 1.10 Provider Interface Components

Note I/O calls, such as Open, can contain both an UNC name and WNet calls.

Multiple UNC Provider

The MUP is a kernel-mode component whose job is to locate UNC names. When an application sends a command containing UNC names, MUP routes each UNC name to one of the registered UNC providers, including LanmanWorkstation and any others that may be installed. When a provider indicates that it can communicate with the server, MUP sends the remainder of the command to the provider.

When applications make I/O calls that contain UNC names, the MUP directs them to the appropriate redirector file system driver. The call is routed to its redirector based on the handle on the I/O call.

Multiple Provider Router

Through the MPR, Windows NT provides an open interface that enables consistent access to third-party network file systems. The key to the MPR is that all file systems, regardless of type and physical location, are accessible through the same set of file system APIs.

Applications, including File Manager, make file system requests through the Windows NT Win32 API. The MPR ensures that requests are directed to the proper file system. Local file requests are sent to the local disk, remote requests to Windows-based servers are sent to the proper server by the Windows redirector, requests to NetWare-based servers are handled by the NetWare Client for Windows NT and sent to the NetWare server, and so on.

Because applications access all types of files through a single set of APIs, any application can access any kind of server without affecting the user.

Distributed Applications and Windows NT

Any application you run on Windows NT can take advantage of networking resources because networking components are built into Windows NT. In addition, Windows NT includes several mechanisms that support and benefit distributed applications.

A *distributed application* is one that has two parts—a front-end to run on the client computer and a back-end to run on the server. In distributed computing, the goal is to divide the computing task into two sections. The front-end requires minimal resources and runs on the client's workstation. The back-end requires large amounts of data, number crunching, or specialized hardware and runs on the server. A connection between the client and the server at a process-to-process level allows data to flow in both directions between the client and server.

Microsoft Mail, Microsoft Schedule+, SQL Server, and SNA Server are examples of distributed applications.

As described in the next section, Windows NT includes NetBIOS and Windows Sockets interfaces for building distributed applications. In addition, Windows NT supports peer-to-peer named pipes, mailslots, and remote procedure calls (RPC). On Windows NT, for example, an electronic mail product could include a messaging service using named pipes and asynchronous communication that runs with any transport protocol or network card.

Of named pipes, mailslots, and RPC, RPC is the most portable mechanism. RPCs use other interprocess communication (IPC) mechanisms—including named pipes and the NetBIOS and Windows Sockets interfaces—to transfer functions and data between client and server computers.

Named pipes and mailslots are implemented to provide backward compatibility with existing LAN Manager installations and applications.

For more information about using distributed applications with Windows NT, see Chapter 8, "Client-Server Connectivity on Windows NT."

NetBIOS and Windows Sockets

Besides redirectors, Windows NT includes two other components that provide links to remote computers—NetBIOS and Windows Sockets. Windows NT includes NetBIOS and Windows Sockets interfaces for building distributed applications. (Windows NT also includes three other interprocess communication mechanisms—named pipes, mailslots, and remote procedure calls—for use by distributed applications. These are described later in this chapter.)

The NetBIOS and Windows Sockets APIs are supplied by separate DLLs. These DLLs communicate with corresponding drivers in the Windows NT Executive. As shown by Figure 1.11, the NetBIOS and Windows Sockets drivers then bypass the Windows NT redirector and communicate with protocol drivers directly using the TDI.

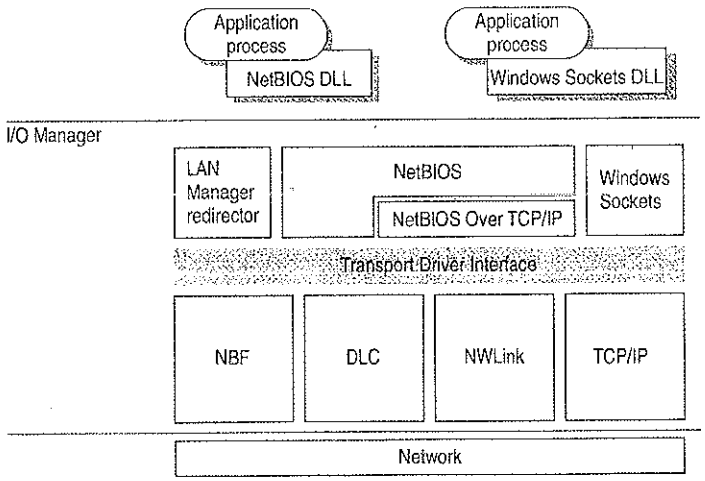


Figure 1.11 NetBIOS and Windows Sockets Support

NetBIOS

NetBIOS is the Network Basic Input/Output System—a session-level interface used by applications to communicate with NetBIOS-compliant transports such as NetBEUI Frame (NBF). The network redirector is an example of a NetBIOS application. The NetBIOS interface is responsible for establishing logical names on the network, establishing sessions between two logical names on the network, and supporting reliable data transfer between computers that have established a session.

This Session-Layer interface was originally developed by Sytek, Inc., for IBM's broadband computer network. At that time, NetBIOS was included on a ROM chip on the network adapter card. Sytek also developed a NetBIOS for IBM's Token-Ring network, this time implemented as a device driver. Several other vendors have since produced versions of this interface.

In order to support the emerging network industry standard, Microsoft developed the NetBIOS interface for MS-Net and LAN Manager products, and also included this interface with the Windows for Workgroups product.

NetBIOS uses a unique logical name to identify a workstation for handling communications between nodes. A NetBIOS name is a unique alphanumeric name consisting of one to 15 characters. To carry on two-way communication between computers, NetBIOS establishes a logical connection, or *session*, between them. Once a logical connection is established, computers can then exchange data in the form of NetBIOS requests or in the form of a Server Message Block (SMB).

Server Message Blocks

The SMB protocol (developed jointly by Microsoft, Intel, and IBM) defines a series of commands used to pass information between networked computers and can be broken into four message types—session control, file, printer, and message. Session control consists of commands that start and end a redirector connection to a shared resource at the server. The file SMB messages are used by the redirector to gain access to files at the server. The printer SMB messages are used by the redirector to send data to a print queue at a server and to get status information about the print queue. The message SMB type allows an application to send messages to or receive messages from another workstation.

The redirector packages network control block (NCB) requests meant for remote computers in a structure known as a system message block (SMB). SMBs can be sent over the network to a remote device. The redirector also uses SMBs to make requests to the protocol stack of the local computer, such as "Create a session with the file server."

The provider DLL listens for SMB messages destined for it and removes the data portion of the SMB request so that it can be processed by a local device.

SMBs provide interoperability between different versions of the Microsoft family of networking products and other networks that use SMBs, including these:

MS OS/2 LAN Manager	DEC PATHWORKS™
Microsoft Windows for Workgroups	Microsoft LAN Manager for UNIX
IBM LAN Server	3Com® 3+Open®
MS-DOS LAN Manager	MS-Net

Windows Sockets

Windows Sockets is a Windows implementation of the widely used UC Berkeley Sockets API. Microsoft TCP/IP, NWLink, and AppleTalk protocols use this interface.

A *socket* provides an endpoint to a connection; two sockets form a complete path. A socket works as a bidirectional pipe for incoming and outgoing data between networked computers. The Windows Sockets API is a networking API tailored for use by programmers using the Microsoft Windows family of products. Windows Sockets is a public specification based on Berkeley UNIX Sockets and aims to do the following:

- Provide a familiar networking API to programmers using Windows or UNIX
- Offer binary compatibility between heterogeneous Windows-based TCP/IP stack and utilities vendors
- Support both connection-oriented and connectionless protocols

Most users will use programs that comply with Windows Sockets, such as FTP or Telnet. (However, developers who are interested in developing a Windows Sockets application can find specifications for Windows Sockets on the Internet.)

Named Pipes and Mailslots

Named pipes and mailslots are actually written as file systems, unlike other IPC mechanisms. Thus, the Registry lists entries for the Named Pipes File System (NPFS) and the Mailslot File System (MSFS). As file systems they share common functionality, such as caching, with the other file systems. Additionally, processes on the local computer can use named pipes and mailslots to communicate with one another without going through networking components. Remote access to named pipes and mailslots, as with all of the file systems, is provided through the redirector.

Named pipes are based on OS/2 API calls, but in Windows NT they include additional asynchronous support and increased security.

Another new feature added to named pipes is impersonation, which allows a server to change its security identifier so that it matches the client's. For example, suppose a database server system uses named pipes to receive read and write requests from clients. When a request comes in, the database server program can impersonate the client before attempting to perform the request. So even if the server program does have authority to perform the function, the client may not, and the request would be denied. (For more information on impersonation, see Chapter 2, "Windows NT Security Model" of the *Windows NT Resource Guide*.)

Mailslot APIs in Windows NT are a subset of those in Microsoft OS/2 LAN Manager. Windows NT implements only second-class mailslots, not first-class mailslots. Second-class mailslots provide *connectionless* messaging for broadcast messages and so on. Delivery of the message is not guaranteed, although the delivery rate on most networks is very high. Second-class mailslots are most useful for identifying other computers or services on a network and for wide-scale notification of a service.

Remote Procedure Calls

Much of the original design work for an RPC facility was started by Sun Microsystems®. This work was continued by the Open Software Foundation (OSF) as part of their overall Data Communications Exchange (DCE) standard. The Microsoft RPC facility is compatible with the OSF/DCE-standard RPC. It is important to note that it is compatible and not compliant. Compliance in this case means starting with the OSF source code and building on it. The Microsoft RPC facility is completely interoperable with other DCE-based RPC systems such as the ones for HP® and IBM AIX® systems.

The RPC mechanism is unique because it uses the other IPC mechanisms to establish communications between the client and the server. RPC can use named pipes, NetBIOS, or Windows Sockets to communicate with remote systems. If the client and server are on the same computer, it can use the Local Procedure Call (LPC) facility to transfer information between processes and subsystems. This makes RPC the most flexible and portable of the Windows NT IPC mechanisms.

RPC is based on the concepts used for creating structured programs, which can be viewed as having a "backbone" to which a series of "ribs" can be attached. The backbone is the mainstream logic of the program, which should rarely change. The ribs are the procedures the backbone calls on to do work or perform functions.

In traditional programs, these ribs are statically linked. By using DLLs, structured programs can dynamically link the ribs. With DLLs, the procedure code and the backbone code are in different modules. The DLL can thus be modified or updated without changes to the backbone. RPC means that the backbone and the ribs can exist on different computers, as shown in Figure 1.12.

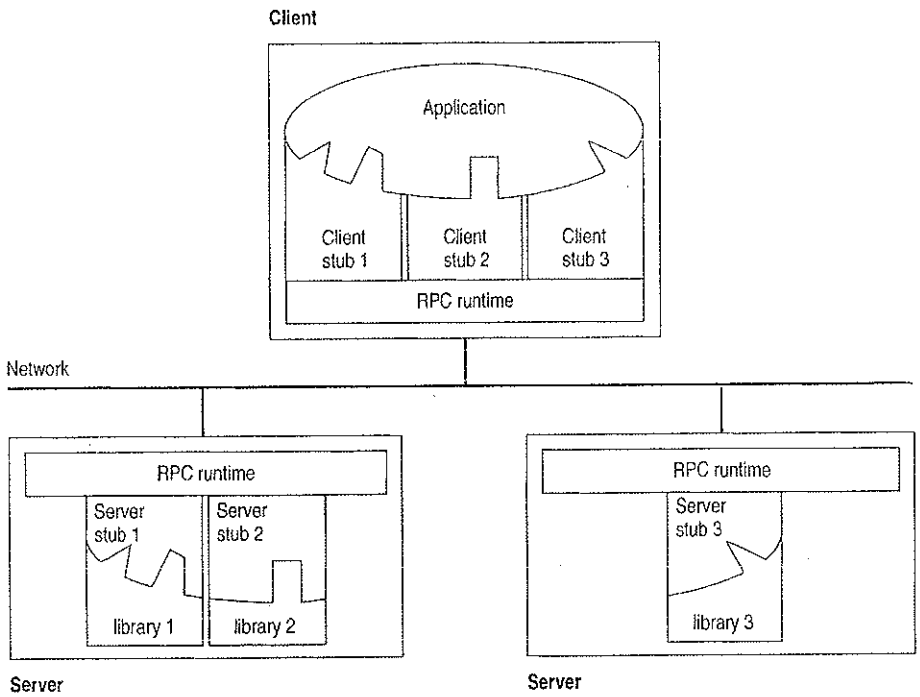


Figure 1.12 Remote Procedure Call Facility

In this figure, the client application was developed with a specially compiled *stub* library. The client application thinks it is calling its own subroutines. In reality, these stubs transfer the data and the function down to a module called the RPC Runtime. This module is responsible for finding the server that can satisfy the RPC command. Once found, the function and data are sent to the server, where it is picked up by the RPC Runtime module on the server. The server piece then loads the needed library for the function, builds the appropriate data structure, and calls the function. The function thinks it is being called by the client application. When the function is completed, any return values are collected, formatted, and sent back to the client via the RPC Runtime modules. When the function returns to the client application it has the appropriate returned data, or it has an indication that the function failed in stream.

Remote Access Service

Windows NT 3.5 Remote Access Service (RAS) connects remote or mobile workers to corporate networks. Optimized for client-server computing, RAS is implemented primarily as a software solution, and is available on all of Microsoft's operating systems.

To understand the RAS architecture, it is important to make the distinction between RAS and remote control solutions, such as Cubix and pcANYWHERE®. RAS is a software-based multi-protocol router; remote control solutions work by sharing screen, keyboard and mouse control over a WAN connection. In a remote control solution, users share a CPU or multiple CPU's on the server. In contrast, a Windows NT RAS server's CPU is dedicated to communications, not to running applications.

Point-to-Point Protocol (PPP)

Windows NT supports the Point-to-Point Protocol (PPP) in RAS. PPP is a set of industry standard framing and authentication protocols. PPP negotiates configuration parameters for multiple layers of the OSI model.

PPP support in Windows NT 3.5 (and Windows 95) means that computers running Windows can dial into remote networks through any server that complies with the PPP standard. PPP compliance also enables a Windows NT Server to receive calls from, and provide network access to, other vendors' remote access software.

The PPP architecture also enables clients to load any combination of IPX, TCP/IP, and NetBEUI. Applications written to the Windows Sockets, NetBIOS, or IPX interface can now be run on a remote Windows NT Workstation. The following illustrates the PPP architecture of RAS.

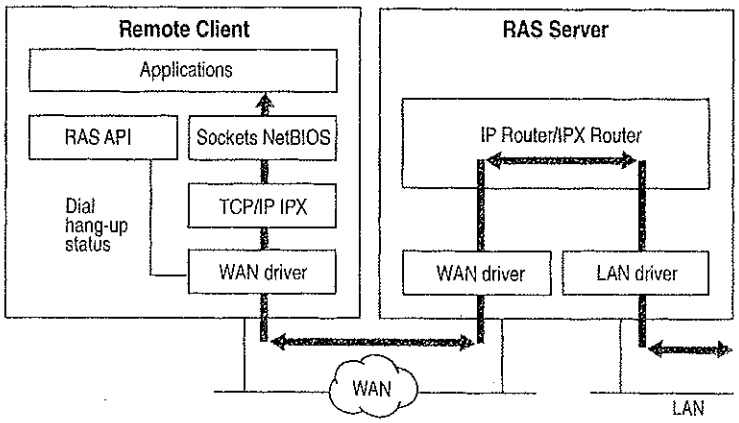


Figure 1.13 PPP Architecture of RAS

RAS Connection Sequence

Understanding the RAS connection sequence will help you understand the PPP protocol.

Upon connecting to a remote computer, PPP negotiation begins.

First, framing rules are established between the remote computer and server. This allows continued communication (frame transfer) to occur.

Next the RAS server authenticates the remote user using the PPP authentication protocols (PAP, CHAP, SPAP). The protocols invoked depend on the security configurations of the remote client and server.

Once authenticated, the Network Control Protocols (NCPs) are used to enable and configure the server for the LAN protocol that will be used on the remote client.

When the PPP connection sequence has completed successfully, the remote client and RAS server can begin to transfer data using any supported protocol, such as Windows Sockets, RPC, or NetBIOS. The following illustrates where the PPP protocol are on the OSI model.

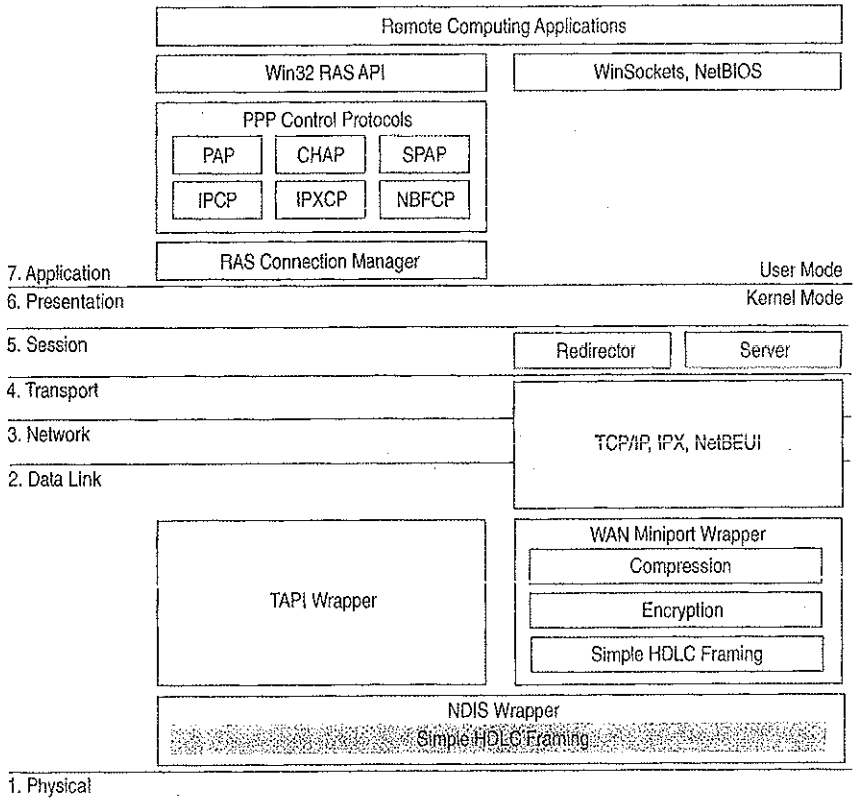


Figure 1.14 Location of the PPP Protocol on the OSI Model

If your remote client is configured to use the NetBIOS gateway or SLIP, this sequence is invalid.

NetBIOS Gateway

Windows NT continues to support NetBIOS gateways, the architecture used in previous version of Windows NT and LAN Manager. Remote users connect using NetBEUI, and the RAS server translates packets, if necessary, to IPX or TCP/IP. This enables users to share network resources in a multi-protocol LAN, but prevents them from running applications which rely on IPX or TCP/IP on the client. The NetBIOS gateway is used by default when remote clients are using NetBEUI. The following illustrates the NetBIOS gateway architecture of RAS.

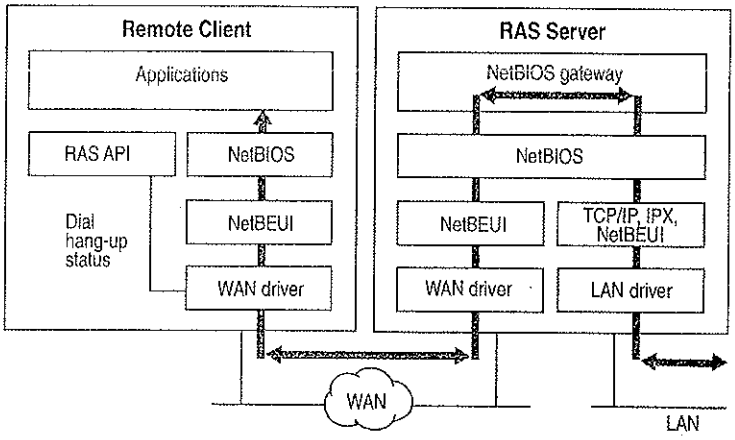


Figure 1.15 NetBIOS Gateway Architecture of RAS

An example of the NetBIOS gateway capability is remote network access for Lotus® Notes® users. While Lotus Notes does offer dial up connectivity, dial up is limited to the Notes application only. RAS complements this connectivity by providing a low-cost, high-performance remote network connection for Notes® users which not only connects Notes, but offers file and print services, and access to other network resources.

Serial Line Internet Protocol (SLIP)

Serial Line Internet Protocol (SLIP), is an older communications standard found in UNIX environments. SLIP does not provide the automatic negotiation of network configuration and encrypted authentication that PPP can provide. SLIP requires user intervention. Windows NT 3.5 RAS can be configured as a SLIP client, enabling Windows NT users to dial into an existing SLIP server. RAS does not provide a SLIP server in Windows NT Server.

See the RASPHONE.HLP online Help file on the Windows NT distribution disks (or, if RAS has been installed, *\systemroot***SYSTEM32**) for more information about RAS.

Services for Macintosh

Through Windows NT Services for Macintosh, Macintosh users can connect to a Windows NT server the same way they would connect to an AppleShare server. Windows NT Services for Macintosh will support an unlimited number of simultaneous AFP™ connections to a Windows NT server, and Macintosh sessions will be integrated with Windows NT sessions. The per-session memory overhead is approximately 15K.

Existing versions of LAN Manager Services for the Macintosh can be easily upgraded to Windows NT Services for Macintosh. OS/2-based volumes that already exist are converted with permissions intact. In addition, graphical installation, administration, and configuration utilities are integrated with existing Windows NT administration tools. Windows NT Services for Macintosh requires System 6.0.7 or higher and is AFP 2.1-compliant; however, AFP 2.0 clients are supported. AFP 2.1 compliance provides support for logon messages and server messages.

Support for Macintosh networking is built into the core operating system for Windows NT Server. Windows NT Services for Macintosh includes a full AFP 2.0 file server. All Macintosh file system attributes, such as resource data forks, 32-bit directory IDs, and so on, are supported. As a file server, all filenames, icons, and access permissions are intelligently managed for different networks. For example, a Word for Windows file will appear on the Macintosh with the correct Word for Macintosh icons. These applications can also be launched from the File Server as Macintosh applications. When files are deleted, there will be no orphaned resource forks left to be cleaned up.

Windows NT Services for Macintosh fully supports and complies with Windows NT security. It presents the AFP security model to Macintosh users and allows them to access files on volumes that reside on CD-ROM or other read-only media. The AFP server also supports both cleartext and encrypted passwords at logon time. The administrator has the option to configure the server not to accept cleartext passwords.

Services for Macintosh can be administered from Control Panel and can be started transparently if the administrator has configured the server to use this facility.

Macintosh-accessible volumes can be created from File Manager. Services for Macintosh automatically creates a Public Files volume at installation time. Windows NT file and directory permissions are automatically translated into corresponding Macintosh permissions.

Windows NT Services for Macintosh has the same functionality as the LAN Manager Services for Macintosh 1.0 MacPrint. In addition, administration and configuration are easier. There is a user interface for publishing a print queue on AppleTalk and a user interface for choosing an AppleTalk printer as a destination device. The Windows NT print subsystem handles AppleTalk despooling errors gracefully, and uses the built-in printer support in Windows NT. (The PPD file scheme of Macintosh Services 1.0 is not used.) Services for Macintosh also has a PostScript-compatible engine that allows Macintoshes to print to any Windows NT printer as if they were printing to a LaserWriter.

Additional Reading

For additional information on topics related to networking and the Windows NT networking model, see the following resources:

ANSI/IEEE standard 802.2 - 1985 (ISO/DIS 8802/2): *IEEE Standards for Local Area Networks—Logical Link Control Standard.*

ANSI/IEEE standard 802.3 - 1985 (ISO/DIS 8802/3): *IEEE Standards for Local Area Networks—Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*; American National Standards Institute; January 12, 1989.

ANSI/IEEE standard 802.4 - 1985 (ISO/DIS 8802/4): *IEEE Standards for Local Area Networks—Token-Passing Bus Access Method and Physical Layer Specifications*; American National Standards Institute; December 17, 1984.

ANSI/IEEE standard 802.5 - 1985 (ISO/DIS 8802/5): *IEEE Standards for Local Area Networks—Token-Ring Access Method and Physical Layer Specifications*; American National Standards Institute; June 2, 1989.

Beatty, Dana. "Programming to the OS/2 IEEE 802.2 API." *OS/2 Notebook*. Ed. Dick Conklin. Redmond, WA: Microsoft Press, 1990.

Haugdahl, J. Scott. *Inside NetBIOS*. Minneapolis: Architecture Technology Corporation, 1990.

Haugdahl, J. Scott. *Inside NetBIOS (2nd Edition)*. Minneapolis, Minn: Architecture Technology Corporation, 1988.

Haugdahl, J. Scott. *Inside Token-Ring (3rd Edition).* Minneapolis, Minn: Architecture Technology Corporation, 1990.

IBM Token-Ring Network Architecture Reference (6165877), November 1985.

IBM Token-Ring Network PC Adapter Technical Reference (69X7830).

International Business Machines. *Local Area Network: Technical Reference (SC30-3383-2).* New York: 1988.

International Standard 7498: *Information processing systems—Open Systems Interconnection—Basic Reference Model (First edition);* American National Standards Institute, November 15, 1984. The OSI model.

Martin, James. *Local Areas Networks: Architecture and Implementations.* Englewood Cliffs, NJ: Prentice Hall: 1989.

Microsoft Corporation, 3Com Corporation. *SMB Specification.* This may be obtained from the files library in the Microsoft Client Server Computing forum on CompuServe (GO MSNETWORK).

Microsoft Corporation. *Microsoft LAN Manager Resource Kit.* Microsoft Corporation, 1992.

Microsoft. *Computer Dictionary.* Redmond, WA: Microsoft Press, 1991.

Microsoft. *Microsoft LAN Manager MS-DLC Protocol Driver.* Redmond, WA: Microsoft Press, 1991.

Microsoft. *Microsoft/3Com LAN Manager Network Driver Interface Specification.* Redmond, WA: Microsoft Press, 1990.

Miller, Mark. *LAN Protocol Handbook.* Redwood City, CA: M & T Books, 1990.

Miller, Mark. *LAN Troubleshooting Handbook.* Redwood City, CA: M & T Books, 1990.

Tanenbaum, Andrew. *Computer Networks (2nd Edition).* Englewood Cliffs, NJ: Prentice Hall, 1988

The Ethernet. A Local Area Network. (Data Link Layer and Physical Layer Specifications); version 2.0, November 1982. Also known as the "Ethernet Blue Book."

CHAPTER 2

Network Interoperability

In addition to Windows-based networking, Windows NT supports network interoperability with computers running a wide range of operating systems and network protocols. This support makes it easy to incorporate computers running Windows NT into existing networks so you can take advantage of the advanced features of Windows NT without disrupting your enterprise.

The networking architecture of Windows NT is protocol-independent, providing standard interfaces for applications—such as Windows Sockets, remote procedure calls (RPC), and NetBIOS—and device drivers. Besides making it easier to implement a particular protocol stack for Windows NT, this architecture also enables a Windows NT computer to run multiple protocols on a single network adapter card. As a result, a Windows NT computer can simultaneously communicate with a number of different network systems.

Of particular interest to most network administrators is how to provide access by and to computers running Windows NT Workstation and Windows NT Server in the following environments:

- Novell NetWare networks
- UNIX networks
- SNA networks for IBM mainframe and midrange computers

This chapter provides an overview of some of the issues and benefits involved in using Windows NT computers in these environments.

Using Windows NT with NetWare

Windows NT computers can easily be integrated into a predominantly NetWare environment, making the benefits of an advanced operating system available to an existing network.

A network administrator contemplating a mixed network environment is naturally concerned about how the various components will be able to communicate with each other. In the case of a mixed Windows-based networking and NetWare environment, the network administrator wants to ensure that Windows NT Workstation computers added to the network are able to use file and print resources on existing NetWare servers, and that existing NetWare clients can access client-server applications running on Windows NT Servers. The following figure shows how the various components of the network relate to each other.

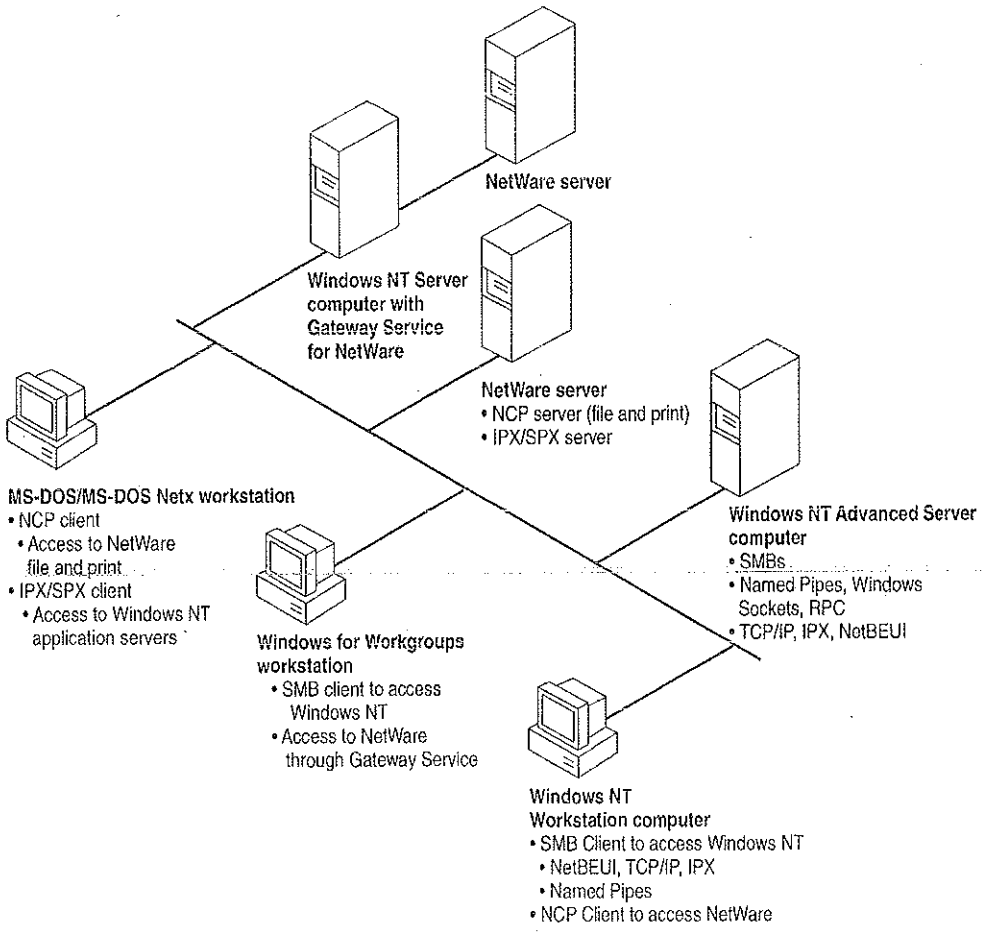


Figure 2.1 Mixed Windows-based and NetWare Environment

The following sections discuss how Windows NT computers can effectively function either as a client of NetWare servers or as an application server for NetWare clients.

Windows NT Servers on a NetWare Network

Many organizations that use NetWare are seeking solutions for downsizing or reengineering existing applications that run on minicomputers or mainframes. NetWare servers are designed to function primarily as file and print servers, so they do not support such business-critical applications well. NetWare servers do not feature preemptive multitasking or protected virtual memory, essential features for client-server applications. On the other hand, Windows NT Server makes an ideal platform for such demanding applications because of its scalability, fault tolerance, 32-bit architecture, and threaded, preemptive multitasking with full memory protection.

NetWare administrators can take advantage of the advanced features of Windows NT Servers on an existing NetWare network without interfering with client systems' access to file and printer resources on NetWare servers. For example, a NetWare administrator can add Windows NT Server computers running SQL Server to the network so client workstations can take advantage of a distributed high-performance relational database system while still being able to use files and printers shared by their usual NetWare servers. Such a solution requires no additional hardware or software to provide the necessary connectivity.

To function as an application server for NetWare clients, a computer running Windows NT Server must be running the built-in NWLink IPX/SPX-compatible protocol stack (NWLink). Connections over NWLink can be made via Remote Procedure Calls (RPC), Windows Sockets, Novell NetBIOS, or the NWLink NetBIOS installed with NWLink. Because NWLink is NDIS-compliant, the Windows NT computer can simultaneously run other protocol stacks, such as NetBEUI Frame (NBF) or TCP/IP, through which it can communicate with non-NetWare computers.

Windows NT Clients on a NetWare Network

Windows NT was designed from the start with integrated network support in mind. Because the network support built into Windows NT is independent of the underlying network system, the same user interface and tools work with all networks that run on Windows NT. For example, with File Manager the user can browse and connect to any NetWare or Windows-networking server on the network.

With the Client Service for NetWare, a Windows NT Workstation computer can access file and print resources on NetWare servers as easily as it accesses resources on Windows-based networking servers. With the Gateway Service for NetWare, a Windows NT Server computer can not only access NetWare file and print resources, but also share these resources with Windows-based networking clients that have no NetWare connectivity software. To the Microsoft networking clients, the NetWare resource looks like any other shared resource on the Windows NT Server computer.

The Windows NT architecture includes an open interface called the multiple provider router (MPR) that enables consistent access to third-party network file systems. The MPR makes all file systems, regardless of type and physical location, accessible through the same set of file-system application programming interfaces (APIs). Applications (and components of the Windows NT shell) make file-system requests through the Windows NT Win32 API. The MPR ensures that requests are directed to the proper file system: local file requests are sent to the local disk, remote requests to Windows-based servers are sent to the proper server by the Windows NT redirector, and requests to NetWare servers are sent to the appropriate server by the Client or Gateway Service for NetWare.

For more information about NWLink and the Client and Gateway Services for NetWare, see the *Windows NT Installation Guide* or *Windows NT Server Services for NetWare Networks.*

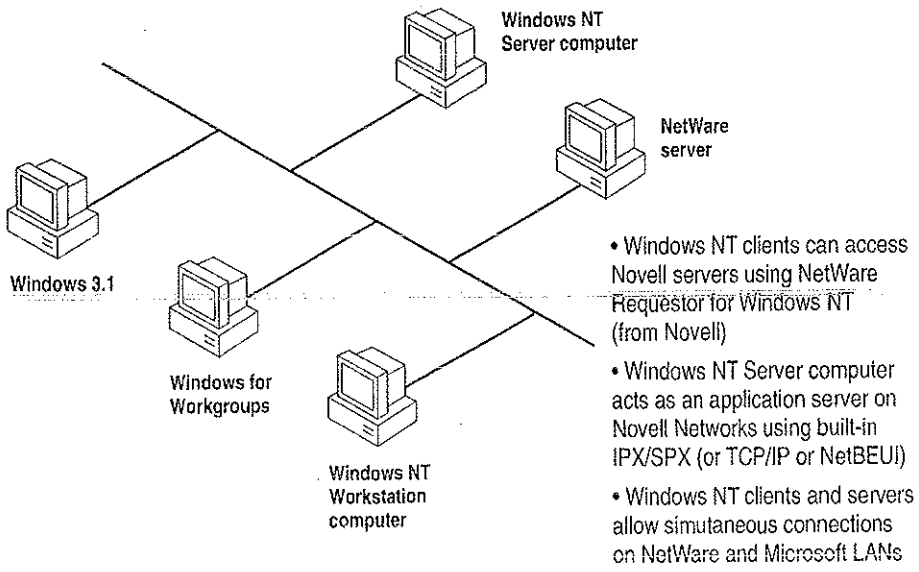


Figure 2.2 Windows NT Computers as NetWare Clients or Application Servers

Additional Considerations Regarding Mixed Networking Environments

Before adding computers running Windows NT (or other non-NetWare operating systems) to a NetWare network, a NetWare administrator should anticipate some of the potential problems that might arise.

One source of possible difficulty for NetWare administrators is that Windows NT NetWare clients do not run NetWare logon scripts. However, Windows NT can run its own logon scripts, and the ability of Windows NT to maintain persistent connections through logon scripts and user profiles provides much the same functionality as NetWare logon scripts in many instances.

Another area of difficulty is backing up Windows NT NetWare clients. Novell servers do not provide tape backup services for their Windows NT clients without third-party software. A Windows NT computer equipped with a supported tape drive can back up other Windows NT computers, as well as NetWare servers and computers running Windows networking software.

Finally, Windows NT can act as a client only for a NetWare server running NetWare version 3.*x* and earlier. Windows NT can access servers running NetWare 4.*x* through the server's Bindery Emulation Mode that emulates version 3.*x*.

Integrating Windows NT and UNIX Systems

With its advanced configuration management tools, Windows NT is especially suited for integrating with most of the UNIX variants that are likely to be found on many networks. Among the Windows NT features that make it easy to interoperate with UNIX systems are:

- Built-in TCP/IP protocol and utilities.
- Support for character and graphics terminal emulation.
- Advanced file transfer and data sharing capabilities.
- Distributed processing support.
- Application portability.

The following sections provide further information about these features. With DHCP and WINS, combined with the DNS server and other TCP/IP tools provided in this resource kit, integrating Windows NT and UNIX systems is easier than ever. For full details on TCP/IP in the Windows NT environment, see Part III, "TCP/IP," and Part IV, "Windows NT and the Internet."

TCP/IP Protocol

At the protocol level, Windows NT includes a fast, robust implementation of the Transport Control Protocol/Internet Protocol (TCP/IP) protocol stack, the most commonly used protocol among UNIX systems. Using TCP/IP, Windows NT computers can communicate with UNIX systems without additional networking software. (TCP/IP also provides efficient communication on wide-area networks, even when no UNIX systems are involved.) The TCP/IP protocol stack for Windows NT is NDIS-compliant and so can be used in conjunction with NetBEUI Frame (NBF) and other NDIS-compliant protocols. It includes an internet protocol (IP) router, serial line internet protocol (SLIP), and point-to-point protocol (PPP) support.

In addition to the TCP/IP protocol itself, Windows NT also includes more than a dozen TCP/IP utilities that make it easier for experienced UNIX users to access UNIX systems from Windows NT and to administer the TCP/IP networking on their own computer. Additional tools are included on the CD-ROM accompanying this resource kit.

Windows NT also provides facilities for integrating computers running Windows NT into networks managed through Simple Network Management Protocol (SNMP), which is commonly used to manage TCP/IP networks. Through its SNMP service, a Windows NT computer can report its current status to an SNMP management system on a TCP/IP network, either in response to a request from a management system or automatically when a significant event occurs on the Windows NT computer.

For more information, see Part III, "TCP/IP."

