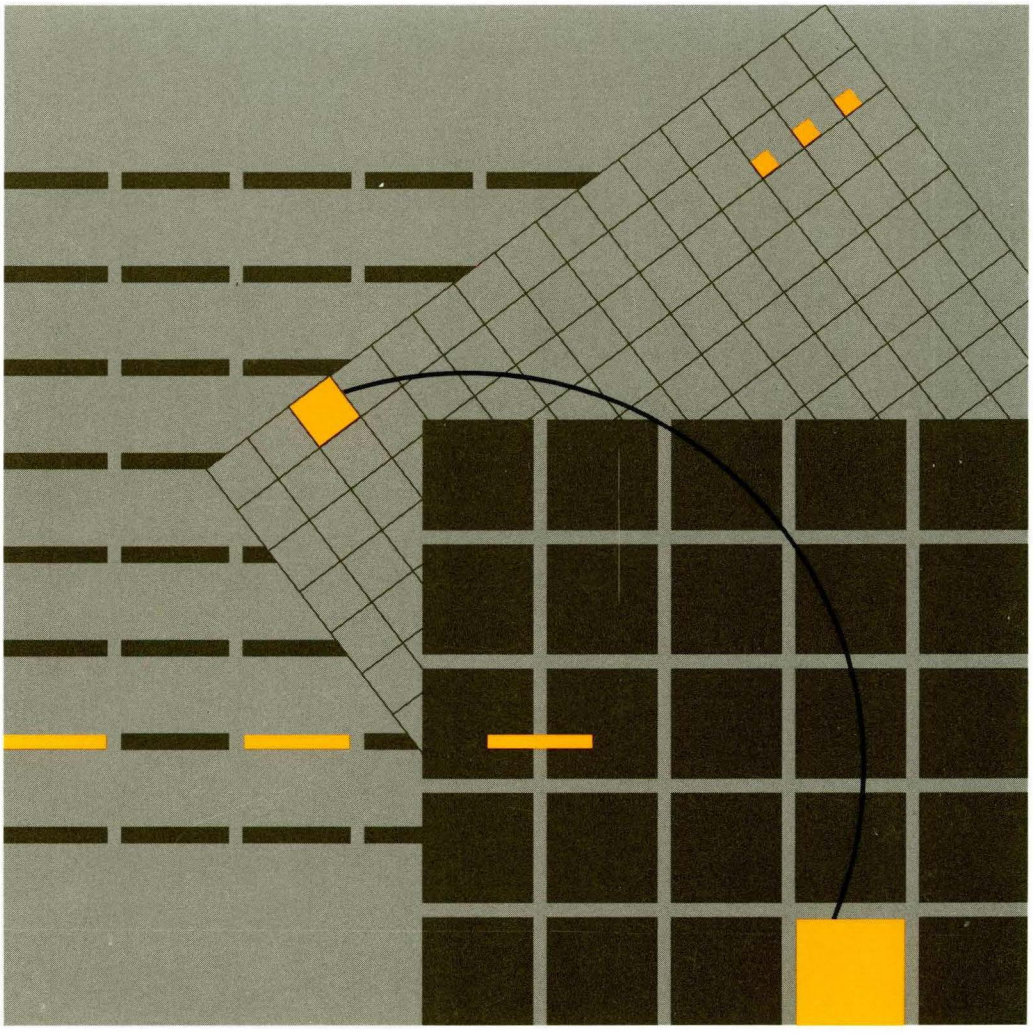
Transmission Control Protocol/
Internet Protocol

SC31-6075-2

TCP/IP Version 1.2 for OS/2: Installation and Maintenance



Transmission Control Protocol/
Internet Protocol

TCP/IP Version 1.2 for OS/2:
Installation and Maintenance

SC31-6075-2

IBM Transmission Control Protocol/ Internet Protocol Version 1.2 for OS/2: Installation and Maintenance

SC31-6075-2

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page xi.

Third Edition (October 1991)

This edition applies to the IBM Transmission Control Protocol/Internet Protocol for OS/2 Version 1.2 licensed program.

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to:

IBM Corporation
Department E15
P.O. Box 12195
Research Triangle Park, North Carolina 27709
U.S.A.

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

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by Sun Microsystems, Massachusetts Institute of Technology, Digital Equipment Corporation, and The University of California.

- Export of this software from the United States of America is assumed to require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

Contents

Figures

Tables

Notices

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the Agreement for IBM Licensed Programs.

Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's program may be used.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send inquiries, in writing, to the IBM Director of Commercial Relations, International Business Machines Corporation, Purchase, New York, 10577.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

The following terms, denoted by an asterisk (*) at their first occurrences in this publication, are trademarks of IBM Corporation in the United States or other countries:

AIX	IBM	Operating System/2
OS/2	Presentation Manager	PS/2
RISC System/6000		

The following terms, denoted by a double asterisk (**) at their first occurrences in this publication, are trademarks of other companies:

Trademark	Owned By
Ethernet	Xerox Corporation
Hayes	Hayes Microcomputer Products, Inc.
Microsoft C	Microsoft Corporation
NCS	Apollo Computer, Inc.
NDIS	3Com Corporation/Microsoft Corporation
Network File System	Sun Microsystems, Inc.
NFS	Sun Microsystems, Inc.
Portmapper	Sun Microsystems, Inc.
PostScript	Adobe Systems, Inc.
UNIX	UNIX System Laboratories, Inc.
VT100	Digital Equipment Corporation
X Window System	Massachusetts Institute of Technology.

About This Book

IBM Transmission Control Protocol/Internet Protocol Version 1.2 for OS/2: Installation and Maintenance describes the installation, configuration, and maintenance of the IBM* Transmission Control Protocol/Internet Protocol for Operating System/2* (TCP/IP for OS/2*) software on a personal computer (PC).

Note: In this book, the abbreviation PC refers to personal computer. See "How the Term "PC" Is Used" on page xv. The abbreviation OS/2 refers to Operating System/2 Standard Edition (OS/2 SE), Version 1.3, Operating System/2 Extended Edition (OS/2 EE), Version 1.3, or Operating System/2, Version 2.0.

Who Should Use This Book

IBM TCP/IP Version 1.2 for OS/2: Installation and Maintenance is intended for network administrators, PC users responsible for installing TCP/IP for OS/2, and system programmers.

If you are installing this product, you should have a working knowledge of computer network technology, the operation of the PC, and knowledge of the OS/2 operating system. Knowledge of the TCP/IP protocols and standard TCP/IP user applications are also helpful. In this book, the term ***protocol*** is a set of rules for handling communication tasks.

If you are not familiar with TCP/IP concepts, see *Internetworking With TCP/IP Volume I: Principles, Protocols, and Architectures*, and *Internetworking With TCP/IP Volume II: Implementation and Internals*.

How to Use This Book

You should read this book when you want to install TCP/IP for OS/2, or when you want to set up TCP/IP for OS/2 servers and functions.

How This Book Is Organized

Read the beginning section of each chapter to familiarize yourself with the topics that you need to know to plan and install this product.

Chapter 1, "Introducing Computer Networks and Protocols," describes computer networks, an internet environment, and protocols supported by TCP/IP for OS/2. Also included in this chapter is an overview of the routing and addressing schemes used by TCP/IP for OS/2.

Chapter 2, "Introducing TCP/IP for Your OS/2 Environment," describes the hardware, software, and other considerations needed to install TCP/IP for OS/2.

Chapter 3, "Installing TCP/IP for OS/2," provides instructions for automated installation of your TCP/IP for OS/2.

Chapter 4, "Host Name Resolution," describes the files used for host name resolution.

Chapter 5, "Manually Modifying Your TCP/IP Configuration," provides information to customize your TCP/IP system.

Chapter 6, "Manually Setting Up the TCP/IP Servers," describes how to customize your TCP/IP servers for the OS/2 environment.

Chapter 7, "Installing and Customizing Network Management Components," describes how to manually set up the Simple Network Management Protocol (SNMP).

Chapter 8, "Installing the Network File System Client," describes how to install Network File System** clients.

Chapter 9, "Installing the Network File System Server," describes how to install Network File System servers.

Chapter 10, "Setting Up an X.25 Interface," describes how to install, configure, and use an X.25 interface.

Chapter 11, "Setting Up a SLIP Line," describes how to connect to another TCP/IP network over a serial line.

Chapter 12, "Setting Up Your Kerberos System," describes how to manually configure, customize, and verify the Kerberos Authentication System for TCP/IP for OS/2.

Chapter 13, "Setting Up the X Window System Server," describes how to setup and use the X Window System** server.

Chapter 14, "Boot Protocol (BOOTP)," describes how to use the BOOTPD and BOOTP commands.

Chapter 15, "Security Issues," provides information concerning security issues in TCP/IP for OS/2.

Appendix A, "Optional Files," contains optional files that can be created for use by TCP/IP for OS/2 applications.

Appendix B, "Sample SLIP.CMD File," contains a sample SLIP.CMD file.

Appendix C, "Sample OS/2 TCP/IP Default Directory Structure," describes the defaults directory structure for the TCP/IP for OS/2 product.

Appendix D, "Management Information Base (MIB) Objects," contains a list of the variables defined by the Management Information Base (MIB).

Appendix E, "MIB2.TBL File: MIB-II Objects," contains a sample MIB variable file.

Appendix F, "Messages and Codes," provides a list of messages and codes for TCP/IP for OS/2.

Appendix G, "Sample BOOTPTAB File," contains a sample BOOTPTAB file.

Appendix H, "Related Protocol Specifications," provides a list of related protocol specifications.

The book also includes a glossary, a bibliography, and an index.

For comments and suggestions on *IBM TCP/IP Version 1.2 for OS/2: Installation and Maintenance*, use the Reader's Comment Form located at the back of this book. Use this form to give IBM information that might improve the book.

What Is New in This Book

The following is a list of the differences between this edition and the previous edition of this book.

- Chapter 1, "Introducing Computer Networks and Protocols," has been expanded to include the X Window System and X.25.
- Chapter 6, "Manually Setting Up the TCP/IP Servers," has been expanded to include information about the LPR driver and RSHD.
- Chapter 7, "Installing and Customizing Network Management Components," is new. The old Chapter 7 "Obtaining Network Status Information," has been moved to *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.
- Chapter 9, "Installing the Network File System Server," is new.
- Chapter 10, "Setting Up an X.25 Interface," is new.
- Chapter 13, "Setting Up the X Window System Server," is new.
- Chapter 14, "Boot Protocol (BOOTP)," is new.
- Appendix D, "Management Information Base (MIB) Objects," is new.
- Appendix E, "MIB2.TBL File: MIB-II Objects," is new.
- Appendix G, "Sample BOOTPTAB File," is new.
- Appendix H, "Related Protocol Specifications," is new.

How the Term "internet" Is Used

In this book, an internet is a logical collection of networks supported by gateways, routers, hosts, and various layers of protocols that permit the network to function as a large, virtual network.

Note: The term internet is used as a generic term for a TCP/IP network, and should not be confused with the Internet (note capital I), which consists of large national backbone networks (such as MILNET, NSFNet, and CREN) and a myriad of regional and local campus networks all over the world.

How the Term "PC" Is Used

In this book, PC refers to a personal computer that can run Operating System/2 Standard Edition (OS/2 SE), Version 1.3, Operating System/2 Extended Edition (OS/2 EE), Version 1.3, or Operating System/2, Version 2.0.

Coding Conventions Used in This Book

The following coding conventions are used throughout this book:

- Uppercase letters represent commands and subcommands that you must type verbatim.
- Lowercase letters represent values that must be entered in lowercase.
- Lowercase italicized letters represent variable parameters for which you supply the values.
- Square brackets [] enclose optional or conditional values.

  Optional values can be omitted. When certain optional values are omitted, default values are used.

  Conditional values can be omitted, depending on the statement.
- Braces {} enclose values from which you must choose a value.

- Vertical line symbols | indicate that you must select a value from either side of the symbol.
- Periods in numbers separate the whole and the fractional portions of the numeral.
- Commands and subcommands appear in bold print within format boxes.

Note: Upper or lowercase italicized letters can also represent screen selections from which you must choose. For example:

Select *Yes* to stop the NFS control program.