Character and Graphics Terminal Support

The TCP/IP Telnet utility is built into the Windows Terminal accessory to make it easy for a Windows NT computer to have character-oriented terminal access to UNIX systems via TCP/IP. Telnet provides basic terminal emulation of TTY (scrolling), as well as emulation of DEC VT-100 (ANSI) and VT-52 terminals.

Even in the traditionally character-oriented UNIX environment, many applications are moving to graphical user interfaces. X Windows is a commonly used standard for graphical interfaces in networked UNIX environments. A number of third-party companies are also developing X Servers to enable Windows NT users to access and run X-based applications on UNIX systems. (In X Windows terminology, an X Server runs on a client workstation to provide graphics output on behalf of an X Client program running on an applications server.) Several third-party vendors are also developing X Client libraries for Windows NT as well; this eventually will enable UNIX (or other systems with X Server capabilities) to access client-server applications running on a Windows NT computer. Companies developing X Servers and X Client libraries for Windows NT include Hummingbird, Congruent, and Digital Equipment Corporation.

File Transfer and Data Sharing

One of the fundamental reasons for connecting computers on a network is to enable them to exchange files and data. Windows NT supports standard facilities for transferring files and sharing data between Windows NT and UNIX systems.

Included with Windows NT itself are both client and server versions of File Transfer Protocol (FTP). FTP makes it possible for Windows NT computers to exchange files with diverse systems, particularly UNIX systems.

Where more advanced data sharing capabilities are required, computers running Windows NT can access data on UNIX systems (including data on remotely mountable file systems, such as NFS, RFS, and AFS) through Microsoft LAN Manager for UNIX (LMU), an implementation of Microsoft Windows networking for servers running UNIX variants. LMU is based on server message blocks (SMBs), a set of protocols developed by Microsoft that are now part of the X/Open standard.

Finally, a number of third-party companies (including NetManage, Beame and Whiteside, Intergraph, and Process Software) have developed versions of Sun's Network File System (NFS) for Windows NT. NFS is a widely used tool for sharing files among various UNIX systems.

Distributed Processing Support

As more and more enterprises adopt the client-server paradigm for their networks, standards-based distributed processing becomes a key factor in the success of that effort. Windows NT provides direct support for several types of industry-standard distributed processing.

The Remote Procedure Call (RPC) facility of Windows NT is wire-compatible with the Open Software Foundation's Distributed Computing Environment (DCE) RPC. Using this RPC, developers can create applications that include not only Windows NT computers, but all systems that support DCE-compatible RPCs, such as systems from Hewlett Packard® and Digital Equipment Corporation.

In addition to RPCs, Windows NT supports Windows Sockets. Windows Sockets provides an API that is compatible with Berkeley-style sockets, a mechanism that is widely used by different UNIX versions for distributed computing.

For more information about RPC and Windows Sockets, see Chapter 1, "Windows NT Networking Architecture."

Perhaps most importantly, Windows Open Services Architecture (WOSA), whose development is being led by Microsoft, specifies an open set of APIs for integrating Windows-based computers with back-end services on a broad range of vendors' systems. WOSA consists of an extensible set of APIs that enable Windows-based desktop applications to access available information without having to know anything about the type of network in use, the types of computers in the enterprise, or the types of back-end services available. As a result, should the network, computers, or services change, desktop applications built using WOSA won't require rewriting. The first two WOSA components address database and electronic messaging: Open Database Connectivity (ODBC) and Messaging API (MAPI). Work is underway for additional standards, including directory, security, and software licensing services.

Common Application Support

For most users, the key measure of interoperability is the ability to run the same applications on multiple platforms. Three key factors are furthering this type of interoperability between UNIX and Windows NT computers.

One factor is the relative ease with which many UNIX independent software vendors (ISVs) are able to port their high-end business and technical applications to the Win32 API of Windows NT. Aiding this process is the fact that most UNIX applications are written in standard C and so are readily adapted to other operating systems (such as Windows NT) for which standard C libraries have been developed. A wide variety of third-party porting aids (including items as diverse asXlibs, GNU tools, and X Client libraries) are available through commercial sources and from Internet. Because application developers are finding it so easy to port their traditionally UNIX-based applications to Windows NT, increasing numbers of such applications will be available for both UNIX platforms and for computers running Windows NT.

Another factor is that Windows NT fully supports programs that conform to the IEEE 1003.1-1990 standard commonly known as POSIX.1 (derived from Portable Operating System Interface). This standard defines a basic set of operating-system services available to character-based applications. Programs that adhere to the POSIX standard can be easily ported from one operating system to another. See Chapter 17, "POSIX Compatibility," of the *Windows NT Resource Guide* for more information.

Another factor is that third-party products from vendors such as Bristol Technologies are available that enable UNIX to run Windows-based applications. Additionally, there are third-party products, such as Consensys Portage, that enable Windows NT to run UNIX-based applications.

Connecting Windows NT and IBM SNA Hosts

A growing trend in many types of enterprises is downsizing mainframe-based applications to run on personal computer client-server networks. Many of these downsized applications will still require access to data and applications residing on IBM System Network Architecture (SNA) hosts, mainframes and midrange computers. Companies have invested large amounts of money, time, and effort in their host systems and so want to be able to make the best use of that investment even as they move toward distributed client-server computing. This section discusses how Windows NT computers can be connected to IBM SNA hosts to leverage the high capacity of SNA hosts in a distributed environment.

Basic Connectivity Using the Built-in DLC Protocol

A computer running Windows NT can communicate with IBM SNA hosts (as well as other network devices) across an Ethernet or token ring LAN through the Data Link Control (DLC) protocol that is built into Windows NT. The DLC protocol device driver enables a basic level of connectivity with other computers running the DLC protocol stack. For example, a Windows NT computer can connect to and communicate with an IBM mainframe through its 37x5 Front-end processor (FEP) using a 3270 terminal emulator and the DLC protocol. See Chapter 7, "Using DLC with Windows NT," for more information.

SNA Server for Windows NT

Although such simple one-to-one connections can suffice for many basic operations, most enterprises require more flexible connectivity between IBM host computers and local area networks (LANs). To meet this need, Microsoft SNA Server exploits client-server architecture to link desktop personal computers to IBM mainframe and midrange computers that are accessible using the Systems Network Architecture (SNA) protocols. The client personal computers can run Windows NT, Windows, MS-DOS, OS/2, or the Macintosh operating system and can use standard LAN protocols to connect to the server; only the computer running SNA Server must run the SNA protocol. Each personal computer user can have multiple 3270 and 5250 sessions for concurrent terminal and printer emulation, including file-transfer and Emulator High-Level Language API (EHLLAPI) applications. SNA Server for Windows NT also provides support for the following APIs for distributed SNA applications:

- Advanced Program-to-Program Communications (APPC) for applications that communicate peer-to-peer with other APPC applications using the LU 6.2 protocol
- Common Programming Interface for Communications (CPI-C) for applications that communicate peer-to-peer with IBM Systems Application Architecture (SAA) applications using the LU 6.2 protocol
- Common Service Verbs (CSV) for applications that communicate with NetView and enable tracing of API calls
- Logical Unit APIs (LUA) for applications (using LUA/Request Unit Interface or LUA/Session Level Interface APIs) that need direct access to LU 0, 1, 2, and 3 data streams

The client-server architecture of SNA Server makes it possible to off-load communications processing from client systems, permitting them to use their system resources more efficiently. Client personal computers do not have to run one protocol to access the LAN and another to access the SNA host. Instead, each personal computer can run Microsoft-based networking (named pipes), TCP/IP, IPX/SPX, AppleTalk®, or Banyan® VINES®, within a single-protocol or mixed network, to access the SNA server. The SNA server routes the connection to the appropriate host computer via the SNA protocol. The SNA server automatically balances the user load across multiple host connections and servers to provide optimal throughput.

The client-server architecture also provides Windows NT-based applications with the ability to access information on IBM mainframes and midrange computers. For example, using SNA Server, mail servers can access PROFS, and Microsoft SQL Server can access DB2 information.

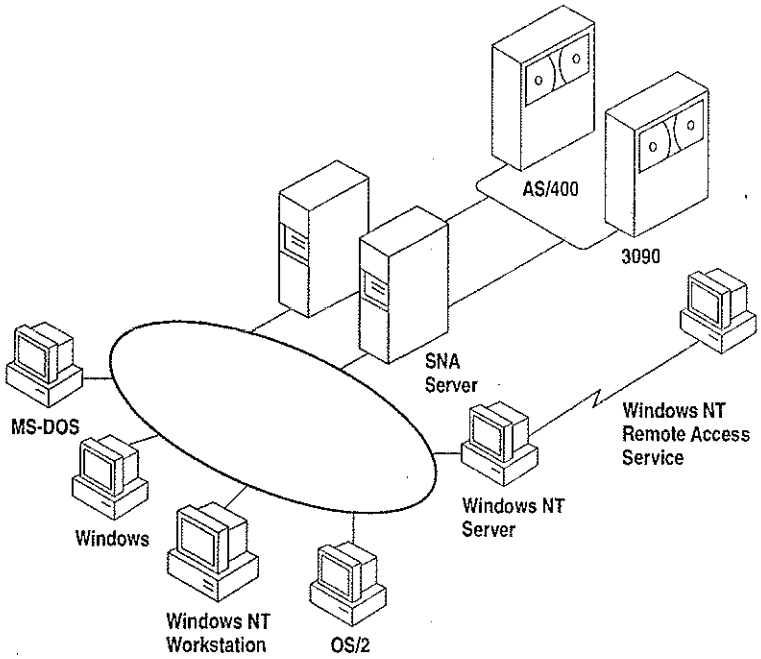


Figure 2.3 SNA Server Connecting LANs to IBM Host Computers

DSPU Support

In addition to standard personal computer connections, SNA Server supports Downstream Physical Units (DSPUs), any SNA device or personal computer running a full PU 2.0 (SNA cluster controller) protocol stack. These systems use the SNA server as a concentrator gateway for connecting to IBM hosts. Examples of some of the DSPU clients that SNA Server supports are IBM OS/2 Communications Manager/2 clients and IBM 3174 cluster controllers. The DSPU protocols that SNA Server supports are DLC over token ring or Ethernet, Synchronous Data Link Control (SDLC), and OSI-standard X.25/QLLC (Qualified Logical Link Control).

NetView Support

SNA Server provides API support for bidirectional communications with NetView, IBM's mainframe-centered network management system. SNA Server can send application- or system-defined Windows NT event-log messages to NetView and can enable Windows NT commands to be executed from the NetView console. For example, if an SNA Server database is stopped on the LAN, an alert can be sent to the NetView console. A data center operator can then send a command from the NetView console to the Windows NT computer to restart the server.

SNA Server also supports Response-Time Monitor (RTM) and user-defined alerts for third-party 3270 emulators.

Centralized Management

Network administrators can administer all SNA servers from a centralized location, such as from a LAN workstation or a NetView console. For example, a company with offices in several cities could have one or more SNA Servers at each site. The MIS department at corporate headquarters can manage all of these SNA servers, performing all administrative functions remotely.

Integration with Windows NT

SNA Server is supported on all the hardware platforms supported by Windows NT. SNA Server relies on the built-in security of Windows NT, so administrators need to manage only a single set of user accounts. SNA Server also is fully integrated with Windows NT system monitoring and management services, and provides automatic server and connection fault tolerance. SNA Server for Windows NT is completely 32-bit and multithreaded for maximum performance, scalability, and reliability.

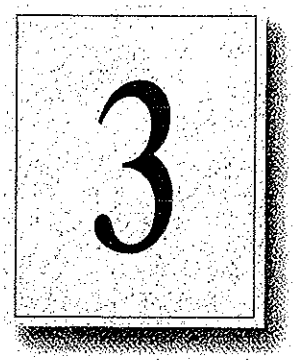
SNA Server is fully compliant with Microsoft's Windows Open Services Architecture (WOSA), providing a consistent interface to enterprise computing environments and hiding the complexities of connectivity from applications.

Server Capabilities

SNA Server provides for as many as 250 simultaneous host connections by each server and up to 2,000 users and 10,000 sessions per server.

CHAPTER 3

Windows NT User Environments



Each user on a Windows NT network works in a unique environment. The user environment is composed of such things as the file and print resources that are available, the configuration of Program Manager icons, screen wallpaper or background, automatic network connections, and applications that run on startup. One important element of the user environment is a directory assigned to a user or to a defined user group on either a workstation or a server where the user can store files. This directory is called a *home directory*.

A user's environment is determined primarily by a *user profile*, which you can create and maintain on a Windows NT Server computer using the User Profile Editor administrative tool. For information on the User Profile Editor, see the *Windows NT Server System Guide*. Some elements of the user environment are more easily controlled by creating a script that is executed whenever the user logs on to a Windows NT Workstation computer or a Windows NT Server computer. Such a script is called a *logon script*.

This chapter explains how to create home directories and logon scripts. It also describes special parameters you can use in logon scripts so the same script runs in different user environments with the expected result for each individual user.

Home Directories

A home directory is a private storage space assigned to a user or group of users. Users typically store their private data in their home directory, and they can normally restrict or grant access to other users. When a user opens a Command Prompt window, the default directory is the user's home directory. The home directory can also be specified as the default working directory for applications.

If hard disk space on your network's client workstations is limited, you might want to assign each user a home directory on a Windows NT Server computer. Or, if you want to limit a user's access to the files and directories on a workstation, you can create a home directory on the workstation and give the user only List permission on all other directories.

Assigning a Home Directory

On a Windows NT Workstation computer, home directories are assigned in User Manager. On a Windows NT Server computer, home directories are assigned in User Manager for Domains. The home directory that is used depends on whether the user logs on to the workstation account or the domain account.

The home directory can be specified by a local path name, such as C:\USERS\BILL, or by a universal naming convention (UNC) name, such as \\MYSERVER\USERS\BILL. The UNC name is the better option for large networks, because the system administrator can more easily see where users' home directories are located.

By default, the home directory is the \USERS\DEFAULT directory that is created during installation of Windows NT. The most common way to assign a home directory is to specify it using the following syntax:

\USERS*accountname*

–Or–

\USERS*groupname*

where *accountname* is the username given to the account or where *groupname* is the name of a local or global group whose members all share the same home directory.

To assign a home directory

1. From the Administrative Tools group in Program Manager, double-click the User Manager or User Manager for Domains icon, depending on whether you are using a Windows NT Workstation computer or Windows NT Server computer.
2. Double-click the name of the user or group whose home directory you want to assign.

   The User Properties dialog box appears.
3. Choose the Profile button to display the User Environment Profile dialog box.
4. Enter the full path specification of the home directory in the Local Path box of the Home Directory group box.

   If you are specifying a remote home directory, specify a disk drive letter and provide the full path (not just the sharename) to the directory. For instance, if the home directory is \JEFFHO on share \\SERVER1\USERS, enter the path \\SERVER1\USERS\JEFFHO.

Note If you want the user to control access to the home directory, give the user Full Control permission for the directory. You will probably also give members of the Administrator or Domain Admins group Full Control permission and give all other users No Access or List permission only. For information on setting directory permissions, see Chapter 4, "File Manager," of the *Windows NT System Guide.*

If you specify a nonexistent directory when you define or modify a user account, Windows NT automatically creates the directory.

When a user logs on to a domain, Windows NT automatically tries to connect to the home directory defined in the user's domain account using the following rules.

- If the computer where the home directory resides is not available, the user's home directory on the local computer is used (if there is one).
- If the home directory specified does not exist or the user does not have a home directory, then the user is connected to the \USERS\DEFAULT directory of the computer that processes the logon.
- If the \USERS\DEFAULT directory does not exist, then the user is connected to the \USERS directory.

Note Windows NT Server connects the user to the home directory specified in the domain user account only when the logon is from a Windows NT or Windows for Workgroups 3.11 client. LAN Manager 2.*x* clients can connect to the home directory by typing the following command at the command prompt:

net use <*drive*>**: /home**

Specifying the Home Directory in a Logon Script or Batch File

Windows NT provides three environment parameters you can use in a logon script or other batch file to specify the location of the home directory, or in Program Manager to specify the working directory of an application. Logon scripts are described later in this chapter. If a home directory has not been defined for the user, the default values are used as shown in the following table.

Table 3.1 Environment Parameters for Logon Scripts and Batch Files

Parameter name	Definition	Default value
%homedrive%	Drive where the home directory is located	Drive where the Windows NT system files are installed
%homepath%	Path name of the home directory	\USERS\DEFAULT
%homeshare%	UNC name of the shared directory containing the home directory, or a local or redirected drive letter	No default value

If the \USERS\DEFAULT directory does not exist on the drive specified by the **%homedrive%** parameter, the value of the **%homepath%** parameter is set by default to the \USERS directory on that drive. If the \USERS directory does not exist, the **%homepath%** parameter is set to the root directory specified by the **%homedrive%** parameter.

When the user opens a Command Prompt window, the default directory is the equivalent of **%homedrive% %homepath%**. If a user's home directory is specified on a remote computer and that computer is not available, the default directory of the Command Prompt on a Windows NT Workstation computer is the user's home directory on the local workstation.

You might also want to specify the working directory of each application as **%homedrive% %homepath%**. That way, all File Open and Save As dialog boxes default to the user's home directory.

Logon Scripts

A logon script is a .BAT, .CMD, or .EXE file that is run automatically when a user logs on at a Windows NT network client running either Windows NT Workstation or MS-DOS. A logon script can automatically configure the user's environment to perform such tasks as making network connections, running applications, and setting environment variables upon startup.

User profiles can do everything that logon scripts can do, and more. However, there are several reasons to use logon scripts instead of, or in addition to, user profiles:

- You have users that use MS-DOS workstations. User profiles work only on Windows NT workstations.
- You want to manage part of the user's environment, such as network connections, without managing or dictating the entire environment.
- You use only personal profiles, and you want to create common network connections for multiple users.
- You already have LAN Manager 2.*x* running on your network, and you want to continue to use the logon scripts you created for that system.
- Logon scripts are easier to create and maintain than user profiles.

You can assign a different logon script to each user or create logon scripts for use by multiple users. Whenever that user logs on, the logon script is downloaded and run. To assign a user a logon script, you designate the name of the logon script file in the user environment profile defined in User Manager on a Windows NT Workstation computer, or User Manager for Domains on a Windows NT Server computer. Specify only the filename, not the full pathname.

The default file extension for logon scripts is .CMD for client workstations running OS/2 2.1 and .BAT for all other client computers. You can define a different file type as the logon script by specifying the file extension. If the same logon script must run at both Intel-based and RISC-based workstations, it must be a .BAT file that runs the appropriate .EXE file or files on the workstation. Use the **%processor%** parameter in the logon script to run the appropriate .EXE file no matter which processor is being used.

You specify the path to the logon script using the Server option of Control Panel. For detailed information, see online Help. By default, Windows NT looks for logon scripts on the primary domain controller in the directory *systemroot*\SYSTEM32\REPL\IMPORT\SCRIPTS, where *systemroot* is the disk drive and directory in which Windows NT Server was installed.

If you use logon scripts in a domain with more than one domain controller, you should replicate the logon scripts to all the backup domain controllers. All servers in a domain can authorize logon requests, and the logon script for a user must be located on the server that approves the user's logon request. By replicating logon scripts, you ensure that logon scripts are always available to users, yet you still need to maintain only one copy of each script.

The filename for each user's logon script is defined with other user account information in User Manager for Domains. If you change the path to the logon scripts, this change is not replicated to the client workstations. The path must be updated manually in the Server option of Control Panel for each client computer.

To simplify the replication of logon scripts, Windows NT Server creates a \SCRIPTS subdirectory under both the default import and export directories used for replication. If you replicate logon scripts, you must be sure to use the Server option of Control panel or Server Manager to change the logon script path to *systemroot*\SYSTEM32\REPL\IMPORT\SCRIPTS or *systemroot*\SYSTEM32\REPL\EXPORT\ SCRIPTS, as appropriate. For more information, see the Server Manager chapter of the *Windows NT Server System Guide*.

When you use replicated logon scripts, you identify one of the domain controllers as the export server and all the others as import servers. The export server for the logon scripts is normally, but does not have to be, the primary domain controller (PDC).

Logon Scripts and LAN Manager 2.x

When a user at a workstation running LAN Manager 2.*x* logs on to a Windows NT Server computer, LAN Manager tries to run the user's logon script. LAN Manager 2.*x* does not, however, recognize the logon script parameters described earlier in this chapter. Logon scripts for LAN Manager 2.*x* workstations should instead use the **NetWkstaGetInfo** or **NetUserGetInfo** parameter to obtain the necessary values.

Logon Scripts and Windows for Workgroups

By default, Windows for Workgroups does not run a logon script when a user logs on to a Windows NT Server computer. To run a logon script from Windows for Workgroups, you must configure Windows for Workgroups to log on to the Windows NT domain on startup.

▶ **To log on to the Windows NT domain on startup from a Windows for Workgroups computer**

1. From Control Panel, double-click the Network option.
2. In the Microsoft Windows Network dialog box, choose the Startup button to display the Startup Settings dialog box.
3. In the Options for Enterprise Networking box, select the Log On To Windows NT or LAN Manager Domain checkbox.

4. In the Domain box, type the name of the Windows NT domain you want to log on to.
5. In the Startup Settings dialog box, choose the OK button.
6. In the Microsoft Windows Network dialog box, choose the OK button.

Windows for Workgroups does not recognize logon script parameters, and application programming interface (API) calls made from a logon script return an error.

Troubleshooting Logon Scripts

Use this list to troubleshoot the most common problems with logon scripts:

- Make sure the logon script is in the directory specified in the Server option of Control Panel. When Windows NT is installed, the logon script directory is as follows:

  ```
  systemroot\system32\repl\import\scripts
  ```

  The only valid path option is a subdirectory of the default logon script directory. If the path is any other directory or it uses the environment variable **%homepath%**, the logon script fails.
- If the logon script is on an NTFS partition, make sure the user has Read permission for the logon script directory. If no permissions have been explicitly assigned, the logon script might fail without providing an error message.
- Make sure the logon script has a filename extension of either .CMD or .BAT. The .EXE extension is also supported, but only for genuine executable programs. If you use a nondefault file extension for your processor, be sure to specify it with the filename of the logon script.

  Attempting to use the .EXE extension for a script file results in the following error message:

  NTVDM CPU has encountered an illegal instruction.

  If this error message appears, close the window in which the logon script is running.
- If the logon script is to run on a Windows for Workgroups computer, make sure the Windows NT domain name is specified as a startup option in the Network option of Control Panel.
- Make sure any new or modified logon scripts have been replicated to all domain controllers. Replication of logon scripts happens periodically, not immediately. To manually force replication, use Server Manager. See the Server Manager chapter of the *Windows NT Server System Guide* for detailed information.

Environment Parameters for Logon Scripts

If you want to use the same logon script for various users, you can use the environment parameters shown in the following table to reduce development and maintenance time.

Table 3.2 Environment Parameters for Logon Scripts

Parameter	Description
%homedir%	Redirected drive letter on user's computer that refers to the share point for the user's home directory
%homedrive%	Local or redirected drive where the home directory is located
%homepath%	Path name of the home directory
%homeshare%	UNC name of the shared directory containing the home directory, or a local or redirected drive letter
%os%	The operating system of the user's workstation
%processor_architecture%	The processor architecture (such as Intel) of the user's workstation
%processor_level%	The type of processor (such as 486) of the user's workstation
%userdomain%	The domain containing the user's account
%username%	The user name of the user

Environment Variables for Logon Scripts

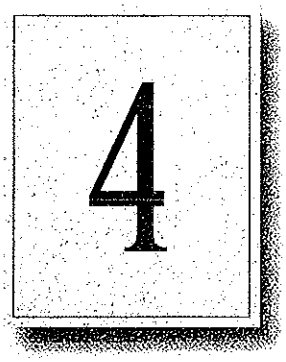
The environment variables shown in the following table can be set by the logon script.

Table 3.3 Environment Variables for Logon Scripts

Variable	Description
ComSpec	Directory for CMD.EXE
LibPath	Directories to search for dynamic link libraries (DLLs)
OS2LibPath	Directories to search for dynamic link libraries (DLLs) under OS/2 subsystem
Path	Directories to search for executable program files
WinDir	Directory in which Windows NT is installed

CHAPTER 4

Network Security and Administration



Each domain and computer in a workgroup maintains its own user accounts information. Even on a multidomain network, if account information for an individual user is coordinated across all parts of the network, the user can access any server or domain with a single logon. If the user's accounts are allowed to become unsynchronized, the following problems can occur:

- The user can't browse a domain or server for which he or she has permissions.
- The user can't access a shared resource.
- The user must type a password each time he or she browses or tries to access a resource.

This chapter provides tips for helping you avoid problems related to network logon. It describes how user accounts and other security information are maintained within workgroups and domains and how security information can be shared by trusted domains.

Before reading this chapter, be sure to read the *Windows NT Server Concepts and Planning Guide* for a thorough discussion of domain organization strategies and user environment management techniques.

Windows NT User Accounts

Windows NT needs only a single logon, even for a heterogeneous networking environment, in part because security in Windows NT is assigned by user rather than by resource. Resource-based security models require a separate password for each resource a user wants to access.

In Windows NT, the network administrator creates an account for each user wanting to use network resources. As described in Chapter 2, "Windows NT Security Model," of the *Windows NT Resource Guide,* Windows NT maintains a user account containing a unique security ID within the user accounts database. Windows NT also keeps track of permissions and user rights for the user. When a person logs on, the *Security Accounts Manager* (SAM) checks the user's logon information against data in its user accounts database to authenticate the logon. Then, when access is granted, the *Local Security Authority* (LSA) creates a security access token for that user.

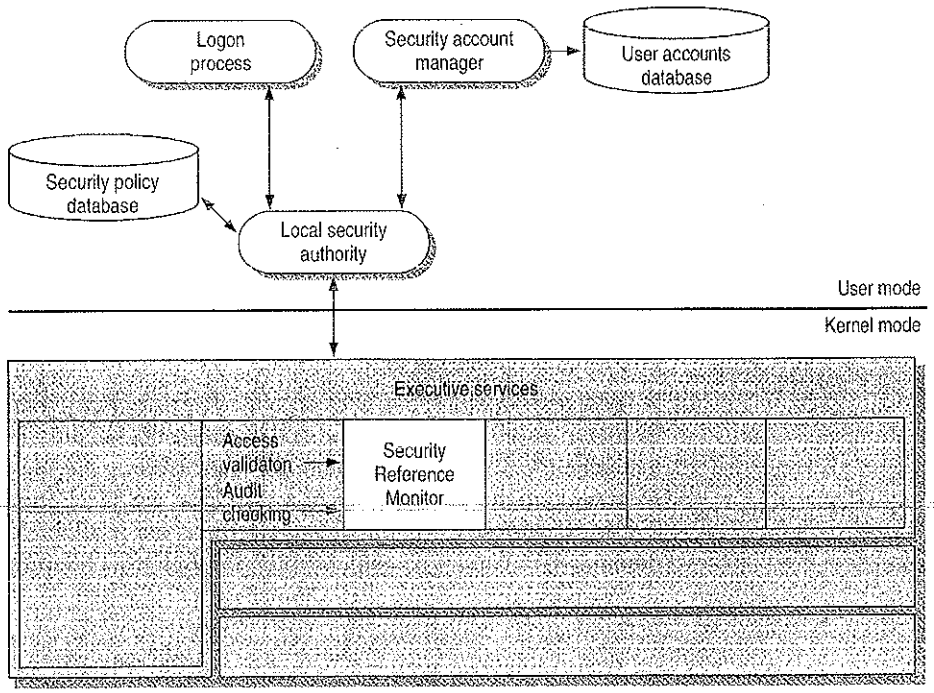


Figure 4.1 Windows NT Security Model

Note A user who forgets his or her password might assume that he or she can gain access to a resource via the Guest account; this is not the case. Because Windows NT recognizes the username, it compares the user's logon information only with the account information for that username. If the password does not match, no access is granted.

By default, the Guest account on Windows NT Server is disabled so that only those users with recognized accounts can access the system. As described in the *Windows NT Server Concepts and Planning Guide*, Windows NT uses the Guest account for people with an unrecognized user account, including users logging on from untrusted domains. Domains and trust relations are explained later in this chapter.

Depending on the way your corporation's network is organized, a given user might, in fact, have more than one account, perhaps one granting access to the local computer or workgroup and another for domains on the network. The user account database used to authenticate a logon doesn't necessarily reside on the user's local computer. Its location depends on whether the computer is part of a workgroup or a domain and whether the user is logging on to the local computer, to the home domain, or to another domain.

In the Windows NT security model, there are two types of user accounts:

- A *global user account* is a normal user account that fits into the Windows NT model described in this chapter. User accounts on Windows NT Workstation computers and on Windows NT Server computers that are not domain controllers are global accounts. Global users are authenticated by the primary domain controller (PDC) or backup domain controller (BDC) on a domain, or through trust relationships.
- A *local user account* is a user account that fully participates in a domain but is available only by remote logon and is authenticated only by user information available locally on the machine that is processing the logon. For example, a local user might be a member of a Windows for Workgroups, LAN Manager 2.*x*, or Novell network. Local user accounts are available only within their domain; they cannot be authenticated through trust relationships.

Workgroups and Domains

A *workgroup* is simply an organizational unit, a way to group computers that don't belong to a domain. In a workgroup, each computer keeps track of its own user and group account information and does not share this information with other computers. Each Windows NT computer that participates in a workgroup maintains its own security policy and security account databases.

Users on a workgroup are considered global users, as explained in the previous section. Logons to another computer are authenticated on the remote computer only by valid username and password.

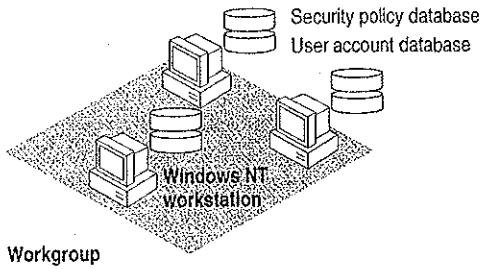


Figure 4.2 Computers Participating in a Workgroup

A workgroup is a good network configuration for a small group of computers with not many user accounts, where network administration is not an issue, or in an environment with a mix of Microsoft networks that does not include Windows NT Server computers.

A *domain* is a group of servers that share common security policy and user account databases. One Windows NT Server computer acts as the primary domain controller (PDC), which maintains the centralized security databases for the domain. Other Windows NT Server computers in the domain function as backup domain controllers and can authenticate logon requests. Domains can also contain Windows NT Server computers that do not act as domain controllers, Windows NT Workstation computers, LAN Manager 2.*x* servers, and other workstations such as those running Windows for Workgroups and MS-DOS. Users of a Windows NT Server domain are authenticated by the primary domain controller or by a backup domain controller. Logon credentials include the username, password, and domain name.

With Windows NT, administrators have full centralized control over security. To eliminate any single point of failure on a Windows NT Server domain, the user account database, including the logon scripts (which are discussed in Chapter 3, "Windows NT User Environments") is automatically replicated to the backup domain controllers.

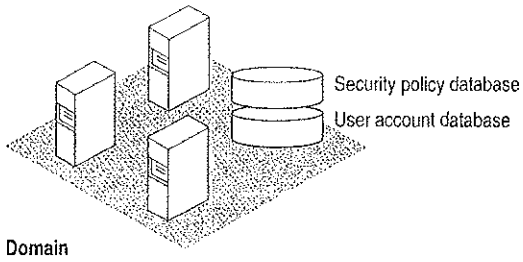


Figure 4.3 Computers Participating in a Domain

Domains and workgroups can interoperate and are identical in terms of browsing. If a Windows NT computer is not participating in a domain, it is by default part of a workgroup (even if the workgroup is only one computer) and can be browsed as part of that workgroup. For more information, see Chapter 5, "Windows NT Browser."

LAN Manager 2.x Domains

A Windows NT computer can connect to standalone LAN Manager 2.*x* servers and LAN Manager 2.*x* servers participating in a LAN Manager 2.*x* domain. LAN Manager 2.*x* and Windows NT computers interoperate because they both use *server message blocks* (SMBs) to communicate between the redirector and server software. The NetBEUI Frame (NBF) and TCP/IP protocols used by Windows NT are also interoperable with NetBEUI and TCP/IP protocols written for LAN Manager 2.*x*.

Note LAN Manager 2.x servers can act as backup domain controllers in a Windows NT Server domain. Both local and global user accounts are replicated to LAN Manager 2.*x* servers acting as BDCs. Because LAN Manager 2.*x* does not support trust relationships or local groups, a LAN Manager 2.*x* server can never be a primary domain controller.

Avoiding Multiple PDCs

A common configuration problem is having multiple PDCs on a domain. This type of configuration problem is described in the following scenario.

A system administrator installs a Windows NT Server computer called \\MAIN_UNIT, which is designated during installation as the PDC of a domain called MyDomain. Later, the system administrator shuts down and turns off the PDC, \\MAIN_UNIT. Then the system administrator installs another server, called \\SECOND_UNIT, which is also installed as the PDC. Because \\MAIN_UNIT is not currently on the network, MyDomain has no PDC, and the installation of \\SECOND_UNIT proceeds without error.

Now the system administrator turns \\MAIN_UNIT back on. When the Netlogon service (described later in this chapter) discovers another PDC on the network, it fails, and \\MAIN_UNIT can no longer participate in the domain.

The system administrator now has a serious problem. It is not possible to simply demote \\MAIN_UNIT from a PDC to a BDC and continue. The *Security ID* (SID) for \\MAIN_UNIT will not be recognized by the current PDC, \\SECOND_UNIT. In fact, \\MAIN_UNIT cannot join MyDomain in any capacity. This happens because when a PDC is created, a unique domain SID is also created. All BDCs and user accounts within the domain share this domain SID as a prefix to their own SIDs. When \\SECOND_UNIT is installed as a PDC, its SID prefix is different from that of \\MAIN_UNIT, and the two computers can never participate in the same domain.

In addition, the system administrator cannot change the name of \\MAIN_UNIT and rejoin MyDomain, because the SID is fixed once the Windows NT Server is installed. If \\MAIN_UNIT is to be the PDC of MyDomain, the system administrator must shut down both \\MAIN_UNIT and \\SECOND_UNIT, start up \\MAIN_UNIT, and then reinstall Windows NT Server on \\SECOND_UNIT, designating it a BDC during setup.

To avoid this problem, \\SECOND_UNIT should be installed as a backup domain controller while \\MAIN_UNIT is running. If \\MAIN_UNIT is taken offline at this point, \\SECOND_UNIT can be promoted to PDC. (In general, it should not be necessary to designate a new PDC unless the original PDC is going to be down for a long time.) When \\MAIN_UNIT is ready to go online again, \\SECOND_UNIT can be demoted to a BDC. The SID for \\MAIN_UNIT is recognized by \\SECOND_UNIT, and when \\MAIN_UNIT is restarted, it becomes the PDC again.

Interdomain Trust Relationships

With Windows NT Server, the user accounts and global groups from one domain can be used in another domain. When a domain is configured to allow accounts from another domain to have access to its resources, it effectively *trusts* the other domain. The *trusted* domain has made its accounts available to be used in the *trusting* domain. These trusted accounts are available on Windows NT Server computers and Windows NT Workstation computers participating in the trusting domain.

Hint By using trust relationships in your multidomain network, you reduce the need for duplicate user account information and reduce the risk of problems caused by unsynchronized account information.

The *trust relationship* is the link between two domains that enables a user with an account in one domain to have access to resources on another domain. The trusting domain is allowing the trusted domain to return to the trusting domain a list of global groups and other information about users who are authenticated in the trusted domain. There is an implicit trust relationship between a Windows NT Workstation participating in a domain and its PDC.

The following figure illustrates a trust relationship between two domains, where the London domain trusts the Topeka domain.

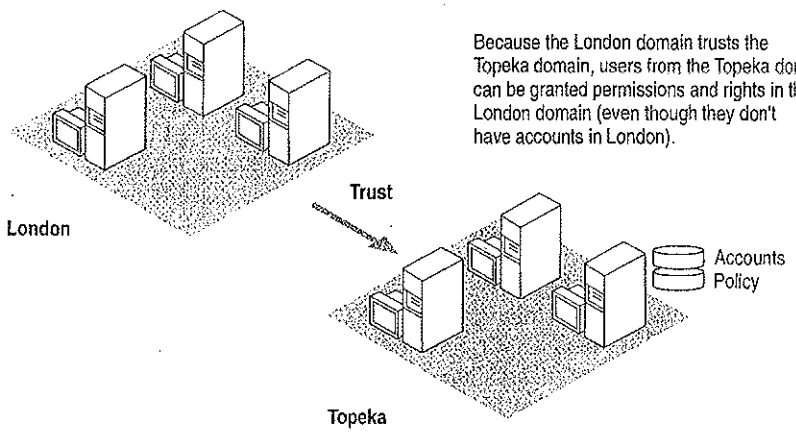


Figure 4.4 Trusted Domain

In this example, the following statements are true because the London domain trusts the Topeka domain:

- Users defined in the Topeka domain can access resources in the London domain without creating an account within that domain.
- Topeka appears in the From box at the initial logon screen of Windows NT computers in the London domain. Thus, a user from the Topeka domain can log on at a computer in the London domain.

When trust relationships are defined, user accounts and global groups can be given rights and permissions in domains other than the domain where these accounts are located. Administration is then much easier, because you need to create each user account only once on your entire network, and then the user account can be given access to any computer on your network (provided you set up domains and trust relationships to allow it).

Note Trust relationships can be configured only between two Windows NT Server domains. Workgroups and LAN Manager 2.*x* domains cannot be configured to use trust relationships.

Changes to Computers in the Trusting and Trusted Domains

When one domain is permitted to trust another, User Manager for Domains creates an interdomain trust account in the Security Accounts Manager (SAM) of the trusted domain. This account is like any other global user account, except that the USER_INTERDOMAIN_TRUST_ACCOUNT bit in the control field for the account is set. The interdomain trust account is used only by the primary domain controller and is invisible in User Manager for Domains. The password is randomly generated and is maintained by User Manager for Domains.

When this trust relationship is established, the Netlogon service on the trusting domain attempts discovery on the trusted domain, as described later in this chapter, and the interdomain trust account is authenticated by a domain controller on the trusted domain.

When one domain trusts another, a trusted domain object is created in the LSA of the trusting domain, and a Secret object is created in the LSA of the trusting domain.

Access to Files in a Trusting Domain

Users from the trusted domain can be given rights and permissions to objects in the trusting domain using File Manager, just as if they were members of the trusting domain. Subject to account privilege, users in the trusted domain can browse resources in the trusting domain.

For example, suppose the London domain trusts the Topeka domain. User EmilyP, who is a member of the Topeka domain, wants to access MYFILE.TXT, which is a file located on a Windows NT Server computer in the London domain. When EmilyP attempts to log on to the server in London, her user account information is not transferred to the London domain's user database. Because London trusts Topeka, the London domain has access to user information in the Topeka domain's user account database. Authenticating a user logon in this manner is called *pass-through authentication,* a concept that is discussed in greater detail later in this chapter.

One-way Trust Relationships

Trust relationships are defined in only one direction. In the previous example, just because the London domain trusts the Topeka domain does not mean that the Topeka domain trusts the London domain. For a two-way trust relationship, each domain must be configured to trust the other.

Trust relationships are not transitive. For example, if the London domain trusts the Topeka domain and the Topeka domain trusts the Melbourne domain, that does not mean that the London domain trusts the Melbourne domain. For the London domain to trust the Melbourne domain, a trust relationship must be explicitly established.

Users and computers from the trusting domain have no special status on the trusted domain. The names of trusting domains do not appear in the From box of the Logon dialog box, nor do users from the trusting domain appear in the File Manager of computers in the trusted domain.

Setting Up Domains

The way you configure your network into domains depends on your administrative resources and the size of your network. This section describes the most common domain models:

- Single domain
- Master domain
- Multiple master domain
- Multiple trust

Single Domain

In the single domain model, there is only one domain. Because there are no other domains, there are no trust relationships to administer. This model is the best implementation for organizations with fewer than 10,000 users in which trust among departments is not an issue. This model offers centralized management of all user accounts, and local groups have to be defined only once. In an organization with multiple domains where there is no need to share information among domains, the best configuration is often multiple single domains.

If, however, you anticipate significant growth in your organization, you might want to consider a more flexible model, such as the multiple master domain model described later in this section. If your organization grows beyond 10,000 users, the single domain model can no longer support all your users, and there might be a great deal of administrative work involved in reconfiguring your user database.

Master Domain

In an organization with fewer than 10,000 users in which trust among departments is an issue, the master domain model is a suitable option. In this model, one domain, the master domain, is trusted by all other domains, but does not trust any of them. Trust relationships among the other domains can be defined and administered as necessary.

The master domain model offers the benefits of both central administration and multiple domains. In an organization with a number of departments, each department can administer its own resources, but user accounts and global groups still need to be defined only once, in the master domain.

As with the single domain model, however, the user population is limited to 10,000, because all user accounts are maintained in one place, the master domain. Further, local groups must be defined for each domain, which can require significantly more administration if you use local groups extensively.

Multiple Master Domain

For large organizations, or those which anticipate substantial growth, the multiple master domain model might be the best solution. In this model, there is more than one master domain, each of which trusts all the other master domains, and all of which are trusted by all the other domains. None of the master domains trusts any of the subdomains.

This model works best when computer resources are grouped in some logical fashion, such as by department or by location. Because a multiple master domain model can support as many as 10,000 users per master domain, it works well for large organizations. And because all the master domains trust each other, only one copy of each user account is needed.

The administrative requirements for a multiple master domain model can be considerably greater than for a single domain or master domain model. Local and global groups might have to be defined several times, there are more trust relationships to manage, and not all user accounts reside in the same domain.

Multiple Trust

In the multiple trust model, all domains trust all other domains. This model is the simplest to understand, but if many domains are involved it is the most complex to administer.

Like the multiple master domain model, the multiple trust model is scalable as the organization grows: it can support as many as 10,000 users for each domain (not for each master domain, as in the multiple master domain model). Because each domain has full control over its own user accounts, the multiple trust model can work well for a company without a centralized management information services (MIS) department. If, however, the organization has many domains, there can be a very large number of trust relationships to manage. And because domain administration is decentralized, it is harder to assure the integrity of global groups that other domains might use.

Local and Global Groups

You can place a set of users with the same administrative requirements into user groups. User groups make system administration much simpler, because you can assign all members of a group the same logon script, file rights and permissions, and user profile. If some aspect of the group's administrative requirements changes, you can make the change in just one place for all the users in the group.

User groups can be local or global. The terms *local group* and *global group* refer not to the contents of the group, but to the scope of the group's availability. A local group is available only on the domain controllers within the domain in which it is created, while a global group is available within its own domain and in any trusting domain. A trusting domain can, therefore, use a global group to control rights and permissions given members of a trusted domain.

Global Groups

A global group contains only individual user accounts (no groups) from the domain in which it is created. Once created, a global group can be assigned permissions and rights, either in its own domain or in any trusting domain. A global group is a good way to export a group of users as a single unit to another domain. For example, in a trusting domain you can grant identical permissions to a particular file to a global group, which then pertain to all individual members of that group.

Global groups are available only on Windows NT Server domains. When Windows NT Server is installed on a computer, it is configured with two predefined global groups:

- Domain Admins
- Domain Users

Local Groups

A local group is a good way to import a group of users and global groups from other domains into a single unit for use in the local domain. A local group can contain user accounts or global groups from one or more domains. The group can be assigned privileges and rights only within its own domain. Local groups created on a Windows NT Workstation computer or a Windows NT Server computer in a workgroup are available only on that computer.

The following predefined local groups are available on Windows NT Workstation and Windows NT Server computers:

- Administrators
- Users
- Guests
- Backup operators
- Replicator

The following additional predefined local groups are available only on Windows NT Server computers acting as primary or backup domain controllers:

- Account operators
- Print operators
- Server operators

Another predefined local group, Power Users, is available only on Windows NT Workstation computers or on Windows NT Server computers that are not acting as domain controllers.

Logons and Authentication

When you log on to a workgroup computer, your logon information is compared with the local user accounts database. When you log on to a computer that participates in a domain, you choose whether to log on locally, or to the domain. (If your domain trusts another domain, you can alternately choose to log on there.)

Note Windows NT Server computers store only domain accounts. To log on to a Windows NT Server computer, you must use a domain account.

For example, suppose AnnM has an account on a domain (MyDomain), as well as an account on a Windows NT workstation (MyWksta) belonging to that domain. When AnnM logs onto her workstation account, the local authentication software uses the information stored in the workstation user accounts database to authenticate the logon. If AnnM logs onto the domain from that workstation, the local authentication software sends the logon request to the domain for authentication. Although they share the same username, each account has a unique security ID.

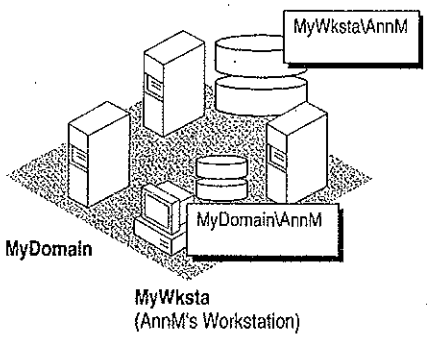


Figure 4.5 Logging On Locally Versus Logging On to the Domain

As described in Chapter 2, "Windows NT Security Model," of the *Windows NT Resource Guide,* the Local Security Authority (LSA) creates a security access token for each user accessing the system. This happens when the user logs on and is authenticated (that is, during interactive logon). The LSA also creates a security access token when a user establishes a connection from a remote computer. This procedure is called a *remote logon.*

For example, suppose AnnM logs on and is authenticated by her local computer and then wants to access a printer controlled by a Windows NT Server computer in domain MyDomain. When she tries to connect to the printer (assuming she hasn't already connected to some other resource in the domain), she is actually performing a remote logon. One of the servers in MyDomain checks the domain's central user accounts database for information to authenticate her account for the domain and then creates a security access token for AnnM, and allows AnnM access.

Note This type of scenario becomes complex when AnnM uses different passwords for different accounts. For example, if her local password doesn't match the password for her domain account, when she tries to browse the domain or connect to a resource in the domain, a message like the following is displayed on the screen:

```
System error 5 has occurred
Access is denied
```

While tools such as File Manager prompt for a valid password, the command-line interface and some applications simply deny access. It is always a better idea to have one set of credentials that apply everywhere in a trusted enterprise.

From an administrative viewpoint, it is important to understand where the user account information is stored. A user's account is either in a private local user accounts database or in a domain user accounts database shared by all the Windows NT Server computers in the domain.

The Netlogon Service

The Netlogon service provides users logging on with a single access point to a domain's primary domain controller and all backup domain controllers. The Netlogon service replicates any changes to the security database to all domain controllers in the domain, including the SAM, BuiltIn, and LSA databases described in Chapter 2, "Windows NT Security Model," of the *Windows NT Resource Guide*. The SAM database is limited only by the number of Registry entries permitted and by the performance limits of the computer hardware. The maximum number of accounts of all types the SAM database supports is 10,000.

The Netlogon service on a Windows NT Server computer fully synchronizes its user database when the domain controller is first installed, or when the domain controller is brought back online after being offline, and the PDC's change log is full when the server returns online.

The Netlogon service accepts logon requests from any client and provides complete authentication information from the SAM database. It can authenticate logon requests as a member of a trusting or trusted domain.

The Netlogon service runs on any Windows NT computer that is a member of a domain. It requires the Workstation service and the "Access This Computer from Network" right, which is set in User Manager on Windows NT Workstation computers or servers, or User Manager for Domains on domain controllers. A domain controller also requires that the Server service be running.

User Authentication

On a Windows NT Workstation computer or a Windows NT Server computer that is not a domain controller, the Netlogon service processes logon requests for the local computer and passes through logon requests to a domain server.

The Netlogon service processes authenticates a logon request in three steps:

1. Discovery
2. Secure channel setup
3. Pass-through authentication (where necessary)

Discovery

When a user logs on to a domain from a Windows NT Workstation computer or a Windows NT Server computer that is not a domain controller, the computer must determine the location of a domain controller in its domain. If the computer is part of a workgroup, not a domain, the Netlogon service terminates. (If the workstation is not connected to a network, Windows NT treats it like a member of a workgroup consisting of one member.)

When a Windows NT Workstation computer or a Windows NT Server computer that is not a domain controller starts up, it attempts to locate a Windows NT Server computer in each trusted domain. (There is an implicit trust between the client and domain controllers in its own domain.) In either case, the server located can be either a primary domain controller (PDC) or a backup domain controller (BDC). The act of locating a domain controller to connect to is called *discovery*. Once a domain controller has been discovered, it is used for subsequent user authentication.

When a domain controller is started up, the Netlogon service attempts discovery with all trusted domains. (Discovery is not necessary on the domain controller's own domain, because it has access to its own SAM database.) Each domain is called three times in intervals of five seconds before discovery fails. If a trusted domain does not respond to a discovery attempt, the domain controller attempts another discovery every 15 minutes until it locates a domain controller on the trusted domain. If the domain controller receives an authorization request for the trusted domain for which discovery has not yet been successful, it attempts another discovery immediately, no matter when the last discovery was attempted.

Secure Communication Channel

Before a connection between two Windows NT computers is allowed, each computer's Netlogon service must be satisfied that the computer at the other end of the connection is identifying itself correctly. To do this, each computer's Netlogon service issues and verifies challenge and challenge response information. When this information is successfully completed, a secure channel is established and a communication session set up between the two computers' Netlogon services. The session can be ended without terminating the secure channel. The secure channel is used to pass subsequent network API calls between the two computers. The secure communication channel is used to pass the username and encrypted password during pass-through authentication. Pass-through authentication is discussed in detail later in this chapter.

The Netlogon service maintains security on these communication channels by using user-level security to create the channel. The following special internal user accounts are created:

- *Workstation trust* accounts, which allow a domain workstation to perform pass-through authentication for a Windows NT Server computer in the domain, as described later in this chapter
- *Server trust* accounts, which allow Windows NT Server computers to get copies of the master domain database from the domain controller
- *Interdomain trust* accounts, which allow a Windows NT Server computer to perform pass-through authentication to another domain

The Netlogon service attempts to set up a secure channel when it is started, as soon as discovery is completed. Failing that, Netlogon retries every 15 minutes or whenever an action requiring pass-through authentication occurs. To reduce network overhead among trusted domains, the Netlogon service on a domain controller creates a secure channel only when it is needed.

Note If the secure channel cannot be created at logon (for example, because the domain controllers are offline), the Netlogon service starts anyway. If the user's interactive logon uses the same domain name and username, the user's interactive logon is successfully completed using cached credentials.

A Windows NT computer stores the information used to authenticate the last several (ten, by default) users who logged on interactively. That way, if all the domain controllers are down at the same time, the last several users who connected to the computer can still log on. Additionally, the credentials of all users who have logged on from the local computer are stored in the local SAM database.

Pass-through Authentication

Pass-through authentication occurs when a user account must be authenticated, but the local computer can't authenticate the account itself. In this case, the username and password are forwarded to a Windows NT Server computer that can authenticate the user, and the user's information is returned to the requesting computer.

Pass-through authentication occurs in the following instances:

- At interactive logon when a user at a Windows NT Workstation computer or a Windows NT Server computer that is not a domain controller is logging onto a domain or trusted domain
- At remote logon when the domain specified is a trusted domain

Figure 4.6 illustrates pass-through authentication. In this example, AnnM wants to access a computer in the London domain. Because the London domain trusts AnnM's home domain (Topeka), it asks the Topeka domain to authenticate AnnM's account information.

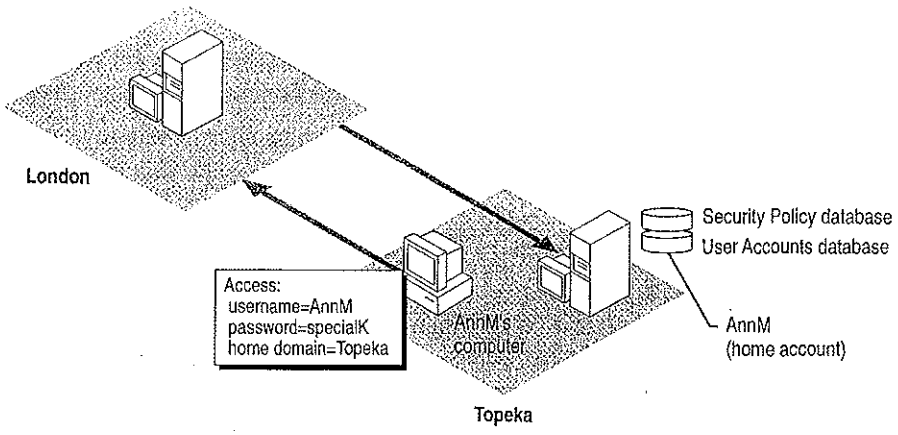


Figure 4.6 Pass-Through Authentication

The Netlogon service provides this pass-through authentication. Each Windows NT computer participating in the domain must be running the Netlogon and Workstation services. (Netlogon is dependent on the Workstation service.) The Netlogon service communicates with the Netlogon service on the remote computer, as illustrated in Figure 4.7.

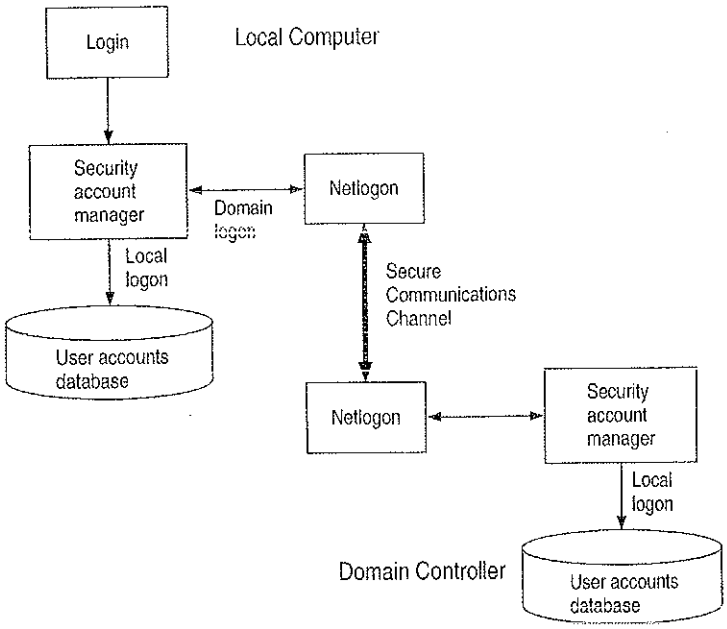


Figure 4.7 Netlogon Requirements for Domain Logons

If the user account is in a trusted domain, the request must first be passed from the computer in the trusting domain to a domain controller in its domain. The domain controller then passes the request to a domain controller in the trusted domain, which authenticates the user account information and then returns the user information by the reverse route.

Interactive Logon

The interactive logon can occur in any user accounts database where a user has an account. Depending on the type of Windows NT computer and how it has been configured, the From box (in the Logon dialog box) lists the local computer and/or domains where user accounts can be authenticated.

Summary of Interactive Logon Authentication

The following table shows the logon options for someone using a Windows NT computer in a workgroup, a domain, and a domain with a trust relationship. The unique identifier used by Windows NT after logon depends on the location of the database used to log on the user. The third column in this table describes the unique identifier used in each case. Any network connection requests sent elsewhere on the network include this unique identifier.

Table 4.1 Summary of Interactive Logon Authentication

Computer is in	User can logon at	Unique identifier
Workgroup	Local database	Computername and username
Domain	Local database Domain database	Computername and username Domain name and username
Domain with a trust relationship	Local database Home domain database Trusted domain database	Computername and username Domain name and username Trusted domain name and username
Domain without a trust relationship	Local database	Computername and username Untrusting domain name and username

Remote Logon

A security access token created at interactive logon is assigned to the initial process created for the user. When the user tries to access a resource on another computer, the security access token is placed in a table in the remote server process. The server process creates a security ID for the user and maps it to the user's security access token. This security ID is sent back to the client redirector and is used in all further server message block (SMB) communication between the server and client. Whenever a resource request comes in from the client, the security ID identifies the user to the server process. The security access token that maps to the user ID identifies the user to the remote security subsystem.

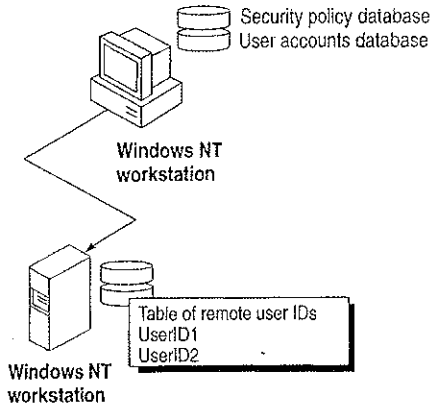


Figure 4.8 Remote Logon

The following list shows the steps in a successful remote logon at a Windows NT Workstation computer or Windows NT Server computer.

1. The username, password, and domain name (the data entered in the Welcome dialog box) of the logged on user are sent from the user's computer to the remote Windows NT server.
2. The authenticating computer's SAM compares the logon username and password with information in the user accounts database.
3. If the access is authorized, the authenticating computer's LSA constructs a security access token and passes it to the server process, which creates a user ID referencing the security access token.
4. The user ID is then returned to the client computer for use in all subsequent requests to the server.

   After the session has been created, the client computer sends requests marked with the user ID it received during session setup. The server matches the user ID with the proper access token kept in an internal table. This security access token at the remote computer is used for access authentication at the remote computer by that user.

Remote Logon at a LAN Manager 2.x Server

Remote logon at a LAN Manager 2.*x* server is basically the same as remote logon to a Windows NT computer. However, instead of comparing the user's logon information against a centralized user accounts database, the LAN Manager 2.*x* server compares the information with its local user accounts database. This database may be the server's own standalone database or a domain database shared by a group of servers. LAN Manager 2.*x* servers cannot use pass-through authentication.

Accessing resources on a LAN Manager 2.*x* server is similar to accessing resources on a Windows NT computer, except that the LAN Manager 2.*x* server does not use a security access token to identify resource requests. Instead, the security ID maps to the username, which is used to process resource requests.

If the LAN Manager 2.*x* server is in the same domain as a Windows NT Server computer, the server logon is identical to that used when accessing another Windows NT Server computer (except that the LAN Manager 2.*x* server does not generate or use security access tokens).

If the LAN Manager 2.*x* server is in another domain, the server logon is identical to logon for a Windows NT Workstation computer that is a member of a workgroup. This is true even for a trusted domain, since LAN Manager 2.*x* servers don't support trust relationships. An account must exist either in the LAN Manager 2.*x* server's domain or at the stand-alone server itself.

Summary of Remote Logon Authentication

This section summarizes the various remote logon scenarios.

- **Workgroup computer connecting to a Windows NT computer in a domain**

  Interactive logon for the user at the workgroup computer (the client) is performed by the local user accounts database.

  The client's username and a function of the password are passed to the specific server in the domain to which the client is trying to connect. This server checks the username and password with information in its local user accounts database. If there is a match, access to this server is allowed.

- **Domain computer connecting to a Windows NT computer in the same domain**

  Interactive logon for the user at the client computer was performed by the domain's user accounts database.

  The client's domain name, username, and a function of the password are passed to the computer being accessed, which passes them to a Windows NT Server computer in the domain.

  The Windows NT Server computer verifies that the domain name for the client matches this domain.

  Next the Windows NT Server computer check the username and password against the domain's user accounts database. If there is a match, access is allowed.

- **Domain client in a trusted domain connecting to a Windows NT computer**

  Interactive logon for the user at the client computer is performed by the domain's user accounts database.

  The client's domain name, username, and a function of the password are passed to the computer being accessed. That computer passes the logon information to a Windows NT Server in the domain.

  The Windows NT Server computer verifies that the client's domain is a trusted domain and then passes the client's identification information to a Windows NT Server computer in that trusted domain.

  A Windows NT Server computer in the trusted domain (that is, the same domain as the client computer) checks the username and password against the domain's user accounts database. If there is a match, access is allowed.

Common Logon Scenarios

The following examples describe various logon scenarios in a Windows NT environment.

Example 1: Logging On to a Member of a Workgroup

For a computer running Windows NT and participating in a workgroup, the logon information is compared with the local user accounts database. When a user logs on, the From box lists only the name of the local computer. The user cannot specify another workgroup or domain for logon. There is no discovery, because the Netlogon service is not running. If the user attempts access to another Windows NT computer, authentication proceeds as discussed in "Example 4: Logging On to an Untrusted Domain," later in this chapter.

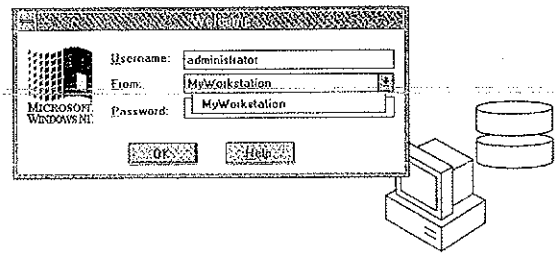


Figure 4.9 Initial Logon and Local Databases for a Windows NT Workstation

After successful authentication, the username and password are cached by the computer's redirector for use when connecting to remote resources.

Example 2: Logging On to the Home Domain

From a Windows NT computer participating in a domain, a user can choose to have his or her logon information authenticated by the local computer or by a domain controller in its domain. If the user account is a domain account, a domain controller's SAM for the home domain or a trusted domain authenticates the logon. The workstation itself connects to a domain with a workstation trust account.

The From box lists the name of the local computer, the name of the home domain in which the computer participates, and the names of any trusted domains.

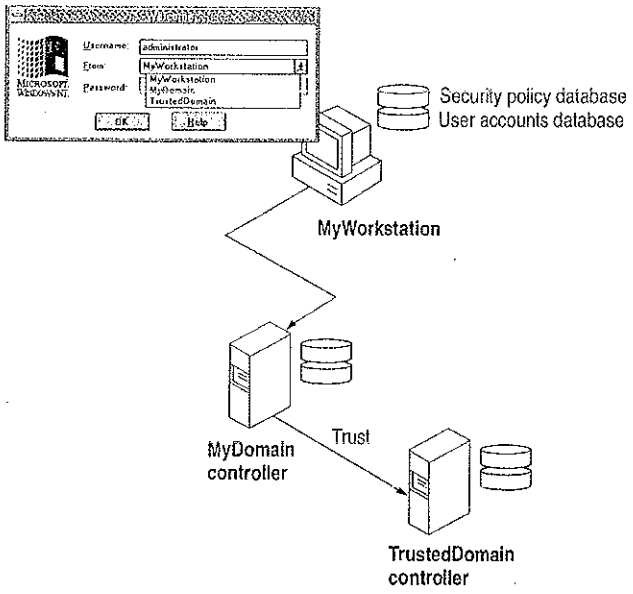


Figure 4.10 Logging On from a Domain Workstation

The security access token generated in an interactive logon is maintained on the computer where the user is logged on.

Example 3: Logging On to a Trusted Domain

When a user at a Windows NT Workstation computer in a domain, or a Windows NT Server computer that is participating in a domain but not as a domain controller, attempts to log on to a trusted domain, the user's credentials are not authenticated on the local computer. The logon request is passed to a domain controller on the trusted domain and is authenticated there.

If the username is not valid and the Guest account of the computer on the computer the user is logging on to is enabled, the user is logged on to the trusted domain as a guest. If the Guest account is disabled, or if the username is valid but the password is not, the logon attempt fails with access denied. The Guest account is used only for remote logons.

The **net use** command prompts for a password if there is no corresponding user account in the trusted domain, or if there is a corresponding user account but the password does match the one supplied by the trusting domain. In situations where the **net use** command would require a password, the **net view** command simply fails with access denied.

The From box lists the domain and trusted domains for this computer.

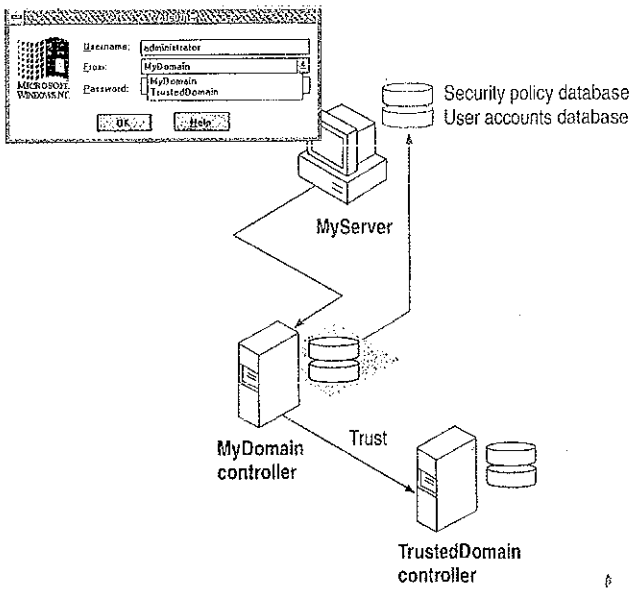


Figure 4.11 Authentication by a Trusted Domain Controller

Example 4: Logging On to an Untrusted Domain

If a client workstation or server connects by remote logon to a Windows NT computer and the domain name specified is not trusted by the domain the client workstation or server that the user is logged on to, the client computer checks its own user account for the username and password supplied. If the credentials are valid, the client logs the user on. If the username is not valid and the client's Guest account is enabled, the computer logs the user on as a guest and passes the credentials to the untrusted domain.

Example 5: Logging on Without Specifying a Domain Name

For workstations running Windows for Workgroups 3.1 or LAN Manager 2.0, the domain of the Windows NT computer being connected to might not be specified. For a user connecting to an individual or workgroup workstation, user credentials are authenticated only on the local computer. If the username is not valid and a Guest account is enabled, the user is logged on as a guest.

If the client is connecting to a domain of which the workstation is a member, user credentials are authenticated first by the workstation itself, and then by a domain controller. If the username is not valid for the domain and the domain controller's Guest account is enabled, the user is logged to the Guest account of the machine being connected to. If the username is valid but the password is not, or if the Guest account is disabled, the user is again prompted for a password, and then the logon attempt fails with access denied.

For a user logging onto a trusted domain from a domain workstation, it is not obvious where the user's domain account is defined. User credentials are authenticated in the following order until the user is successfully logged on: first by the workstation itself, then by the local domain server, and finally by the trusted domain. If all these logon attempts fail, the user is connected, if possible, to the local workstation's Guest account.

Troubleshooting Logon Problems

This section discusses the two categories of typical problems users might face that relate to logons:

- Problems when trying to view a server's shared resources
- Problems when trying to access one of those resources

Viewing a Server's Shared Resources

Suppose AnnM logs on to a Windows NT domain with the password Yippee. She wants to view the shared resources on a server named \\PRODUCTS, but her password there is Yahoo. Because of this situation, Ann sees the following message displayed on the screen:

```
Error 5: Access has been denied.
```

AnnM asks the administrator of \\PRODUCTS to change her password, but the administrator leaves the User Must Change Password At Next Logon checkbox checked. When AnnM tries to view the server's shared resources this time, she sees the following message displayed on the screen:

```
Error 2242: The password of this user has expired.
```

When the administrator of \\PRODUCTS clears the User Must Change Password At Next Logon checkbox, AnnM is finally able to see the server's shared resources.

Accessing a Server's Shared Resources

Suppose AnnM is logged on to a Windows NT domain with the password Yippee but wants to connect to a shared directory on \\PRODUCTS, where her password is Yahoo. Even though \\PRODUCTS has a Guest account because there is an account for AnnM, she is not allowed to gain access via the Guest account. Instead, Windows NT prompts AnnM for the valid password on \\PRODUCTS.

On the other hand, JeffH wants to access the same shared directory and has no account on \\PRODUCTS. He is allowed access to this resource via the Guest account for \\PRODUCTS and is assigned the permissions associated with that account.

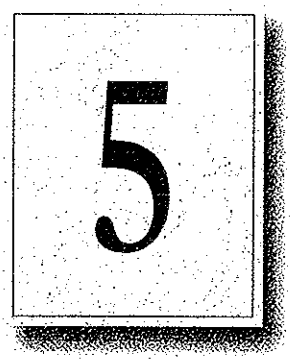
WAN Environments

In a WAN environment, timeout parameters are automatically tuned by both Windows NT Workstation and Windows NT Server. Session setup times out after 45 seconds.

Using the LMHOSTS file, a directed mailslot can be sent directly to a computer's internet protocol (IP) address to establish a trust relationship. For information on the LMHOSTS file, see Chapter 15, "Setting Up LMHOSTS."

CHAPTER 5

Windows NT Browser



Users on a Windows NT network often need to know what domains and computers are accessible from their local computer. Viewing all the network resources available is known as *browsing*. The Windows NT Browser system maintains a list, called the *browse list,* of all the domains and servers available. For instance, when a user attempts to connect to a network drive using File Manager, the list of servers that is displayed in the Shared Directories box of the Connect Network Drive dialog box is the browse list, and it is provided by a browser in the local computer's domain.

Note For the purposes of this discussion, the term *server* refers to any computer that can provide resources to the rest of the network. A Windows NT Workstation computer, for instance, is a server in the context of the Browser system if it can share file or print resources with other computers on the network. The computer does not have to be actually sharing resources to be considered a server. In this chapter, specific references to Windows NT Server computers are always made explicitly.

The Windows NT browser system consists of a *master browser*, *backup browsers*, and client systems. The master browser maintains the browse list and periodically sends copies to the backup browsers. When a browser client needs information, it obtains the current browse list by remotely sending a **NetServerEnum2** application programming interface (API) call to either the master browser or a backup browser. (A **NetServerEnum** API call is also supported for compatibility with Microsoft LAN Manager networks.)

The centralized browser architecture reduces the number of *broadcast datagrams.* A datagram is a network packet that is sent to a mailslot on a specified computer (a *directed datagram)* or to a mailslot on any number of computers (a *broadcast datagram).* The centralized browser architecture also reduces the demands on the client's CPU and memory.

Specifying a Browser Computer

Whether a computer running Windows NT Workstation computer or a Windows NT Server computer can become a browser is determined in the Registry by the MaintainServerList entry under the HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Services\Browser\Parameters key. The possible values for the MaintainServerList entry are shown in the following table:

Table 5.1 Values for the MaintainServerList Entry

Value	Meaning
No	This computer will never be a browser.
Yes	This computer will become a browser. At startup, the server tries to contact the master browser to get a current browse list. If the master browser cannot be found, this computer forces a browser election, and can become the master browser. For more information on browser elections, see "Determining Browser Roles," later in this chapter. This is the default value for Windows NT Server computers.
Auto	This computer is a *potential browser*. Whether it becomes a browser depends on the number of existing browsers. This computer is notified by the master browser if it should become a backup browser. This is the default value for Windows NT Workstation computers.

On any computer with a value of Yes or Auto for the MaintainServerList, Windows NT Setup configures the Browser service to start automatically when the computer starts.

Another setting in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\Parameters key in the registry has a bearing on which servers become master browsers and backup browsers. Setting the IsDomainMasterBrowser entry to True or Yes on a computer makes that computer a *preferred master browser*. A preferred master browser computer has an advantage over other computers in master browser elections. Also, whenever a preferred master browser computer is started, it forces a browser election. For more information on browser elections, see "Determining Browser Roles," later in this chapter.

Number of Browsers in Domains and Workgroups

In a Windows NT Server domain, every Windows NT Server computer is a browser. One Windows NT Server computer in the domain, the primary domain controller if there is one, is the master browser, and the other Windows NT Server computers are backup browsers. If there is more than one Windows NT Server computer in the domain, no Windows NT Workstation computer will ever be a master browser in the domain.

In a workgroup containing Windows NT Workstation computers, there is always one master browser. If there are at least two Windows NT Workstation computers in the workgroup, there is also one backup browser. For every 32 Windows NT Workstation computers in the workgroup, there is another backup browser.

Determining Browser Roles

At certain times in each domain or workgroup, it is necessary to force an election of the master browser. This section explains how the election works.

When a Windows NT computer needs to force a master browser election, it notifies the other browsers on the system by broadcasting an *election datagram.* The election datagram contains the sending browser's election version and election criteria, as explained later in this section. The election version is a constant value that identifies the version of the browser election protocol.

When a browser receives an election datagram, the receiving browser examines the datagram and first compares the election version with its own. If the receiving browser has a higher election version than any other browser, it wins the election regardless of the election criteria. If the election versions are identical for both computers, the election criteria are compared.

The election criteria is a 4-byte hexadecimal value. If there is a tie on the basis of election version, the tie is broken by the value of the election criteria.

- If the browser has a higher election criteria than the issuer of the election datagram, the browser issues its own election datagram and enters the "election in progress" state.
- If the browser does not have a higher election criteria than the issuer of the election datagram, the browser attempts to determine which system is the new master browser.

Specific groups of bytes are masked and their values set according to the following list:

```
Operating System Type:              0xFF000000
    Windows NT Server:              0x20000000
    Windows NT Workstation:         0x10000000
    Windows for Workgroups:         0x01000000

Election Version:                   0x00FFFF00
Per Version Criteria:               0x000000FF
    Primary Domain Controller:      0x00000080
    WINS client:                0x00000020
    Preferred Master browser            0x00000008
    Running Master browser:         0x00000004
    MaintainServerList=yes          0x00000002
    Running Backup Browser          0x00000001
```

If there is still a tie, the browser that has been running longest is the winner. If there is still a tie, the browser that has a lexically lower name is the winner. For example, a server with a name of A becomes master browser instead of a server with a name of B.

When a browser receives an election datagram indicating that it wins the election, the browser enters the *running election* state. In the running election state, the browser sends an election request after a delay based on the browser's current browser role:

- Master browsers delay for 200ms.
- Backup browsers delay for 400ms
- All other browsers delay for 800ms.

The browser broadcasts up to four election datagrams. If, after four election datagrams, no other browser has responded with an election criteria that would win the election, the browser becomes the master browser. If the browser receives an election datagram indicating that another system would win the election, the browser demotes itself to backup browser. To avoid unnecessary network traffic, a browser that has lost an election does not broadcast any unsent election datagrams.

Browsers

The master browser and backup browsers in each domain have certain duties to maintain the browse list.

Role of Master Browsers

The master browser maintains the browse list, the list of all servers in the master browser's domain or workgroup, and the list of all domains on the network. For a domain that spans more than one subnetwork, the master browser maintains the browse list for the portion of the domain on its subnetwork.

Individual servers announce their presence to the master browser by sending a directed datagram called a *server announcement* to the domain or workgroup's master browser. Computers running Windows NT Server, Windows NT Workstation, Windows for Workgroups, and LAN Manager servers send server announcements. When the master browser receives a server announcement from a computer, it adds that computer to the browse list.

The master browser also returns lists of backup browsers (in the local subnetwork of a TCP/IP-based network, if the domain spans more than one subnetwork) to computers running Windows NT Server, Windows NT Workstation, and Windows for Workgroups. If a TCP/IP subnetwork comprises more than one domain, each domain has its own master browser and backup browsers. On networks using the NetBEUI Frame (NBF) or NWLink IPX/SPX-compatible network protocol, name queries are sent across routers, so there is always only one master browser for each domain.

When a computer starts and the computer's MaintainServerList registry entry is set to Auto, the master browser must tell that computer whether or not to become a backup browser.

When a computer first becomes a master browser, it can force all servers to register with it if its browse list is empty. The master browse computer does this by broadcasting a *RequestAnnouncement* datagram. All computers that receive a RequestAnnouncement datagram must respond by sending a server announcement at a random time within the next 30 seconds. The randomized delay ensures that the network and the master browser itself are not overwhelmed with responses.

When a master browser receives a server announcement from another computer that claims to be the master browser, the receiving master browser demotes itself and forces an election. This action ensures that there is always only one master browser in each domain or workgroup.

Note The list of servers that the master browser maintains is limited to 64K of data. This limits the number of computers that can be in a browse list in a single workgroup or domain to 2000-3000 computers.

Role of Domain Master Browsers

The primary domain controller (PDC) of a domain is given a bias in browser elections to ensure that it becomes the master browser. The browser service running on a domain's primary domain controller has the special additional role of being the *domain master browser.*

For a domain that uses TCP/IP and spans more than one subnetwork, each subnetwork functions as an independent browsing entity, with its own master browser and backup browsers. To browse across the WAN to other subnetworks, at least one browser running Windows NT Server is required on the domain for each subnetwork. On the subnetwork with the PDC, this Windows NT Server computer is typically the PDC, which functions as the domain master browser.

When a domain spans multiple subnetworks, the master browsers for each subnetwork announces itself as the master browsers to the domain master browser using a directed MasterBrowserAnnouncement datagram. The domain master browser then sends a remote **NetServerEnum** API call to each master browser to collect each subnetwork's list of servers. The domain master browser merges the server list from each subnetwork master browser with its own server list to form the browse list for the domain. This process is repeated every 15 minutes to ensure that the domain master browser has a complete browse list of all the servers in the domain.

The master browser on each subnetwork also sends a remote **NetServerEnum** API call to the domain master browser to obtain the complete browse list for the domain. This browse list is thus available to browser clients on the subnetwork.