How Numbers Are Used in This Book

In this book, numbers over four digits are represented in metric style. A space is used rather than a comma to separate groups of three digits. For example, the number sixteen thousand, one hundred forty-seven is written 16 147.

Where to Find More Information

The following is a list of related publications that you might want to read for more information about TCP/IP for OS/2.

- *IBM TCP/IP Tutorial and Technical Overview*
- *Internetworking With TCP/IP Volume I: Principles, Protocols, and Architectures*
- *Internetworking With TCP/IP Volume II: Implementation and Internals*
- *IBM Operating System/2 System Administrator's Guide for Communications*
- *Introducing IBM's TCP/IP Products for OS/2, VM, and MVS*
- *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference*
- *IBM TCP/IP Version 1.2 for OS/2: User's Guide*
- *IBM TCP/IP Version 1.2 for OS/2: Quick Reference Guide.*

For more information about related publications, see the "Bibliography" at the back of this book.

Chapter 1. Introducing Computer Networks and Protocols

Chapter 1. Introducing Computer Networks and Protocols

This chapter introduces the concepts of computer networks and an internet environment. The protocols used by TCP/IP are listed by layer, and then described. Routing and addressing guidelines are also described.

Computer Networks

A computer network is a group of connected nodes that are used for data communication. A computer network configuration consists of data processing devices, software, and transmission media that are linked for information interchange.

Nodes are the functional units, located at the points of connection among the data circuits. A node, or end point, can be a host computer, a communication controller, a cluster controller, a video display terminal, or another peripheral device.

Computer networks can be local area networks (LANs), which provide direct communication among data stations on the user's local premises, or wide area networks (WANs), which provide communication services to a geographic area larger than that served by a LAN. Typically, WANs operate at a slower rate of speed than LANs.

Different types of networks provide different functions. Network configurations vary, depending on the functions required by the organization. Different organizations implement different types of networks. The technology used by these networks varies not only from organization to organization, but often varies within the same company.

Networks can differ at any or all layers. At the physical layer, networks can run over various network interfaces, such as token ring, Ethernet**, PC Network, X.25, and serial line. Networks can also vary as to the architectures they use to implement network strategies. Some of the more common architectures used today are OSI, TCP/IP, SNA, and ISDN. Networks use different protocols to communicate over the different physical interfaces available. In addition to these differences, networks can all use different software packages to implement various functions.

To exchange information among these different networks, the concept of an internet emerged.

Internet Environment

An internet is a logical collection of networks supported by gateways, routers, bridges, hosts, and various layers of protocols. An internet permits different physical networks to function as a single, large virtual network, and permits dissimilar computers to communicate with each other, regardless of their physical connections. Processes within gateways, routers, and hosts originate and receive packet information. Protocols specify a set of rules and formats required to exchange these packets of information.

Protocols are used to accomplish different tasks in TCP/IP software. To understand TCP/IP, you should be familiar with the following terms and relationships.

A ***client*** is a computer or process that requests services on the network. A ***server*** is a computer or process that responds to a request for service from a client. A ***user*** accesses a service, which allows the use of data or some other resource.

A ***datagram*** is the basic unit of information, consisting of one or more data packets that are passed across an internet at the transport level.

A ***gateway*** is a functional unit that connects two computer networks of different network architectures. A ***router*** is a device that connects networks at the ISO Network Layer. A router is protocol-dependent and connects only networks operating the same protocol. Routers do more than transmit data; they also select the best transmission paths and optimum sizes for packets. A ***bridge*** is a router that connects two or more networks and forwards packets among them. The operations carried out by a bridge are done at the physical layer and are transparent to TCP/IP and TCP/IP routing.

A ***host*** is a computer, connected to a network, which provides an access point to that network. A host can be a client, a server, or a client and server simultaneously. In a communication network, computers are both the sources and destinations of the packets. The ***local host*** is the computer to which a user's terminal is directly connected without the use of an internet, for example, a PC running TCP/IP. A ***foreign host*** is any host on the network including the local host. A ***remote host*** is any foreign host not including the local host. A host is identified by its internet address.

An ***internet address*** is a unique 32-bit address identifying each node in an internet. An internet address consists of a network number and a local address. Internet addresses are represented in dotted-decimal notation and are used to route packets through the network.

Mapping relates internet addresses to physical hardware addresses in the network. For example, the Address Resolution Protocol (ARP) is used to map internet addresses to token ring or Ethernet physical hardware addresses.

A ***network*** is the combination of two or more nodes and the connecting branches among them. A ***physical network*** is the hardware that makes up a network. A ***logical network*** is the abstract organization overlaid on one or more physical networks. An internet is an example of a logical network.

Packet refers to the unit or block of data of one transaction between a host and its network. A packet usually contains a network header, at least one high-level protocol header, and data blocks. Generally, the format of the data blocks does not affect how packets are handled. Packets are the exchange medium used at the internetwork layer to send and receive data through the network.

A ***port*** is an end point for communication between applications, generally referring to a logical connection. A port provides queues for sending and receiving data. Each port has a port number for identification. When the port number is combined with an internet address, a ***socket*** address results.

Protocol refers to a set of rules for achieving communication on a network.

TCP/IP Protocols and Functions

This section categorizes the TCP/IP protocols and functions by their functional group (network layer, internetwork layer, transport layer, and application layer). Figure 1 shows the relationship of these protocols and functions within the TCP/IP layered architecture for OS/2.

- Network Layer
  - X.25 Protocol
  - Serial Line Internet Protocol (SLIP)
- Internetwork Layer
  - Internet Protocol (IP)
  - Internet Control Message Protocol (ICMP)
  - Routing Information Protocol (RIP)
  - Address Resolution Protocol (ARP)
- Transport Layer
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
- Application Layer
  - Telnet
  - File Transfer Protocol (FTP)
  - Trivial File Transfer Protocol (TFTP)
  - Simple Mail Transfer Protocol (SMTP)
  - Domain Name System (DNS)
  - Simple Network Management Protocol (SNMP)
  - Kerberos Authentication System
  - Remote Printing (LPR and LPD)
  - Talk
  - Finger
  - RouteD
  - X Window System
  - Remote Procedure Call (RPC)
  - Network File System (NFS**)
  - Remote Execution Protocol (REXEC)
  - Network Computing System (NCS)
  - Socket Interfaces.

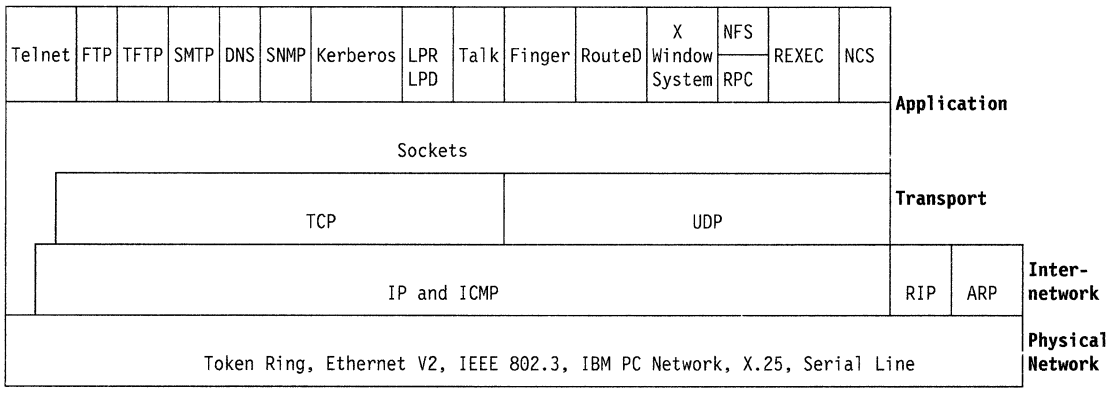


Figure 1. The TCP/IP Layered Architecture

Network Protocols

This section describes the protocols that compose the network layer available in TCP/IP for OS/2. Network protocols define how data is transported over a physical network. These network protocols are not defined by TCP/IP. After a TCP/IP packet is created, the network protocol adds a transport dependent network header before the packet is sent out onto the network.

X.25 Protocol

You can use an X.25 network to establish a TCP/IP connection between two hosts. X.25, recommended as a communication interface standard by the International Telegraph and Telephone Consultative Commitee (CCITT), defines the interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE). A DTE is a computer or workstation connected to a network. A DCE is the equipment at the point of the connection to the network, such as a modem.

For more information about TCP/IP over X.25, see RFC 877.

Serial Line Internet Protocol (SLIP)

In TCP/IP for OS/2, the Serial Line Internet Protocol (SLIP) allows you to set up a point-to-point connection between two TCP/IP hosts over a serial line; for example, a serial cable or an RS-232 connection into a modem and over a telephone line. You can use SLIP to access a remote TCP/IP network from your local host, or to route datagrams between two TCP/IP networks.

For more information about SLIP, see RFC 1055.

Internetwork Protocols

Protocols in the internetwork layer provide connection services for TCP/IP. These protocols connect physical networks and transport protocols. This section describes the internetwork protocols in TCP/IP.

For more information about TCP/IP in general, see Request For Comments (RFCs) 1118, 1180, 1206, 1207, and 1208. See Appendix H, "Related Protocol Specifications" for a list of other related RFCs.

Internet Protocol (IP)

Internet Protocol (IP) provides the interface from the transport level (host-to-host, TCP or UDP) protocols to the physical-level protocols. IP is the basic transport mechanism for routing IP packets to the next gateway, router, or destination host.

IP provides the means to transmit blocks of data (or packets) from sources to destinations. Sources and destinations are hosts identified by internet addresses. Outgoing packets automatically have an IP header prefixed to them, and incoming packets have their IP header removed before being sent to the higher-level protocols. This protocol provides the universal addressing of hosts in an internet network.

IP does not ensure a reliable communication, because it does not require acknowledgments from the sending host, the receiving host, or intermediate hosts. IP does not provide error control for data; it provides only a header checksum. IP treats each packet as an independent entity unrelated to any other packet. IP does not perform retransmissions or flow control. A higher-level protocol that uses IP must implement its own reliability procedures.

For more information about IP, see RFC 791.

Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP) passes control messages between gateways, routers, and hosts. For example, ICMP messages can be sent in any of the following situations:

- When a host checks to see if another host is available (PING).
- When a packet cannot reach its destination.
- When a gateway or router can direct a host to send traffic on a shorter route.
- When a host requests a netmask or a time stamp.
- When a gateway or router does not have the buffering capacity to forward a packet.

ICMP provides feedback about problems in the communication environment; it does not make IP reliable. ICMP does not guarantee that an IP packet will be delivered reliably or that an ICMP message will be returned to the source host when an IP packet is not delivered or is incorrectly delivered.

For more information about ICMP, see RFC 792.

Routing Information Protocol (RIP)

Routing Information Protocol (RIP) is used by gateways, routers, and hosts to exchange routing information. This information can be used to maintain routing table entries.

For more information about RIP, see RFC 1058.

Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) maps internet addresses to hardware addresses. TCP/IP uses ARP to collect and distribute the information for mapping tables.

ARP is not directly available to users or applications. When an application sends an internet packet, IP requests the appropriate address mapping. If the mapping is not in the mapping table, an ARP broadcast packet is sent to all the hosts on the network requesting the physical hardware address for the host.

For more information about ARP, see RFC 826.

Transport Protocols

The transport layer of TCP/IP consists of transport protocols, which allow communication between application programs. This section describes the transport protocols in TCP/IP.

Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) provides a reliable vehicle for delivering packets between hosts on an internet. TCP takes a stream of data, breaks it into datagrams, sends each one individually using IP, and reassembles the datagrams at the destination node. If any datagrams are lost or damaged during transmission, TCP detects this fact and resends the missing datagrams. The received data stream is a reliable copy of the transmitted data stream.

TCP is designed to guarantee that data is not damaged, lost, duplicated, or delivered out of order. The receiving process has control over the rate at which it receives the data.

For more information about TCP, see RFC 793.

User Datagram Protocol (UDP)

User Datagram Protocol (UDP) provides an unreliable mode of communication between source and destination hosts. UDP is built upon the service of the IP protocol in the internetwork layer. UDP provides a procedure for application programs to send data to other programs with a minimum of protocol overhead.