Note Windows NT workgroups cannot span multiple subnetworks. Any Windows NT workgroup that spans subnetworks actually functions as two separate workgroups, with identical names.

Role of Backup Browsers

Backup browsers call the master browser every 15 minutes to get the latest copy of the browse list, as well as a list of domains. Each backup browser caches these lists and returns the list of servers to any clients that send a remote **NetServerEnum** API call to the backup browser. If the backup browser cannot find the master browser, it forces an election.

How Computers Announce Themselves

When a computer is started, it announces itself by sending a server announcement to the domain or workgroup's master browser every minute. As the computer continues running, the time between server announcements is increased until it eventually becomes once every 12 minutes.

If the master browser has not received a server announcement from a computer for three announcement periods, the computer is removed from the browse list.

Note There might be up to a 36-minute delay between the time a server goes down and the time it is removed from the browse list.

Domain Announcements

Client computers sometimes need to retrieve lists of domains, as well as lists of servers in those domains. The Windows NT **NetServerEnum** API has a level of information to allow this.

When a browser becomes a master browser, it broadcasts a DomainAnnouncement datagram every minute for the first five minutes, and then broadcasts once every 15 minutes after that. Master browsers on other domains receive these DomainAnnouncement datagrams and add the specified domain to the browse list.

DomainAnnouncement datagrams contain the name of the domain, the name of the domain master browser, and whether the master browser is running Windows NT Server or Windows NT Workstation. If the master browser if running Windows NT Server, the datagram also specifies whether that browser is the domain's PDC.

If a domain has not announced itself for three consecutive announcement periods, the domain is removed from the browse list.

Note A domain might be down for as long as 45 minutes before it is removed from the browse list.

The domain master browser augments this list of domains with the list of domains that have registered a domain NetBIOS address with the Windows Internet Name Service (WINS). Checking against WINS ensures that the browser maintains a complete list of domain names in an environment with subnetworks. For information on special NetBIOS names, see "Managing Special Names" in Chapter 14, "Installing and Configuring WINS Servers."

How Clients Receive Browser Information

When an application running on a client issues a **NetServerEnum** API call, the client sends the API call to a browser.

If this is the first time a **NetServerEnum** API call has been issued by an application running on the client, the client must first determine which computers are the browsers in its workgroup or domain. The client does this by sending a QueryBrowserServers directed datagram. This request is processed by the master browser for the domain and subnetwork on which the client is located. The master browser then returns a list of browsers active in the workgroup or domain being queried. The client selects the names of three browsers from the list, and then stores these names for future use. For future **NetServerEnum** API calls, a browser is chosen randomly from the three browser names that were saved by the client.

If the client cannot find the master browser after three attempts, the client issues a *ForceElection* broadcast to the domain being queried. A ForceElection broadcast forces the election of a new master browser in the domain. To indicate that the master browser could not be found, the client then returns an error (ERROR_BAD_NETPATH) to the application. For more information on browser elections, see "Determining Browser Roles," earlier in this chapter.

Browser Failures

When a server fails, it stops announcing itself. When the master browser does not receive a server announcement for three of the server's current announcement periods, the master browser removes the non-browser from the browse list. It might take up to an additional 15 minutes for the backup browsers to retrieve the updated browse list from the master browser, so it could take as long as 51 minutes from the time a server fails to when it is removed from all browse lists.

Because a backup browser announces itself in the same way as a server, the procedure when a backup browser fails is the same as that for a server. If the name of this backup browser has been given to any clients, attempts made by those clients to contact this backup browser fail. The client then retries the **NetServerEnum** API call on another backup browser on the client's list of browsers. If all the backup browsers that a client knows have failed, the client attempts to get a new list of backup browsers from the master browser. If the client is unable to contact the master browser, it forces a browser election.

When a master browser fails, the backup browsers detect the failure within 15 minutes. After a master browser failure is detected, the first backup browser to detect the failure forces an election to select a new master browser. In addition, it is possible that between the time the master browser fails and the election of a new master browser happens, the domain will disappear from the list of domains in the browse list. If a client performs its first **NetServerEnum** API call after the old master browser has failed but before a backup browser detects the failure, the client forces an election. If a master browser fails and there are no backup browsers, browsing in the workgroup or domain will not function correctly.

When a domain master browser fails, other master browsers see only servers on the same local subnetwork. Eventually, all servers that are not on the local subnetwork are removed from the browse list.

Browser Components

The Browser system consists of two components:

- Browser service
- Datagram Receiver

The Browser service is the user-mode portion that is responsible for maintaining the browse list, remotely making API calls, and managing the various roles a browser can have. It resides within the LanmanServer service (*\systemroot*\SYSTEM32\SERVICES.EXE) and is supported by *\systemroot*\SYSTEM32\BROWSER.DLL. The browser's registry entries are under the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet \Services\Browser key.

The datagram receiver is the kernel-mode portion of the browser, and is simply a datagram and mailslot receiver. It receives directed and broadcast datagrams of interest to the workstation and server services. It provides kernel-level support for the **NetServerEnum** API, as well as support for remote mailslot reception (second-class datagram-based mailslot messages) and the request announcement services.

The datagram receiver file is *\systemroot*\SYSTEM32\BROWSER.SYS. The datagram receiver's registry entries are in the HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Services\DGRcvr key.

Mailslot Names

All browser datagrams destined for LAN Manager, Windows for Workgroups, Windows NT Workstation, or Windows NT Server computers are sent to the mailslot name \MAILSLOT\LANMAN.

Browser datagrams that are destined only for Windows NT Workstation or Windows NT Server computers are sent to the mailslot name \MAILSLOT\MSBROWSE.

LAN Manager Interoperability

In order for Windows NT browsers and LAN Manager browsers to work together, you might have to perform some configuration tasks.

Making Windows NT Servers Visible to LAN Manager Clients

To make a Windows NT server visible to LAN Manager clients, you must configure the Windows NT server to announce itself to LAN Manager 2.*x* servers. You can do this by using the Networks option in Control Panel or by changing the LMannounce entry in the Registry.

▸ **To make a Windows NT server visible to LAN Manager clients using the Control Panel**

1. On the Windows NT computer, double-click the Network option in Control Panel to display the Network Settings dialog box.
2. Select Sever from the Installed Network Software box, and then choose the Configure button to display the Server dialog box.
3. Select Make Browser Broadcasts to LAN Manager 2.*x* Clients check box, and then choose the OK button.

▸ **To make a Windows NT browser visible to LAN Manager clients using the Windows NT Registry**

1. Run the REGEDT32.EXE file from File Manager or Program Manager to start the Registry Editor.
2. Locate the following key:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
3. Change the value of the LMannounce entry to **1**.

For more information about the Windows NT Registry, see Chapters 10 through 14 in the *Windows NT Resource Guide*.

Making LAN Manager Domains Visible to Windows NT Browsers

You can make up to four LAN Manager-only domains visible to a Windows NT Browser. You can do this by using the Control Panel or configuring the Registry of the Windows NT browser. The LAN Manager domains you add to the Windows NT browse list this way will be visible to all members of the Windows NT browser's domain.

▸ **To make LAN Manager domains visible to a Windows NT browser using the Control Panel**

1. On the Windows NT computer, double-click the Networks option in Control Panel to display the Network Settings dialog box.
2. Select Computer Browser from the Installed Network Software box, and then choose the Configure button to display the Browser Configuration dialog box.
3. For each LAN Manager domain you want to add, type the LAN Manager domain name in the box on the left, and then choose the Add button.
4. When finished adding up to four domains, choose the OK button.

▸ **To make LAN Manager domains visible to a Windows NT browser using the Windows NT Registry**

1. Run the REGEDT32.EXE file from File Manager or Program Manager of the Windows NT browser to start the Registry Editor.
2. Locate the following key:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters
3. In the OtherDomains entry, add the names of the LAN Manager domains that you want to be made visible to the Windows NT browser.

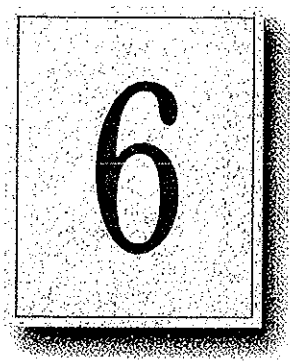
For more information about the Windows NT Registry, see chapters 10 through 14 in the *Windows NT Resource Guide*.

PART II

Using Windows NT Networking

CHAPTER 6

Using NBF with Windows NT



NetBEUI Frame (NBF) is the implementation of the NetBIOS Extended User Interface (NetBEUI) protocol driver used in Windows NT. This protocol provides compatibility with existing LANs that use the NetBEUI protocol.

This chapter describes how NBF handles connection-oriented and connectionless network traffic, and it also describes NBF's unique method for handling resources to create a virtually infinite number of connections. The topics include the following:

- Overview of NetBEUI and NBF
- NBF and network traffic
- NBF and sessions
- Session limits

Overview of NetBEUI and NBF

The NetBEUI protocol, first introduced by IBM in 1985, was written to the NetBIOS interface and designed as a small, efficient protocol for use on department-sized LANs of 20 to 200 workstations. This original design assumed that broader connectivity services could be added by including gateways as the network grew. (As described later in this chapter, NBF breaks the session limit that restricted NetBEUI's reach.)

The NetBEUI protocol provides powerful flow control and tuning parameters plus robust error detection. Microsoft has supported the NetBEUI protocol in all of its networking products since Microsoft's first networking product, MS-Net, was introduced in the mid-1980s.

NetBEUI is the precursor to the NetBEUI Frame (NBF) protocol included with Windows NT. NBF provides compatibility with existing LAN Manager and MS-Net installations, and with IBM LAN Server installations. On Windows NT, the NetBIOS interface is supported under MS-DOS, 16-bit Windows, and Win32 subsystem environments.

NBF and Network Traffic

The NBF protocol, like NetBEUI, provides for both connectionless or connection-oriented traffic. Connectionless communications can be either unreliable or reliable. NBF and NetBEUI provide only *unreliable connectionless*, not reliable connectionless communications.

Unreliable communication is similar to sending a letter in the mail. No response is generated by the receiver of the letter to ensure the sender that the letter made it to its destination. In comparison, reliable connectionless communications is like a registered letter whose sender is notified that the letter arrived.

Connection-oriented communications provide reliable communications between two computers in a way that is analogous to a phone call, where two callers connect, a conversation occurs, and then the connection is dropped when the conversation ends. A reliable connection requires more overhead than connectionless communications do.

NBF communicates via the NDIS interface at the Logical Link Control (LLC) sublayer. A connection at the LLC sublayer is called a *link*, which is uniquely defined by the adapter's address and the destination service access point (DSAP). A service access point (SAP) can be thought of as the address of a port to a layer as defined by the OSI model. Because NBF is a NetBIOS implementation, it uses the NetBIOS SAP (0xF0). While the 802.2 protocol governs the overall flow of data, the primitives are responsible for passing the data from one layer to the next. The primitives are passed through the SAPs between layers.

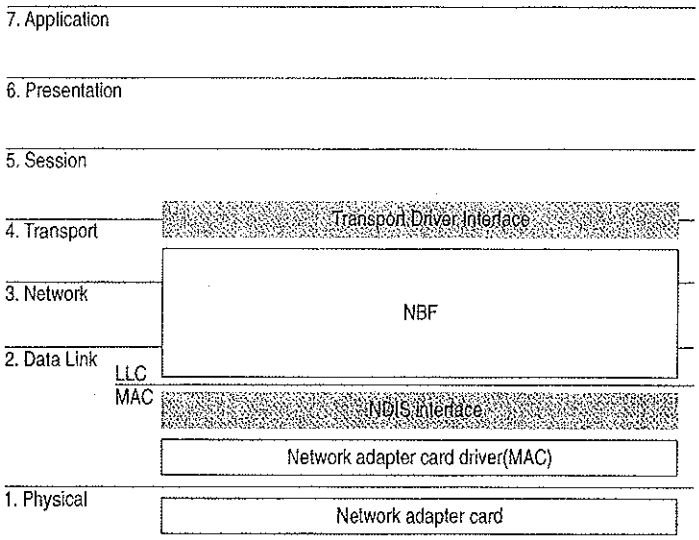


Figure 6.1 NBF Communicates via the NDIS Interface at the LLC Sublayer

Connectionless Traffic

For connectionless traffic that requires a response from a remote computer, NBF sends out a certain number of frames, depending on the command. The total number is based on *retry* Registry value entries, such as **NameQueryRetries**. The time between sending each frame is determined by *timeout* Registry entries, such as **NameQueryTimeout**.

Three types of NetBIOS commands generate connectionless traffic: name claim and resolution, datagrams, and miscellaneous commands. These commands are sent as UI (Unnumbered Information) frames at the LLC sublayer.

To see how Windows NT uses retry and timeout values from the Registry, consider what happens when Windows NT registers computernames via NBF using the NetBIOS **Add.Name** command. When NBF receives the **Add.Name** command, it broadcasts ADD_NAME_QUERY frames a total of **AddNameQueryRetries** times and sends these broadcasts at a time interval of **AddNameQueryTimeout**. This allows computers on the network enough time to inform the sending computer whether the name is already registered as a unique name on another computer or a group name on the network.

Note All Registry values discussed in this chapter are found under the following Registry path:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Nbf
```

Connection-Oriented Traffic

The **net use** command is an example of a connection-oriented communication, as illustrated in Figure 6.2.

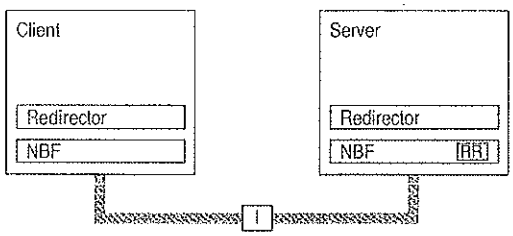


Figure 6.2 Connection-oriented Network Traffic

When a user types **net use** at the command line to connect to a shared resource, NBF must first locate the server by sending UI-frames, and then initialize the link. This is handled by the redirector when it makes a connection to the NBF drivers via the Transport Driver Interface (TDI) boundary. NBF begins the sequence by generating a NetBIOS Find Name frame. Once the server is found, a session is set up with UC Class-II frames following the standard 802.2 protocol (802.2 governs the overall flow of data).

The client computer sends an SABME (Set Asynchronous Balance Mode Extended) frame, and the server returns a UA (Unnumbered Acknowledgment) frame. Then the client sends an RR (Receive Ready) frame, notifying the server that it is ready to receive I-frames whose sequence number is currently 0. The server acknowledges this frame.

Once the LLC-level session is established, additional NetBEUI-level information is exchanged. The client sends a Session Initialize frame, and then the server responds with a Session Confirm frame. At this point, the NetBEUI-level session is ready to handle application-level frames (Server Message Blocks, or SMBs).

Reliable transfer is achieved with link-oriented frames by numbering the I-frames. This allows the receiving computer to determine whether the frames were lost and in what order they were received.

NBF uses two techniques to improve performance for connection-oriented traffic: use of adaptive sliding windows and use of link timers. These techniques are described in the next two sections.

Adaptive Sliding Window Protocol

NBF uses an adaptive sliding window algorithm to improve performance while reducing network congestion and providing flow control. A sliding window algorithm allows a sender to dynamically tune the number of LLC frames sent before an acknowledgment is requested. Figure 6.3 shows frames traveling through a two-way pipe.

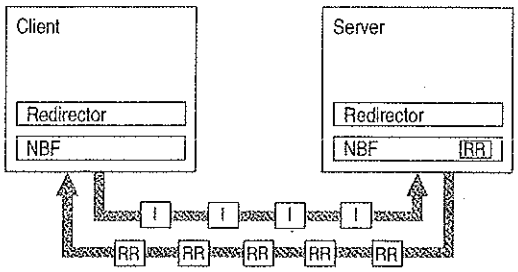


Figure 6.3 Adaptive Sliding Window

If the sender could feed only one frame into the pipe and then had to wait for an acknowledgment (ACK), the sender's pipe would be underused. If the sender can send multiple frames before an ACK is returned, the sender can keep the pipe full, thereby using the full bandwidth of the pipe. The frames would travel forward, and then ACKs for the received frames would travel back. The number of frames that the sender is allowed to send before it must wait for an ACK is referred to as the *send window.* In general, NBF has no receive window, unless it detects that the remote is a version of IBM LAN Server, which never polls; in this case, NBF uses a receive window based on the value of **MaximumIncomingFrames** in the Registry.

The adaptive sliding window protocol tries to determine the best sizes for the send window for the current network conditions. Ideally, the windows should be big enough so that maximum throughput can be realized. However, if the window gets too big, the receiver could get overloaded and drop frames. For big windows, dropped frames cause significant network traffic because more frames have to be retransmitted. Lost frames might be a problem on slow links or when frames have to pass over multiple hops to find the receiving station. Lost frames coupled with large send windows generate multiple retransmissions. This traffic overhead might make an already congested network worse. By limiting the send window size, traffic is throttled, and congestion control is exercised.

Link Timers

NBF uses three timers: the response timer (T1), the acknowledgment timer (T2), and the inactivity timer (Ti). These timers help regulate network traffic and are controlled by the values of the **DefaultT1Timeout**, **DefaultT2Timeout**, and **DefaultTiTimeout** Registry entries, respectively.

The response timer is used to determine how long the sender should wait before it assumes the I-frame is lost. After T1 milliseconds, NBF sends an RR frame that has not been acknowledged and doubles the value for T1. If the RR frame is not acknowledged after the number of retries defined by the value of **LLCRetries**, the link is dropped.

Where the return traffic does not allow the receiver to send an I-frame within a legitimate time period, the acknowledgment timer begins, and then the ACK is sent. The value for this timer is set by the T2 variable, with a default value of 150 milliseconds. If the sender has to wait until the T2 timer starts in order to receive a response, the link might be underused while the sender waits for the ACK. This rare situation can occur over slow links. On the other hand, if the timer value is too low, the timer starts and sends unnecessary ACKs, generating excess traffic. NBF is optimized so that the last frame the sender wants to send is sent with the POLL bit turned on. This forces the receiver to send an ACK immediately.

The inactivity timer, Ti, is used to detect whether the link has gone down. The default value for Ti is 30 seconds. If Ti milliseconds pass without activity on the link, NBF sends an I-frame for polling. This is then ACKed, and the link is maintained.

Note Remember that T2 <= T1 <= Ti.

NBF and Sessions

Each process within Windows NT that uses NetBIOS can communicate with up to 254 different computers. The implementation of NetBIOS under Windows NT requires the application to do a few more things than have traditionally been done on other platforms, but the capacity for doing up to 254 sessions from within each process is well worth the price. Prior implementations of NetBIOS had the 254-session limit for the entire computer, including the workstation and server components.

Note that the 254-session limit does not apply to the default workstation or server components. The workstation and server services avoid the problem by writing directly to the TDI rather than calling NetBIOS directly. This is a handle-based (32-bit) interface.

NBF also has a unique method of handling resources to create a virtually infinite (memory permitting) number of connections, as described in the next section.

Session Limits

The 254-session limit is based on a key variable in the NetBIOS architecture called the *Local Session Number* (LSN). This is a one-byte number (0 to 255) with several numbers reserved for system use. When two computers establish a session via NBF, there is an exchange of LSNs.

The LSNs on the two computers might be different. They do not have to match, but a computer always uses the same LSN for a given session. This number is assigned when a program issues a CALL NCB (Network Control Block). The number is actually shared between the two computers in the initial frame sent from the calling computer to the listening computer. Figure 6.4 shows this session-creation frame exchange.

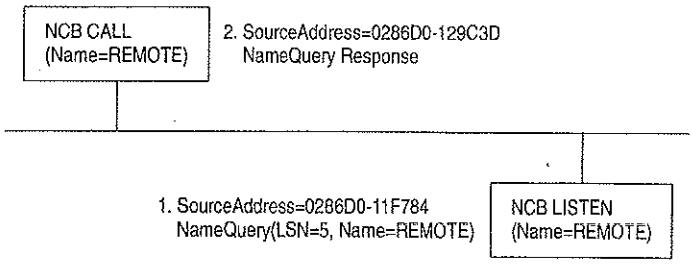


Figure 6.4 Broadcast of NameQuery

The initial frame is a NameQuery frame. In previous implementations of NBF, this frame was broadcast onto the network. All computers read the frame and check to see if they have the name in their name space and if there is a LISTEN NCB pending on the name. If there is a LISTEN NCB pending, the computer assigns a new LSN for itself, and then adds it to the response frame and satisfies the LISTEN NCB, which now contains just the LSN used on that computer. Even though both computers know the LSN of the other, the information is not used. The more important information for the two communicating partners is the network addresses that are part of the frames. As the frames are exchanged, each partner picks up the address of the other in the source address component of the frame received. The NBF protocol keeps the network address of the remote partner so that subsequent frames can be addressed directly.

Note This process applies for NBF connections. NetBIOS connections established via TCP/IP and RFC1001/1002 or NBP are handled differently.

Windows NT has to use the same NameQuery frame to establish connections with remote computers via NBF; otherwise, it would not be able to talk to existing workstations and servers. The NameQuery frame transmitted must contain the 1-byte-wide LSN to be used.

Breaking the 254-Session Limit

NBF breaks the 254-session barrier by using a combination of two matrices, one maintained by NBF, and one maintained by NetBIOS.

The NBF system maintains a two-dimensional matrix, as shown in Figure 6.5. Along the side of this matrix are the LSN numbers 1 to 254. Across the top are the network addresses for the different computers that it has sessions with. In the cell defined by the LSN and network address is the TDI handle, which relates back to the process that established the connection (either the CALL or LISTEN).

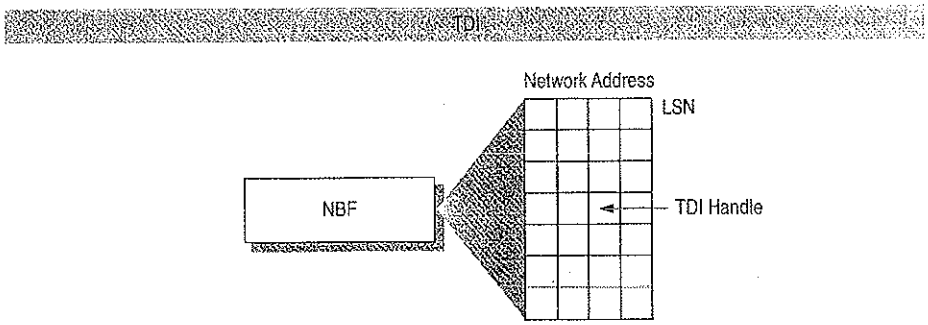


Figure 6.5 NBF and Its LSN Matrix

Note The matrix concept and its contents are for illustration purposes only. The physical storage algorithm and exact contents are beyond the scope of this chapter.

The NameQuery frame from Windows NT contains the LSN number associated with the TDI handle that satisfies either the NCB CALL or the LISTEN. In the case of a CALL, it is not broadcast but is addressed directly to the recipient.

The remaining mystery is how NBF gets the network address of the recipient to add to its matrix when doing the CALL. (It's easy on the LISTEN side because the address is in the NameQuery frame received.)

As shown in Figure 6.6, NBF uses two NameQuery frames.

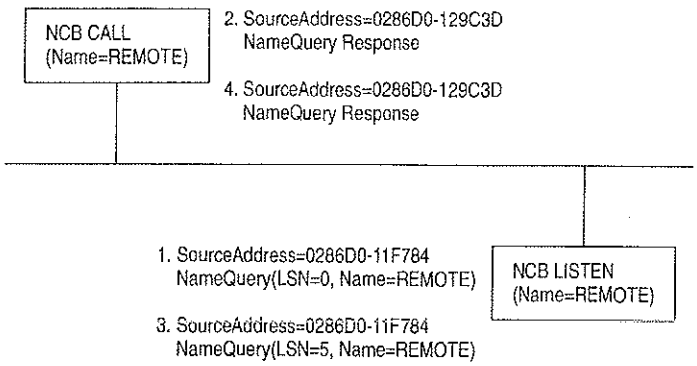


Figure 6.6 Two NameQuery Frames in Windows NT NBF

For the numbered items in Figure 6.6:

1. The first frame is the FindName format of the NameQuery. However, an LSN of 0 is special; it indicates that it is a FindName. The FindName is broadcast; when the remote computer responds to the frame, NBF has the network address it needs to add an entry to the table.
2. The second NameQuery is then sent directly to the remote station, with the LSN filled in as a CALL command. The FindName will be successfully returned by the remote computer, even if no LISTEN NCB is posted against the name.
3. If no LISTEN NCB is posted against the name, frame (3) is sent.
4. The same frame is responded to by frame (4).

NBF must also address another problem—the LSN from the NBF table cannot be the one returned to the process issuing the CALL or LISTEN commands. NBF may have established connections with multiple remote computers with LSN=5, for example. Windows NT must return each process an LSN number that uniquely defines its session.

As stated earlier, NBF uses the TDI handle to know which LSN and network address to send frames to, and each process has its own set of LSNs available to it. Therefore, there must be a component between the originating process and the TDI interface of NBF that translates a process ID and an LSN into a TDI handle. The component in the middle is called NETBIOS.SYS.

This concept is illustrated in Figure 6.7, although the table maintained by NETBIOS.SYS is actually 254 LSNs per LANA number per process. (In Windows NT, each binding path is represented by a LANA number). In reality, each process can have up to 254 sessions per LANA number, not just a total of 254 sessions.

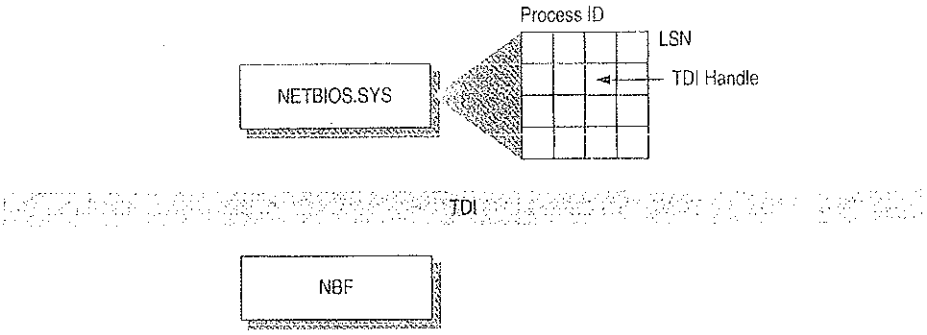


Figure 6.7 NETBIOS.SYS Matrix

NETBIOS.SYS builds a second matrix that has LSNs down the side, process IDs along the top, and TDI handles in the cells. It is the LSN from this table that is passed back to the originating process.

Figure 6.8 presents a top-down view of the architecture.

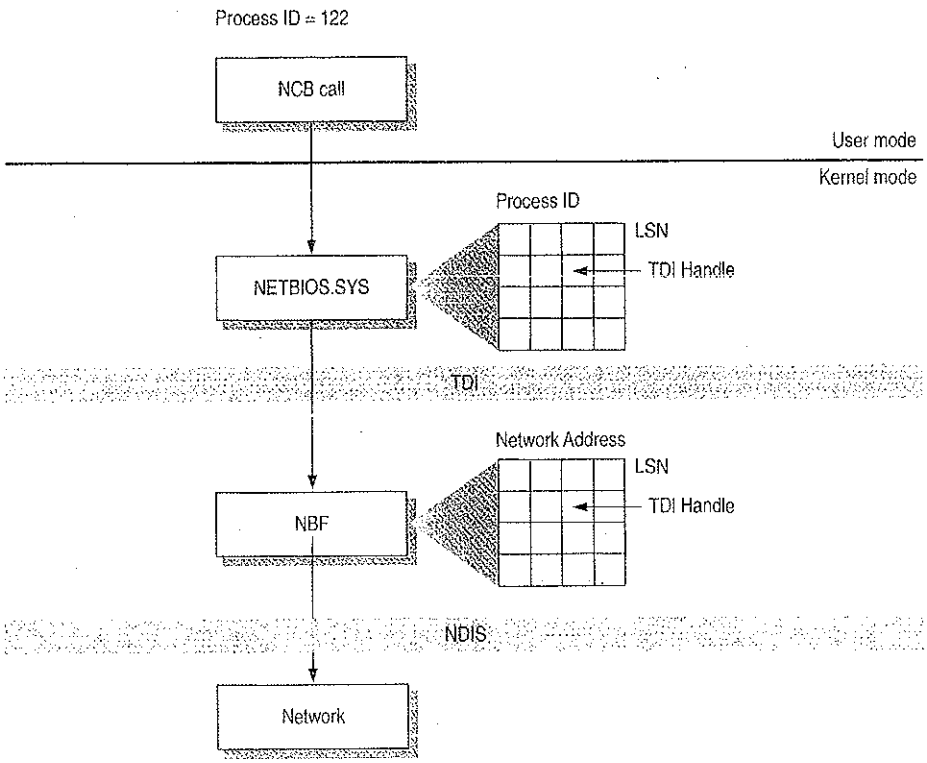


Figure 6.8 Another View of the NetBIOS Architecture

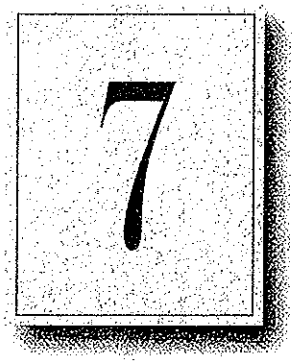
For example, suppose a process needs to establish a session with a remote computer. Before the process can issue the CALL NCB, it must issue a RESET NCB. This command signals NETBIOS.SYS to allocate space in its TDI handle table, among other things. Once the RESET is satisfied, the process issues a CALL NCB to make a connection with a specific remote computer. This NCB is directed down to the NETBIOS.SYS device driver. The driver opens a new TDI handle to NBF and sends the command to NBF.

NBF issues the first NAME_QUERY with LSN=0 to find the remote computer. When the remote computer responds, the network address is extracted from the frame, and a column in the NBF table is created. The second NAME_QUERY with an LSN is sent directly to the remote computer. When that frame is returned successfully, NBF returns from the TDI call to the NETBIOS.SYS driver with a successful status code.

NETBIOS.SYS then fills in the LSN from its table into the NCB and satisfies it back to the calling process.

CHAPTER 7

Using DLC with Windows NT



A Data Link Control (DLC) protocol interface device driver is included in Windows NT Workstation and Windows NT Server. The DLC protocol is traditionally used to provide connectivity to IBM mainframes. It is also used to provide connectivity to local area network printers that are directly attached to the network, instead of to a specific computer.

This chapter provides details about the DLC protocol device driver for Windows NT.

Overview

The Data Link Control (DLC) protocol driver provided with Windows NT allows the computer to communicate with other computers running the DLC protocol stack (for example, an IBM mainframe) and other network peripherals (for example, printers such as a Hewlett-Packard HP 4Si that use a network adapter card to connect directly to the network).

Windows NT DLC contains an 802.2 Logical Link Control (LLC) Finite State Machine, which is used when transmitting and receiving type 2 connection-oriented frames. DLC can also transmit and receive type 1 connectionless frames, such as Unnumbered Information (UI) frames. Type 1 and 2 frames can be transmitted and received simultaneously.

Windows NT DLC works with either token ring or Ethernet MAC drivers and can transmit and receive Digital.Intel.Xerox (DIX) format frames when bound to an Ethernet MAC.

The DLC interface can be accessed from 32-bit Windows NT-based programs and from 16-bit MS-DOS–based and 16-bit Windows-based programs. The 32-bit interface conforms largely to the CCB2 interface, the segmented 16-bit pointers being replaced with flat 32-bit pointers. The 16-bit interface conforms to the CCB1 interface.

Note For definitions of the CCB interfaces, see the *IBM Local Area Network Technical Reference.*

Loading the DLC Driver on Windows NT

The DLC driver can be loaded when the system is first installed, or any time thereafter, using the Network option in Control Panel.

The order of the bindings section is significant to DLC because an adapter is specified at the DLC interface as a number—typically 0 or 1 (although Windows NT DLC can support up to 16 physical adapters). The number corresponds to the index of the adapter in the DLC bindings section. If you have only one network adapter card installed, DLC applications use a value of 0 to refer to this adapter, and you need not make any changes to the bindings.

If you have more than one adapter card, you might want to modify the bindings.

To change the order of the bindings

1. From the Network Control Panel, choose Bindings.
2. From the Show Bindings For box, choose DLC Protocol.

   You will see a list of bindings, such as the following:

   ```
   DLC Protocol -> ARC Built-in Ethernet Adapter Driver ->
      [01] ARC Built-in Ethernet Adapter
   DLC Protocol -> IBM Token Ring Adapter Driver ->
      [02] IBM Token Ring Adapter
   ```

   The numbers in brackets refer to the order in which the adapters were installed. In this example, DLC currently refers to the Ethernet adapter as adapter #0 and the Token Ring adapter as adapter #1.

   If you have software (such as a 3270 emulator program) that allows you to specify an adapter number at run time, you might decide to keep the current setup and change the adapter number when you run the software. Typically, however, the software uses adapter #0, expecting an IBM Token Ring card to be the primary adapter. In this case, you will need to change the order of the bindings list.
3. To change the order of an item in the list, highlight the item, and then use the up- and down-arrow buttons to reposition it in the list.

   For example, suppose you wanted to change the above bindings so that the IBM Token Ring adapter corresponds to adapter #0 and the ARC Ethernet adapter corresponds to adapter #1. Highlight the line containing IBM Token Ring Adapter Driver, and click once on the up-arrow button. The bindings are now correctly ordered for your application software, and you do not need to modify the program configuration.
4. Choose OK to keep the modified bindings list.

DLC Driver Parameters in the Registry

Unlike other Windows NT protocol drivers, DLC does not bind to a MAC driver until an adapter open command is issued. When an adapter is opened for the first time, the DLC protocol driver writes some default values into the Registry for that adapter. These values control the various timers that DLC uses, whether DIX frames should be used over an Ethernet link, and whether bits in a destination address should be swapped (used when going over a bridge that swaps destination addresses).

The timer entries in the Registry are supplied because program-supplied timer values might not be sufficient. There are three timers used by DLC link communication:

- T1 is the response timer.
- T2 is the acknowledgment delay timer.
- Ti is the inactivity timer.

Each timer is split into two groups—**T*x*TickOne** and **T*x*TickTwo**, where *x* is 1, 2, or i.

Typically, these timer values are set when a program opens an adapter and/or creates a Service Access Point (SAP).

The Registry contains entries used to modify timer values. Registry entries for DLC are found in the following location:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DLC\Parameters
   \<Adapter Name>
```

When you edit a timer entry value, the change takes effect the next time the adapter is opened (for example, by rerunning the application). For more information, including the ranges and default values for the timers, see "DLC System Driver Entries" in Chapter 14, "Registry Value Entries."

Communicating with SNA Hosts Using DLC and SNA

One of the major uses of the DLC protocol today is connecting personal computers to SNA hosts, that is, IBM mainframe or midrange computers such as the AS/400. With the increased popularity of local area networks in the mid-1980s, IBM introduced two new connectivity options for its hosts. With the Token Ring Interface Connection (TIC), any SNA host can communicate with a token ring network. With the LAN Interface Connection (LIC), an AS/400 computer can communicate with an Ethernet network.

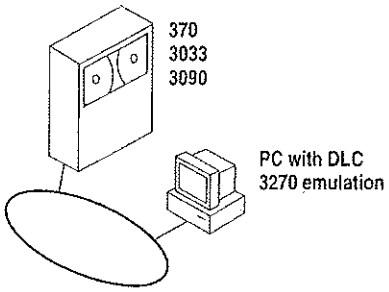


Figure 7.1 Mainframe Connectivity Path Using Token Ring

The SNA hosts already possessed a rich protocol stack in Systems Network Architecture (SNA). SNA provides equivalent functionality to the OSI Network, Transport, Session, and Presentation levels (although functionality might differ at each level). Because the DLC layer and the OSI Data Link layer are almost identical in functionality, a programming interface was developed for the DLC layer and exposed to programmers wanting to use this level of interface. The interface is described in the IEEE 802.2 standard.

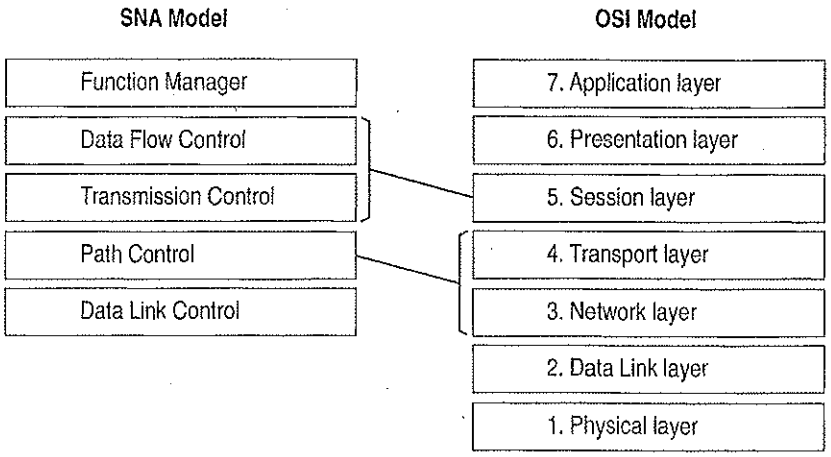


Figure 7.2 Comparison of SNA and OSI Models

SNA Server uses the DLC protocol device driver when communicating to mainframes via the token ring interface. Detailed configuration and installation information is provided in the *Microsoft SNA Server Installation Guide* and the *Microsoft SNA Server Administration Guide.*

Using DLC to Connect to HP Printers

DLC is used to provide connectivity to local area network printers that are directly attached to the network, not to a specific computer.

Printing via the DLC protocol device driver starts by creating a printer that uses the HPMON.DLL printer driver. All commands are performed in the Print Manager utility.

▷ **To connect to a printer that is directly attached to the network**

1. From the Printer menu in Print Manager, choose Create Printer.
2. In the Print To box, select Other.
3. In the Print Destinations dialog box, select Hewlett-Packard Network Port.
4. In the Add Hewlett-Packard Network Peripheral Port dialog box, select the network adapter card that will communicate with the printer.

From the Add Hewlett-Packard Network Peripheral Port dialog box, you can cause Windows NT to automatically search for printers connected to your network. You can also adjust the DLC Timers for this application. DLC timers are described in "DLC Driver Parameters in the Registry," earlier in this chapter.

For more specific information, see the online Help associated with the Add Hewlett-Packard Network Peripheral Port dialog box.

Changing the Locally Administered Address

There might be times when you want to change or override the network address of the network adapter card when running the DLC protocol. You might want to do this, for example, when communicating directly to a mainframe. Certain configurations of mainframe software require the network address of the devices connecting to it to follow a set format, so it might be necessary to change the card's network address. You can do this through the Registry Editor.

Note The following example is for an IBM Token Ring adapter. This parameter is supported on other network adapters as well, but not necessarily all.

The following instructions do not apply when connecting to a mainframe via SNA Server. The modifications needed to the network address are handled during the installation process.

To change the address of an adapter card

1. From the File menu of Program Manager, choose the Run command.
2. In the Command Line box of the Run dialog box, type REGEDT32.EXE, and then choose the OK button.
3. When the Registry Editor starts, select the following key:

   ```
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ibmTOKMC01
   ```

4. From the Edit menu, choose Add Value. For Value, type **NetworkAddress,** and select REG_SZ for data. Choose OK.
5. Type the 12-digit Locally Administered Address (LAA) that you need to communicate to the mainframe. If you don't know this address, see your network administrator or operations group.
6. Exit the Registry Editor and restart your computer.

   (You must restart the computer for the modification to take effect.)
7. From the command prompt, run the following command to report the active MAC address:

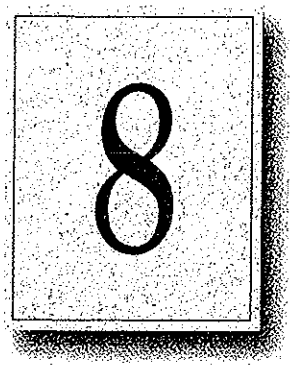
   ```
   net config rdr
   ```

   If the MAC address is the one you entered in the Registry Editor, the LAA has taken effect.

For more information about using Registry Editor, see Chapter 11, "Registry Editor and Registry Administration," of the *Windows NT Resource Guide.* For information about specific DLC-related Registry Entries, see Chapter 14, "Registry Value Entries," of the *Windows NT Resource Guide.*

CHAPTER 8

Client-Server Connectivity on Windows NT



Client-server computing systems must be able to access data that resides on different hardware platforms, different operating systems, different network operating systems, and different database management systems (DBMSs). This chapter discusses specifically how client workstations communicate with databases stored on Windows NT computers. Primarily, this chapter covers details about MS-DOS, Windows, Windows NT Workstation, and OS/2 client workstations.

This chapter explains client-server connectivity on Windows NT using Microsoft SQL Server as an example. For information on other client-server databases developed for Windows NT, see the appropriate vendor documentation.

SQL Server

Microsoft SQL Server 4.21 has been completely reengineered for Windows NT. SQL Server includes the following enhancements and performance improvements that were not part of previous versions of SQL Server:

- A new Symmetric Server architecture allows SQL Server to scale from notebook computers to symmetric multiprocessor servers, with support for Intel-based and RISC-based computers. This architecture dynamically balances the processor load across multiple CPUs and provides a preemptive multithreaded design for improved performance and reliability.
- Windows NT provides preemptive scheduling, virtual paged memory management, symmetric multiprocessing, and asynchronous I/O, the foundation of a mission-critical database server platform. Integration with the Windows NT operating system improves operational control and ease of use. Administrators can manage multiple SQL Servers across distributed networks using graphical tools for configuration, security, database administration, performance monitoring, event notification, and unattended backup.
- Unified logon security with Windows NT security means that authorized users do not have to maintain separate SQL Server logon passwords and can bypass a separate logon process for SQL Server. Additionally, SQL Server applications can take advantage of Windows NT security features, which include encrypted passwords, password aging, domain-wide user accounts, and Windows-based user administration.
- Windows NT provides an ideal platform for building powerful 32-bit client-server applications for Microsoft SQL Server. The *Microsoft SQL Server Programmer's Toolkit* contains a 32-bit Win32-based version of the Microsoft DB-Library™ application programming interface.
- Microsoft SQL Server is fully interoperable with Microsoft SQL Server for OS/2, as well as with SYBASE SQL Server for the UNIX and VMS operating systems. Existing applications will work unchanged. Microsoft SQL Server operates across all corporate network environments, including Novell NetWare and TCP/IP-based LANs.

The key to enterprise interoperability is network independence. Microsoft SQL Server can support clients communicating over multiple heterogeneous networks simultaneously, with no need for additional integration products. SQL Server communicates on named pipes (over either NetBEUI or TCP/IP network protocols) with Windows, Windows NT, MS-DOS, and OS/2 clients. In addition, SQL Server can simultaneously support TCP/IP Sockets for communication with Macintosh, UNIX, or VMS clients and SPX Sockets for communications in a Novell NetWare environment. It also supports DECnet™ Sockets, AppleTalk, and Banyan VINES. Microsoft SQL Server leverages the power, ease of use, and scalability offered by the Windows NT operating system to manage large databases for mission-critical applications.

Data Access Mechanisms

Figure 8.1 illustrates the key interfaces used to access data in a Microsoft SQL Server client-server environment. These include application programming interfaces (APIs), data stream protocols, interprocess communication (IPC) mechanisms, network protocols, and the Tabular Data System (TDS) protocol.

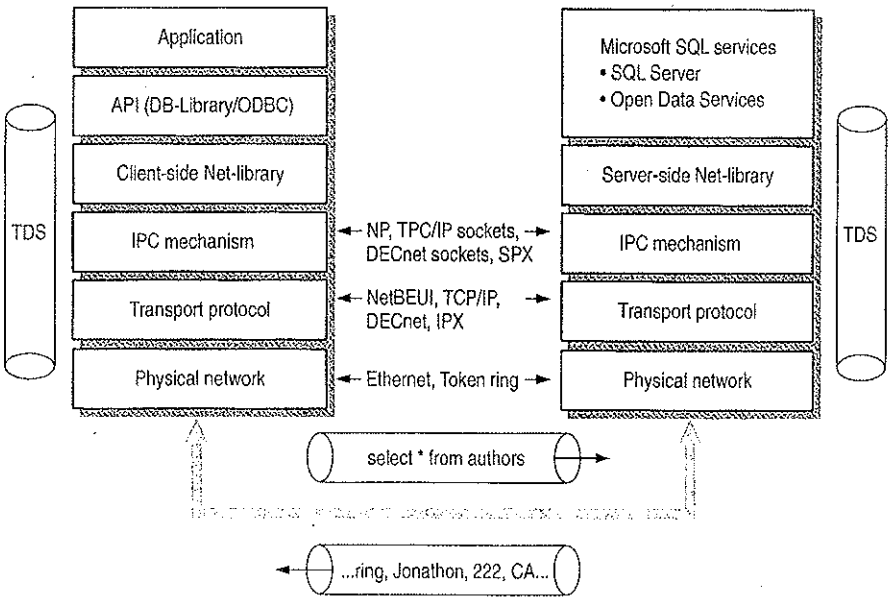


Figure 8.1 Levels and Interfaces Within the Microsoft SQL Server Architecture

The following sections describe each of these interfaces to SQL Server.

Application Programming Interfaces

Each back-end database typically has its own application programming interface (API) through which it communicates with clients. A client application needing to access multiple back-end databases must be able to transform requests and data transfers into each of the corresponding APIs. Client-server applications communicate with Microsoft SQL Server through two APIs—Open Database Connectivity (ODBC) and DB-Library.

ODBC is an API for generic database connectivity for Windows and Windows NT platforms. It is designed to be a general-purpose call-level interface (CLI) for any database, including nonrelational DBMSs. The ODBC interface provides the needed functionality for applications that must access multiple DBMSs from different vendors. Application developers can develop, compile, and ship an application without targeting a specific DBMS, provided that DBMS-specific features are not used. ODBC ensures interoperability by forcing all clients to adhere to a standard interface. The ODBC driver automatically interprets a command for a specific data source.

DB-Library is a set of API calls designed specifically so multiplatform client applications can interact with Microsoft SQL Server. DB-Library provides the needed functionality for applications requiring client support for MS-DOS and OS/2, as well as for Microsoft Windows and Windows NT. It is also equivalent to the SYBASE Open Client interface on UNIX, VMS, and Macintosh systems.

Data Stream Protocols

Every DBMS uses a logical data stream protocol that enables the transfer of requests, data, status, error messages, and so on, between the DBMS and its clients. The API uses interprocess communication (IPC) mechanisms supported by the operating system and network to package and transport this logical protocol.

The data stream protocol for Microsoft SQL Server is called Tabular Data Stream (TDS). TDS is also used by Open Data Services and SYBASE® software to transfer requests and responses between the client and the server. Because TDS is a logical data stream protocol, it requires physical network IPC mechanisms to transmit the data. The Net-Library architecture described later in this chapter provides a method of sending TDS across a physical network connection.

Data stream protocols are typically proprietary, developed and optimized to work exclusively with a particular DBMS. An application accessing multiple databases must, therefore, be able to use multiple data stream protocols. Using ODBC helps resolve this problem for application developers.

With ODBC implementations, the data stream protocol differences are resolved at the driver level. Each driver emits the data stream using the protocol established by the server. The SQL Server ODBC driver emits TDS directly; it does not translate or otherwise encapsulate DB-Library function calls.

Interprocess Communication Mechanisms

The choice of IPC mechanism is constrained by the operating system and network being used. For example, Microsoft SQL Server for OS/2 uses named pipes as its IPC mechanism, SYBASE SQL Server on UNIX uses TCP/IP sockets, and SYBASE on VMS uses DECnet Sockets. In a heterogeneous environment, multiple IPC mechanisms might be used on a single computer.

SQL Server for Windows NT can communicate over multiple IPC mechanisms. SQL Server communicates on named pipes (over either NetBEUI or TCP/IP network protocols) with Windows, Windows NT, MS-DOS, and OS/2 clients. It can also simultaneously support TCP/IP Sockets for communication with Macintosh, UNIX, or VMS clients and SPX sockets for communications in a Novell NetWare environment. SQL Server also supports Banyan VINES, DECnet Sockets, and AppleTalk.

Network Protocols

A network protocol is used to transport the data stream protocol over a network. It can be considered as the plumbing that supports the IPC mechanisms used by the data stream protocol, as well as supporting basic network operations such as file transfers and print sharing.

Back-end databases can reside on a local area network (LAN) that connects it with the client application, or it can reside at a remote site, connected via a wide area network (WAN) and/or gateway. In both cases, it is possible that the network protocols or physical network supported by the various back-end databases are different from those supported by the client or each other. In these cases, a client application must use different network protocols to communicate with various back-end databases.

The network transport protocols supported within SQL Server include NetBEUI, TCP/IP, SPX/IPX using NWLink, DECnet, AppleTalk, and VINES IP.

Net-Library Architecture

Microsoft SQL Server Net-Library architecture for client-server applications is based on the Net-Library concept that abstracts the client and server applications from the underlying network protocols being used. Figure 8.2 shows how SQL Server and related products can be accessed from practically any network environment.

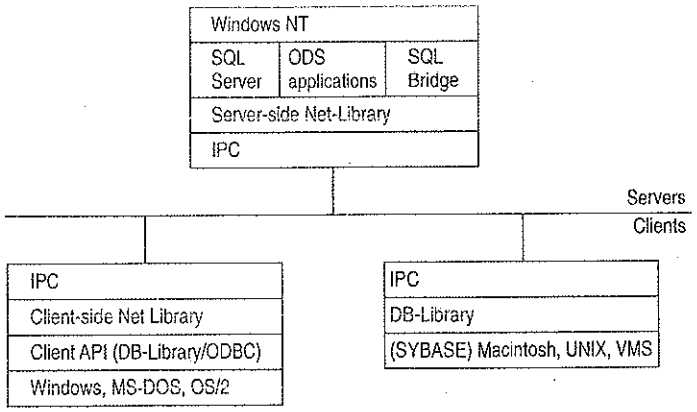


Figure 8.2 Net-Library Architecture

The Net-Library architecture provides a method of sending TDS (used by Microsoft SQL Server, Open Data Services, and SYBASE) via an IPC across a physical network connection. The Net-Library architecture also provides a transparent interface to the DB-Library APIs and the SQL Server driver for ODBC.

Net-Libraries are linked dynamically at run time. With the Microsoft Windows NT, Windows, and OS/2 operating systems, Net-Libraries are implemented as DLLs, and multiple Net-Libraries can be loaded simultaneously. With MS-DOS, Net-Libraries are implemented as terminate-and-stay-resident (TSR) programs, and only one can be loaded at a time.

The Net-Library architecture can be divided into two components—server-side Net-Libraries and client-side Net-Libraries.

Server-Side Net-Library Architecture

Microsoft SQL Server uses the server-side Net-Library architecture that was first introduced with Microsoft SQL Bridge. It can accept client requests across multiple network protocols at the same time.

Figure 8.3 illustrates the integration of server-side Net-Libraries with the various SQL Server–based products on the Windows NT platform.

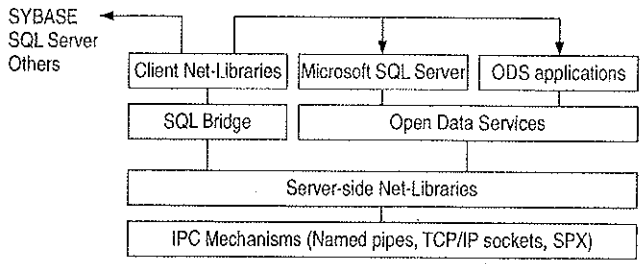


Figure 8.3 Server-Side Net-Library Architecture on the Windows NT Platform

The default Net-Library is named pipes.

When a server-side Net-Library is loaded by an application such as SQL Server, the Net-Library implements a network-specific way of establishing communication with clients and, in some cases, registers its presence on the network. SQL Server looks at the Windows NT Registry to determine which Net-Library to load on startup and which parameters to pass to it. The SQL Server Monitor process also uses a server-side Net-Library to communicate with clients and to search the following Registry key for network-specific parameters:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SQLServer\Server
```

At startup, SQL Server specifies a value for the *server_name* parameter in the SRV_CONFIG structure of Open Data Services. This value identifies which Registry key SQL Server will search for values of the **ListenOn** and *connection_string* Registry entries. (By default, SQL Server looks in HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SQLServer\Server.)

Each *connection_string* Registry value is read and passed on to the associated Net-Library (for example, named pipes) that is listed in the **ListenOn** field in the Server subkey. Each Net-Library acts upon the *connection_string* differently.

If there is no *connection_string* associated with the Net-Library, SQL Server does one of the following:

- If the Registry entry is under the SQL#Server\Server subkey, no connection string is passed as the default.
- If the Registry entry is not under SQL#Server\Server, *server_name* is passed as the default.

If the *server_name* subkey and the SQL#Server\Server subtree do not exist, or the Registry cannot be accessed, SQL Server assumes that the named pipes DLL (for the default Net-Library) is loaded, and no parameter is passed. (Named pipes access can be turned off by using the Registry Editor to explicitly delete the named pipes entry from the SQL#Server\Server subkey.)

Remote stored procedure calls and the Microsoft SQL Administrator tool also use the DB-Library/Net-Library architecture under Windows NT.

Client-Side Net-Library Architecture

When a call is made to open a connection to SQL Server, the API involved (DB-Library or the SQL Server driver for ODBC) determines which client-side Net-Library should be loaded to communicate with SQL Server or Open Data Services. (This process is described in more detail later in this chapter.)

Figure 8.4 shows client-side Net-Libraries used to communicate with SQL Server on the server side.

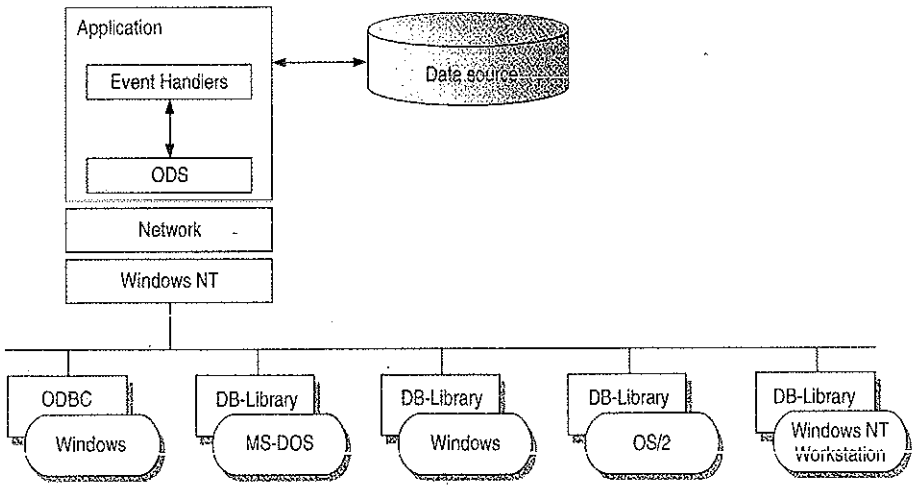


Figure 8.4 Client-Side Net-Library Architecture

Win32 DB-Library Architecture

Internally, a separate operating system thread is spawned for each connection that DB-Library makes with SQL Server. Each instance of the DB-Library DLL that is loaded by a calling process gets a private data area, while sharing code.

The Win32 DB-Library architecture differs from the implementation with Windows 3.*x*. In Windows 3.*x*, the DB-Library DLL has a single data segment that is shared among all calling processes. W3DBLIB.DLL maintains DB-Library connections as a linked list of connections in a single data segment. This architecture is required, because in Windows 3.*x* DLLs have a single data segment that is shared among all calling processes. This necessitates the initialization and clean up of the DB-Library DLL data structures through calls to the **dbinit** and **dbwinexit** functions.

The DB-Library functions for Win32 are located in NTWDBLIB.DLL, and the named pipe Net-Library is located in DBNMPNTW.DLL. (Be sure to set the PATH environment variable to include the directory where the DLLs reside.)

Another file, NTWDBLIB.LIB, contains import definitions that your applications for the Win32 API use. Set the LIB environment variable to include the directory where NTWDBLIB.LIB resides.

DB-Library resolves server names differently depending on the client platform.

Resolving Server Names for Clients Based on Windows, MS-DOS, OS/2, and Windows NT

When **dbopen** (the DB-Library function that initiates a client conversation with SQL Server) is called with the name of a SQL Server to connect to, DB-Library uses configuration information to determine which client-side Net-Library to load.

The client-side Net-Library configuration is stored in the following locations:

Client	Net-Library configuration is stored in
Windows 3.*x*	WIN.INI
MS-DOS	Environment variable
OS/2	OS2.INI
Windows NT	Windows NT Registry

DB-Library scans the **[SQLSERVER]** section of WIN.INI, OS2.INI, or the \SQLServer\Client\ConnectTo subtree of the Windows NT Registry looking for a logical name that matches the *servername* parameter specified in the call to **dbopen**. All items in the **[SQLSERVER]** section of the .INI file or in the Registry subtree have this format:

```
logical-name=Net-Lib-DLL-name[,network-specific-parameters]
```

Note Although some Net-Libraries need values for *network-specific-parameters*, this is optional for others that instead use defaults or determine the network-specific information required themselves.

DB-Library uses the following logic to determine which Net-Library to load:

- If a matching logical name is found in the .INI file or in the Windows NT Registry, DB-Library loads the specified Net-Library DLL. If network-specific parameters are present in the .INI entry or the Windows NT Registry, these are passed unmodified by DB-Library to the Net-Library DLL.
- If no matching logical name is found in the .INI file or in the Windows NT Registry, the DLL name (and optionally, the network-specific parameters) of the entry named DSQUERY will be used to load the required Net-Library. So, if you don't have a specific server name but do have a DSQUERY entry, that entry will be used as the default.
- If there is neither a specific logical name nor a DSQUERY entry in the .INI file or in the Windows NT Registry, DB-Library loads the named pipes Net-Library (for example, DBNMPP3.DLL for the Windows operating system) and passes it the *servername* parameter from **dbopen**. With Microsoft SQL Server using named pipes, you typically never need to make a .INI entry. If you use any other Net-Library, you must make at least one entry.

The following examples illustrate this logic:

- **forecast=dbnmp3**

  The Windows named pipe Net-Library is used, and it connects to SQL Server \\Forecast using the standard named pipe, \pipe\sql\query.

- **sales=dbnmp3,\\server1\pipe\sql2\query**

  The Windows named pipe Net-Library is used, and it connects to \\server1, where SQL Server has been started using an alternate named pipe, \pipe\sql2\query.

Note SQL Server can be directed to use an alternate pipe by adding an entry to the ListenOn field in the Registry under the following tree:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SQLServer\Server
```

- **dsquery=dbmsspx3**

  The SPX Net-Library is used, and the *servername* parameter from **dbopen** is used. This Net-Library requires no specific network connection information because it queries the NetWare Bindery to determine the location of the server running the Network Manager service specified in the *servername* parameter.

- **unixsrv=sybtcpw,131,107.005.21,3180**

  The SYBASE TCP/IP Net-Library is used, and DB-Library passes the IP address and port number contained in the initialization string to the Net-Library.

Note The SQL Server ODBC driver uses the same Net-Libraries as DB-Library to communicate with SQL Server, Open Data Services, and SQL Bridge.

Resolving Server Names for MS-DOS–Based Clients

With MS-DOS, only one Net-Library TSR can be loaded, so there is no .INI configuration. Instead, MS-DOS environment variables are used to specify any network-specific connection information. Environment variables have the following format:

```
logical-name=network-specific-parameters
```

The Net-Library used is the currently loaded TSR. If the *servername* parameter passed to **dbopen** corresponds to a currently set environment variable, DB-Library passes the information contained in the environment string to the currently loaded Net-Library. In turn, Net-Library uses this information to determine server location and network-specific information parameters, if present. If no environment variable matches the *servername* passed to **dbopen**, DB-Library passes the *servername* parameter from **dbopen** to the currently loaded Net-Library.

New DB-Library Function Identifies SQL Servers

DB-Library version 4.20.20 and later includes a new function (**dbserverenum**) that enables applications to identify SQL Servers available on the network, regardless of which network operating system is being used. For details on the **dbserverenum** function, see the *Microsoft SQL Server Programmer's Reference for C*.

Configuration of the Net-Library

The Net-Library files and IPCs for each network protocol supported by Microsoft SQL Server are listed in the following table. These files are installed automatically using the SQL Server Setup utility on the server side and the SQL Client Configuration Utility on the Windows, Windows NT, MS-DOS, and OS/2 client side. The AUTOEXEC.BAT file is used to load the MS-DOS client Net-Library.

The server-side Net-Library is used by SQL Server and ODS applications. If SQL Server and ODS are on the same computer, ODS uses an alternate pipe.

Table 8.1 shows which files you need when installing SQL Server on various network operating systems with various network protocols. Use the following table to determine exactly which files need to be in place for servers and clients.

You can also use this table for troubleshooting, should there be difficulty in connecting a client workstation to Microsoft SQL Server.

Table 8.1 Server-Side and Client-Side Net-Library Files

Network interface	Network protocol	Network clients supported	Client-side Net-Library	Server-side Net-Library	Comments
Named Pipes	NetBEUI or TCP/IP	LAN Manager, Windows for Workgroups, and Windows NT clients	DBNMPIPE.EXE (MS-DOS), DBNMP3.DLL (Windows), DBNMPP.DLL (OS/2), DBNMPNTW.DLL (Windows NT)	SSNMPNTW.DLL	This network setup provides SQL Server Integrated Security with the Windows NT User Account Database.
	NWLink	Windows NT clients	DBNMPNTW.DLL (Windows NT)	SSNMPNTW.DLL	
Windows Sockets	TCP/IP	UNIX and MAC clients	Part of SYBASE Open Client	SSMSSOCN.DLL	This configuration provides multiple vendor integration.

Table 8.1 Server-Side and Client-Side Net-Library Files *(continued)*

Network interface	Network protocol	Network clients supported	Client-side Net-Library	Server-side Net-Library	Comments
		PC clients: FTP PC/TCP, HP ARPA Services, Wollongong PathWay, Novell LAN WorkPlace, AT&T® StarGroup, Sun PC-NFS, DEC PATHWORKS (DECnet), Microsoft TCP/IP for LAN Manager, and so on	DBMSSOCN.DLL (Windows NT), DBMSSOC3.DLL (Windows), DBMSSOC.EXE (MS-DOS)	SSMSSOCN.DLL	The corresponding Net-Libraries are available from SYBASE.
Windows Sockets	NWLink (IPX/SPX)	Novell NetWare 3.10+ (MS-DOS and Windows) and OS/2 Requestor, NSD004 (OS/2) clients	DBMSSPX.EXE (DOS), DBMSSPX3.DLL (Windows), DBMSSPXP.DLL (OS/2)	Novell: SSMSSPXN.DLL	The servername is registered with the Novell bindery service.
		NWLink	DBMSSPXN.DLL (Windows NT)		
VINES Sockets	VINES IP	Banyan VINES, 4.11 (rev.5)+ and Windows NT clients	DBMSVINE.EXE (DOS), DBMSVIN3.DLL (Windows), DBMSVINP.DLL (OS/2), DBMSVINN.DLL (Windows NT)	Banyan VINES: SSMSVINN.DLL	Registers to StreetTalk as the given service. Banyan VINES will automatically handle lookups of partial names or nicknames.

Notes NWLink is a Microsoft implementation of the IPX/SPX protocol. Alternative software available through Novell is fully expected sometime in the near future.

Using NetBEUI as the network protocol, the client workstation always uses a broadcast to locate the SQL Server(s) on the network. Also, with TCP/IP the client workstation always uses a broadcast to locate the SQL Server(s), provided that the servername and IP address are not located in the LMHOST file on the workstations.

Novell Connectivity

As shown by Table 8.1, in a Novell NetWare environment, SQL Server requires NWLink (installed through Network Control Panel) and the SSMSSPXN.DLL. This DLL is automatically installed on the server side, with the appropriate Registry entries, when you use SQL Server Setup and choose Change Network Support, then NWLink IPX/SPX.

The following is a sample of what is added to the Registry for Microsoft SQL Server on a Novell Network:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SQLServer\Server
ListenOn: REG_MULTI_SZ: SSNMPNTW, \\.\pipe\sql\query
    SSMSSPXN, CORAL (computername)
```

Windows and OS/2 client workstations require the Novell NetWare 3.10 or higher level of IPX. The SQL Client Configuration Utility that ships with SQL Server is used to specify the default network that the Windows and OS/2 clients will use. By choosing Novell IPX/SPX, the required DBMSSPX3.DLL is automatically installed on the Windows client side, and DBMSSPXP.DLL is installed on the OS/2 client side. This adds the appropriate entries in the WIN.INI file or the OS/2.INI file, respectively.

The following is a sample of what is added to the WIN.INI for Windows clients communicating with Microsoft SQL Server on a Novell Network:

```
[SQLSERVER]
DSQUERY=DBMSSPX3
```

MS-DOS clients require the same level of IPX that the Windows workstations do. DBMSSPX.EXE must be installed on the MS-DOS computer. This TSR can be loaded either manually or from AUOTEXEC.BAT.

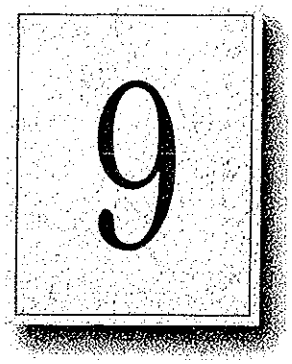
Windows NT client workstations use NWLink, which is installed through Network Control Panel. After installation, use the Client Configuration Utility to specify that the default network is Novell IPX/SPX. This, in turn, installs the required DBMSSPXN.DLL on the Windows NT client side.

The following is a sample Registry entry for Windows NT clients communicating with Microsoft SQL Server on a Novell Network:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SQLServer\Client\ConnectTo
DSQUERY: REG_SZ: DBMSSPXN
```

CHAPTER 9

Using Remote Access Service



Windows NT 3.5 Remote Access Service (RAS) connects remote or mobile workers to corporate networks. Optimized for client-server computing, Remote Access Service (RAS) is implemented primarily as a software solution, and is included in all of Microsoft's operating systems.

The goals in designing RAS were to make it:

- Secure
- Interoperable
- Economical
- Scalable
- High performance
- Easy to use
- Extensible

RAS Capabilities and Functionality

RAS provides transparent network access for computer running Windows NT, Windows for Workgroups, MS-DOS version 3.1 or later (RAS version 1.1a), and MS OS/2 version 3.1 (RAS version 1.1).

Users run the RAS graphical phone book on a remote computer, and then initiate a connection to the RAS server using a local modem, X.25, or ISDN card. The RAS server, running on a Windows NT Server-based computer connected to the corporate network, authenticates the users and services the sessions until terminated by the user or network administrator. All services that are typically available to a LAN-connected user (including file- and print-sharing, database access and messaging) are enabled via the RAS connection. The following figure depicts the RAS architecture:

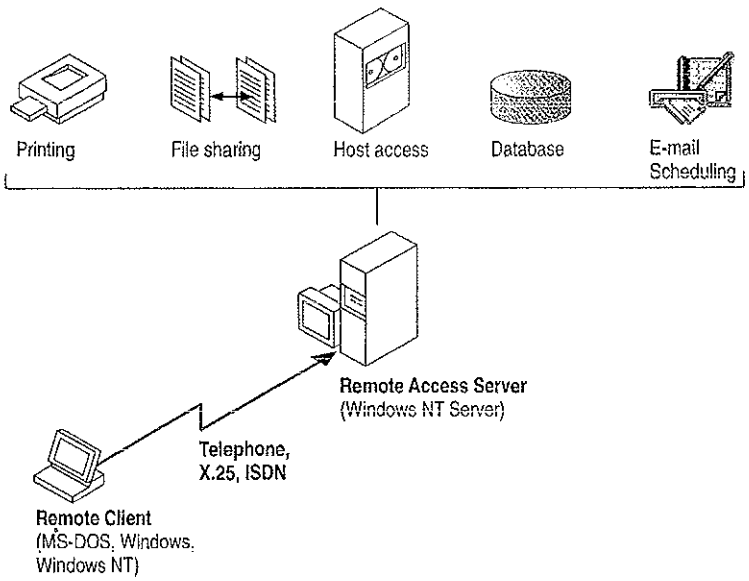


Figure 9.1 RAS Architecture

Note that the remote clients use standard tools to access resources. For example, the Windows File Manager is used to make drive connections, and Print Manager is used to connect printers. Connections made while LAN-connected via these tools are persistent, so users don't need to re-connect to network resources during their remote sessions. Since drive letters and UNC (Universal Naming Convention) names are fully supported via RAS, most commercial and custom applications work without any modification.

Connectivity is achieved in one of three ways: via a standard modem, ISDN card, or X.25. The asynchronous modem is the most popular means of connecting, with ISDN emerging as a high-speed alternative. X.25 is a standard for many companies doing business internationally.

Remote Access Versus Remote Control

In understanding the RAS architecture, it is important to make the distinction between RAS and remote control solutions, such as Cubix and pcANYWHERE. RAS is a software-based multi-protocol router; remote control solutions work by sharing screen, keyboard and mouse over the wire. In a remote control solution, users share a CPU or multiple CPU's on the server. The RAS server's CPU is dedicated to communications, not to running applications.

This architectural difference has significant implications in two areas: scalability and software applications architecture.

In the area of scalability, consider the differing approach to increasing the capacity or performance of a remote-control server. For best performance, an additional or upgraded CPU or computer would need to be purchased for every port to be added or upgraded. With RAS, additional ports can be added without upgrading the server computer. When it does require an upgrade, the RAS Server would generally get additional RAM, a less costly approach than with remote-control. With Windows NT, a single server can scale to support hundreds of remote users, using far fewer hardware resources than a remote control solution.

In software applications architecture, the RAS client normally executes applications from the remote workstation. Contrast this with the remote control client, which runs applications from the host-side CPU. The RAS arrangement is better suited to graphical, client-server—based applications, and because network traffic is reduced, the user achieves higher performance. Remote control, however, can be useful in non-client-server environments.

RAS Features in Windows NT 3.5

Microsoft's Remote Access Server first shipped with LAN Manager 2.1 in 1991. It was included with the Windows NT 3.1 operating system, and has now been significantly enhanced for Windows NT 3.5. RAS features the following capabilities:

- Multiprotocol routing via PPP support
- Internet support
- Improved integration with NetWare® networks
- Increased number of simultaneous connections
- Software data compression
- Data encryption
- Availability of the RAS APIs

Multi-protocol Routing via PPP Support

The underlying RAS architecture allows clients to run any combination of the network protocols NetBEUI, TCP/IP, or IPX during a RAS session. This means that Windows Sockets and NetWare-aware, as well as NetBIOS applications, can be run remotely. The Point-to-Point Protocol (PPP) is used as the framing mechanism on the wire. Using PPP enables a high degree of interoperability with existing remote access services.

Internet Support

RAS enables Windows NT and the next version of Windows, Windows95, to provide complete services to the Internet. A Windows NT Server 3.5-based computer can be configured as an Internet service provider, offering dial-up Internet connections to a client workstation running Windows NT 3.5 or Windows95. A computer running Windows NT Workstation 3.5 can dial into an Internet-connected computer running Windows NT Server 3.5, or to any one of a variety of industry-standard PPP or SLIP-based Internet servers.

Improved Integration with NetWare Networks

Windows NT 3.5 and RAS fully integrate into a NetWare network. The RAS clients are running IPX and/or NetBIOS, so all applications that typically work when directly connected to the network, continue to work when remotely connected. The RAS server now supports IPX routing, enabling remote clients to gain access to all NetWare resources via the RAS server.

Increased Number of Simultaneous Connections

Windows NT Server 3.5 supports up to 256 simultaneous connections. The Windows NT Workstation provides a single RAS connection, primarily for personal use or for very small networks.

Software Data Compression

Software data compression in RAS allows users to boost their effective throughput. Data is compressed by the RAS client, sent over the wire in a compressed format, and then decompressed by the server. In typical use, RAS software compression doubles effective throughput.

Data Encryption

Remote Access Service provides data encryption, in addition to password encryption, to provide privacy for sensitive data. While most RAS users do not need encryption, government agencies, law enforcement organizations, financial institutions, and others benefit from it. Microsoft RAS uses the RC4 encryption algorithm of RSA Data Security Inc.

RAS APIs

In April 1994, Microsoft published the 16-bit and 32-bit RAS APIs, which allow corporate developers and solution providers to create custom, remote-enabled applications that can establish a remote connection, use network resources, and re-connect in the event of a communications link failure. Applications developed using these tools will be compatible with Windows95, Windows NT Workstation and Server 3.5, and Windows for Workgroups 3.11.

Security

Microsoft's RAS provides security at the operating system, file system, and network layers, as well as data encryption and event auditing. Some of the security features are inherited from the Windows NT operating system, while others are specific to RAS itself. Every stage of the process—such as user authentication, data transmission, resource access, logoff and auditing—can be secured. The next section describes RAS security in detail.

Windows NT Security

Windows NT, the host for RAS, is a secure operating environment. Windows NT was designed to meet the requirements for C-2 level (U.S. Department of Defense) security, meaning that access to system resources can be discretely controlled, and all access to the system can be recorded and audited. A Windows NT Server-based computer, provided it is secured physically, can be locked-down using software. Any access to the system requires a password and leaves an audit trail.

Windows NT Server provides for enterprise-wide security using a *trusted domain, single-network logon* model. A domain is simply a collection of servers that are administered together. Trusted domains establish relationships whereby the users and groups of one domain can be granted access to resources in a trusting domain. This eliminates the need for duplicate entry of user accounts across a multi-server network. Finally, under the single-network-logon model, once a user is authenticated, the user carries access credentials. Anytime the user attempts to gain access to a resource anywhere on the network, Windows NT automatically presents the user's credentials. If trusted domains are used, the user may never have to present a password after initial logon, even though his account exists on one server in one domain only.

The single-network logon model extends to RAS users. RAS access is granted from the pool of all Windows NT user accounts. An administrator grants a single user, group of users, or all users the right to dial into the network. Then, users use their domain login to connect via RAS. Once the user has been authenticated by RAS, they can use resources throughout the domain and in any trusted domains.

Finally, Windows NT provides the Event Viewer for auditing. All system, application, and security events are recorded to a central secure database which, with proper privileges, can be viewed from anywhere on the network. Any attempts to violate system security, start or stop services without authorization, or gain access to protected resources, is recorded in the Event Log and can be viewed by the administrator.

Authentication

Authentication is an important concern for many corporations. This section answers some of the most frequently-asked questions, such as:

- How can our system insure the privacy of passwords?
- Can our system include a security mechanism in addition to that provided by RAS and Windows NT?
- Is the call-back feature supported?

Authentication Protocols

The Challenge Handshake Authentication Protocol (CHAP) is used by the Remote Access Server to negotiate the most secure form of encrypted authentication supported by both server and client. CHAP uses a challenge-response mechanism with one-way encryption on the response. CHAP allows the RAS server to negotiate downward from the most-secure to the least-secure encryption mechanism, and protects passwords transmitted in the process.

Table 9.1 Security Levels and RAS Encryption Protocols

Level of security	Type of encryption	RAS encryption protocol
High	One-way	CHAP, MD5
Medium	Two-way	SPAP
Low	Clear-text	PAP

CHAP allows different types of encryption algorithms to be used. Specifically, RAS uses DES and RSA Security Inc.'s MD5. Microsoft RAS uses DES encryption when both the client and the server are using RAS. DES encryption, the U.S. government standard, was designed to protect against password discovery and playback. Windows NT 3.5, Windows for Workgroups, and Windows95 will *always* negotiate DES-encrypted authentication when communicating with each other. When connecting to third-party remote access servers or client software, RAS can negotiate SPAP or clear-text authentication if the third party product does not support encrypted authentication.

MD5, an encryption scheme used by various PPP vendors for encrypted authentication, can be negotiated by the Microsoft RAS client when connecting to other vendors' remote access servers. MD5 is not available in the RAS server.

SPAP, the Shiva Password Authentication Protocol, is a two-way (reversible) encryption mechanism employed by Shiva. Windows NT Workstation 3.5, when connecting to a Shiva LAN Rover, uses SPAP; as does a Shiva client connecting to a Windows NT Server 3.5. This form of authentication is more secure than clear text, but less secure than CHAP.

PAP uses clear-text passwords and is the least sophisticated authentication protocol. It is typically negotiated if the remote workstation and server cannot negotiate a more secure form of validation.

The Microsoft RAS server has an option that prevents clear-text passwords from being negotiated. This option enables system administrators to enforce a high level of security.

Third-party Security Hosts

RAS supports third-party security hosts. The security host sits between the remote user and the RAS Server.

The security host generally provides an extra layer of security by requiring a hardware key of some sort in order to provide authentication. Verification that the remote user is in physical possession of the key takes place before they are given access to the RAS Server. This open architecture allows customers to choose from a variety of security hosts to augment the security in RAS.