Like IP, UDP does not offer reliable datagram delivery or duplication protection. UDP does provide checksums for both the header and data portions of a datagram. However, applications that require reliable delivery of streams of data should use TCP.

For more information about UDP, see RFC 768.

Applications, Functions, and Protocols

Applications are provided with TCP/IP for OS/2 that allow users to use network services. These applications are included in the application layer of TCP/IP. The application layer is built upon the services of the transport layer. This section describes the applications, functions, and protocols in TCP/IP.

Telnet Protocol

Telnet Protocol provides a standard method to interface terminal devices and terminal-oriented processes with each other. Telnet is built upon the services of TCP in the transport layer. Telnet provides duplex communication and sends data either as ASCII characters or as binary data.

Telnet is commonly used to establish a logon session on a foreign host. Telnet can also be used for terminal-to-terminal communication and interprocess communication.

For more information about the Telnet Protocol, see RFCs 854, 856, 857, 885, and 1091.

File Transfer Protocol (FTP)

File Transfer Protocol (FTP) makes it possible to transfer data between local and foreign hosts or between two foreign hosts. FTP is built upon the services of TCP in the transport layer. FTP transfers files as either ASCII characters or as binary data. ASCII characters are used to transfer files that contain text characters.

FTP provides functions such as listing remote directories, changing the current remote directory, creating and removing remote directories, and transferring one or more files in a single request. Security is handled by passing user and account passwords to the foreign hosts.

For more information about FTP, see RFC 959.

Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol (TFTP) is designed only to read and write files to and from a foreign host. TFTP is built upon the services of UDP in the transport layer. TFTP allows you to limit drive and directory access.

TFTP, like FTP, can transfer files as either ASCII characters or as binary data. However, unlike FTP, TFTP cannot be used to list or change directories at a foreign host, and it has no provisions for user authentication.

For more information about TFTP, see RFC 783.

Simple Mail Transfer Protocol (SMTP)

Simple Mail Transfer Protocol (SMTP) is an electronic mail protocol with both client (sender) and server (receiver) functions.

SMTP is implemented with the Sendmail program in an OS/2 environment. You do not interface directly with SMTP. Instead, electronic mail software is used to create mail, which in turn uses SMTP to send the mail to its destination.

For more information about SMTP, see RFCs 821, 822, and 974.

Domain Name System (DNS)

Domain Name System (DNS) uses a hierarchical-naming system for naming hosts. Each host name is composed of domain labels separated by periods. Local network administrators have the authority to name local domains within an internet. Each label represents an increasingly higher domain level within an internet. The fully qualified domain name of a host connected to one of the larger internets generally has one or more subdomains. For example:

host.subdomain.subdomain.rootdomain
or
host.subdomain.rootdomain

Domain names often reflect the hierarchy level used by network administrators to assign domain names. For example, the domain name `eng.mit.edu` is the lowest level domain name, which is a subdomain of `mit.edu`. The subdomain `mit.edu` is a subdomain of `edu`.

Figure 2 on page 10 is an example of the DNS used in the hierarchy naming structure across an internet.

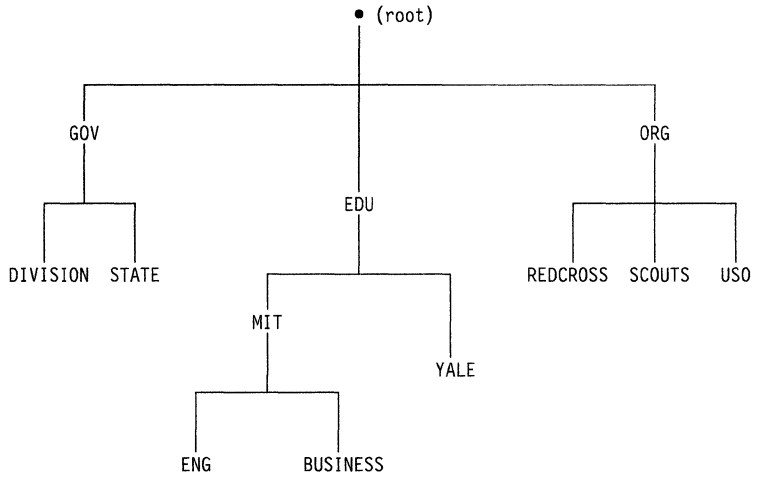


Figure 2. Hierarchical Tree

You can refer to hosts in your domain by host name only; however, a name server requires a fully qualified domain name. The local resolver combines the host name with the domain name before sending the address resolution request to the domain name server.

TCP/IP for OS/2 uses the local resolver functions of a local name resolution file. This file, called HOSTS, resides in the ETC directory and contains entries that allow you to map symbolic names to internet addresses. If a RESOLV file exists in the ETC directory, the resolver sends the request to the foreign name server before using the local HOSTS file.

When using the HOSTS file on a small internet, it is not necessary to use the hierarchical-naming system used by the larger internets. The following example is a token ring network of three users and their entries in the HOSTS file.

```
129.5.24.1 Host1 vjsPC PC1 mathdept
129.5.24.3 PC3 normasPC Host3 # This is Norma's PC
129.5.24.4 PC4 budsPC
```

A carriage return must be entered at the end of each line.

In this example, each time the user enters the *host_name* of `Host1` or the *aliases* `vjsPC`, `PC1`, or `mathdept`, the local name resolver translates it to the internet address of `129.5.24.1`. For more information about the format of network addresses, see "Network Address Format" on page 14.

For more information about DNS, see RFCs 1034 and 1035.

Simple Network Management Protocol (SNMP)

Simple Network Management Protocol (SNMP) provides a means for managing an internet environment. SNMP allows network management by elements, such as gateways, routers, and hosts. Network elements act as servers and contain management agents, which perform the management functions requested. Network management stations act as clients; they run the management applications, which monitor and control the network. SNMP provides a means of communicating between these elements and stations to send and receive information about network resources.

For more information about SNMP, see RFCs 1155, 1157, 1187, and 1213.

Kerberos Authentication System

The Kerberos Authentication System provides additional security by allowing authorization checking at the user level rather than at the node level. This system allows client and server pairs to verify that the partner is authorized to participate in the function being performed.

For more information about Kerberos, see the MIT papers "Kerberos: An Authentication Service for Open Networks" and "Kerberos Authentication and Authorization System."

Remote Printing (LPR and LPD)

TCP/IP for OS/2 provides both client and server support for remote printing. This application allows you to spool files remotely to a line printer daemon (LPD). The line printer client (LPR) sends the file to be printed to a specified print server host and to a specified printer.

For more information about LPR and LPD, see RFC 1179.

Talk

Talk allows you to send interactive messages, as opposed to the batch mail capabilities of SMTP. When a local host sends a Talk request to a foreign host, the user on the foreign host is notified that there is a connection request. The user on the foreign host must respond with a Talk message to the local host. Message exchange can then occur between the local and foreign hosts.

Finger Protocol (FINGER)

The Finger Protocol (FINGER) provides an interface for querying the current status of a remote host or a user ID on a remote host. FINGER uses TCP as the underlying protocol.

For more information about FINGER, see RFC 1196.

RouteD

RouteD uses the Routing Information Protocol (RIP) to dynamically create and maintain network routing tables. The RIP protocol arranges to have gateways and routers periodically broadcast their routing tables to neighbors. Using this information, a RouteD server can update a host's routing tables. For example, RouteD determines if a new route has been created, if a route is temporarily unavailable, or if a more efficient route exists.

For more information about RouteD, see RFC 1058.

X Window System

The X Window System Protocol is designed to support network transparent windowing and graphics. TCP/IP for OS/2 provides server support to X Window System client applications.

For more information about X Window System, see RFC 1013.

File Transfer Protocol Application Programming Interface (FTP API)

The File Transfer Protocol (FTP) Application Programming Interface (API) allows applications to have a client interface for file transfer. Applications written to this interface can communicate with multiple FTP servers at the same time. A maximum of 256 simultaneous connections are supported. The interface also allows third-party transfers between pairs of FTP servers. Consecutive third-party proxy transfers are allowed between any sequence of pairs of FTP servers.

The API tracks the servers to which an application is currently connected. When a new request for FTP service is requested, API checks whether there is a connection to the server. If the connection does not exist, it is established. If the server has dropped the connection since last use, it is reestablished.

FTP API provides functions, such as listing remote directories, changing the current remote directory, creating and removing remote directories, and transferring one or more files in a single request. Security is handled by passing user and account passwords to the foreign hosts.

For more information about FTP, see RFC 959.

Remote Procedure Call (RPC)

The Remote Procedure Call Protocol (RPC) is a programming interface that allows programs to execute subroutines on a foreign host. RPCs are high-level program calls, which can be used in place of the lower-level calls that are based on sockets.

For more information about RPC, see RFC 1057.

Network File System (NFS)

The Network File System (NFS) allows you to manipulate files on remote TCP/IP hosts as if they reside on your local host. NFS is based on the NFS protocol, and uses the Remote Procedure Call (RPC) protocol to communicate between the client and the server. The files to be accessed reside on the server host, and are made available to the user on the client host.

NFS supports a hierarchical file structure. The directory and subdirectory structure can be different for individual client systems.

For more information about NFS, see RFC 1094.

Remote Execution Protocol (REXEC)

Remote Execution Protocol allows you to execute a command on a foreign host and receive the results on the local host. Remote Execution Protocol provides automatic logon and user authentication depending on the parameters set by the user.

Network Computing System (NCS)

Network Computing System (NCS) is a programmer tool kit that allows programmers to distribute processing power to other hosts. NCS is similar to RPC in that it allows a higher level of program calls. TCP/IP provides a programming interface to NCS.

For more information about NCS, see *Network Computing System (NCS) Reference*.

Socket Interfaces

Socket interfaces allow users to write their own applications to supplement those supplied by TCP/IP for OS/2. Most of these additional applications communicate with either TCP or UDP. Some applications are written to communicate directly with IP. To write applications that use the socket interfaces of TCP/IP for OS/2, you must be able to compile and link the programs using the Microsoft C** compiler, Version 6.00A.

Sockets are duplex, which means that data can be transmitted and received simultaneously. Sockets allow you to send to, and receive from, the socket as if you are writing to and reading from any other network device.

Routing

The routing functions in an internet are performed at the internetwork layer. Routing is the process of deciding where to send a packet based on its destination address. Two kinds of routing are involved in communications within an internet: direct and indirect.

Direct routing is used when the source and destination nodes are on the same logical network within an internet. The source node maps the destination internet address into a hardware address and sends packets to the destination node using this address. This mapping is normally performed through a translation table. If a match cannot be found for a destination internet address, ARP is invoked to determine this address.

Indirect routing is used when the source and destination nodes are on different logical networks within an internet. The source node sends packets to a gateway or router on the same network using direct routing. From there, the packets are forwarded through intermediate gateways or routers, as required, until they arrive at the destination network. Direct routing is then used to forward the packets to the destination host on that network. Each gateway, router, and host in an internet has a routing table that defines the address of the next gateway to other networks (as well as other nodes on other networks) in an internet.

Internet Addressing

Each internet host is assigned at least one unique internet address. This address is used by the IP and other higher-level protocols. When gateway hosts are used, more than one address may be required. Each interface to an internet is assigned its own unique address. Internet addresses are used to route packets through the network.

Addresses within an internet consist of a network number and a local address. A unique network number is assigned to each network when it connects to another internet. If a local network is not going to connect to other internets, any convenient network number is assigned. Some networks are divided into subnets. For information about subnetting, see "Subnetwork Address Format" on page 15.

Hosts that exchange packets on the same physical network should have the same network number. Hosts on different physical networks might also have the same network number. If hosts have the same network number, part of the local address is used as a subnetwork number. All host interfaces to the same physical network are given the same subnetwork number.

An internet can provide standards for assigning addresses to networks, broadcasts, and subnetworks. Examples of these standard formats are described in the following sections.

Network Address Format

A standard internet address uses a two-part, 32-bit address field. The first part of the address field contains the network address; the second part contains the local address. The four different types of address fields are classified as A, B, C, or D, depending on the bit allocation.

Figure 3 represents a class A address. Class A addresses have a 7-bit network number and a 24-bit local address, with the highest order bit set to 0.

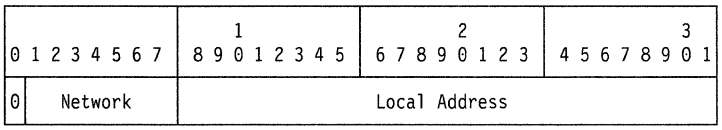


Figure 3. Class A Address

Figure 4 represents a class B address. Class B addresses have a 14-bit network number and a 16-bit local address with the highest order bits set to 10.

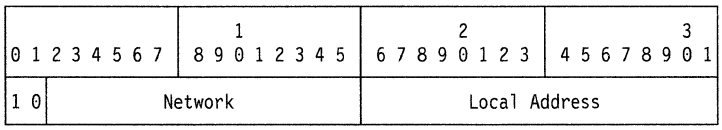


Figure 4. Class B Address

Figure 5 represents a class C address. Class C addresses have a 21-bit network number and an 8-bit local address with the three highest order bits set to 110.

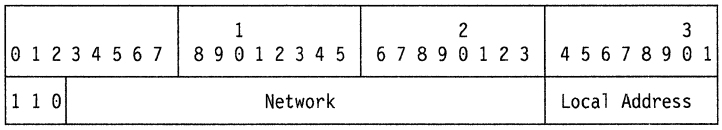


Figure 5. Class C Address

Figure 6 represents a class D address. Class D networks have a multicast address that is sent to selected hosts on the network. The four highest order bits are set to 1110.

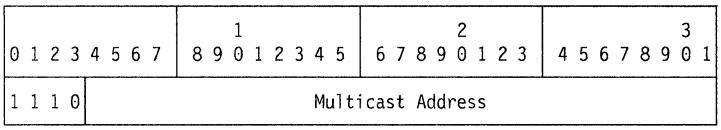


Figure 6. Class D Address

Note: Class D addresses are not supported in TCP/IP for OS/2.

A commonly used notation for internet host addresses is the dotted-decimal, which divides the 32-bit address into four 8-bit fields. The value of each field is specified as a decimal number, and the fields are separated by periods (for example, 010.002.000.052 or 10.2.0.52).

Address examples in this book use dotted-decimal notation in the following forms:

Class A *nnn.lll.lll.lll*
Class B *nnn.nnn.lll.lll*
Class C *nnn.nnn.nnn.lll*

where *nnn* represents part or all of a network number and *lll* represents part or all of a local address.

Broadcast Address Format

TCP/IP uses IP broadcasting to send datagrams to all the TCP/IP hosts on a network or subnetwork. A datagram sent to the broadcast address is received by all the hosts on the network and processed as if the datagram was sent directly to the host's IP address. The IP broadcast address is formed by setting all the host bits to ones.

For more information about IP broadcasting, see RFCs 919 and 922.

Subnetwork Address Format

The subnetwork capability of TCP/IP divides a single network into multiple logical networks (subnets). For instance, an organization can have a single internet network address that is known to users outside the organization, yet configure its internal network into different departmental subnets. Subnetwork addresses enhance local routing capabilities, while reducing the number of network numbers required.

For a subnet, the local address part of an internet address is divided into a subnet number and a host number, for example:

network_number subnet_number host_number

where:

network_number Is the network portion of the internet address.
subnet_number Is a field of a constant width for a given network.
host_number Is a field that is at least 1-bit wide.

If the width of the *subnet_number* field is 0, the network is not organized into subnets, and addressing to the network is done with an internet network address (*network_number*).

Figure 7 represents a class B address with a 6-bit subnet field.

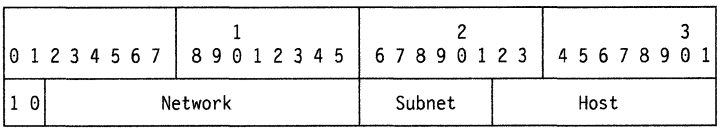


Figure 7. Class B Address with Subnet

The bits that identify the subnet are specified by a bit mask. A bit mask is a pattern of binary digits used to assign subnet addresses. The subnet bits are not required to be adjacent in the address. However, the subnet bits generally are contiguous and located as the most significant bits of the local address.

For more information about Subnetwork Address Format, see RFC 950.

Chapter 2. Introducing TCP/IP for Your OS/2 Environment

Chapter 2. Introducing TCP/IP for Your OS/2 Environment

This chapter contains an introduction to TCP/IP for OS/2. Additional background information can be found in *Introducing IBM's TCP/IP Products for OS/2, VM, and MVS*.

This chapter also describes the hardware and software requirements for installing TCP/IP for OS/2.

Overview of TCP/IP for OS/2

TCP/IP for OS/2 allows PCs running OS/2 to attach to networks running TCP/IP. With TCP/IP for OS/2 your PC can be used as a stand-alone PC, yet access and provide services on a network.

Multitasking

TCP/IP for OS/2 uses the multitasking feature in OS/2. Multitasking allows you to work on two or more applications at the same time. This means TCP/IP clients and servers can run as background tasks in the TCP/IP for OS/2 environment; enabling clients and servers to send and receive, or to request and service several functions at the same time.

You should not run TCP/IP for OS/2 servers as detached processes. Because detached processes do not process output calls to your PC, you do not receive any messages from the servers.

Clients and Servers

TCP/IP for OS/2 clients and servers are programs started by you. Clients request services on the network. Servers respond to requests for service from clients.

Hosts and Routers

When you install TCP/IP for OS/2, you can configure your PC as a host or router, or both. If you install a second network adapter board to connect to another network, you can configure your PC as a router. If you configure your PC to handle both functions, you may require additional memory to provide both packet routing and network services efficiently. If you configure your PC as a router, your performance could also be affected.

System Requirements

This section describes the hardware and software environments that are required for TCP/IP for OS/2.

Hardware Environment

TCP/IP for OS/2 requires the following hardware.

- Computer Models

  Any PC, with the appropriate fixed disk and memory that OS/2 SE 1.3, OS/2 EE 1.3, or OS/2 2.0 supports is also supported by TCP/IP for OS/2.

  Note: To use an X.25 interface with TCP/IP for OS/2, you need at least a PS/2 Model 50 (micro channel) or higher.

- Computer Accessories

  You require the following accessories to install TCP/IP for OS/2:

  - A 3.5-inch or 5.25-inch diskette drive
  - A color or monochrome monitor
  - A keyboard (IBM PC AT or Enhanced)
  - A mouse (optional)
  - A Hayes** AT-compatible modem (optional)

- Network Adapters

  TCP/IP for OS/2 conforms to the Network Driver Interface Specification (NDIS**) and has been tested with the following network adapters.

  - IBM PC Network Adapter II
  - IBM PC Network Adapter II/A
  - IBM PS/2 Adapter/A for Ethernet
  - IBM Token Ring Network Adapter
  - IBM Token Ring Network Adapter II
  - IBM Token Ring Network Adapter/A
  - IBM Token Ring Network 16/4 Adapter
  - IBM Token Ring Network 16/4 Adapter/A
  - 3Com Etherlink II Adapter
  - 3Com Etherlink/MC Adapter
  - Western Digital Ethercard PLUS Adapter
  - Western Digital Ethercard PLUS/A Adapter
  - Ungerman-Bass NIUpc Adapter
  - Ungerman-Bass NIUps Adapter

  To run X.25, you need the following network adapter:

  - IBM X.25 Interface Coprocessor/2

- Network Communication Media

  TCP/IP for OS/2 accommodates several kinds of media. Set up your network communication media using one or more of the following network communication devices:

  - IBM PC Network LAN
  - IBM Token Ring LAN
  - Ethernet LAN
  - Serial Line
  - X.25 Link

  Once the LANs are in place, you can interconnect them with gateways to form an internet. If a serial line is used, the following line speeds are supported:

  - 1200 bps
  - 2400 bps
  - 4800 bps
  - 9600 bps
  - 19 200 bps.

  Note: All line speeds are given in bits per second.

Software Environment

TCP/IP for OS/2 requires OS/2 SE, Version 1.3, OS/2 EE, Version 1.3, or OS/2, Version 2.0 installed on your PC. TCP/IP for OS/2 using an X.25 network requires OS/2 EE 1.3 and Communications Manager installed on your PC.

To install TCP/IP for OS/2, you should know how to create and edit files, and manipulate directories on your fixed disk using OS/2. You should be familiar with the OS/2 Presentation Manager*. You should also be familiar with the OS/2 Communications Manager to use the X.25 network.

If you plan to write your own applications, which use the programming interfaces of TCP/IP for OS/2, or modify the source code provided, you need the Microsoft C compiler, Version 6.00A.

Chapter 3. Installing TCP/IP for OS/2

Chapter 3. Installing TCP/IP for OS/2

This chapter contains information necessary to install TCP/IP for OS/2 on your PC.

System Requirements

You must have Operating System/2 SE, Version 1.3, Operating System/2 EE, Version 1.3, or Operating System/2, Version 2.0, and at least one network adapter or modem installed on your PC. For additional information about the required hardware and software environments, see Chapter 2, "Introducing TCP/IP for Your OS/2 Environment."

The steps you should follow during installation depend on the following:

- The operation of your PC as a host or router
- Your network communication media
- The models of network adapters and the number of network connections planned for your PC.

Installation

The Installation and Configuration Automation Tool (ICAT) program is used for automated installation. ICAT is contained on a diskette that is a part of your TCP/IP for OS/2 base product.

During the installation process, ICAT prompts you for all the information you need to get TCP/IP for OS/2 running on your PC.

ICAT is a Presentation Manager application that uses standard input and output conventions. If at any time you need further assistance to understand a data entry field, place the cursor on the field, and press the F1 key. To get help on the current screen, press the F1 key, or click on the F1=Help selection available at the bottom of each ICAT screen.

Starting ICAT

The following steps describe how to start the ICAT program.

1. Insert the diskette labeled "TCP/IP Version 1.2 for OS/2, diskette B-1" into drive A.
2. Type `A:ICAT` at an OS/2 command prompt, and press the Enter key.

The "TCP/IP - ICAT" introduction window is displayed.

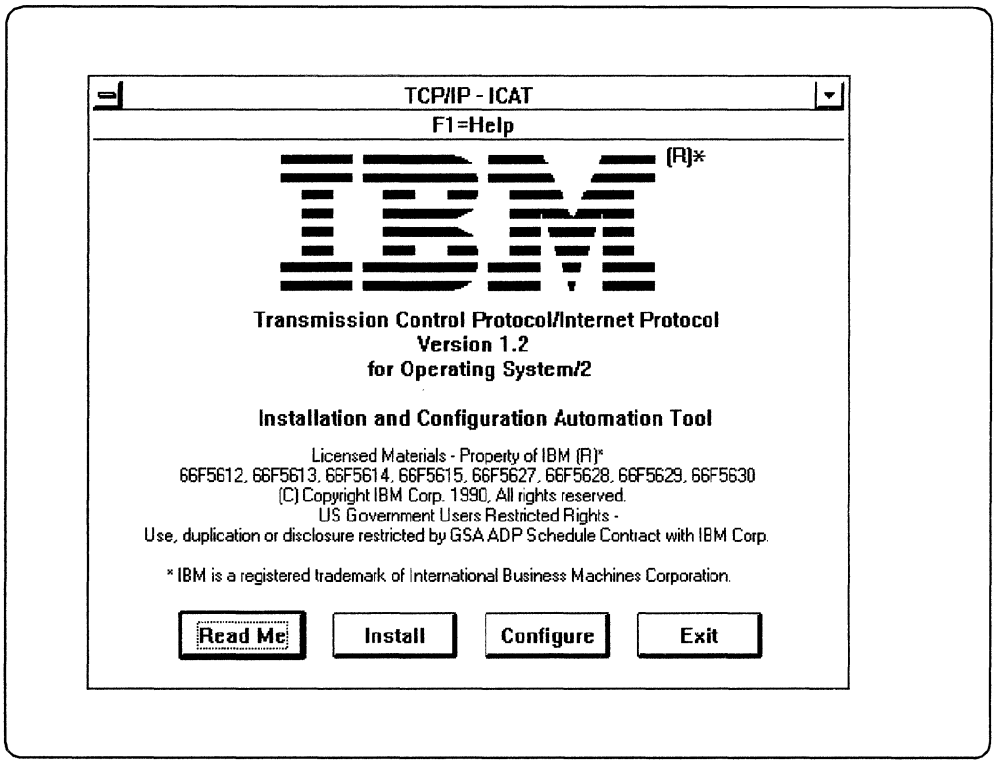


Figure 8. TCP/IP - ICAT

The following button selections are displayed at the bottom of the window.