As an additional measure of security, RAS offers call-back. Call-back security enables administrators to require remote users to dial from a specific predetermined location (e.g. telephone number at home) or to call back a user from any location, in order to use low-cost communications lines. In the case of secured call back, the user initiates a call, and connects with the RAS Server. The RAS Server then drops the call, and calls back a moment later to the pre-assigned call-back number. This security method will generally thwart most impersonators.

Network Access Restrictions

Remote access to the network under RAS is controlled by the system administrator. In addition to the tools provided with Windows NT Server (authentication, trusted domains, event auditing, C2 security design, etc.), the RAS Admin tool gives an administrator the ability to grant or revoke remote access privileges on a user-by-user basis. This means that even though RAS is running on a Windows NT Server-based computer, access to the network must be explicitly granted for each user who is to be authorized to enter the network via RAS.

This process ensures that remote access must be explicitly granted, and provides a convenient means for setting call back restrictions.

Microsoft's RAS provides an additional measure of security. The RAS Administrator provides a switch that allows access to be granted to all resources that the RAS host computer can see, or just resources local to the computer. This allows a customer to tightly control what information is available to remote users, and to limit their exposure in the event of a security breach.

Data Encryption

Data encryption protects data and ensures secure dial-up communications. This is especially important for financial institutions, law-enforcement and government agencies, and corporations that require secure data transfer. For installations where total security is required, the RAS administrator can set the RAS server to force encrypted communications. Users connecting to that server automatically encrypt all data sent.

Interoperability

Because LAN's are evolving quickly from islands of information to fully-connected networks of diverse operating systems, protocols, and file systems, Microsoft has defined interoperability as a key feature in Windows NT and RAS and has concentrated on the following areas to ensure smooth integration into the heterogeneous networks of today and tomorrow:

- Flexible hardware options
- PPP, an underlying protocol for interoperability
- A ramp to the Internet
- Seamless integration with NetWare networks
- Interoperability with other third-party remote access vendors

Flexible Hardware Options

Microsoft's Remote Access Service offers the broadest hardware support of any remote access vendor. Currently, over 1,700 computers, 300 modems, and 11 multi-port serial adapters are supported. By selecting a remote access solution with very broad hardware support, customers can gain flexibility in their system design. A complete listing of the hardware devices supported by RAS can be found in the Windows NT Hardware Compatibility List (HCL). The HCL ships with Windows NT, and can also be found on the Microsoft Download Service (206-936-MSDL) or on CompuServe (GO WINNT).

Point-to-Point Protocol: The Enabling Technology

Previous versions of RAS functioned as NetBIOS gateways. Users would make their connections using NetBEUI/ NetBIOS, and then inherit other protocols from the server. This method enabled users to share network resources in a multi-vendor LAN environment, but limited them from running applications which relied on the presence of a protocol other than NetBEUI on the client-side. The enhanced architecture is as follows:

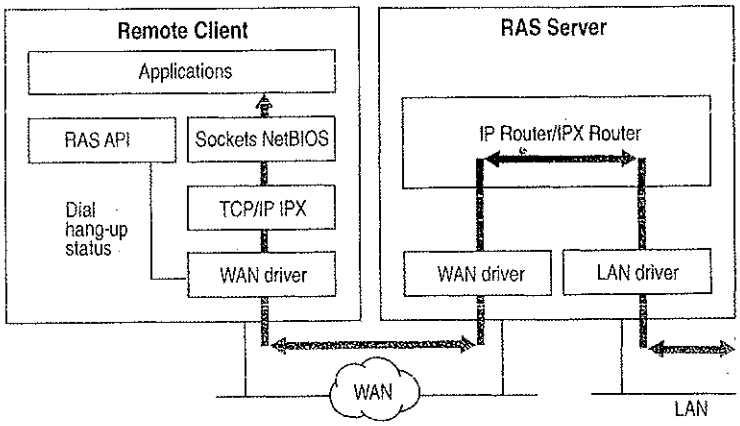


Figure 9.2 PPP Architecture

While this architecture continues to support the NetBIOS gateway, it also offers some exciting new possibilities. This architecture enables clients to load any combination of NetBEUI, IPX, and TCP/IP. Applications written to the Windows Sockets, NetBIOS, or IPX interface can now be run on a Windows NT Workstation. This architecture will be the basis for the RAS client in Windows95 as well.

Multi-protocol routing is just one of the benefits of Microsoft's move to the Point-to-Point Protocol (PPP) in RAS. The Point-to-Point Protocol is a set of industry standard protocols that enable remote access solutions to interoperate in a multi-vendor network. PPP support in Windows NT 3.5 and Windows95 means that workstations running Windows can dial into remote networks through any industry-standard PPP server. It also enables a Windows NT Server to receive calls from, and provide network access to, other vendors' remote access workstation software.

And while multi-protocol support is an important new feature of Microsoft's RAS, NetBIOS gateway support continues to be an important part of its feature set.

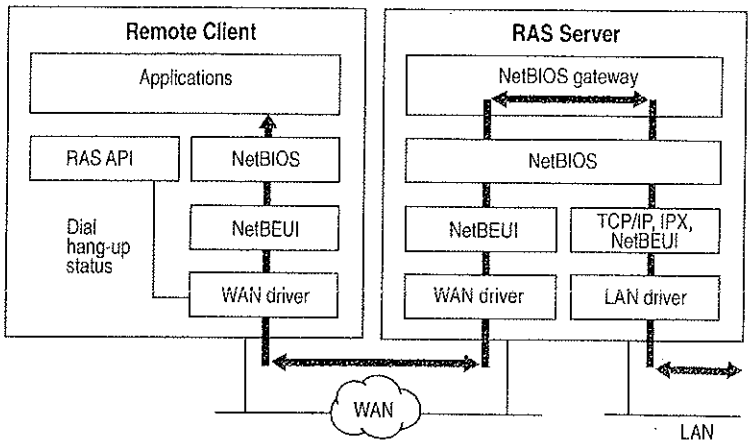


Figure 9.3 NetBIOS Gateway Architecture

An example of the NetBIOS gateway capability is remote network access for Lotus Notes users. While Lotus Notes does offer dial-up connectivity, dial up is limited to the Notes application only. RAS complements this connectivity by providing a low-cost, high-performance remote network connection for Notes® users, which not only connects Notes, but offers file and print services, and access to other network resources.

Many customers who are interested in PPP interoperability, are also concerned with SLIP. SLIP, the Serial Line Internet Protocol, is an older communications standard found in UNIX environments. SLIP does not provide automatic negotiation of network configuration; it requires user intervention. It also does not support encrypted authentication. Microsoft supports SLIP on the client side, so that the clients running Windows NT 3.5 may dial into an existing SLIP server. RAS does not provide a SLIP server in Windows NT version 3.5.

Using Terminal and Script Settings for Remote Logons

When you use RAS to connect to a remote computer, the remote computer will probably require a specific series of commands and responses to successfully log you on to the remote system. The sequence is identical each time you log on to the remote system.

If both the remote server and client are Windows NT 3.5 computers, connection and logon can be completely automated using Windows NT built-in security. If you log on to the Windows NT RAS client using a username and password that is valid on the remote network, and select the Authenticate Using Current User Name and Password check box in the Edit Phone Book Entry dialog box, Windows NT RAS will automatically connect to the remote Windows NT 3.5 RAS server.

If the remote computer you are logging on to is not a Windows NT 3.5 computer, you must configure the Security settings for each RAS entry to handle the log on requirements for the remote device you are connecting to. The remote logon will be either manual using a Terminal screen that allows you to interact with the remote computer, or you can automate the remote log on using scripts that are stored in SWITCH.INF or PAD.INF (for X.25 networks).

This section explains how to use the RAS Terminal screen and also describes how to create and use automatic scripts for logon to remote computers.

Using RAS Terminal for Remote Logons

If a remote computer you dial in to requires a log on procedure, you must configure the Security settings for that RAS entry to use a RAS Terminal log on as described in the procedure below. After RAS connects to the remote system, a character-based window will appear and display the log on sequence from the remote computer. You use this screen to interact with the remote computer for logging on. Alternatively, you can automate the manual log on through RAS Terminal as described in the next section, "Automating Remote Log Ons Using SWITCH.INF Scripts."

To configure a Windows NT 3.5 RAS entry to use Terminal after dialing

1. In Remote Access, select the entry you want to connect to.
2. Choose the Edit button.
3. If the Security button is not visible, choose the Advanced button.
4. Choose the Security button. (In Windows NT 3.1 and Windows for Workgroups 3.11, this button is labeled Switch).
5. In the After Dialing box, select Terminal. (In Windows NT 3.1 and Windows for Workgroups 3.11, this box is labeled Post-Connect).
6. Choose the OK button until you return to the main Remote Access Screen.

After you dial and connect to this entry, the After Dial Terminal screen will appear and you will see prompts from the remote computer. You then log on to the remote computer using the After Dial Terminal dialog box. After you have completed all interaction with the remote computer, choose the Done button to close the After Dial Terminal dialog box.

Automating Remote Log Ons Using SWITCH.INF Scripts

You can use the SWITCH.INF file (or PAD.INF on X.25 networks) to automate the log on process instead of using the manual RAS Terminal describe in the previous section, "Using RAS Terminal for Remote Log Ons."

Creating Scripts for RAS

SWITCH.INF is like a set of small batch files (scripts) contained in one file. A SWITCH.INF script has four elements: a section header, commands, responses, and comments.

Section headers divide SWITCH.INF into individual scripts. A section header starts a script.

Each line in a script is a command or a response. A command comes from the local RAS client. The commands you can issue from a Windows NT computer are listed below.

A response is from the remote device or computer. To write an automatic script, you must know the required responses for a specific device. The commands and responses must be in the exact order the remote device expects them. Branching statements, such as GOTO or IF, are not supported. The required sequence of commands and responses for a specific intermediary device should be in the documentation for the device, or if you are connecting to a commercial service, from the support staff of that service.

The SWITCH.INF file can contain scripts for each intermediary devices or online service that the RAS user will call. The scripts are activated by configuring Remote Access phonebook entries as described below in the section "Activating SWITCH.INF Scripts."

Note RAS permits you to embed your username and password only in clear text in the SWITCH.INF file. The ability to use macros that obtain your username and password from your own RAS phone book file (*username*.PBK) will be included in an upcoming, interim release of Windows NT. This functionality may be available by the time you are reading this. Check the RASPHONE.HLP file on your current system for the availability of these macros and for more information about creating scripts with SWITCH.INF.

Section Headers

A section header marks the beginning of a script for a certain intermediary device and must not exceed 31 characters. The section header is enclosed in square brackets. For example:

```
[Route 66 Login]
```

Comment Lines

Comment lines must have a semicolon (;) in column one and can appear anywhere in the file. Comment lines can contain information for those who maintain the SWITCH.INF file. For example:

```
; This script was created by MariaG on September 29, 1995
```

Commands

A command comes from the local computer. A response comes from the remote device or computer.

You use the **COMMAND=** statement to send commands to the intermediary device. The **COMMAND=** statement can be used three ways, as described below:

COMMAND=

COMMAND= by itself causes a 2-second delay, depending on CPU speed and whether or not caching software like SMARTDRV.DRV is running. Using **COMMAND=** as a delay is important because the intermediary device may not be able to process all commands if they are send at once.

COMMAND=*custom string*

This sends *custom string* but will also cause a slight delay of several hundred milliseconds (depending on CPU speed and caching software installed) to give the intermediary device time to process *custom string* and prepare for the next command.

COMMAND=*custom string <cr>*

This causes *custom string* to be sent instantaneously because of the carriage return (*<cr>*) at the end of the line.

You must consult the documentation from the remote device to determine the required strings to be send with the **COMMAND=** command.

Response Related Keywords

Each command line is followed by one or more response lines. You must consult the documentation from the remote device to determine the possible response strings.

In addition to the response strings you obtain for the remote device (or online service), response lines can contain one of the following keywords:

OK=*custom response string <macro>*
The script continues to the next line.

CONNECT=*custom response string <macro>*
Used at the end of a successful script.

ERROR=*custom response string <macro>*
Causes RAS to display a generic error message.

ERROR_DIAGNOSTICS=*custom response string* **<diagnostics>**
Causes RAS to display the specific cause for an error returned by the device. Not all devices report specific errors. Use **ERROR=** if you device does not return specific errors.

NoResponse
Used when no response will come from the remote device.

These commands are usually combined. **CONNECT=** is usually the last line executed unless an **ERROR** line follows it and the intermediary device reports an error.

RAS on the local computer always expects a response from the remote device and will wait until a response is received unless a **NoResponse** statement follows the **COMMAND=** line. If there is no statement for a response following a **COMMAND=** line, the **COMMAND=** line will execute and stop.

Reserved Macro Words

Reserved macro keyword are enclosed in angle brackets

<cr>
Inserts a carriage return.

<lf>
Inserts a line feed.

<match>
Reports a match if the string enclosed in quotation marks is found in the device response. For example, <match> "Smith" matches Jane Smith and John Smith III.

<?>
Inserts a wildcard character, for example, CO<?><?>2 matches COOL2 or COAT2, but not COOL3.

<hXX> (XX are hexadecimal digits)
Allows any hexadecimal character to appear in a string including the zero byte, <h00>.

<ignore>
Ignores the rest of a response from the macro on. For example, <cr><lf>CONNECTV-<ignore> reads the following responses as the same: "crlfCONNECTV-1.1" and "crlfCONNECTV-2.3."

<diagnostics>
Passes specific error information from a device to RAS. This enables RAS to display the specific error to RAS users. Otherwise, a nonspecific error message will appear.

Activating SWITCH.INF Scripts

You can configure a RAS entry to execute a SWITCH.INF script before dialing, after dialing, or both. For example, to automate a remote log on to a remote host, you would first create the script in SWITCH.INF, and then configure the RAS entry to use the created script after dialing.

To activate a script in Windows NT 3.5

1. In Remote Access, select the entry you want to connect to.
2. Choose the Edit button.
3. If the Security button is not visible, choose the Advanced button.

   Choose the Security button. (In Windows NT 3.1 and Windows for Workgroups 3.11, this button is labeled Switch).

   In the After Dialing box, select the name of the script. The section header in SWITCH.INF is what will appear as the name of the script. (In Windows NT 3.1 and Windows for Workgroups 3.11, this box is labeled Post-Connect).
4. Choose the OK button until you return to the main Remote Access Screen.

When you dial this entry, the selected script will execute after RAS dials and connects to the remote host.

Troubleshooting Scripts Using DEVICE.LOG

Windows NT 3.1 and 3.5 (and Windows for Workgroups 3.11) allow you to log all information passed between RAS, the modem, and the intermediate device, including errors reported by the intermediate device. This can allow you to find errors that prevent your scripts from working.

The DEVICE.LOG file is created by turning logging on in the registry. The DEVICE.LOG file is in the SYSTEM32\RAS subdirectory of your Windows NT directory.

▸ **To create DEVICE.LOG**

1. Hang up any connections, and then exit from Remote Access.
2. Start the Registry Editor by running the REGEDT32.EXE program.
3. Go to HKEY_LOCAL_MACHINE, and then access the following key:
   \SYSTEM\CurrentControlSet\Services\RasMan\Parameters

   Change the value of the Logging parameter to 1. When changed, the parameter should look like this:

   Logging:REG_DWORD:0x1

Logging begins when you restart Remote Access or start the Remote Access Server service (if your computer is receiving calls). You do not need to shutdown and restart Windows NT.

If an error is encountered during script execution, execution halts. You must determine the problem by looking in DEVICE.LOG, make the necessary corrections to the script, and then restart RAS.

To turn logging on in Windows for Workgroups 3.11, edit the SYSTEM.INI file and in the [Remote Access] section add the line LOGGING=1. The text file DEVICE.LOG will be created automatically in the Windows directory when RAS is started.

Using Scripts with Other Microsoft RAS Clients

Microsoft RAS version 1.0 does not have the capability to invoke RAS Terminal or use scripts in .INF files.

Microsoft RAS version 1.1 supports PAD.INF only. Note that the syntax used in the PAD.INF file differs slightly different from subsequent versions of Microsoft RAS.

Microsoft RAS for Windows for Workgroups version 3.11, Windows NT version 3.1 and version 3.5 support RAS Terminal and scripts in SWITCH.INF and PAD.INF.

Resource Directory

This resource directory provides contact information on many of the vendors that provide RAS-related equipment and support. It is not intended as an all-inclusive list of RAS-related products.

Digiboard
6400 Flying Cloud Drive
Eden Prairie, MN 55344
(612) 943-9020
Multi-port Serial Adapters, ISDN Adapters

Eicon Technology Corp.
2196 - 32nd Avenue (Lachine)
Montreal, Quebec H8T 3H7
Canada
(514) 631-2592
X.25 Adapters

NetManage, Inc.
20823 Stevens Creek Blvd.
Cupertino, CA 95014
Phone: (408) 973-7171
Fax: (408) 257-6405
Terminal Emulation, File Transfer, X Windows, E-mail, NFS, TN3270, BIND, SNMP

Security Dynamics
One Alewife Center
Cambridge, MA 02140 USA
Phone (617) 547-7820
Fax (617) 354 8836
Advanced network security and authorization products

Digital Pathways Inc
201 Ravendale Drive
Mountain View, CA 94043-5216
Phone (415) 964 0707
Fax (415) 961 7487
Advanced network security and authorization products

Racal
480 Spring Park Place
Suite 900
Herndon, Virginia 22070
Phone (703) 437 9333
Fax (703) 471 0892
Advanced network security and authorization products

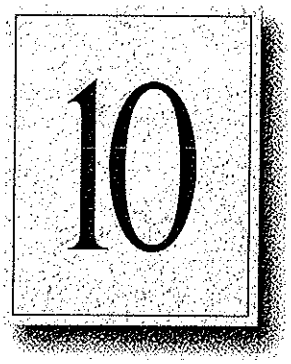
SpartaCom, Inc.
10, avenue du Québec
Bât. F4
B.P. 537
F-91946 Courtaboeuf Cedex
France
Phone (33-1) 69.07.17.80
Fax (33-1) 69.29.09.19

PART III

TCP/IP

CHAPTER 10

Overview of Microsoft TCP/IP for Windows NT



Transmission Control Protocol/Internet Protocol (TCP/IP) is a networking protocol that provides communication across interconnected networks made up of computers with diverse hardware architectures and various operating systems. TCP/IP can be used to communicate with Windows NT systems, with devices that use other Microsoft networking products, and with non-Microsoft systems, such as UNIX systems.

This chapter introduces Microsoft TCP/IP for Windows NT. The topics in this chapter include the following:

- Advantages of adding TCP/IP to a Windows NT configuration
- Microsoft TCP/IP core technology and third-party add-ons
- Windows NT solutions in TCP/IP internetworks

For more detailed information on TCP/IP and its integration with Microsoft Windows NT and other networking products, see Chapter 12, "Networking Concepts for TCP/IP."

Advantages of Adding TCP/IP to a Windows NT Configuration

The TCP/IP protocol family is a standard set of networking protocols, or rules, that govern how data is passed between computers on a network. TCP/IP is used to connect the Internet, the worldwide internetwork connecting over two million universities, research labs, U.S. defense installations, and corporations. These same protocols can be used in private internetworks that connect several local area networks.

Microsoft TCP/IP for Windows NT enables enterprise networking and connectivity on Windows NT computers. Adding TCP/IP to a Windows NT configuration offers the following advantages:

- A standard, routable enterprise networking protocol that is the most complete and accepted protocol available. All modern operating systems offer TCP/IP support, and most large networks rely on TCP/IP for much of their network traffic.
- A technology for connecting dissimilar systems. Many standard connectivity utilities are available to access and transfer data between dissimilar systems, including File Transfer Protocol (FTP) and Terminal Emulation Protocol (Telnet). Several of these standard utilities are included with Windows NT.
- A robust, scalable, cross-platform client-server framework. Microsoft TCP/IP supports the Windows Sockets 1.1 interface, which is ideal for developing client-server applications that can run with Windows Sockets-compliant stacks from other vendors. Many public-domain Internet tools are also written to the Windows Sockets standard. Windows Sockets applications can also take advantage of other networking protocols such as Microsoft NWLink, the Microsoft implementation of the IPX/SPX protocols used in Novell® NetWare® networks.
- The enabling technology necessary to connect Windows NT to the global Internet. TCP/IP, Point to Point Protocol (PPP), and Windows Sockets 1.1 provide the foundation needed to connect and use Internet services.

Microsoft TCP/IP Core Technology and Third-Party Add-Ons

Microsoft TCP/IP provides all the elements necessary to implement these protocols for networking. Microsoft TCP/IP includes the following:

- Core TCP/IP protocols, including the Transmission Control Protocol (TCP), Internet Protocol (IP), User Datagram Protocol (UDP), Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP). This suite of Internet protocols provides a set of standards for how computers communicate and how networks are interconnected. Support is also provided for PPP and Serial-Line IP (SLIP), which are protocols used for dial-up access to TCP/IP networks, including the Internet.
- Support for application interfaces, including Windows Sockets 1.1 for network programming, remote procedure call (RPC) for communicating between systems, NetBIOS for establishing logical names and sessions on the network, and network dynamic data exchange (Network DDE) for sharing information embedded in documents across the network.
- Basic TCP/IP connectivity utilities, including **finger**, **ftp**, **lpr**, **rcp**, **rexec**, **rsh**, **telnet**, and **tftp**. These utilities allow Windows NT users to interact with and use resources on non-Microsoft hosts, such as UNIX workstations.
- TCP/IP diagnostic tools, including **arp**, **hostname**, **ipconfig**, **lpq**, **nbtstat**, **netstat**, **ping**, **route**, and **tracert**. These utilities can be used to detect and resolve TCP/IP networking problems.
- Services and related administrative tools, including the FTP Server service for transferring files between remote computers, Windows Internet Name Service (WINS) for dynamically registering and querying computer names on an internetwork, Dynamic Host Configuration Protocol (DHCP) service for automatically configuring TCP/IP on Windows NT computers, and TCP/IP printing for accessing printers connected to a UNIX computer or connected directly to the network via TCP/IP.
- Simple Network Management Protocol (SNMP) agent. This component allows a Windows NT computer to be administered remotely using management tools such as SunNet Manager or HP Open View. SNMP can also be used to monitor DHCP servers and to monitor and configure WINS servers.

- The client software for simple network protocols, including Character Generator, Daytime, Discard, Echo, and Quote of the Day. These protocols allow a Windows NT computer to respond to requests from other systems that support these protocols. When these protocols are installed, a sample QUOTES files is also installed in the *\systemroot*\SYSTEM32\DRIVERS\ETC directory.
- Path MTU Discovery, which provides the ability to determine the datagram size for all routers between Windows NT computers and any other systems on the WAN. Microsoft TCP/IP also supports the Internet Gateway Multicast Protocol (IGMP), which is used by new workgroup software products.

The following figure shows the elements of Microsoft TCP/IP alongside the variety of additional applications and connectivity utilities provided by Microsoft and other third-party vendors.

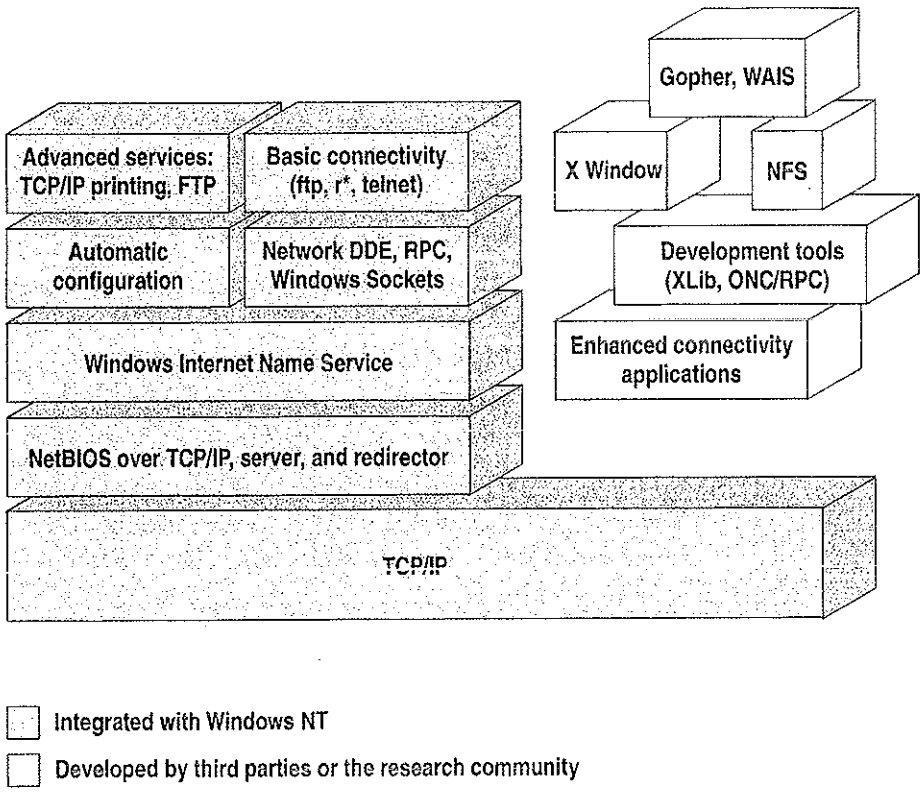


Figure 10.1 Microsoft TCP/IP Core Technology and Third-party Add-ons

TCP/IP standards are defined in *Requests for Comments* (RFCs), which are published by the Internet Engineering Task Force (IETF) and other working groups. The relevant RFCs supported in this version of Microsoft TCP/IP (and for Microsoft Remote Access Service) are described in the following table.

Table 10.1 Requests for Comments (RFCs) Supported by Microsoft TCP/IP

RFC	Title
768	User Datagram Protocol (UDP)
783	Trivial File Transfer Protocol (TFTP)
791	Internet Protocol (IP)
792	Internet Control Message Protocol (ICMP)
793	Transmission Control Protocol (TCP)
826	Address Resolution Protocol (ARP)
854	Telnet Protocol (TELNET)
862	Echo Protocol (ECHO)
863	Discard Protocol (DISCARD)
864	Character Generator Protocol (CHARGEN)
865	Quote of the Day Protocol (QUOTE)
867	Daytime Protocol (DAYTIME)
894	IP over Ethernet
919, 922	IP Broadcast Datagrams (broadcasting with subnets)
959	File Transfer Protocol (FTP)
1001, 1002	NetBIOS Service Protocols
1034, 1035	Domain Name System (DOMAIN)
1042	IP over Token Ring
1055	Transmission of IP over Serial Lines (IP-SLIP)
1112	Internet Gateway Multicast Protocol (IGMP)
1122, 1123	Host Requirements (communications and applications)
1134	Point to Point Protocol (PPP)
1144	Compressing TCP/IP Headers for Low-Speed Serial Links
1157	Simple Network Management Protocol (SNMP)

Table 10.1 Key Requests for Comments (RFCs) Supported by Microsoft TCP/IP *(continued)*

RFC	Title
1179	Line Printer Daemon Protocol
1188	IP over FDDI
1191	Path MTU Discovery
1201	IP over ARCNET
1231	IEEE 802.5 Token Ring MIB (MIB-II)
1332	PPP Internet Protocol Control Protocol (IPCP)
1334	PPP Authentication Protocols
1533	DHCP Options and BOOTP Vendor Extensions
1534	Interoperation Between DHCP and BOOTP
1541	Dynamic Host Configuration Protocol (DHCP)
1542	Clarifications and Extensions for the Bootstrap Protocol
1547	Requirements for Point to Point Protocol (PPP)
1548	Point to Point Protocol (PPP)
1549	PPP in High-level Data Link Control (HDLC) Framing
1552	PPP Internetwork Packet Exchange Control Protocol (IPXCP)
1553	IPX Header Compression
1570	Link Control Protocol (LCP) Extensions
Draft RFCs	NetBIOS Frame Control Protocol (NBFCP); PPP over ISDN; PPP over X.25; Compression Control Protocol

All RFCs can be found on the Internet via ds.internic.net.

In this version of Windows NT, Microsoft TCP/IP does not include a complete suite of TCP/IP connectivity utilities, Network File System (NFS) support, or some TCP/IP server services (daemons) such as **routed** and **telnetd**. Many such applications and utilities that are available in the public domain or from third-party vendors are compatible with Microsoft TCP/IP.

Tip For Windows for Workgroups computers and MS-DOS–based computers on a Microsoft network, you can install the new version of Microsoft TCP/IP—32 for Windows for Workgroups and the Microsoft Network Client version 2.0 for MS-DOS from the Windows NT Server 3.5 compact disc. This software includes the DHCP and WINS clients and other elements of the new Microsoft TCP/IP software. For information about installing these clients, see Chapter 9, "Network Client Administrator," in the *Windows NT Server Installation Guide.*

Windows NT Solutions in TCP/IP Internetworks

When TCP/IP is used as a transport protocol with Windows NT, Windows NT computers can communicate with other kinds of systems without additional networking software. Microsoft TCP/IP in combination with other parts of Windows NT provides a scalable solution for enterprise networks that include a mix of system types and software on many platforms.

This section summarizes how TCP/IP works with Windows NT to provide enterprise networking solutions. For information about how the elements discussed in this section fit within the networking architecture, see "TCP/IP and Windows NT Networking" in Chapter 12, "Networking Concepts for TCP/IP."

Using TCP/IP for Scalability in Windows Networks

TCP/IP delivers a scalable internetworking technology widely supported by hardware and software vendors.

When TCP/IP is used as the enterprise-networking protocol, the Windows-based networking solutions from Microsoft can be used on an existing internetwork to provide client and server support for TCP/IP and connectivity utilities. These solutions include:

- Microsoft Windows NT Workstation 3.5, with enhancements to support wide area networks (WAN), TCP/IP printing, extended LMHOSTS file, Windows Sockets 1.1, FTP Server service software, and DHCP and WINS client software.
- Microsoft Windows NT Server 3.5, with the same enhancements as Windows NT, plus DHCP server and WINS server software to support the implementation of these new protocols.
- Microsoft TCP/IP-32 for Windows for Workgroups 3.11, with Windows Sockets support, can be used to provide access for Windows for Workgroups computers to Windows NT, LAN Manager, and other TCP/IP systems. Microsoft TCP/IP-32 includes DHCP and WINS client software.
- Microsoft LAN Manager, including both client and server support for Windows Sockets, and MS-DOS–based connectivity utilities. The Microsoft Network Client 2.0 software on the Windows NT Server compact disc includes new Microsoft TCP/IP support with DHCP and WINS clients.

The current version of TCP/IP for Windows NT also supports IP routing in systems with multiple network adapters attached to separate physical networks (multihomed systems).

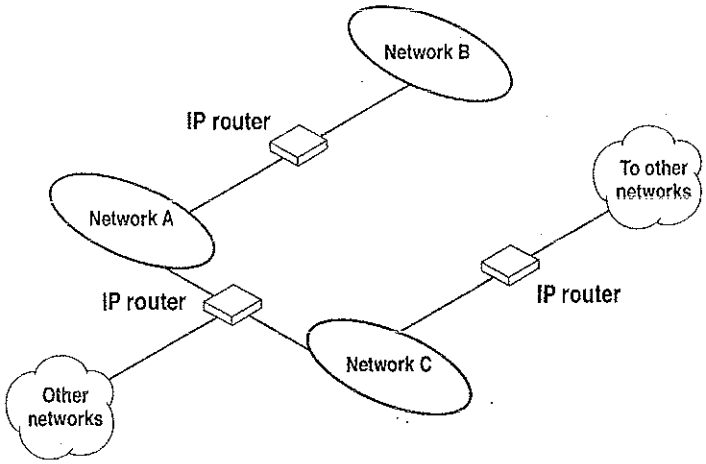


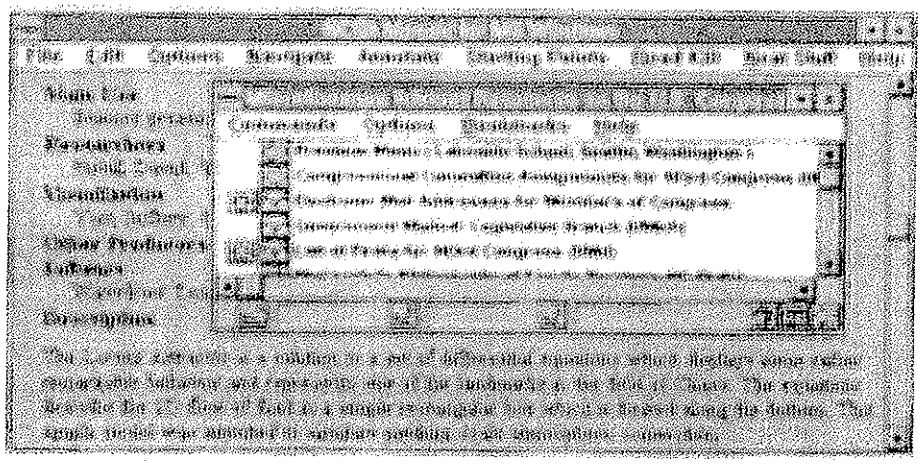
Figure 10.2 TCP/IP for Windows NT Supports IP Routing for Multihomed Systems

Using TCP/IP for Connectivity to the Internet

Microsoft TCP/IP provides Windows-based networking with a set of internetworking protocols based on open standards.

Microsoft TCP/IP for Windows NT includes many common connectivity applications such as **ftp**, **rsh**, and **telnet** that support file transfer, remote process execution, and terminal emulation for communication on the Internet and between non-Microsoft network systems.

TCP/IP applications created by researchers and other users, such as Gopher and NCSA Mosaic, are in the public domain or are available through other vendors as both 16-bit and 32-bit Windows-based applications. Any of these applications that follow the Windows Sockets 1.1 standard are compatible with Windows NT. Such applications allow a Windows NT computer to act as a powerful Internet client using the extensive internetworking components with public-domain viewers and applications to access Internet resources.



Tip Public-domain Windows-based utilities such as LPR and Gopher can be obtained on the Internet via ftp.cica.indiana.edu in the /pub/win3/nt or /pub/win3/winsock directory, or via the same directories on ftp.cdrom.com.

TCP/IP for Heterogeneous Networking

Because most modern operating systems (in addition to Windows NT) support TCP/IP protocols, an internetwork with mixed system types can share information using simple networking applications and utilities. With TCP/IP as a connectivity protocol, Windows NT can communicate with many non-Microsoft systems, including:

- Internet hosts
- Apple® Macintosh® systems
- IBM mainframes
- UNIX systems
- Open VMS™ systems
- Printers with network adapters connected directly to the network

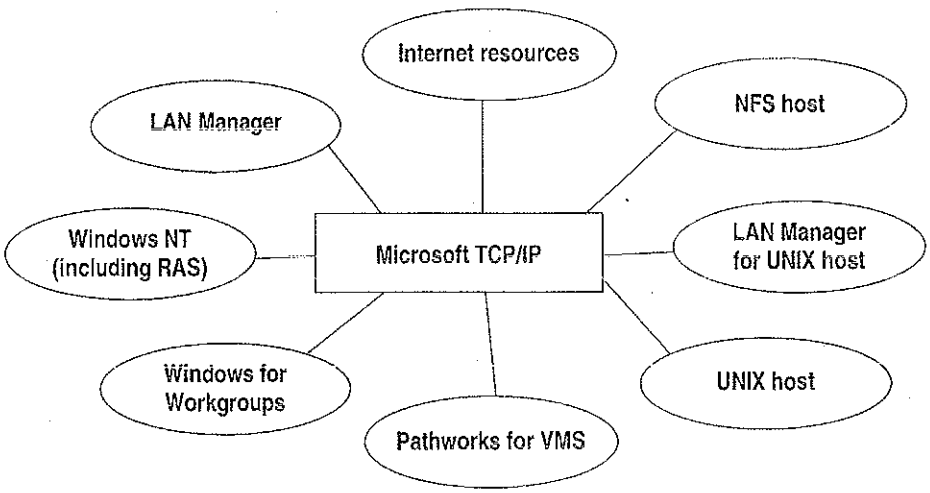


Figure 10.3 Microsoft TCP/IP Connectivity

Microsoft TCP/IP provides a framework for interoperable heterogeneous networking. The modular architecture of Windows NT networking with its transport-independent services contributes to the strength of this framework. For example, Windows NT supports these transport protocols, among many others:

- IPX/SPX for use in NetWare environments, using the Microsoft NWLink transport. Besides providing interoperability with NetWare networks, IPX/SPX is a fast LAN transport for Windows-based networking as well.
- TCP/IP for internetworks based on IP technologies. TCP/IP is the preferred transport for internetworks and provides interoperability with UNIX and other TCP/IP-based networks.

- NetBEUI as the protocol for local area networking on smaller networks and compatibility with existing LAN Manager and IBM LAN Server networks.
- AppleTalk for connecting to and sharing resources with Macintosh systems.

Other transport protocols provided by third-party vendors, such as DECnet and OSI, can also be used by Windows NT networking services.

Windows NT provides standard network programming interfaces through the Windows Sockets, RPC, and NetBIOS interfaces. Developers can take advantage of this heterogeneous client-server platform to create custom applications that will run on any system in the enterprise. An example of such a service is Microsoft SQL Server, which uses Windows Sockets 1.1 to provide access to NetWare, MS-DOS–based, Windows NT, and UNIX clients.

Using TCP/IP with Third-Party Software

TCP/IP is a common denominator for heterogeneous networking, and Windows Sockets is a standard used by application developers. Together they provide a framework for cross-platform client-server development. TCP/IP-aware applications from vendors that comply with the Windows Sockets standards can run over virtually any TCP/IP implementation.

The Windows Sockets standard ensures compatibility with Windows-based TCP/IP utilities developed by more than 30 vendors. This includes third-party applications for the X Window System, sophisticated terminal emulation software, NFS, electronic mail packages, and more. Because Windows NT offers compatibility with 16-bit Windows Sockets, applications created for Windows 3.*x* Windows Sockets run over Windows NT without modification or recompilation.

For example, third-party applications for X Window provide strong connectivity solutions by means of X Window servers, database servers, and terminal emulation. With such applications, a Windows NT computer can work as an X Window server platform while retaining compatibility with applications created for Windows NT, Windows 3.1, and MS-DOS on the same system. Other third-party software includes X Window client libraries for Windows NT, which allow developers to write X Window client applications on Windows NT that can be run and displayed remotely on X Window server systems.

The Windows Sockets API is a networking API used by programmers creating applications for both the Microsoft Windows NT and Windows operating systems. Windows Sockets is an open standard that is part of the Microsoft Windows Open System Architecture (WOSA) initiative. It is a public specification based on Berkeley UNIX sockets, which means that UNIX applications can be quickly ported to Microsoft Windows and Windows NT. Windows Sockets provides a single standard programming interface supported by all the major vendors implementing TCP/IP for Windows systems.