Selection	Description
Read Me	Displays additional product release information. It is important that you review the Read Me information before installing or configuring the TCP/IP for OS/2 product.
Install	Displays the "TCP/IP Installation Tool" window.
Configure	Displays the "TCP/IP Configuration Tool" window.
Exit	Exits you from the ICAT program.

To install your TCP/IP for OS/2 programs, select *Install*.

Installing TCP/IP

If you are replacing an existing installation of TCP/IP for OS/2 with Version 1.2, follow the procedures described in "Removing TCP/IP for OS/2" on page 40 to remove the earlier installation, before proceeding with your installation.

The following is an example of the "TCP/IP Installation Tool" window that is displayed as a result of selecting *Install* from the "TCP/IP - ICAT" introduction window.

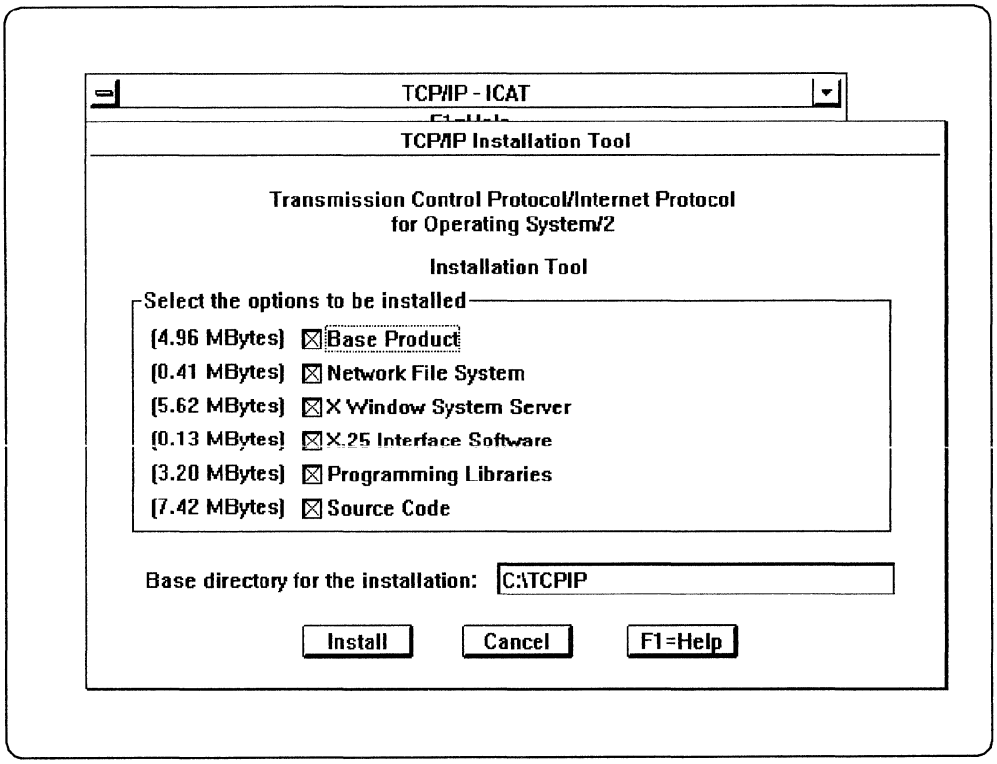


Figure 9. TCP/IP Installation Tool

The following options are displayed.

Option	Description
Base Product	The executable modules that are needed to run TCP/IP and the TCP/IP for OS/2 clients and servers.
Network File System	The executable modules that are needed to run the Network File System Client and Server.
X Window System Server	The executable modules that are needed to run the X Window System Server.
X.25 Interface Software	The executable modules that are needed to run TCP/IP for OS/2 over an X.25 interface.
Programming Libraries	The libraries and header files for each of the TCP/IP application programming interfaces.
TCP/IP Source Code	The source code to customize or change the TCP/IP for OS/2 products.

Note: These products are purchased as a total package or individually. The sizes of the products shown in Figure 9 may have changed since the printing of this manual.

The default base directory for the installation is C:\TCPIP. To change the name of the base directory for the installation of TCP/IP for OS/2, use your Tab key to scroll to the "Base directory for the installation" field, and type the name of the directory.

When you use the "TCP/IP Configuration Tool" window, you are requested to enter the name of the base directory where you installed TCP/IP. Keep a record of the name you have chosen for the base directory.

Press the Tab key to move from the base directory field to the buttons at the bottom of the window.

The following button selections are displayed at the bottom of the window.

Selection	Description
Install	Executes the installation process for the products that you selected.
Cancel	Returns you to the "TCP/IP - ICAT" introduction window without installing any of the selected products.
F1=Help	Provides help information.

If you choose the *Install* selection, you are prompted to insert the required diskettes into drive A. If you are installing a product that requires CONFIG.SYS changes, ICAT asks you if you want to update your CONFIG.SYS after the diskettes have been copied. Upon completion of the automated installation process, a message is displayed, verifying a successful installation.

The "TCP/IP - ICAT" introduction window is redisplayed.

If a message is displayed stating that installation was not successful, repeat the installation process and correct any errors that are indicated during the process.

The TCP/IP for OS/2 programs are installed, by default, into the C:\TCPIP subdirectory. For example, the TCPIP.LIB file resides in the directory path C:\TCPIP\LIB and contains the programming interface libraries of TCP/IP for OS/2. For more information about the default directory structure, see Appendix C, "Sample OS/2 TCP/IP Default Directory Structure."

Before You Configure Your TCP/IP Software

You must have certain information available before configuring your system. Review the ICAT program documentation provided in the following sections of this chapter to determine the information that you need for your specific network configuration.

You are now ready to configure your TCP/IP software.

Configuring TCP/IP

The following is an example of the "TCP/IP Configuration Tool" window that is displayed as a result of selecting *Configure* from the "TCP/IP - ICAT" introduction window.

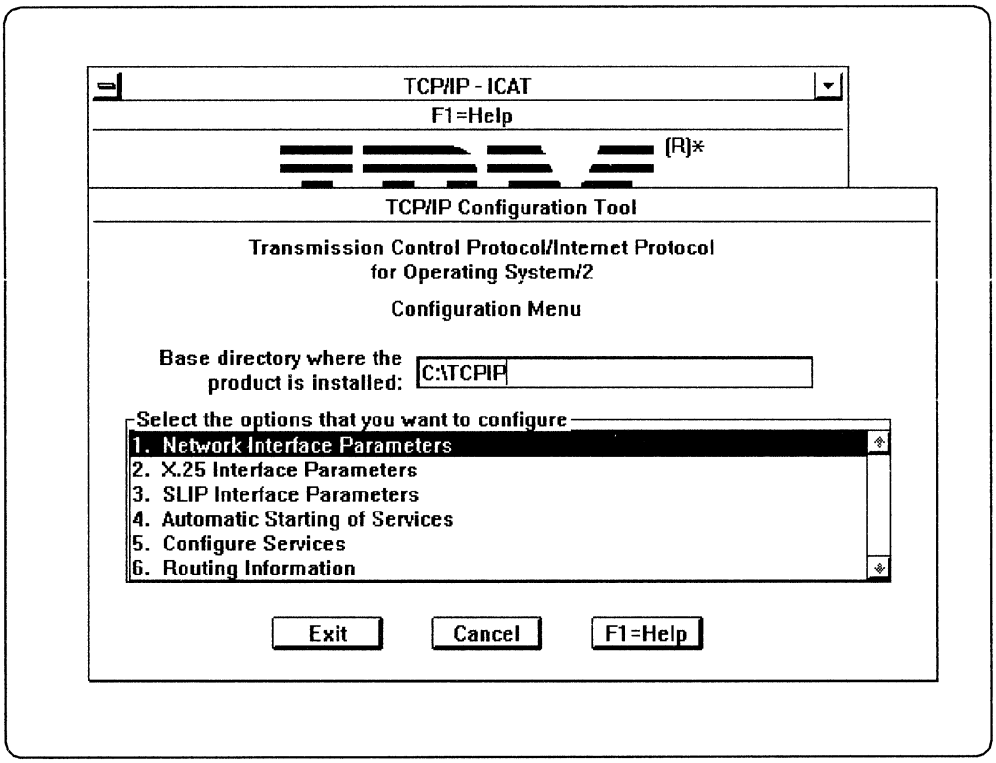


Figure 10. TCP/IP Configuration Tool

At the top of the window in the "Base directory where the product is installed" field, type the name of the installation directory that you specified in the "TCP/IP Installation Tool" window.

The following configuration selections are displayed.

Selection	Description
Network Interface Parameters	Displays the "Configure Network Interface Parameters" window.
X.25 Interface Parameters	Displays the "Configure X.25 Interface Parameters" window.
SLIP Interface Parameters	Displays the "Configure SLIP Interface Parameters" window.
Automatic Starting of Services	Displays the "Configure Automatic Starting of Services" windows.
Configure Services	Displays the "Configure Services" window.
Routing Information	Displays the "Configure Routing Information" window.

Use any of the following methods to make your screen selections.

- Use the mouse and place the cursor on the desired selection, and double click.
- Use the following keyboard functions:
  - Press the directional arrows to scroll to the desired item, and press the Enter key.
  - Press the directional arrows to scroll to the desired item, and press the space bar.
- Use the Tab key to move from one group of selections to another group of selections.

The following button selections are displayed at the bottom of the window.

Selection	Description
Exit	Saves the current configuration values and returns you to the "TCP/IP-ICAT" introduction window.
Cancel	Discards any changes made and returns you to the "TCP/IP-ICAT" introduction window.
F1=Help	Provides help information.

Configuring Network Interface Parameters

The following is an example of the "Configure Network Interface Parameters" window that is displayed as a result of selecting *Network Interface Parameters* from the "TCP/IP Configuration Tool" window.

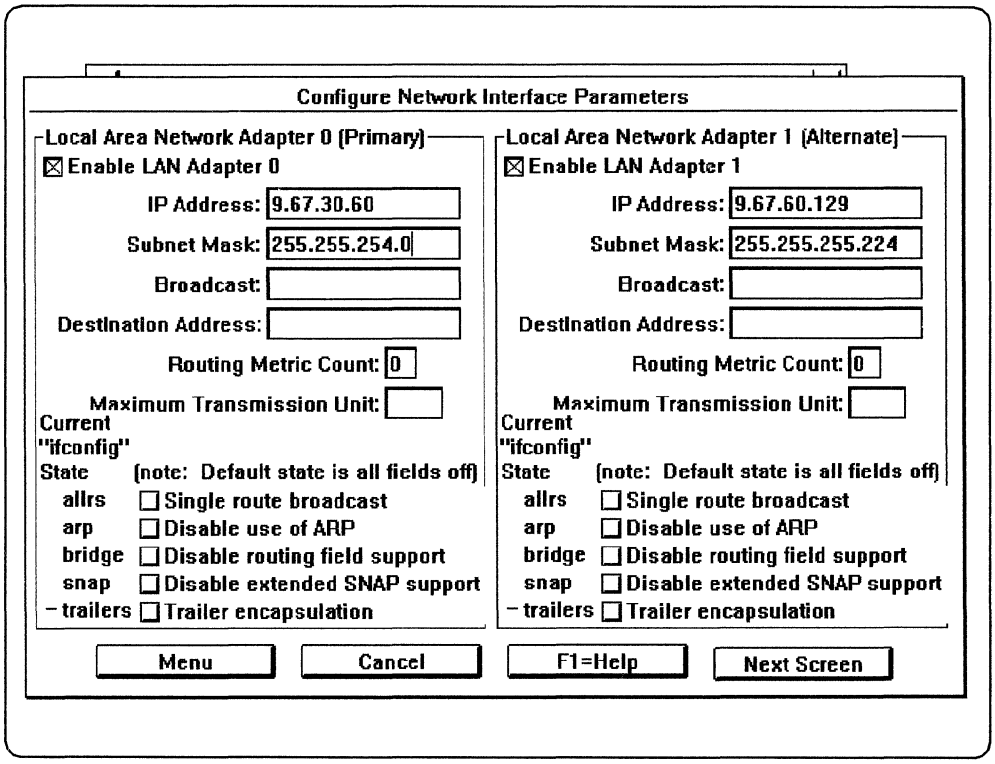


Figure 11. Configure Network Interface Parameters

When you configure your network adapters, ICAT creates or modifies the SETUP.CMD file for you. You can configure up to four network adapters. An example of the format of the network adapter entries is displayed in Figure 11.

The IFCONFIG parameters that you can select are displayed at the bottom of the window. A description of the parameters is also displayed. If you select one of the IFCONFIG options, the current state changes. For more information about the IFCONFIG parameters, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide.*

Use any of the following methods to make your screen selections at the top of the window.

- Use the mouse and place the cursor in the entry field of the desired selection; type your entry, and single click on your next entry field.
- Press the directional arrows to scroll to the desired item; type your entry in the entry field. Use the Tab key to move from one group of selections to another group of selections.

Use any of the following methods to make your screen selections at the bottom of the window.

- Use the mouse and place the cursor in the box of the desired selection, and single click.
- Press the directional arrows to scroll to the desired item, and press the space bar.
- Use the Tab key to move from one group of selections to another group of selections.

The following button selections are displayed at the bottom of the window.

Selection	Description
Menu	Retains the current values, and redisplays the "TCP/IP Configuration Tool" window. The values are not saved until you exit from the "TCP/IP Configuration Tool" window.
Cancel	Discards any changes made and returns you to the "TCP/IP Configuration Tool" window.
F1 = Help	Provides help information.
Next Screen	Displays the next screen.
Previous Screen	Returns you to the previous screen.

Configuring X.25 Interface Parameters

The following is an example of the "Configure X.25 Interface Parameters" window that is displayed as a result of selecting *X.25 Interface Parameters* from the "TCP/IP Configuration Tool" window.

When you configure your X.25 interface, ICAT creates or modifies the X25.CMD file for you.

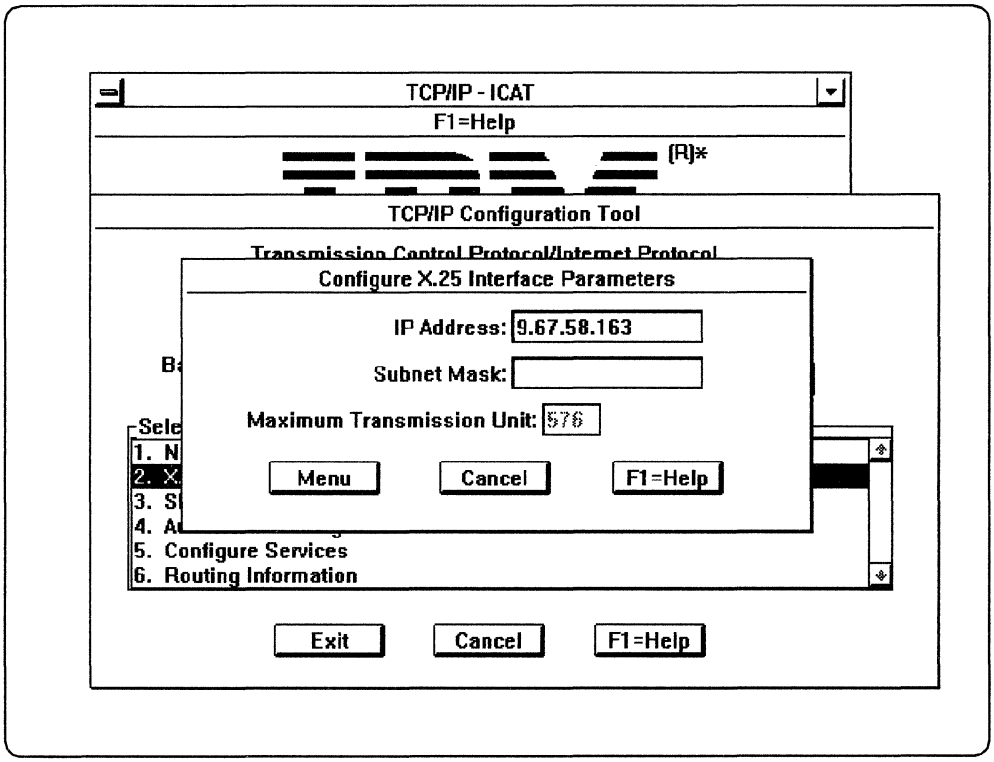


Figure 12. Configure X.25 Interface Parameters

The following button selections are displayed at the bottom of the window.

Selection	Description
Menu	Retains the current values, and redisplays the "TCP/IP Configuration Tool" window. The values are not saved until you exit from the "TCP/IP Configuration Tool" window.
Cancel	Discards any changed values and returns you to the "TCP/IP Configuration Tool" window.
F1=Help	Provides help information.

Configuring SLIP Interface Parameters

The following is an example of the "Configure SLIP Interface Parameters" window that is displayed as a result of selecting *SLIP Interface Parameters* from the "TCP/IP Configuration Tool" window.

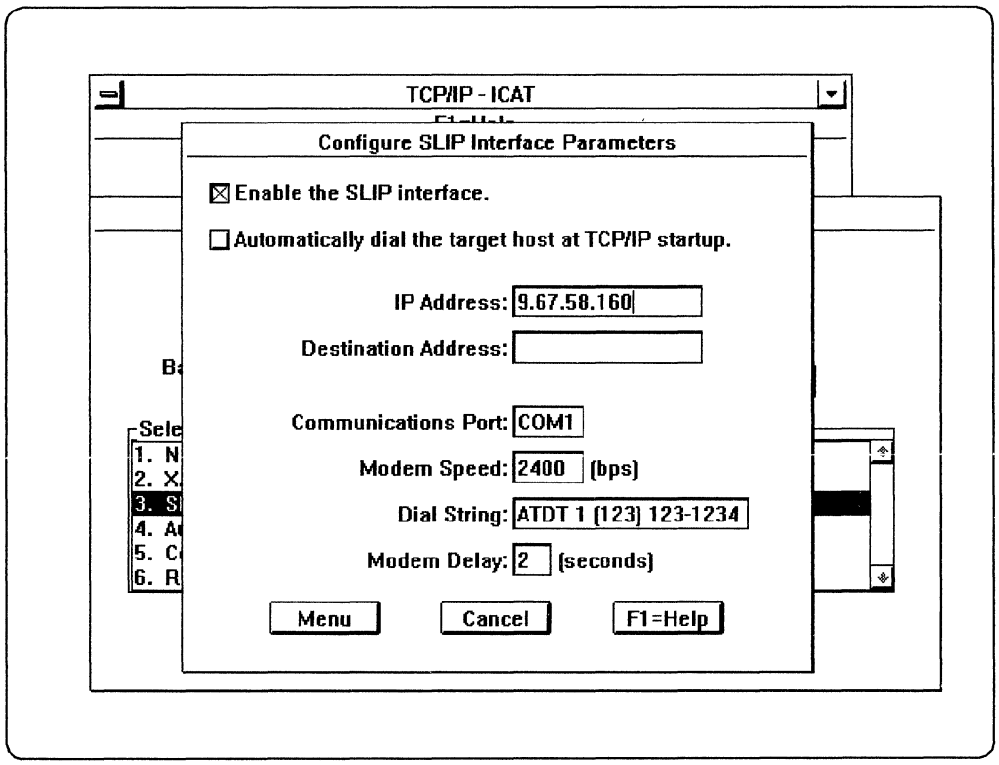


Figure 13. Configure SLIP Interface Parameters

When you configure your SLIP Interface, ICAT creates or modifies the SLIP.CMD and SETUP.CMD files for you. For an example of a SLIP.CMD file, see Appendix B, "Sample SLIP.CMD File."

Note: You must click on the "Enable the SLIP Interface" selection to use the adapter.

The following button selections are displayed at the bottom of the window.

Selection	Description
Menu	Retains the current values, and redisplays the "TCP/IP Configuration Tool" window. The values are not saved until you exit from the "TCP/IP Configuration Tool" window.
Cancel	Discards any changed values and returns you to the "TCP/IP Configuration Tool" window.
F1=Help	Provides help information.

Configuring Automatic Starting of Services

The following are examples of the "Configure Automatic Starting of Services" windows that are displayed when you select *Automatic Starting of Services* from the "TCP/IP Configuration Tool" window.

When you configure the services that you want automatically started, ICAT creates or modifies the TCPSTART.CMD and STARTUP.CMD files.

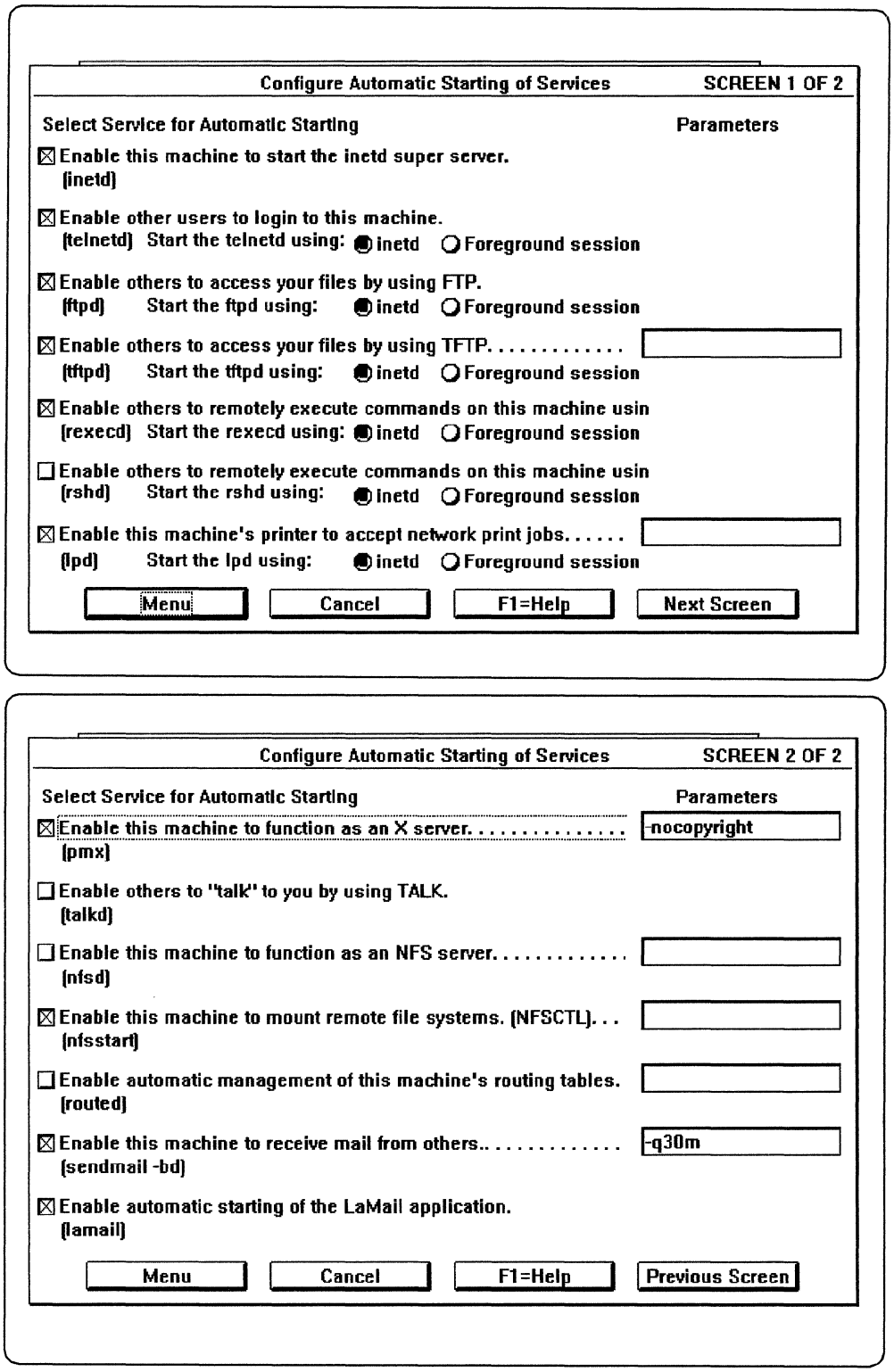


Figure 14. Configure Automatic Starting of Services

Select the TCP/IP services that you want to automatically start when you initially start TCP/IP. The TCP/IP services are invoked from the TCPSTART.CMD file. See Chapter 6, "Manually Setting Up the TCP/IP Servers," for more information about parameters.