The Windows NT TCP/IP utilities use Windows Sockets, as do 32-bit TCP/IP applications developed by third parties. Windows NT also uses the Windows Sockets interface to support Services for Macintosh and IPX/SPX in NWLink. Under Windows NT, 16-bit Windows-based applications created under the Windows Sockets standard will run without modification or recompilation. Most TCP/IP users will use programs that comply with the Windows Sockets standard, such as **ftp** or **telnet** or third-party applications.

The Windows Sockets standard allows a developer to create an application with a single common interface and a single executable that can run over many of the TCP/IP implementations provided by vendors. The goals for Windows Sockets are the following:

- Provide a familiar networking API to programmers using Windows NT, Windows for Workgroups, or UNIX
- Offer binary compatibility between vendors for heterogeneous Windows-based TCP/IP stacks and utilities
- Support both connection-oriented and connectionless protocols

Typical Windows Sockets applications include graphic connectivity utilities, terminal emulation software, Simple Mail Transfer Protocol (SMTP) and electronic mail clients, network printing utilities, SQL client applications, and corporate client-server applications.

If you are interested in developing a Windows Sockets application, specifications for Windows Sockets are available on the Internet from ftp.microsoft.com, on CompuServe® in the MSL library, and in the Microsoft Win32 Software Developers Kit.

▷ **To get a copy of the Windows Sockets specification via anonymous FTP**

1. Make sure you have write permission in your current working directory.
2. At the command prompt, start **ftp**, and then connect to **ftp.microsoft.com** (or **198.105.232.1**).
3. Log on as **anonymous**.
4. Type your electronic mail address for the *password*.
5. Type **cd \advsys\winsock\spec11**, and then press ENTER.
6. Use the **dir** command to see the list of available file types. If you want binary data such as in the Microsoft Word version, type **bin**, and then press ENTER.
7. Determine the file with the format you want [for example, ASCII (.TXT), PostScript® (.PS), or Microsoft Word (.DOC)], and then type **get winsock.***ext* where *ext* is the format that you want, such as **winsock.doc** for the Microsoft Word version.

▸ **To get a copy of the Windows Sockets specification from CompuServe**

1. At the command prompt, type **go msl**, and then press ENTER.
2. Browse using the keywords **windows sockets**.
3. Choose the file with the format you want [ASCII (.TXT), PostScript (.PS), or Microsoft Word for Windows (.DOC)], and then type **get winsock**.*ext*.

There is also an electronic mailing list designed for discussion of Windows Sockets programming.

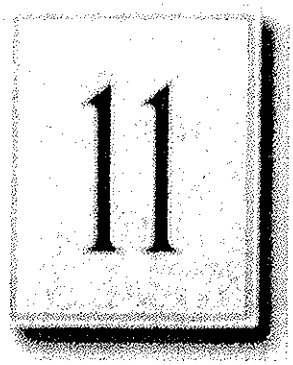
▸ **To subscribe to the Windows Sockets mailing list**

- Send electronic mail to listserv@sunsite.unc.edu with a message body that contains **subscribe winsock** *user's-email-address*.

  You can use the same procedure to subscribe to two mailing lists called **winsock-hackers** and **winsock-users**.

CHAPTER 11

Installing and Configuring Microsoft TCP/IP and SNMP



This chapter explains how to install Microsoft TCP/IP and the SNMP service for Windows NT and how to configure the protocols on your computer.

The TCP/IP protocol family can be installed as part of Custom Setup when you install Windows NT, following the steps described in this chapter. Also, if you upgrade to a new version of Windows NT, Setup automatically installs the new TCP/IP protocol and preserves your previous TCP/IP settings. This chapter assumes that Windows NT has been successfully installed on your computer but TCP/IP has not been installed.

The following topics appear in this chapter:

- Before installing Microsoft TCP/IP
- Installing TCP/IP
- Configuring TCP/IP
- Configuring TCP/IP to use DNS
- Configuring advanced TCP/IP options
- Configuring SNMP
- Removing TCP/IP components
- Configuring Remote Access Service (RAS) for use with TCP/IP

You must be logged on as a member of the Administrators group to install and configure all elements of TCP/IP.

Before Installing Microsoft TCP/IP

Important The values that you use for manually configuring TCP/IP and SNMP must be supplied by the network administrator.

Check with your network administrator to find out the following information before you install Microsoft TCP/IP on a Windows NT computer:

- Whether you can use Dynamic Host Configuration Protocol (DHCP) to configure TCP/IP. You can choose this option if a DHCP server is installed on your internetwork. You cannot choose this option if this computer will be a DHCP server. For information, see "Using Dynamic Host Configuration Protocol" later in this chapter.
- Whether this computer will be a DHCP server. This option is available only for Windows NT Server. For information, see Chapter 13, "Installing and Configuring DHCP Servers."
- Whether this computer will be a Windows Internet Name Service (WINS) server. This option is available only for Windows NT Server. For information, see Chapter 14, "Installing and Configuring WINS Servers."
- Whether this computer will be a WINS proxy agent. For information, see "Windows Internet Name Service and Broadcast Name Resolution" in Chapter 12, "Networking Concepts for TCP/IP."

If you cannot use DHCP for automatic configuration, you need to obtain the following values from the network administrator so you can configure TCP/IP manually:

- The IP address and subnet mask for each network adapter card installed on the computer. For information, see "IP Addressing" in Chapter 12, "Networking Concepts for TCP/IP."
- The IP address for the default local gateways (IP routers).
- Whether your computer will use Domain Name System (DNS) and, if so, the IP addresses and DNS domain name of the DNS servers on the internetwork. For information, see "Domain Name System Addressing" in Chapter 12 "Networking Concepts for TCP/IP."
- The IP addresses for WINS servers, if WINS servers are available on your network.

You need to know the following information before you install the Simple Network Management Protocol (SNMP) service on your computer, as described in "Configuring SNMP" later in this chapter:

- Community names in your network
- Trap destination for each community
- IP addresses or computer names for SNMP management hosts

Although the Windows NT SNMP management agent supports management consoles over both IPX and UDP protocols, SNMP must be installed in conjunction with the other TCP/IP services. Once SNMP is installed, no additional configuration is needed to manage over IPX. If IPX is installed, SNMP automatically runs with it.

Installing TCP/IP

You must be logged on as a member of the Administrators group for the local computer to install and configure TCP/IP.

▷ To install Microsoft TCP/IP on a Windows NT computer

1. Double-click the Network icon in Control Panel to display the Network Settings dialog box.
2. Choose the Add Software button to display the Add Network Software dialog box.
3. Select TCP/IP Protocol And Related Components from the Network Software box, and then choose the Continue button.
4. In the Windows NT TCP/IP Installation Options dialog box, select the options for the TCP/IP components you want to install, as described in the table that follows this procedure. If any TCP/IP elements have been installed previously, they are dimmed and not available. When you have selected the options you want, choose the Continue button.

   While you are installing or configuring TCP/IP, you can read the hint bar at the bottom of each TCP/IP dialog box for information about a selected item, or choose the Help button to get detailed online information.

   Windows NT Setup displays a message prompting for the full path to the Windows NT distribution files.

5. In the Windows NT Setup dialog box, enter the full path to the Windows NT distribution files, and then choose the Continue button.

   You can specify a drive letter for floppy disks, a CD-ROM drive, or a shared network directory, or you can specify the Universal Naming Convention (UNC) path name for a network resource, such as \\NTSETUP\MASTER.

   All necessary files are copied to your hard disk.

   Note If you are installing from floppy disks, Windows NT Setup might request disks more than once. This behavior is normal and not an error condition.

6. If you selected the options for installing the SNMP and FTP Server services, you are automatically requested to configure these services.

   Follow the directions provided in the online Help for these dialog boxes. For additional details, see "Configuring SNMP" later in this chapter, and see also Chapter 16, "Using the Microsoft FTP Server Service."

7. In the Network Settings dialog box, choose the OK button.

   If you selected the Enable Automatic DHCP Configuration option and a DHCP server is available on your network, all configuration settings for TCP/IP are completed automatically, as described in "Using Dynamic Host Configuration Protocol" later in this chapter.

   If you did not check the Enable Automatic DHCP Configuration option, continue with the configuration procedures described in "Configuring TCP/IP Manually" later in this chapter. TCP/IP must be configured in order to operate.

   If you checked the DHCP Server Service or WINS Server Service options, you must complete the configuration steps described in Chapter 13, "Installing and Configuring DHCP Servers," and Chapter 14, "Installing and Configuring WINS Servers."

Table 11.1 Windows NT TCP/IP Installation Options

Option	Usage
TCP/IP Internetworking	Includes the TCP/IP protocol, NetBIOS over TCP/IP and Windows Sockets interfaces, and the TCP/IP diagnostic utilities. These elements are installed automatically.
Connectivity Utilities	Installs the TCP/IP utilities. Select this option to install the connectivity utilities described in Appendix A, "TCP/IP Utilities Reference."
SNMP Service	Installs the SNMP service. Select this option to allow this computer to be administered remotely using management tools such as Sun Net Manager or HP Open View. This option also allows you to monitor statistics for the TCP/IP services and WINS servers using Performance Monitor, as described in Chapter 17, "Using Performance Monitor with TCP/IP Services."
TCP/IP Network Printing Support	Enables this computer to print directly over the network using TCP/IP. Select this option if you want to print to UNIX print queues or TCP/IP printers that are connected directly to the network, as described in Chapter 18, "Internetwork Printing with TCP/IP." This option must be installed if you want to use the **Lpdsvr** service so that UNIX computers can print to Windows NT printers.
FTP Server Service	Enables files on this computer to be shared over the network with remote computers that support FTP and TCP/IP (especially non-Microsoft network computers). Select this option if you want to use TCP/IP to share files with other computers, as described in Chapter 16, "Using the Microsoft FTP Server Service."
Simple TCP/IP Services	Provides the client software for the Character Generator, Daytime, Discard, Echo, and Quote of the Day services. Select this option to allow this computer to respond to requests from other systems that support these protocols.

Table 11.1 Windows NT TCP/IP Installation Options *(continued)*

Option	Usage
DHCP Server Service	Installs the server software to support automatic configuration and addressing for computers using TCP/IP on your internetwork. This option is available only for Windows NT Server. Select this option if this computer is to be a DHCP Server, as described in Chapter 13, "Installing and Configuring DHCP Servers." If you select this option, you must manually configure the IP address, subnet mask, and default gateway for this computer.
WINS Server Service	Installs the server software to support WINS, a dynamic name resolution service for computers on a Windows internetwork. This option is available only for Windows NT Server. Select this option if this computer is to be installed as a primary or secondary WINS server, as described in Chapter 14, "Installing and Configuring WINS Servers." Do not select this option if this computer will be a WINS proxy agent.
Enable Automatic DHCP Configuration	Turns on automatic configuration of TCP/IP parameters for this computer. Select this option if there is a DHCP server on your internetwork to support dynamic host configuration. This option is the preferred method for configuring TCP/IP on most Windows NT computers. This option is not available if the DHCP Server Service or WINS Server Service option is selected.

If you have trouble installing Microsoft TCP/IP on your computer, follow the suggestions in the error messages displayed on the screen. You can also use diagnostic utilities such as **ping** to isolate network hardware problems and incompatible configurations. For information, see Chapter 19, "Troubleshooting TCP/IP."

After TCP/IP is installed, the *\systemroot*\SYSTEM32\DRIVERS\ETC directory contains several files, including default HOSTS, NETWORKS, PROTOCOLS, QUOTES, and SERVICES files plus a sample LMHOSTS.SAM file that describes the format for this file.

Configuring TCP/IP

For TCP/IP to work on your computer, it must be configured with the IP addresses, subnet mask, and default gateway for each network adapter on the computer. Microsoft TCP/IP can be configured using two different methods:

- If there is a DHCP server on your internetwork, it can automatically configure TCP/IP for your computer using DHCP.
- If there is no DHCP server, or if you are configuring a Windows NT Server computer to be a DHCP server, you must manually configure all TCP/IP settings.

These configuration methods are described in the following sections.

Using DHCP

The best method for ensuring easy and accurate installation of TCP/IP is to use automatic DHCP configuration, which uses DHCP to configure your local computer with the correct IP address, subnet mask, and default gateway.

You can take advantage of this method for configuring TCP/IP if there is a DHCP server installed on your network. The network administrator can tell you if this option is available. You cannot use DHCP configuration for a server that you are installing as a DHCP server or a WINS server. You must configure TCP/IP settings manually for DHCP servers, as described in "Configuring TCP/IP Manually" later in this chapter.

▶ **To configure TCP/IP using DHCP**

1. Make sure the Enable Automatic DHCP Configuration option is checked in either the Windows NT TCP/IP Installation Options dialog box or the TCP/IP Configuration dialog box.
2. When you restart the computer after completing TCP/IP installation, the DHCP server automatically provides the correct configuration information for your computer.

If you subsequently attempt to configure TCP/IP in the Network Settings dialog box, the system warns you that any manual settings will override the automatic settings provided by DHCP. As a general rule, you should not change the automatic settings unless you specifically want to override a setting provided by DHCP. For detailed information about DHCP, see "Dynamic Host Configuration Protocol" in Chapter 12, "Networking Concepts for TCP/IP."

Configuring TCP/IP Manually

After the Microsoft TCP/IP protocol software is installed on your computer, you must manually provide valid addressing information if you are installing TCP/IP on a DHCP server or a WINS Server, or if you cannot use automatic DHCP configuration.

For a WINS server computer that has more than one network adapter card, WINS always binds to the first adapter in the list of adapters bound by TCP/IP. Make sure that this adapter address is not set to 0, and that the binding order of IP addresses is not disturbed.

You must be logged on as a member of the Administrators group for the local computer to configure TCP/IP.

Caution Be sure to use the values for IP addresses and subnet masks that are supplied by your network administrator to avoid duplicate addresses. If duplicate addresses do occur, this can cause some computers on the network to function unpredictably. For more information, see "IP Addressing" in Chapter 12, "Networking Concepts for TCP/IP."

To manually configure the TCP/IP protocol

1. Complete one of the following tasks:

   If you are installing TCP/IP, perform the following steps.

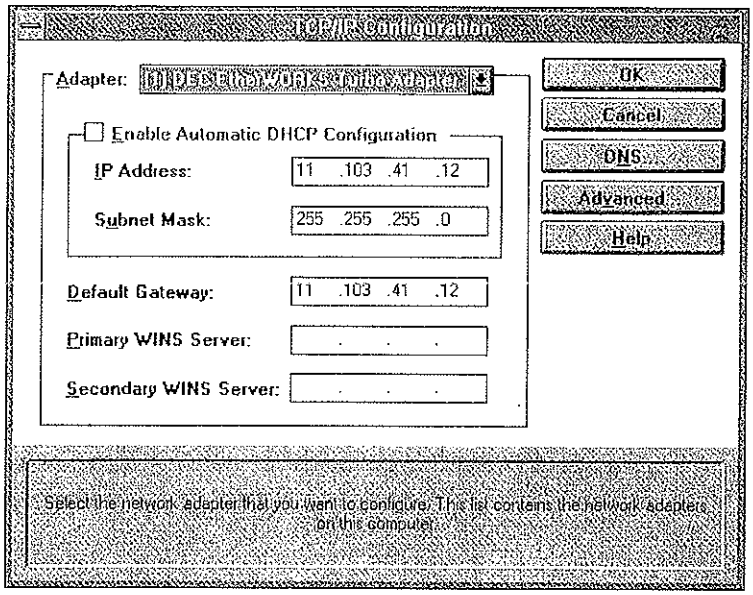
   - Complete all options in the Windows NT TCP/IP Installation Options dialog box, and then choose OK to display the Network Settings dialog box.
   - Choose the OK button to display the Microsoft TCP/IP Configuration dialog box.

   –Or–

   If you are reconfiguring TCP/IP, perform the following steps.

   - Double-click the Network option in Control Panel to display the Network Settings dialog box.

- In the Installed Network Software box, select TCP/IP Protocol, and then choose the Configure button to display the TCP/IP Configuration dialog box.



2. In the Adapter box, select the network adapter for which you want to set IP addresses.

   The Adapter list contains all network adapters to which IP is bound on this computer. This list includes all adapters installed on this computer.

   You must set specific IP addressing information for each bound adapter with correct values provided by the network administrator. The bindings for a network adapter determine how network protocols and other layers of network software work together.

3. For each bound network adapter, type values in the IP Address and Subnet Mask boxes.

   - The value in the IP Address box identifies the IP address for your local computer or, if more than one network card is installed in the computer, for the network adapter card selected in the Adapter box.

- The value in the Subnet Mask box identifies the network membership for the selected network adapter and its host ID. This allows the computer to separate the IP address into host and network IDs. The subnet mask defaults to an appropriate value, as shown in the following table:

Table 11.2 Subnet Mask Defaults

Address class	Range of first octet in IP address	Subnet mask
Class A	1–126	255.0.0.0
Class B	128–191	255.255.0.0
Class C	192–223	255.255.255.0

4. For each network adapter on the computer, type the correct IP address value in the Default Gateway box, as provided by the network administrator.

   This value specifies the IP address of the default gateway (or IP router) used to forward packets to other networks or subnets. This value should be the IP address of your local gateway.

   This parameter is required only for systems on internetworks. If this parameter is not provided, IP functionality is limited to the local subnet unless a route is specified with the TCP/IP **route** utility, as described in Appendix A, "TCP/IP Utilities Reference."

   If your computer has multiple network cards, additional default gateways can be added using the Advanced Microsoft TCP/IP Configuration dialog box, as described later in this chapter.

5. If there are WINS servers installed on your network and you want to use WINS in combination with broadcast name queries to resolve computer names, type IP addresses in the boxes for the primary and, optionally, the secondary WINS servers.

   The network administrator should provide the correct values for these parameters. These are global values for the computer, not just individual adapters.

   If an address for a WINS server is not specified, this computer uses name query broadcasts (the b-node mode for NetBIOS over TCP/IP) plus the local LMHOSTS file to resolve computer names to IP addresses. Broadcast resolution is limited to the local network.

Note WINS name resolution is enabled and configured automatically for a computer that is configured with DHCP.

On a WINS server, NetBIOS over TCP/IP (NETBT.SYS) uses WINS on the local computer as the primary name server, regardless of how name resolution might be configured. Also, NetBIOS over TCP/IP binds to the first IP address on a network adapter and ignores any additional addresses.

For overview information about name resolution options, see "Name Resolution for Windows Networking" in Chapter 12 "Networking Concepts for TCP/IP." For detailed information about installing and configuring WINS servers, see Chapter 14, "Installing and Configuring WINS Servers."

6. If you want to configure the advanced TCP/IP options for multiple gateways and other items, choose the Advanced button, and then continue with the configuration procedure, as described in "Configuring Advanced TCP/IP Options" later in this chapter.
7. If you want to use DNS for host name resolution, choose the DNS button, and then continue with the configuration procedure, as described in the next section.
8. If you do not want to configure DNS or advanced options, or if you have completed the other configuration procedures, choose the OK button. When the Network Settings dialog box is displayed again, choose the OK button.

   Microsoft TCP/IP has been configured. If you are installing TCP/IP for the first time, you must restart the computer for the configuration to take effect. If you are changing your existing configuration, you do not have to restart your computer.

After TCP/IP is installed, the *\systemroot*\SYSTEM32\DRIVERS\ETC directory contains a default HOSTS file and a sample LMHOSTS.SAM file. The network administrator might require that replacement HOSTS and LMHOSTS files be used instead of these default files.

Configuring TCP/IP to Use DNS

Although TCP/IP uses IP addresses to identify and reach computers, users typically prefer to use computer names. DNS is a naming service generally used in the UNIX networking community to provide standard naming conventions for IP workstations. Windows Sockets applications and TCP/IP utilities, such as **ftp** and **telnet**, can also use DNS in addition to the HOSTS file to find systems when connecting to foreign hosts or systems on your network.

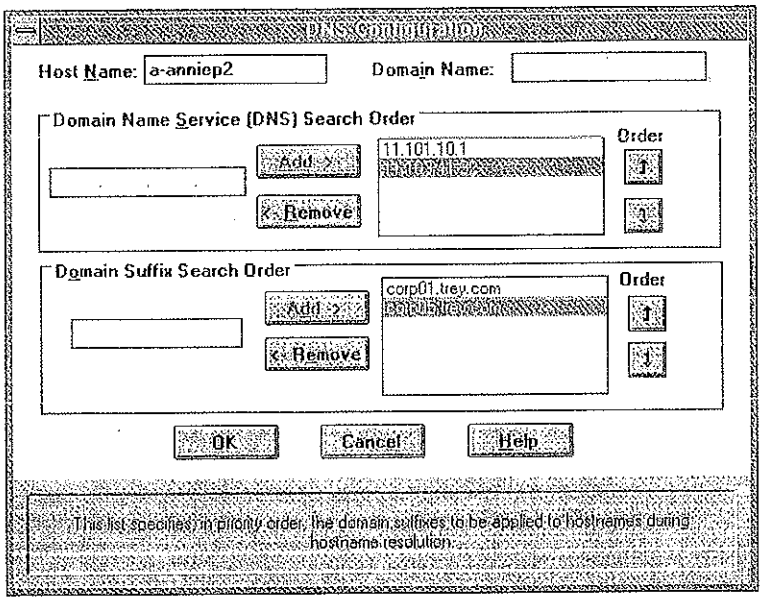
Contact the network administrator to find out whether you should configure your computer to use DNS. Usually, you can use DNS if you are using TCP/IP to communicate over the Internet or if your private internetwork uses DNS to distribute host information. For information, see "Domain Name System Addressing" in Chapter 12, "Networking Concepts for TCP/IP."

Microsoft TCP/IP includes DNS client software for resolving Internet or UNIX system names. Microsoft Windows networking provides dynamic name resolution for NetBIOS computer names via WINS servers and NetBIOS over TCP/IP.

DNS configuration is global for all network adapters installed on a computer.

To configure TCP/IP DNS connectivity

1. Double-click the Network option in Control Panel to display the Network Settings dialog box.
2. In the Installed Network Software box, select TCP/IP Protocol, and then choose the Configure button to display the TCP/IP Configuration dialog box.
3. Choose the DNS button to display the DNS Configuration dialog box.



Names are displayed in the Host Name box and Domain Name box.

4. Complete one or both of the following optional tasks:
   - Type a new name in the Host Name box (usually your computer name).

     The host name can be any combination of A–Z letters, 0–9 numerals, and the hyphen (-) character.

     Note Some characters that can be used in Windows NT computer names, particularly the underscore, cannot be used in host names.

     By default, this value is the Windows NT computer name, but the network administrator can assign another host name without affecting the computer name. The host name is used to identify the local computer by name for authentication by some utilities. Other TCP/IP-based utilities, such as **rexec**, can use this value to learn the name of the local computer. Host names are stored on DNS servers in a table that maps names to IP addresses for use by DNS.

- Type a new name in the Domain Name box.

  The DNS Domain Name can be any combination of A–Z letters, 0–9 numerals, and the hyphen (-) plus the period (.) character used as a separator.

  The DNS Domain Name is usually an organization name followed by a period and an extension that indicates the type of organization, such as microsoft.com. The DNS Domain Name is used with the host name to create a fully qualified domain name (FQDN) for the computer. The FQDN is the host name followed by a period (.) followed by the domain name. For example, this could be **corp01.research.trey.com**, where **corp01** is the host name and **research.trey.com** is the domain name. During DNS queries, the local domain name is appended to short names.

  Note A DNS domain is not the same as a Windows NT or LAN Manager domain.

5. In the Domain Name System (DNS) Search Order box, type the IP address of the DNS server that will provide name resolution, and then choose the Add button to move the IP address to the list on the right.

   The network administrator should provide the correct values for this parameter.

   You can add up to three IP addresses for DNS servers. The servers running DNS will be queried in the order listed. To change the order of the IP addresses, select an IP address to move, and then use the up- and down-arrow buttons. To remove an IP address, select the IP address, and then choose the Remove button.

6. In the Domain Suffix Search Order box, type the domain suffixes to add to your domain suffix search list, and then choose the Add button.

   This list specifies the DNS domain suffixes to be appended to host names during name resolution. You can add up to six domain suffixes. To change the search order of the domain suffixes, select a domain name to move, and then use the up- and down-arrow buttons. To remove a domain name, select the domain name, and then choose the Remove button.

7. When you are done setting DNS options, choose the OK button.

8. When the TCP/IP Configuration dialog box reappears, choose the OK button. When the Network Settings dialog box reappears, choose the OK button.

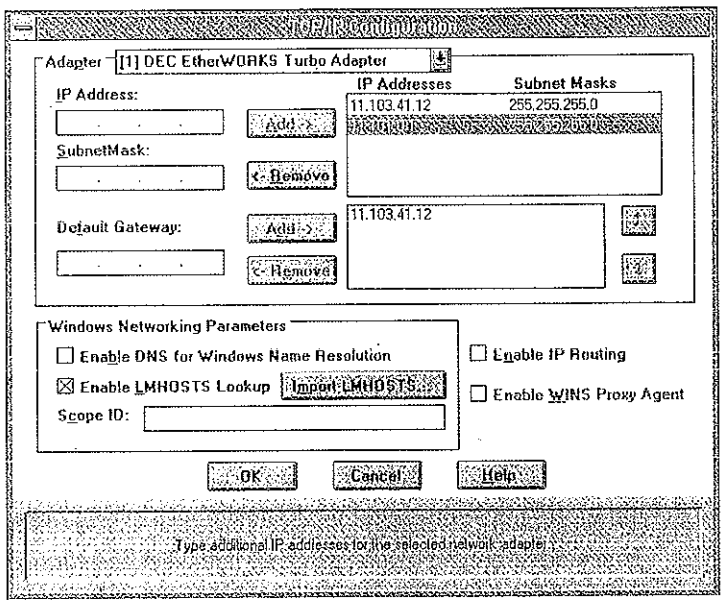
   The settings take effect after you restart the computer.

Configuring Advanced TCP/IP Options

If your computer has multiple network adapters connected to different networks using TCP/IP, you can choose the Advanced button in the TCP/IP Configuration dialog box to configure options for the adapters or to configure alternate default gateways.

To configure or reconfigure advanced TCP/IP options

1. Double-click the Network option in Control Panel to display the Network Settings dialog box.
2. In the Installed Network Software box, select TCP/IP Protocol, and then choose the Configure button to display the TCP/IP Configuration dialog box.



3. Choose the Advanced button to display the Advanced Microsoft TCP/IP Configuration dialog box.
4. In the Adapter box, select the network adapter for which you want to specify advanced configuration values.

   The IP address and default gateway settings in this dialog box are defined only for the selected network adapter.

5. In the IP Address and SubnetMask boxes, type an additional IP address and subnet mask for the selected adapter, and then choose the Add button to move the IP address to the list on the right.

   The network administrator should provide the correct values for this parameter.

   Optionally, if your network card uses multiple IP addresses, repeat this process for each additional IP address. You can specify up to five additional IP addresses and subnet masks for identifying the selected network adapter. This can be useful for a computer connected to one physical network that contains multiple logical IP networks.

6. In the Default Gateway box, type the IP address for an additional gateway that the selected adapter can use, and then choose the Add button to move the IP address to the list on the right.

   Repeat this process for each additional gateway. The network administrator should provide the correct values for this parameter.

   This list specifies up to five additional default gateways for the selected network adapter.

   To change the priority order for the gateways, select an address to move and use the up- or down-arrow buttons. To remove a gateway, select it, and then choose the Remove button.

7. If you want to use DNS for DNS name resolution on Windows networks, select the Enable DNS For Windows Name Resolution option.

   If this option is selected, the system finds the DNS server by using the IP address specified in the DNS Configuration dialog box, as described earlier in this chapter. Selecting this option enables DNS name resolution for use by Windows networking applications.

8. If you want to use the LMHOSTS file for NetBIOS name resolution on Windows networks, select the Enable LMHOSTS Lookup option.

   If you already have a configured LMHOSTS file, choose the Import LMHOSTS button and specify the directory path for the LMHOSTS file you want to use. By default, Windows NT uses the LMHOSTS file found in *\systemroot*\SYSTEM32\DRIVERS\ETC.

   For any method of name resolution used in a Windows NT network, the LMHOSTS file is consulted last after querying WINS or using broadcasts, but before DNS is consulted.

9. In the Scope ID box, type the computer's scope identifier, if required on an internetwork that uses NetBIOS over TCP/IP.

   To communicate with each other, all computers on a TCP/IP internetwork must have the same scope ID. Usually, this value is left blank. A scope ID might be assigned to a group of computers that will communicate only with each other and no other systems. Such computers can find each other if their scope IDs are identical. Scope IDs are used only for communication based on NetBIOS over TCP/IP.

   A computer can have only one scope ID, even if it has more than one adapter card with access to more than one network. If such a multihomed computer is a DHCP client, with DHCP servers on each network, the scope ID of the two networks should be identical. If they are not identical, the last adapter card to be configured will write its scope ID to the Registry, which could result in unexpected behavior and a loss of connectivity to one of the networks. It is best in this case to set the scope ID manually. Any manually configured value overrides values provided by the DHCP server.

   The network administrator should provide the correct value, if required.

10. To turn on static IP routing, check the Enable IP Routing option.

    This option allows this computer to participate with other static routers on a network. You should check this option if you have two or more network cards and your network uses static routing, which also requires the addition of static routing tables. For information about creating static routing tables, see the **route** utility in Appendix A, "TCP/IP Utilities Reference."

    This option is not available if your computer has only one network adapter and one IP address. Also, this option does not support routers running the Routing Information Protocol (RIP).

11. If you want this computer to be used to resolve names based on the WINS database, select the Enable WINS Proxy Agent option.

    This option allows the computer to answer name queries for remote computers, so other computers configured for broadcast name resolution can benefit from the name resolution services provided by a WINS server.

    This option is available only if you entered a value for a primary WINS server in the TCP/IP Configuration dialog box, as described in "Configuring TCP/IP" earlier in this chapter. However, the proxy agent cannot be run on a computer that is also a WINS server.

    Consult with the network administrator to determine whether your computer should be configured as a WINS proxy agent, as only a few computers on each subnetwork should be configured for this feature.

12. When you are done setting advanced options, choose the OK button. When the TCP/IP Configuration dialog box reappears, choose the OK button. When the Network Settings dialog box reappears, choose the OK button to complete advanced TCP/IP configuration.

You must restart the computer for the changes to take effect.

Configuring SNMP

The SNMP service is installed when you select the SNMP Service option in the Windows NT TCP/IP Installation Options dialog box. After the SNMP service software is installed on your computer, you must configure it with valid information for SNMP to operate.

You must be logged on as a member of the Administrators group for the local computer to configure SNMP.

The SNMP configuration information identifies communities and trap destinations.

- A *community* is a group of hosts to which a Windows NT computer running the SNMP service belongs. You can specify one or more communities to which the Windows NT computer using SNMP will send traps. The community name is placed in the SNMP packet when the trap is sent.

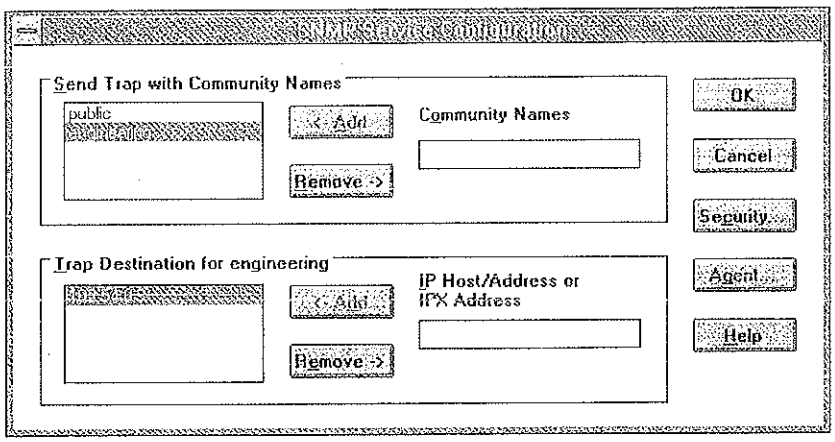
  When the SNMP service receives a request for information that does not contain the correct community name and does not match an accepted host name for the service, the SNMP service can send a trap to the trap destination(s), indicating that the request failed authentication.
- *Trap destinations* are the names or IP addresses of hosts to which you want the SNMP service to send traps with the selected community name.

You might want to use SNMP for statistics, but might not care about identifying communities or traps. In this case, you can specify the "public" community name when you configure the SNMP service.

▶ **To configure the SNMP service**

1. Double-click the Network option in Control Panel to display the Network Settings dialog box.

2. In the Installed Network Software box, select SNMP Service, and then choose the Configure button to display the SNMP Service Configuration dialog box.



3. To identify each community to which you want this computer to send traps, type the name in the Community Names box. After typing each name, choose the Add button to move the name to the Send Traps With Community Names list on the left.

   Typically, all hosts belong to public, which is the standard name for the common community of all hosts. To delete an entry in the list, select it, and then choose the Remove button.

Note Community names are case sensitive.

---

4. To specify hosts for each community you send traps to, after you have added the community and while it is still highlighted, type the hosts in the IP Host/Address Or IPX Address box. Then choose the Add button to move the host name or IP address to the Trap Destination for the *selected community* list on the left.

   You can enter a host name, its IP address, or its IPX address.

   To delete an entry in the list, select it, and then choose the Remove button.

5. To enable additional security for the SNMP service, choose the Security button. Continue with the configuration procedure, as described in the next section, "Configuring SNMP Security."

6. To specify Agent information (comments about the user, location, and services), choose the Agent button, and then continue with the configuration procedure, as described in "Configuring SNMP Agent Information" later in this chapter.

7. When you have completed all procedures, choose the OK button. When the Network Settings dialog box reappears, choose the OK button.

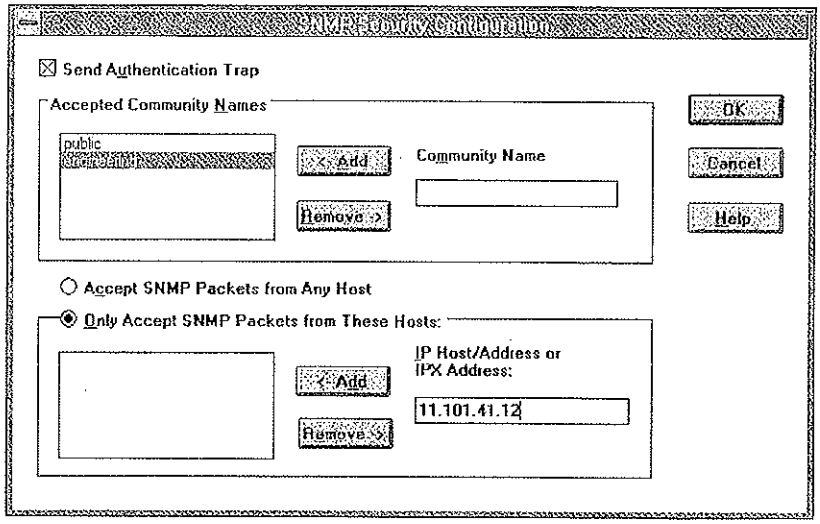
   The Microsoft SNMP service has been configured and is ready to start. It is not necessary to reboot the computer.

Configuring SNMP Security

SNMP security allows you to specify the communities and hosts a computer will accept requests from, and to specify whether to send an authentication trap when an unauthorized community or host requests information.

To configure SNMP security

1. Double-click the Network option in Control Panel to display the Network Settings dialog box.
2. In the Installed Network Software list box, select SNMP Service, and then choose the Configure button.
3. In the SNMP Service Configuration dialog box, choose the Security button.



4. If you want to send a trap for failed authentications, select the Send Authentication Trap check box in the SNMP Security Configuration dialog box.
5. In the Community Name box, type the community names in which you will accept requests. Choose the Add button after typing each name to move the name to the Accepted Community Names list on the left.

   A host must belong to a community that appears on this list for the SNMP service to accept requests from that host. Typically, all hosts belong to public, which is the standard name for the common community of all hosts. To delete an entry in the list, select it, and then choose the Remove button.

6. Select an option to specify whether to accept SNMP packets from any host or from only specified hosts.
   - If the Accept SNMP Packets From Any Host option is selected, no SNMP packets are rejected on the basis of source host ID. The list of hosts under Only Accept SNMP Packets From These Hosts has no effect.
   - If the Only Accept SNMP Packets From These Hosts option is selected, SNMP packets will be accepted only from the hosts listed. In the IP Host/Address Or IPX Address box, type the host names, IP addresses, or IPX addresses of the hosts from which you will accept requests. Then choose the Add button to move the host name or IP address to the list box on the left. To delete an entry in the list, select it, and then choose the Remove button.
7. Choose the OK button.

   The SNMP Service Configuration dialog box reappears.

   To specify Agent information (comments about the user, location, and services), choose the Agent button. Continue with the configuration procedure, as described in the next section.
8. After you complete all procedures, choose the OK button. When the Network Settings dialog box reappears, choose the OK button.

   The Microsoft SNMP service and SNMP security have been configured and are ready to start. You do not need to reboot the computer.

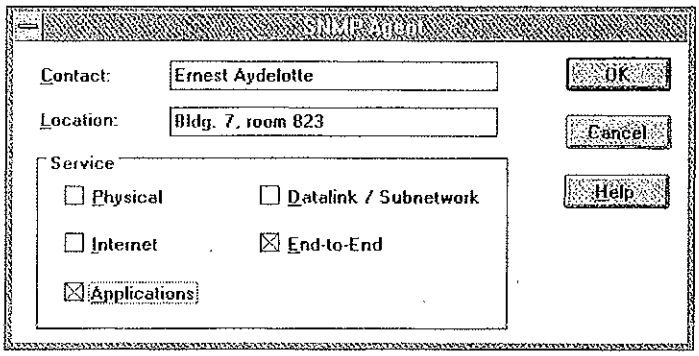
Configuring SNMP Agent Information

SNMP agent information allows you to specify comments about the user and the physical location of the computer and to indicate the types of service to report. The types of service that can be reported are based on the computer's configuration.

▶ **To configure SNMP agent information**

1. Double-click the Network option in Control Panel to display the Network Settings dialog box.
2. In the Installed Network Software list box, select SNMP Service, and then choose the Configure button to display the SNMP Service Configuration dialog box.

3. Choose the Agent button to display the SNMP Agent dialog box.



4. In the Contact box and Location box, type the computer user's name and the computer's physical location.

   These comments are used as text only. They cannot include embedded control characters.

5. In the Service group box, select all options that indicate network capabilities provided by your Windows NT computer.

   SNMP must have this information to manage the enabled services.

If you have installed additional TCP/IP services, such as a bridge or router, you should consult RFC 1213 for additional information.