Use any of the following methods to make your screen selections on the right side of the window.

- Use the mouse and place the cursor in the entry field of the desired selection; type your entry, and single click on your next entry field.
- Press the directional arrows to scroll to the desired item; type your entry in the entry field.
- Use the Tab key to move down to the buttons at the bottom of the windows.

Use any of the following methods to make your screen selections on the left side of the window.

- Use the mouse and place the cursor in the box of the desired selection, and single click.
- Press the directional arrows to scroll to the desired item, and press the space bar.
- Use the Tab key to move from one group of selections to another group of selections.

The following button selections are displayed at the bottom of the windows.

Selection	Description
Menu	Retains the current values, and redisplays the "TCP/IP Configuration Tool" window. The values are not saved until you exit from the "TCP/IP Configuration Tool" window.
Cancel	Discards any changes made and returns you to the "TCP/IP Configuration Tool" window.
F1=Help	Provides help information.
Next Screen	Displays the next screen.
Previous Screen	Returns you to the previous screen.

Configuring Services

The following is an example of the "Configure Services" window that is displayed as a result of selecting *Configure Services* from the "TCP/IP Configuration Tool" window.

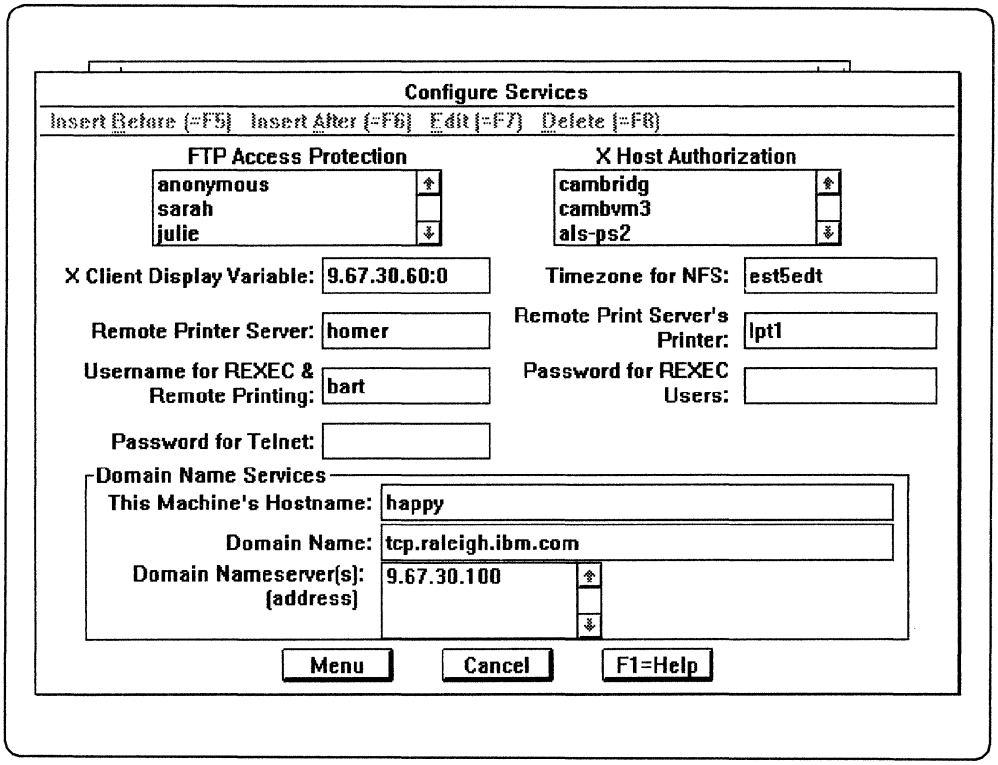


Figure 15. Configure Services

The TCP/IP services that can be configured are:

Service	Description
FTP Access Protection	Specifies the names of the users and passwords that can access your FTP server. The TRUSERS file that is used by the FTP server is created or modified.
X Host Authorization	Specifies the X client hosts on the network that are authorized to connect to the X server. The X0HOSTS file that is used by the X server is created or modified.
X Client Display Variable	Specifies the DISPLAY environment variable used by X client utilities to indicate to which X server to direct their requests. An environmental variable called DISPLAY is added to your CONFIG.SYS file.
Timezone for NFS	Specifies the TZ environment variable used by the NFS server and client. The TZ environmental variable is added to your CONFIG.SYS file.
Remote Printer Server	Specifies the host name of the LPD server. An environment variable called LPR_SERVER is added to your CONFIG.SYS file.
Remote Print Server's Printer	Specifies the name of the printer that is used by LPR. An environment variable called LPR_PRINTER is added to your CONFIG.SYS file.

Username for REXEC & Remote Printing	Specifies the name used to recognize REXEC and remote print requests submitted by other TCP/IP hosts to your PC. An environment variable called USER is added to your CONFIG.SYS file.
Password for REXEC Users	Specifies the password used to validate REXEC requests submitted by other TCP/IP hosts to your PC. An environment variable called PASSWD is added to your CONFIG.SYS file.
Password for Telnet	Specifies the name of the password that is used by the Telnet server. An environment variable called TELNET.PASSWORD.ID is added to your CONFIG.SYS file.
Domain Name Services	Specifies the host name, the domain name, and the name server addresses. A RESOLV file that is used by clients is created and an environment variable called HOSTNAME is added to your CONFIG.SYS.

Use the function keys that are displayed on the action bar at the top of the screen to make your screen selections for the FTP Access Protection, Domain Name Services, and X Host Authorization boxes.

The following buttons are displayed at the bottom of the window:

Selection	**Description**
Menu	Retains the current values, and redisplays the "TCP/IP Configuration Tool" window. The values are not saved until you exit from the "TCP/IP Configuration Tool" window.
Cancel	Discards any changes made and returns you to the "TCP/IP Configuration Tool" window.
F1=Help	Provides help information.

Configuring Routing Information

The following is an example of the "Configure Routing Information" window that is displayed as a result of selecting *Routing Information* from the "TCP/IP Configuration Tool" window.

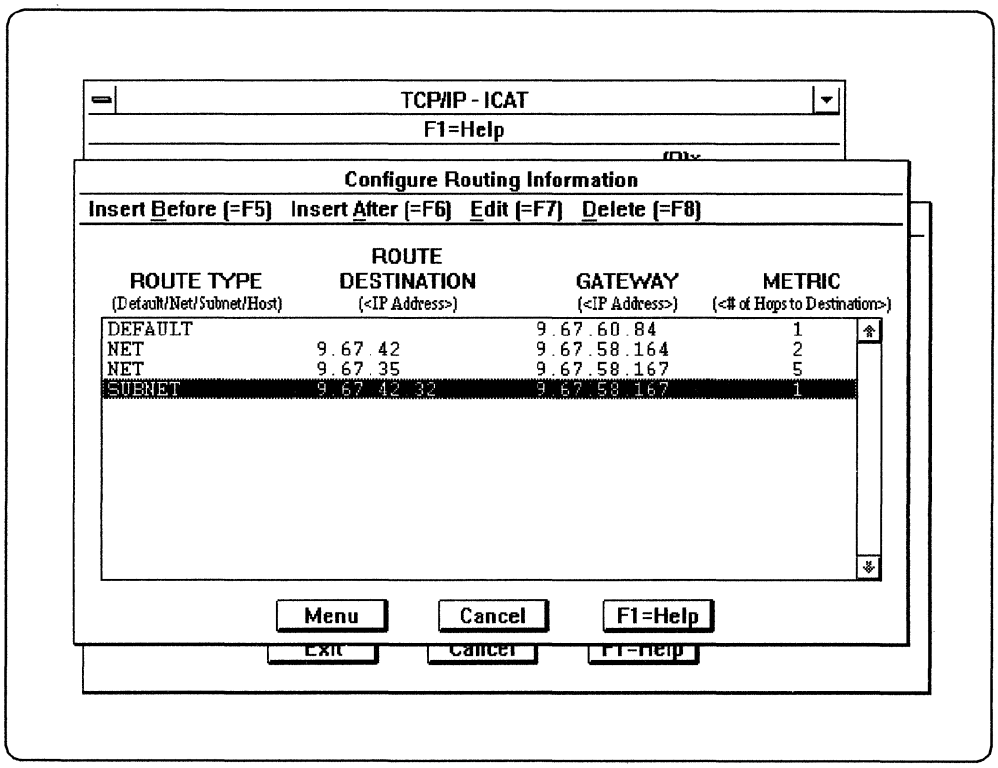


Figure 16. Configure Routing Information

The Configure Routing Information window is used to define routes that are placed in the SETUP.CMD file. See Appendix A, "Optional Files," or *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for the routing information you need to type in the entry fields of this window. An example of the routing entry's format is displayed at the top of the window.

To make your entries, use the function keys displayed at the top of the window to perform the desired function, or press the F10 key to go to the action bar and select the desired action.

Use the Tab key to move to the buttons at the bottom of the window.

The following button selections are displayed at the bottom of the window.

Selection	Description
Menu	Retains the current values, and redisplays the "TCP/IP Configuration Tool" window. The values are not saved until you exit from the "TCP/IP Configuration Tool" window.
Cancel	Discards any changes made and returns you to the "TCP/IP Configuration Tool" window.
F1=Help	Provides help information.

Starting TCP/IP for OS/2

When you exit the "TCP/IP Configuration Tool" window, the changes you made to your configuration are saved to the appropriate files.

You must reboot your system to activate the changes to your configuration. You should shut down all tasks before you reboot your system.

When the system restarts, TCP/IP is started from your CONFIG.SYS. If your configuration included the automatic starting of any services, your STARTUP.CMD calls TCPSTART.CMD, which starts the services you requested.

During the configuration process, several files were modified. Backup copies are created with file names ending in .BK*n*. For example, the first backup copy of the CONFIG.SYS file would be CONFIG.BK1.

Testing Your Installation

You can use the PING command to test your installation. PING sends out an ICMP echo request to a specified destination and waits for an echo reply. The receiver echoes the frames it receives back to the originator.

Enter the PING command from an OS/2 command prompt.

The following is the format of the PING command.

PING [-?] | [-d] [-r] [-v] *host* [*data_size* [*npackets*]]

The parameters for the PING command are:

Parameter	Description
-?	Displays help information.
-d	Starts the socket-level debugging process.
-r	Bypasses the routing table.
-v	Specifies verbose output.
host	Specifies the foreign host to which you want to send the echo request.
data_size	Sets the number of data bytes for the echo request (the default number of data bytes is 56, with an additional 8-byte header attached).
npackets	Sets the number of echo requests that are sent to the foreign host. These parameters are position dependent; you cannot specify the number of packets without specifying the data size. **Note:** If you do not specify *npackets*, the echo requests is sent continuously. Any of the following actions discontinues the echo request. • Pressing the Ctrl key and the C key simultaneously. • Pressing the Ctrl key and the Break key simultaneously. • Closing the task.

PING can be used to verify the following parts of the TCP/IP for OS/2 installation.

- PING *your_internet_address*

  This form of the PING command verifies that TCP/IP is running.

- PING *remote_internet_address*

  This form of the PING command verifies that you can be addressed.

- PING *remote_host_name*

  This form of the PING command verifies the operation of the name server or host file and whether it can reach that host.

You are now ready to use the applications and functions provided with TCP/IP for OS/2.

Removing TCP/IP for OS/2

This section describes how to remove TCP/IP for OS/2 from your PC.

Procedure

To remove TCP/IP for OS/2 from your PC, do the following steps.

1. Remove any TCP/IP related information from your CONFIG.SYS.

   Note: For CONFIG.SYS statements that contain references to multiple directories, such as SET PATH, LIBPATH and SET HELP, remove the reference to the directory in which TCP/IP is installed. Do not delete the entire statement.