Table 11.3 SNMP Service Options

Option	Description
Physical	Select this option if this Windows NT computer manages any physical TCP/IP device, such as a repeater.
Datalink/Subnetwork	Select this option if this Windows NT computer manages a TCP/IP subnetwork or datalink, such as a bridge.
Internet	Select this option if this Windows NT computer acts as an IP gateway.
End-to-End	Select this option if this Windows NT computer acts as an IP host. This option should be selected for all Windows NT installations.
Applications	Select this option if this Windows NT computer includes any applications that use TCP/IP, such as electronic mail. This option should be selected for all Windows NT installations.

6. Choose the OK button.
7. When the SNMP Service Configuration dialog box reappears, choose the OK button. When the Network Settings dialog box reappears, choose the OK button.

SNMP is now ready to operate. You do not need to restart the computer.

Removing TCP/IP Components

If you want to remove the TCP/IP protocol or any of the services installed on a computer, use the Network option in Control Panel to remove it.

When you remove network software, Windows NT warns you that the action permanently removes that component. You cannot reinstall a component that has been removed until after you restart the computer.

▸ **To remove any TCP/IP component**

1. Double-click the Network option in Control Panel to display the Network Settings dialog box.
2. In the Installed Network Software list, select the component that you want to remove.
3. Choose the Remove button to permanently remove the component.

Configuring RAS for Use with TCP/IP

Windows NT users who install Remote Access Service (RAS) for remote networking maintain all the benefits of TCP/IP networking, including access to the WINS and DNS capabilities of Microsoft TCP/IP. RAS clients can be configured to use Point to Point Protocol (PPP) or Serial Line Internet Protocol (SLIP) to allow TCP/IP dial-up support for existing TCP/IP internetworks and the Internet. When PPP is configured on a Windows NT Remote Access server, it can function as a router for RAS clients. SLIP client software is provided to support older implementations; it does not support multiple protocols.

As with all network services, you install RAS by using the Network option in Control Panel. During RAS installation and configuration, you can specify the network protocol settings to use for RAS connections, which also enables you to specify TCP/IP configuration settings. When the network administrator installs a Microsoft RAS server, IP addresses are reserved for use by RAS clients.

Users with RAS client computers can use the Remote Access program to enter and maintain names and telephone numbers of remote networks. RAS clients can connect to and disconnect from these networks through the Remote Access program. You can also use the Remote Access Phone Book application to select the network protocols to use for a specific Phone Book entry. If TCP/IP is installed, the Phone Book automatically selects TCP/IP over PPP as the protocol.

If a RAS client computer has a serial COM port, you can use the Remote Access Phone Book application to configure SLIP for use with a selected Phone Book entry. If you configure a RAS client computer to use the SLIP option, when you dial in for a connection to the selected Phone Book entry, the Terminal screen appears, and you can begin an interactive session with a SLIP server. When you use SLIP, Remote Access Phone Book bypasses user authentication. You will not be asked for a username and password.

For complete information about setting up RAS servers and clients and using RAS with Windows NT, see *Windows NT Server Remote Access Service*.

CHAPTER 12

Networking Concepts for TCP/IP

This chapter describes how TCP/IP fits in the Windows NT network architecture and explains the various components of the Internet Protocol suite and IP addressing. As part of the discussion on name resolution in Windows-based networking, this chapter also describes NetBIOS over TCP/IP (NBT) and Domain Name System (DNS). For additional information about these topics, see the books listed in the "Welcome" section of this manual.

This chapter also provides conceptual information about two key features for Microsoft TCP/IP: Dynamic Host Configuration Protocol (DHCP) and Windows Internet Name Service (WINS).

The following topics appear in this chapter:

- TCP/IP and Windows NT networking
- Internet protocol suite
- IP addressing
- Name resolution for Windows-based networking
- SNMP

TCP/IP and Windows NT Networking

The architecture of the Microsoft Windows NT operating system with integrated networking is protocol-independent. This architecture, illustrated in the following figure, provides Windows NT file, print, and other services over any network protocol that uses exports from the TDI interface. The protocols package network requests for applications in their respective formats, and then send the requests to the appropriate network adapter via the *network device interface specification* (NDIS) interface. The NDIS specification allows multiple network protocols to reside over a wide variety of network adapters and media types.

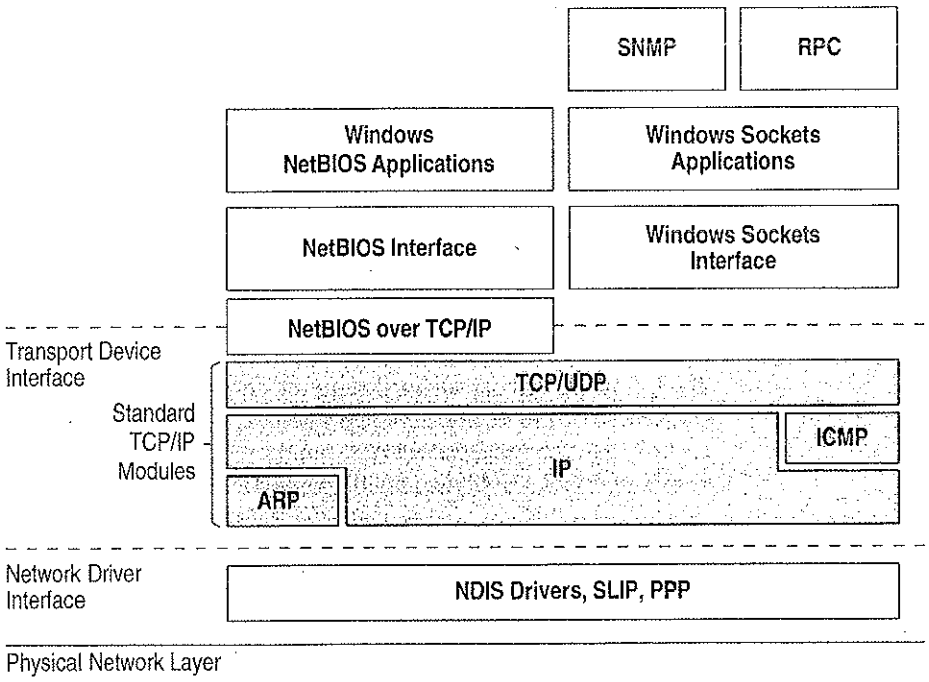


Figure 12.1 Architectural Model of Windows NT with TCP/IP

Under the Windows NT transport-independent architecture, TCP/IP is a protocol family that can be used to offer Windows-based networking capabilities. The TCP/IP protocol gives Windows NT, Windows for Workgroups, and LAN Manager computers transparent access to each other and allows communication with non-Microsoft systems in the enterprise network.

Internet Protocol Suite

TCP/IP refers to the Internet suite of protocols. It includes a set of standards that specify how computers communicate and gives conventions for connecting networks and routing traffic through the connections.

The Internet protocols are a result of a Defense Advanced Research Projects Agency (DARPA) research project on network interconnection in the late 1970s. It was mandated on all United States defense long-haul networks in 1983 but was not widely accepted until it was integrated with 4.2 Berkeley Software Distribution (BSD) UNIX. The popularity of TCP/IP is based on the following features:

- *Robust client-server framework.* TCP/IP is an excellent client-server application platform, especially in wide area network (WAN) environments.
- *Information sharing.* Thousands of academic, military, scientific, and commercial organizations share data, electronic mail, and services on the Internet using TCP/IP.
- *General availability.* Implementations of TCP/IP are available on nearly every popular computer operating system. Source code is widely available for many implementations. Vendors for bridges, routers, and network analyzers all offer support for the TCP/IP protocol suite within their products.

The following discussion introduces the components of the IP protocol suite. Some knowledge of the architecture and interaction between TCP/IP components is useful for both administrators and users, but most of the details discussed here are transparent when you are actually using TCP/IP.

Transmission Control Protocol and Internet Protocol

Transmission Control Protocol (TCP) and Internet Protocol (IP) are only two members of the IP protocol suite. IP is a protocol that provides packet delivery for all other protocols within the TCP/IP family. IP provides a best-effort, connectionless delivery system for computer data. That is, IP packets are not guaranteed to arrive at their destination, nor are they guaranteed to be received in the sequence in which they were sent. The protocol's checksum feature confirms only the IP header's integrity. Thus, responsibility for the data contained within the IP packet (and the sequencing) is assured only by using higher-level protocols.

Perhaps the most common higher-level IP protocol is TCP. TCP supplies a reliable, connection-based protocol over (or encapsulated within) IP. TCP guarantees the delivery of packets, ensures proper sequencing of the data, and provides a checksum feature that validates both the packet header and its data for accuracy. In the event that the network either corrupts or loses a TCP/IP packet during transmission, TCP is responsible for retransmitting the faulty packet. This reliability makes TCP/IP the protocol of choice for session-based data transmission, client-server applications, and critical services, such as electronic mail.

This reliability has a price. TCP headers require the use of additional bits to provide proper sequencing of information, as well as a mandatory checksum to ensure reliability of both the TCP header and the packet data. To guarantee successful data delivery, the protocol also requires the recipient to acknowledge successful receipt of data.

Such acknowledgments (or ACKs) generate additional network traffic, diminishing the level of data throughput in favor of reliability. To reduce the impact on performance, most hosts send an acknowledgment for every other segment or when an ACK timeout expires.

User Datagram Protocol

If reliability is not essential, User Datagram Protocol (UDP), a TCP complement, offers a connectionless datagram service that guarantees neither delivery nor correct sequencing of delivered packets (much like IP). Higher-level protocols or applications might provide reliability mechanisms in addition to UDP/IP. UDP data checksums are optional, providing a way to exchange data over highly reliable networks without unnecessarily consuming network resources or processing time. When UDP checksums are used, they validate the integrity of both the header and data. ACKs are also not enforced by the UDP protocol; this is left to higher-level protocols.

UDP also offers one-to-many service capabilities, because it can be either broadcast or multicast.

Address Resolution Protocol and Internet Control Message Protocol

Two other protocols in the IP suite perform important functions, although these are not directly related to the transport of data: Address Resolution Protocol (ARP) and Internet Control Message Protocol (ICMP). ARP and ICMP are maintenance protocols that support the IP framework and are usually invisible to users and applications.

IP packets contain both source and destination IP addresses, but the hardware address of the destination computer system must also be known. IP acquires a system's hardware address by broadcasting a special inquiry packet (an ARP *request packet*) containing the IP address of the system with which it is attempting to communicate. All of the ARP-enabled nodes on the local IP network detect these broadcasts, and the system that owns the IP address in question replies by sending its hardware address to the requesting computer system in an ARP reply packet. The hardware/IP address mapping is then stored in the requesting system's ARP cache for subsequent use. Because the ARP reply can also be broadcast to the network, it is likely that other nodes on the network can use this information to update their own ARP caches. (You can use the **arp** utility to view the ARP tables.)

ICMP allows two nodes on an IP network to share IP status and error information. This information can be used by higher-level protocols to recover from transmission problems or by network administrators to detect network trouble. Although ICMP packets are encapsulated within IP packets, they are not considered to be a higher-level protocol (ICMP is required in every TCP/IP implementation). The **ping** utility makes use of the ICMP *echo request* and *echo reply* packets to determine whether a particular IP node (computer system) on a network is functional. For this reason, the **ping** utility is useful for diagnosing IP network or gateway failures.

IP Addressing

A *host* is any device attached to the network that uses TCP/IP. To receive and deliver packets successfully between hosts, TCP/IP relies on three values, that the user provides: IP address, subnet mask, and default gateway.

The network administrator provides each of these values for configuring TCP/IP on a computer. Windows NT users on networks with DHCP servers can take advantage of automatic system configuration and do not need to manually configure TCP/IP parameters. This section provides details about IP addresses, subnet masks, and IP gateways.

IP Addresses

Every host interface, or node, on a TCP/IP network is identified by a unique IP address. This address is used to identify a host on a network; it also specifies routing information in an internetwork. The *IP address* identifies a computer as a 32-bit address that is unique across a TCP/IP network. An address is usually represented in dotted-decimal notation, which depicts each octet (eight bits, or one byte) of an IP address as its decimal value and separates each octet with a period. An IP address looks like this:

```
102.54.94.97
```

Important Because IP addresses identify nodes on an interconnected network, each host on the internetwork must be assigned a unique IP address, valid for its particular network.

Network ID and Host ID

Although an IP address is a single value, it contains two pieces of information: the network ID and the host (or system) ID for your computer.

- The *network ID* identifies a group of computers and other devices that are all located on the same logical network, which are separated or interconnected by routers. In internetworks (networks formed by a collection of local area networks), there is a unique network ID for each network.
- The *host ID* identifies your computer within a particular network ID. (A host is any device that is attached to the network and uses TCP/IP.)

Networks that connect to the public Internet must obtain an official network ID from the InterNIC to guarantee IP network ID uniqueness. The InterNIC can be contacted via electronic mail at info@internic.net (for the United States, 1–800–444–4345 or, for Canada and overseas, 619–455–4600). Internet registration requests can be sent to hostmaster@internic.net. You can also use FTP to connect to is.internic.net, then log in as **anonymous**, and then change to the /INFOSOURCE/FAQ directory.

After receiving a network ID, the local network administrator must assign unique host IDs for computers within the local network. Although private networks not connected to the Internet can choose to use their own network identifier, obtaining a valid network ID from InterNIC allows a private network to connect to the Internet in the future without reassigning addresses.

The Internet community has defined address *classes* to accommodate networks of varying sizes. Each network class can be discerned from the first octet of its IP address. The following table summarizes the relationship between the first octet of a given address and its network ID and host ID fields. It also identifies the total number of network IDs and host IDs for each address class that participates in the Internet addressing scheme. This sample uses w.x.y.z to designate the bytes of the IP address.

Table 12.1 IP Address Classes

Class	w values[1,2]	Network ID	Host ID	Available networks	Available hosts per net
A	1–126	w	x.y.z	126	16,777,214
B	128–191	w.x	y.z	16,384	65,534
C	192–223	w.x.y	z	2,097,151	254

1 Inclusive range for the first octet in the IP address.

2 The address 127 is reserved for loopback testing and interprocess communication on the local computer; it is not a valid network address. Addresses 224 and above are reserved for special protocols (IGMP multicast and others), and cannot be used as host addresses.

A network host uses the network ID and host ID to determine which packets it should receive or ignore and to determine the scope of its transmissions (only nodes with the same network ID accept each other's IP-level broadcasts).

Because the sender's IP address is included in every outgoing IP packet, it is useful for the receiving computer system to derive the originating network ID and host ID from the IP address field. This task is done by using subnet masks, as described in the following section.

Subnet Masks

Subnet masks are 32-bit values that allow the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID. Like an IP address, the value of a subnet mask is frequently represented in dotted-decimal notation. Subnet masks are determined by assigning 1's to bits that belong to the network ID and 0's to the bits that belong to the host ID. Once the bits are in place, the 32-bit value is converted to dotted-decimal notation, as shown in the following table.

Table 12.2 Default Subnet Masks for Standard IP Address Classes

Address class	Bits for subnet mask	Subnet mask
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

The result enables TCP/IP to determine the host and network IDs of the local computer. For example, when the IP address is 102.54.94.97 and the subnet mask is 255.255.0.0, the network ID is 102.54 and the host ID is 94.97.

Although configuring a host with a subnet mask might seem redundant after examining the previous tables (since the class of a host is easily determined), subnet masks are also used to further segment an assigned network ID among several local networks.

For example, suppose a network is assigned the Class-B network address 144.100. This is one of over 16,000 Class-B addresses capable of serving more than 65,000 nodes. However, the worldwide corporate network to which this ID is assigned is composed of 12 international LANs with 75 to 100 nodes each. Instead of applying for 11 more network IDs, it is better to use subnetting to make more effective use of the assigned ID 144.100. The third octet of the IP address can be used as a subnet ID, to define the subnet mask 255.255.255.0. This arrangement splits the Class-B address into 254 subnets: 144.100.1 through 144.100.254, each of which can have 254 nodes. (Host IDs 0 and 255 should not be assigned to a computer; they are used as broadcast addresses, which are typically recognized by all computers.) Any 12 of these network addresses could be assigned to the international LANs in this example. Within each LAN, each computer is assigned a unique host ID, and they all have the subnet mask 255.255.255.0.

The preceding example demonstrates a simple (and common) subnet scheme for Class-B addresses. Sometimes it is necessary to segment only portions of an octet, using only a few bits to specify subnet IDs (such as when subnets exceed 256 nodes). Each user should check with the local network administrator to determine the network's subnet policy and the correct subnet mask. For all systems on the local network, the subnet mask must be the same for that network ID.

Important All computers on a logical network must use the same subnet mask and network ID; otherwise, addressing and routing problems can occur.

Routing and IP Gateways

TCP/IP networks are connected by *gateways* (or routers), which have knowledge of the networks connected in the internetwork. Although each IP host can maintain static routes for specific destinations, usually the default gateway is used to find remote destinations. (The *default gateway* is needed only for computers that are part of an internetwork.)

When IP prepares to send a packet, it inserts the local (source) IP address and the destination address of the packet in the IP header and checks whether the network ID of the destination matches the network ID of the source. If they match, the packet is sent directly to the destination computer on the local network. If the network IDs do not match, the routing table is examined for static routes. If none are found, the packet is forwarded to the default gateway for delivery.

The default gateway is a computer connected to the local subnet and other networks that has knowledge of the network IDs for other networks in the internetwork and how to reach them. Because the default gateway knows the network IDs of the other networks in the internetwork, it can forward the packet to other gateways until the packet is eventually delivered to a gateway connected to the specified destination. This process is known as *routing*.

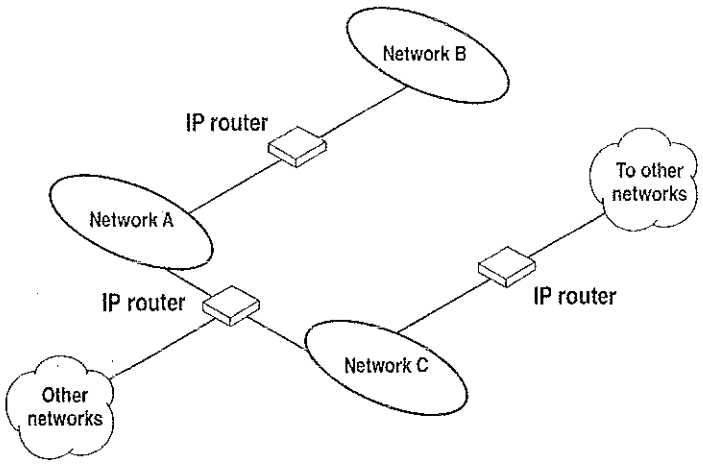


Figure 12.2 Internetwork Routing Through Gateways

On networks that are not part of an internetwork, IP gateways are not required. If a network is part of an internetwork and a system does not specify a default gateway (or if the gateway computer is not operating properly), only communication beyond the local subnet is impaired. Users can add static routes by using the **route** utility to specify a route for a particular system. Static routes always override the use of default gateways.

If the default gateway becomes unavailable, the computer cannot communicate outside its own subnet. Multiple default gateways can be assigned to prevent such a problem. When a computer is configured with multiple default gateways, retransmission problems result in the system trying the other routers in the configuration to ensure internetworking communications capabilities. To configure multiple default gateways in Windows NT, you must provide an IP address for each gateway in the Advanced Microsoft TCP/IP Configuration dialog box, as described in Chapter 11, "Installing and Configuring Microsoft TCP/IP and SNMP."

Dynamic Host Configuration Protocol

Assigning and maintaining IP address information can be an administrative burden for network administrators responsible for internetwork connections. Contributing to this burden is the problem that many users do not have the knowledge necessary to configure their own computers for internetworking and must therefore rely on their administrators.

The Dynamic Host Configuration Protocol (DHCP) was established to relieve this administrative burden. DHCP provides safe, reliable, and simple TCP/IP network configuration, ensures that address conflicts do not occur, and helps conserve the use of IP addresses through centralized management of address allocation. DHCP offers dynamic configuration of IP addresses for computers. The system administrator controls how IP addresses are assigned by specifying *lease* durations, which specify how long a computer can use an assigned IP address before having to renew the lease with the DHCP server.

As an example of how maintenance tasks are made easy with DHCP, the IP address is released automatically for a DHCP client computer that is removed from a subnet, and a new address for the new subnet is automatically assigned when that computer reconnects on another subnet. Neither the user nor the network administrator needs to intervene to supply new configuration information. This is a most significant feature for mobile computer users with portables that are docked at different computers, or for computers that are moved to different offices frequently.

The DHCP client and server services for Windows NT are implemented under Requests for Comments (RFCs) 1533, 1534, 1541, and 1542.

The following illustration shows an example of a DHCP server providing configuration information on two subnets. If, for example, ClientC is moved to Subnet 1, the DHCP server automatically supplies new TCP/IP configuration information the next time that ClientC is started.

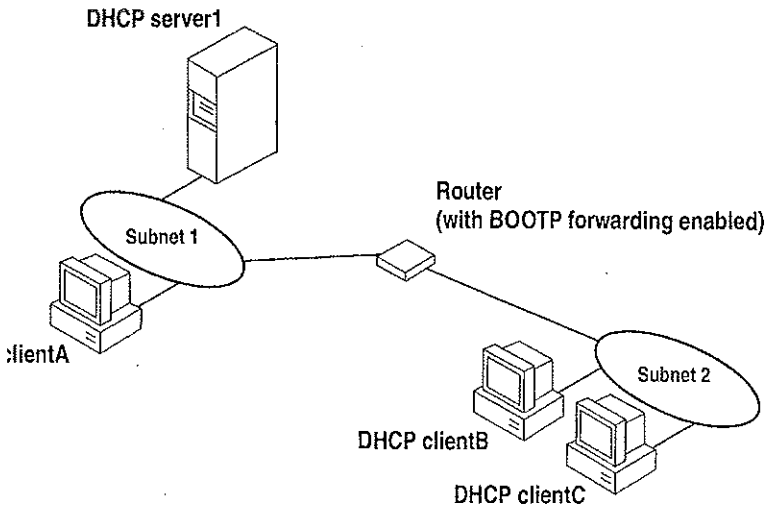


Figure 12.3 DHCP Clients and Servers on a Routed Network

DHCP uses a client-server model and is based on leases for IP addresses. During system startup (the *initializing* state), a DHCP client computer sends a *discover message* that is broadcast to the local network and might be relayed to all DHCP servers on the private internetwork. Each DHCP server that receives the discover message responds with an *offer message* containing an IP address and valid configuration information for the client that sent the request.

The DHCP client collects the configuration offerings from the servers and enters a *selecting* state. When the client enters the *requesting* state, it chooses one of the configurations and sends a *request message* that identifies the DHCP server for the selected configuration.

The selected DHCP server sends a *DHCP acknowledgment message* that contains the address first sent during the discovery stage, plus a valid lease for the address and the TCP/IP network configuration parameters for the client. After the client receives the acknowledgment, it enters a *bound* state and can now participate on the TCP/IP network and complete its system startup. Client computers that have local storage save the received address for use during subsequent system startup. As the lease approaches its expiration date, it attempts to renew its lease with the DHCP server, and is assigned a new address if the current IP address lease cannot be renewed.

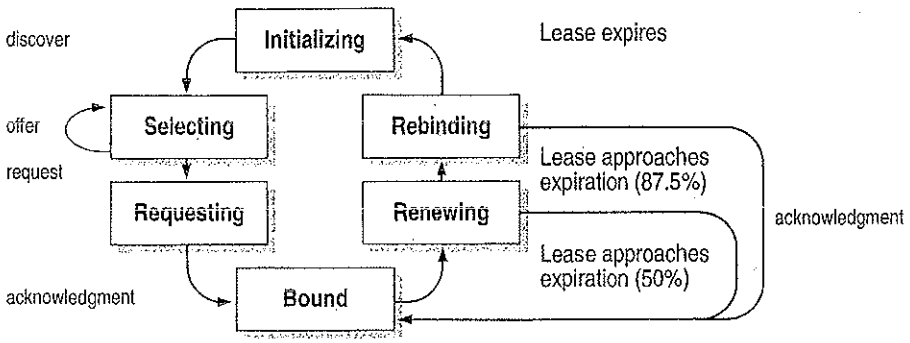


Figure 12.4 DHCP Client State Transition During System Startup

In Windows NT Server, the network administrator uses DHCP Manager to define local policies for address allocation, leases, and other options. For information about using this tool, see Chapter 13, "Installing and Configuring DHCP Servers." For information about the steps for setting up TCP/IP using DHCP, see "Configuring TCP/IP" in Chapter 11, "Installing and Configuring Microsoft TCP/IP and SNMP." For information about setting up DHCP relaying, see the documentation for your router.

Name Resolution for Windows-Based Networking

Configuring Windows NT with TCP/IP requires the IP address and computer name, which are unique identifiers for the computer on the network. The IP address, as described earlier in this chapter, is the unique address by which all other TCP/IP devices on the internetwork recognize that computer. For TCP/IP and the Internet, the computer name is the globally known system name plus a DNS domain name. (On the local network, the computer name is the NetBIOS name that was defined during Windows NT Setup.)

Computers use IP addresses to identify each other, but users usually find it easier to work with computer names. A mechanism must be available on a TCP/IP network to resolve computer names to IP addresses. To ensure that both computer name and address are unique, the Windows NT computer using TCP/IP registers its name and IP address on the network during system startup. A Windows NT computer can use one or more of the following methods to ensure accurate name resolution in TCP/IP internetworks:

- Windows Internet Name Service

  Windows NT computers can use WINS if one or more WINS servers are available that contain a dynamic database mapping computer names to IP addresses. WINS can be used in conjunction with broadcast name resolution for an internetwork where other name resolution methods are inadequate. As described in the following section, WINS is a NetBIOS over TCP/IP (NBT) mode of operation defined in RFC 1001/1002 as p-node.

- Broadcast name resolution

  Windows NT computers can also use broadcast name resolution, which is a NetBIOS over TCP/IP mode of operation defined in RFC 1001/1002 as b-node. This method relies on a computer making IP-level broadcasts to register its name by announcing it on the network. Each computer in the broadcast area is responsible for challenging attempts to register a duplicate name and for responding to name queries for its registered name.

- DNS name resolution

  The Domain Name System (DNS) provides a way to look up name mappings when connecting a computer to foreign hosts using NetBIOS over TCP/IP or Windows Sockets applications, such as FTP. DNS is a distributed database designed to relieve the traffic problems that arose with the exploding growth of the Internet in the early 1980s.

- An LMHOSTS file to specify the NetBIOS computer name and IP address mappings, or a HOSTS file to specify the DNS name and IP address

  On a local computer, the HOSTS file (used by Windows Sockets applications to find TCP/IP host names) and LMHOSTS file (used by NetBIOS over TCP/IP to find Microsoft networking computer names) can be used to list known IP addresses mapped with corresponding computer names. The LMHOSTS file is still used for name resolution in Windows NT for small-scale networks or remote subnets where WINS is not available.

This section provides details about name resolution in Windows NT after first presenting some background information about the modes of NetBIOS over TCP/IP that can be used in Microsoft networks.

NetBIOS over TCP/IP and Name Resolution

NetBIOS over TCP/IP (NBT) is the session-layer network service that performs name-to-IP address mapping for name resolution. This section describes the modes of NBT, as defined in RFCs 1001 and 1002 to specify how NetBIOS should be implemented over TCP/IP.

The modes of NBT define how network resources are identified and accessed. The two most important aspects of the related naming activities are registration and resolution. *Registration* is the process used to acquire a unique name for each node (computer system) on the network. A computer typically registers itself when it starts. *Resolution* is the process used to determine the specific address for a computer name.

The NBT modes include the following:

- *b-node*, which uses broadcasts to resolve names
- *p-node*, which uses point-to-point communications with a name server to resolve names
- *m-node*, which uses b-node first (broadcasts), and then p-node (name queries) if the broadcast fails to resolve a name
- *h-node*, which uses p-node first for name queries, and then b-node if the name service is unavailable or if the name is not registered in the WINS database

If WINS servers are specified by either a DHCP server or the TCP/IP configuration specified in the Network option of Control Panel, Windows NT 3.5 defaults to h-node. Otherwise, the default node type is b-node, unless another node type has been set as an option by the DHCP server.

For DHCP users on a Windows NT network, the node type is assigned by the DHCP server. A DHCP client computer can have only one NetBIOS node type, no matter how many adapter cards it has. On a multihomed computer with access to more than one network, the node type must be the same on both networks. When WINS servers are in place on the network, NBT resolves names on a client computer by communicating with the WINS server. If you want to configure a multihomed computer with some network adapter cards using b-node and some using h-node, configure WINS server addresses for the adapter cards that are to run in h-mode. The presence of a WINS address on an adapter card effectively overrides the b-node setting.

When WINS servers are not in place, NBT uses b-node broadcasts to resolve names. NBT in Windows NT can also use LMHOSTS files and DNS for name resolution, depending on how TCP/IP is configured on a particular computer. In Windows NT 3.5, the NETBT.SYS module provides the NBT functionality that supports name registration and resolution modes.

Windows NT version 3.5 supports all of the NBT modes described in the following sections. NBT is also used with the LAN Manager 2.*x* Server message protocol.

B-Node

The b-node mode uses broadcasts for name registration and resolution. That is, if NT_PC1 wants to communicate with NT_PC2, it broadcasts to all machines that it is looking for NT_PC2, and then it waits a specified time for NT_PC2 to respond. B-node has two major problems:

- In a large environment, it loads the network with broadcasts.
- Routers do not forward broadcasts, so computers that are on opposite sides of a router never hear the requests.

P-Node

The p-node mode addresses the issues that b-node does not solve. In a p-node environment, computers neither create nor respond to broadcasts. All computers register themselves with the WINS server, which is a NetBIOS Name Server (NBNS) with enhancements. The WINS server is responsible for knowing computer names and addresses and for ensuring no duplicate names exist on the network. All computers must be configured to know the address of the WINS server.

In this environment, when NT_PC1 wants to communicate with NT_PC2, it queries the WINS server for the address of NT_PC2. When NT_PC1 gets the appropriate address from the WINS server, it goes directly to NT_PC2 without broadcasting. Because the name queries go directly to the WINS server, p-node avoids loading the network with broadcasts. Because broadcasts are not used and because the address is received directly, computers can span routers.

The most significant problems with p-node are the following:

- All computers must be configured to know the address of the WINS server (although this is typically configured via DHCP)
- If for any reason the WINS server is down, computers that rely on the WINS server to resolve addresses cannot get to any other systems on the network, even if they are on the local network

M-Node

The m-node mode was created primarily to solve the problems associated with b-node and p-node. This mode uses a combination of b-node and p-node. In an m-node environment, a computer first attempts registration and resolution using b-node. If that is successful, it then switches to the p-node. Because this uses b-node first, it does not solve the problem of generating broadcast traffic on the network. However, m-node can cross routers. Also, because b-node is always tried first, computers on the same side of a router continue to operate as usual if the WINS server is down.

M-node uses broadcasts for performance optimization, because in most environments local resources are used more frequently than remote resources. Also, in a Windows NT network, m-node can cause problems with NetLogon in routed environments.

H-Node

The h-node mode, which is currently in RFC draft form, is also a combination of b-node and p-node that uses broadcasts as a last effort. Because p-node is used first, no broadcasts are generated if the WINS server is running, and computers can span routers. If the WINS server is down, b-node is used, so computers on the same side of a router continue to operate as usual.

The h-node mode does more than change the order for using b-node and p-node. If the WINS server is down so that local broadcasts (b-node) must be used, the computer continues to poll the WINS server. As soon as the WINS server can be reached again, the system switches back to p-node. Also, optionally on a Windows network, h-node can be configured to use the LMHOSTS file after broadcast name resolution fails.

The h-node mode solves the most significant problems associated with broadcasts and operating in a routed environment. For Microsoft TCP/IP users who configure TCP/IP manually, h-node is used by default, unless the user does not specify addresses for WINS servers when configuring TCP/IP.

B-Node with LMHOSTS and Combinations

Another variation is also used in Microsoft networks to span routers without a WINS server and p-node mode. In this mode, b-node uses a list of computers and addresses stored in an LMHOSTS file. If a b-node attempt fails, the system looks in LMHOSTS to find a name and then uses the associated address to cross the router. However, each computer must have this list, which creates an administrative burden in maintaining and distributing the list. Both Windows for Workgroups 3.11 and LAN Manager 2.*x* used such a modified b-node system. Windows NT uses this method if WINS servers are not used on the network. In Windows NT, some extensions have been added to this file to make it easier to manage (as described in Chapter 15, "Setting Up LMHOSTS"), but modified b-node is not an ideal solution.

Some sites might need to use both b-node and p-node modes at the same site. Although this configuration can work, administrators must exercise extreme caution in doing so, using it only for transition situations. Because p-node hosts disregard broadcasts and b-node hosts rely on broadcasts for name resolution, the two hosts can potentially be configured with the same NetBIOS name, leading to unpredictable results. Notice that if a computer configured to use b-node has a static mapping in the WINS database, a computer configured to use p-node cannot use the same computer name.

Windows NT computers can also be configured as WINS proxy agents to help the transition to using WINS. For more details, see the next section.

Windows Internet Name Service and Broadcast Name Resolution

WINS provides a distributed database for registering and querying dynamic computer name-to-IP address mappings in a routed network environment. If you are administering a routed network, WINS is your best first choice for name resolution, because it is designed to solve the problems that occur with name resolution in complex internetworks.

WINS reduces the use of local broadcasts for name resolution and allows users to easily locate systems on remote networks. Furthermore, when dynamic addressing through DHCP results in new IP addresses for computers that move between subnets, the changes are automatically updated in the WINS database. Neither the user nor the network administrator needs to make manual accommodations for name resolution in such a case.

The WINS protocol is based on and is compatible with the protocols defined for NBNS in RFCs 1001/1002, so it is interoperable with any other implementations of these RFCs.

This section provides an overview of how WINS and name query broadcasts provide name resolution on Windows networks. For information about setting up WINS servers, see Chapter 14, "Installing and Configuring WINS Servers."

WINS in a Routed Environment

WINS consists of two components: the WINS server, which handles name queries and registrations, and the client software, which queries for computer name resolution.

Windows-based networking clients (WINS-enabled Windows NT or Windows for Workgroups 3.11 computers) can use WINS directly. Non-WINS computers on the internetwork that are b-node compatible as described in RFCs 1001 and 1002 can access WINS through proxies, which are WINS-enabled computers that listen to name query broadcasts and then respond for names that are not on the local subnet or are p-node computers.

On a Windows NT network, users can browse transparently across routers. To allow browsing without WINS, the network administrator must ensure that the users' primary domain has Windows NT Server or Windows NT Workstation computers on both sides of the router to act as master browsers. These computers need correctly configured LMHOSTS files with entries for the domain controllers across the subnet.

With WINS, such strategies are not necessary because the WINS servers and proxies transparently provide the support necessary for browsing across routers where domains span the routers.

The following figure shows a small internetwork, with three local area networks connected by a router. Two of the subnets include WINS name servers, which can be used by clients on both subnets. WINS-enabled computers, including proxies, access the WINS server directly, and the computers using broadcasts access the WINS server through proxies. Proxies only pass name query packets and verify that registrations do not duplicate existing systems in the WINS database. Proxies, however, do not register b-node systems in the WINS database.

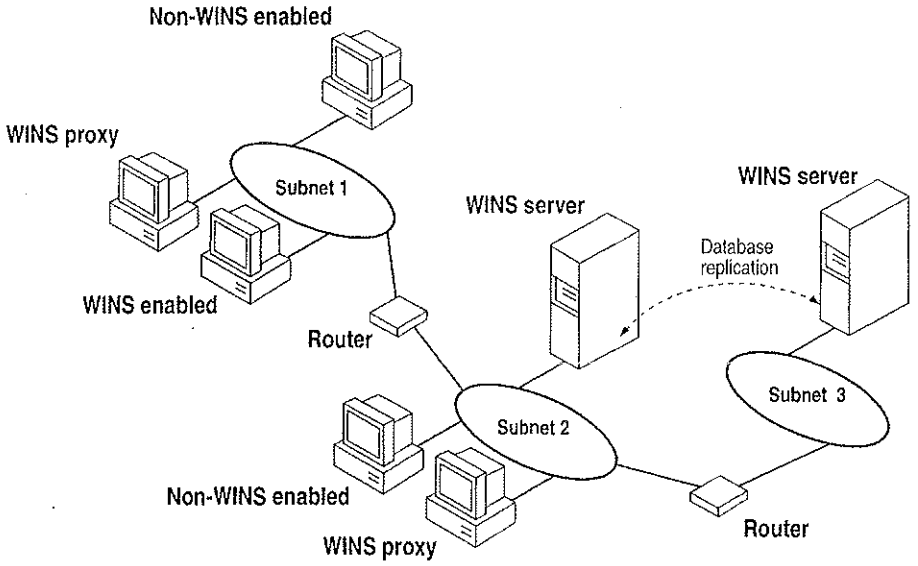


Figure 12.5 Example of an Internetwork with WINS Servers

The proxy communicates with the WINS server to resolve names (rather than maintaining its own database) and then caches the names for a certain time. The proxy serves as an intermediary, by either communicating with the WINS server or supplying a name-to-IP address mapping from its cache. The following illustration shows the relationships among WINS servers and clients, including proxies for non-WINS computers and the replication between WINS servers.

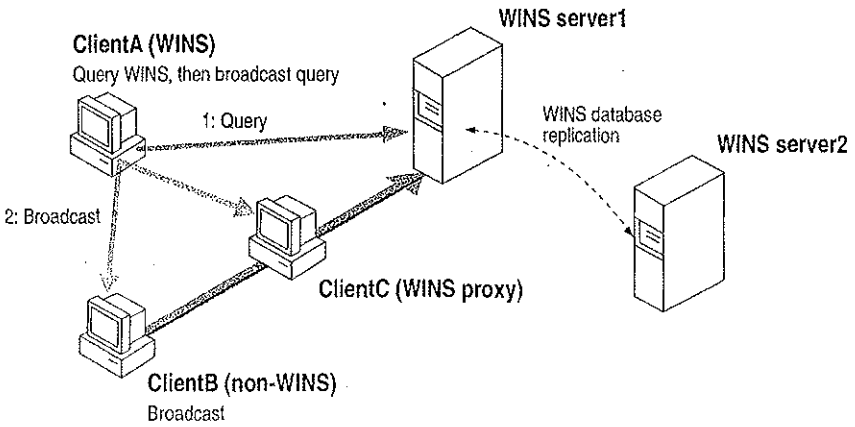


Figure 12.6 Example of Clients and Servers Using WINS