2. If the file C:\STARTUP.CMD contains the line CALL TCPSTART.CMD, delete the line.

3. Save any configuration-related files in a directory other than the directory in which your current TCP/IP software resides. Configuration-related files include:

   - BIN\SETUP.CMD
   - BIN\TCPSTART.CMD
   - ETC\SENDMAIL.CF
   - BIN\SLIP.CMD
   - BIN\X25.CMD
   - The files listed in Appendix A, "Optional Files."

4. Reboot your machine.

5. Remove the directory structure in which TCP/IP is installed.

Chapter 4. Host Name Resolution

Chapter 4. Host Name Resolution

This chapter describes the files that are used for host name resolution. The files, which reside in the ETC directory, are:

- RESOLV
- HOSTS

This chapter also provides examples of the file content of the RESOLV file and the HOSTS file.

Overview of Name Resolution

When a TCP/IP for OS/2 service or application receives a symbolic name that represents a host address, it calls a host name resolver routine to resolve the symbolic name into an internet address. The host name resolver routine queries a domain name server or a local HOSTS file, or both, to perform the name resolution.

The resolver sends the request to the foreign name server before using the local HOSTS file. If a RESOLV file exists in the ETC directory, the host name resolver routine first tries to resolve the name through name servers specified in that file.

If resolution through a name server fails or if a RESOLV file does not exist, the host name resolver routine tries to resolve the name locally by searching the HOSTS file in the ETC directory for a match of the symbolic host name.

If a match is found, the routine returns the corresponding internet address. If a match is not found, a message is displayed stating that the host is unknown.

RESOLV File

The following example shows the format of the RESOLV file.

domain *domain_name*
nameserver *internet_address*

The following is an example of the content of a RESOLV file.

```
domain eng.mit.edu
nameserver 129.34.128.245
nameserver 129.34.128.246
```

Because the Domain Name Server requires a fully qualified domain name, the host name resolver routine scans the host name to see if it contains a period. If a period is not present, the routine appends the domain name specified by the *domain* statement in the RESOLV file. Otherwise, the name is assumed to be fully qualified and is passed *as is* to the name server.

You can have a maximum of one *domain* entry and three *nameserver* entries. The routine queries the first name server specified in the RESOLV file. If the specified name server does not respond, the routine queries the next name server specified, if any, until either a response is received or the last name server specified fails to respond.

If a name server responds with the internet address associated with the symbolic host name, that address is returned to the requesting service or application.

Note: For each name server that does not respond, a time-out of 60 to 80 seconds occurs.

HOSTS File

The following example shows the format of the HOSTS file.

internet_address host_name [*alias(es)*] [*# comment*] <*carriage return*>

When using the HOSTS file on a small internet, it is not necessary to use the hierarchical-naming system used by the larger internets. The following example is a token ring network of three users and their entries in the HOSTS file.

```
129.5.24.1 Host1 vjsPC PC1 mathdept
129.5.24.3 PC3 normasPC Host3 # This is Norma's PC
129.5.24.4 PC4 budsPC
```

You must enter a carriage return at the end of each line.

In this example, each time you enter the *host name* of `Host1` or the *aliases* `vjsPC`, `PC1`, or `mathdept`, the local name resolver translates it to the internet address of `129.5.24.1`. For more information about the format of network addresses, see "Network Address Format" on page 14.

Note: When both the RESOLV and HOSTS files exist in the ETC directory, you should keep the HOSTS file up to date. As name servers are modified, the HOSTS file can become outdated.

Chapter 5. Manually Modifying Your TCP/IP Configuration

Chapter 5. Manually Modifying Your TCP/IP Configuration

TCP/IP for OS/2 allows you to change your TCP/IP configuration to suit your specific requirements.

This chapter describes an alternative to using ICAT to modify your TCP/IP configuration.

Configuring the Network Interface

The IFCONFIG command assigns an address to a network interface and also configures network interface parameters.

You must use the IFCONFIG command to define the network address of each interface present on the machine.

You can also use the IFCONFIG command to redefine an interface's address or other operating parameters.

Warning: Do not attempt to modify the configuration of a network interface unless you are an experienced TCP/IP user.

The following example shows the format of the IFCONFIG command.

```
IFCONFIG interface [[af] [address [dest_address]] [up | down]
        [netmask mask]] [metric n] [mtu n]  [trailers | -trailers]
        [arp | -arp] [bridge | -bridge] [snap | -snap] [-allrs]
        [broadcast broadcast_address] [802.3 | -802.3]
        [icmpred | -icmpred]
```

The parameters of the IFCONFIG command are:

Parameter	Description
interface	The name of the interface you are configuring (lan0, lan1, lan2, lan3, sl, or x25).
af	Name of the address family supported. You must specify the address family (af), because an interface can receive transmissions in different protocols, and each protocol requires a separate naming scheme. However, specifying the address family can change the interpretation of the remaining parameters. Specify only `inet`, which is the default.
address	The address assigned to a particular interface in the standard dotted-decimal notation.
dest_address	Specifies the address of the correspondent on the receiving end of a point-to-point link.
up	Enables an interface after the interface has been marked down with an IFCONFIG statement. Interfaces are automatically marked up when the first address is set on an interface.

down

Marks an interface down. When an interface is marked down, the system does not attempt to transmit messages through that interface. In some cases, the reception of messages is also disabled.

This action does not automatically disable routes using the interface.

netmask *mask*

This parameter is used for networks only. The *mask* value specifies how much of the internet address to reserve for use as a subnetwork address.

For example, the subnetwork capability of TCP/IP divides a single network into multiple logical networks. An organization can have a single internet network address that is known to users outside the organization, yet configure its internal network into different departmental subnets.

The subnet, or local address, portion of an internet address is then divided into a subnet number and a host number, for example:

network_number subnet_number host_number

where:

network_number Is the network portion of the internet address.

subnet_number Is the subnet number portion of the local address.

host_number Is the host number portion of the local address.

The *mask* value includes the network portion of the local address and the subnet portion, which is taken from the host field of the address. The *mask* can be specified as a single hexadecimal number with a leading 0x, or with a dotted-decimal notation address.

The *mask* contains 1s for the bit positions in the 32-bit address that are to be used for the network and subnet parts, and 0s for the host part. The *mask* should contain at least the standard network portion, but the bits of the netmask do not have to be contiguous. The subnet field should be contiguous with the network portion.

For an example of the ROUTE command with the subnet parameter, see "ROUTE—Modifying Routing Tables" on page 50.

metric *n*

Sets the metric for the interface to *n*. The value *n* represents a number and should be between 0 and 15. The default is 0 (directly connected). The routing metric is used by the Routing Information Protocol (RIP).

Metrics that are greater in value make a route less favorable. Metrics are counted as the number of hops to the destination network or host.

mtu *n*

Sets the maximum transmission unit of the interface to *n*. The value *n* represents a number. The default MTU value is 1500.

Notes:

1. When using a PCNet adapter the MTU should be set to a maximum of 1462.
2. When using an Ethernet adapter on an IEEE 802.3 network, the MTU should be set to a maximum of 1492.
3. When using a Token Ring 16/4 Adapter/A card on a 16 megabyte token ring, the MTU should be set to a maximum of 4400.
4. When using an X.25 co-processor adapter, the MTU should be set to a maximum of 576.

trailers

Requests the use of a trailer link level encapsulation when sending.

For example, if a network interface supports trailers, the system, when possible, encapsulates outgoing messages, which minimize the number of memory-to-memory copy operations that the receiver must perform.

On networks that support the Address Resolution Protocol (ARP), this parameter indicates that the system should request that other systems use trailers when sending to this host. Trailer encapsulations are sent to other hosts that have made such requests.

-trailers

Disables trailer link level encapsulation. This is the default.

arp

Enables ARP in mapping between network level addresses and physical, or station addresses.

ARP is currently implemented for mapping between internet addresses and Ethernet addresses or IBM token ring addresses.

-arp

Disables Address Resolution Protocol.

bridge

Enables routing field support.

-bridge

Disables routing field support.

snap

Sends token ring headers with the extended snap format. This is the Institute of Electrical and Electronic Engineers (IEEE) standard and is necessary to communicate with machines using the extended snap format, such as AIX*. The snap parameter is the configuration default.

-snap

Does not send token ring headers with the extended snap format.

-allrs

Sets the token ring broadcast indicator to Single-Route Broadcast. The default is All-Routes Broadcast. See *IBM OS/2 LAN Technical Reference* for more information.

broadcast *broadcast_address*

Specifies the address to use to represent broadcasts to the network. The default *broadcast address* is an internet address with a local address that has a value of all 1s.

802.3	Enables Ethernet 802.3
-802.3	Disables Ethernet 802.3. Enables Ethernet DIX 2. This is the default.
icmpred	Allows TCP/IP to add routes obtained by the ICMP redirects. This is the default.
-icmpred	Prevents TCP/IP from adding routes obtained by ICMP redirects.

The IFCONFIG command displays the current configuration for a network interface when only an interface is supplied. If a protocol family is specified using *af*, IFCONFIG reports only the details specific to that protocol family.

To receive help for the command syntax, use the IFCONFIG command alone, without specifying an interface, address, or parameter.

ROUTE—Modifying Routing Tables

Use the ROUTE command to manually configure the network routing tables.

Warning: Do not attempt to configure the network routing tables unless you are an experienced TCP/IP user.

The following example shows the format of the ROUTE command.

ROUTE [?] [-f] [-h] [{add | delete} {net | subnet | host} {*destination* | default} *router* [*metric*]]

The parameters of the ROUTE command are:

Parameter	Description
?	Displays help information.
-f	Flushes the routing tables of all network and subnet route entries. If this is used in conjunction with another parameter, the tables are flushed before the parameter's application.
-h	Flushes the routing tables of all host route entries. If this is used in conjunction with another parameter, the tables are flushed before the parameter's application.
add	Adds a route.
delete	Deletes a route.
net	Specifies that a network is to be added or deleted.
subnet	Specifies that a subnet is to be added or deleted.
host	Specifies that a host destination is to be added or deleted.
destination	Specifies the internet address of the destination host, network, or subnet.
default	Specifies all destinations not defined with another routing table entry.

router Specifies the internet address of the next hop in the path to the destination.

metric Specifies the number of hops to the destination.

Note: *Metric* is required for add commands.

The following is an example of a ROUTE command.

```
ROUTE -fh add net 129.34.10.0 129.34.10.60 1
```

In the example, the ROUTE -fh command clears the routing table of all entries and adds a route to the network 129.34.10.0 through the router 129.34.10.60, specifying the number of hops to the destination host as 1.

The following is the response that is displayed as a result of issuing the ROUTE -fh command illustrated in the preceding example.

```
add net 129.34.10.0: router 129.34.10.60
```

You can use the NETSTAT -r command to display the current routing tables.

The subnet parameter is used in place of net, when using a netmask that subnets down into the fourth byte of an address.

For example, if you have the address 192.67.59.*nn*, where *nn* represents any number, and you need to support at least 4 networks with no more than 62 hosts for each net, use a netmask of 255.255.255.192. The following example illustrates the ROUTE command with the subnet parameter.

```
ROUTE add subnet 192.67.59.64 192.67.59.66 1
```

In the example, any packets addressed for the address range 192.67.59.65 through 192.67.59.126 are sent to 192.67.59.66, which is one hop away.

Chapter 6. Manually Setting Up the TCP/IP Servers

Chapter 6. Manually Setting Up the TCP/IP Servers

This chapter describes how to manually customize TCP/IP for the OS/2 environment.

Only one instance of each server should be run on a PC at one time. If you try to start a second server of the same type, a message is displayed informing you that the address is already in use. For example, if you have an FTP server running, you cannot start a second FTP server on your PC.

INETD

This section describes how to set up the INETD server. INETD is a super server that allows you to start multiple servers from a single OS/2 session, and use the applicable server when needed. INETD supports the following servers:

- FTPD
- LPD
- REXECD
- RSHD
- TELNETD
- TFTPD

You can use INETD as an alternative to starting each individual server. However, you cannot specify the parameters used with LPD and TFTPD.

Warning: Do not attempt to activate servers by two methods. If you use INETD, which can start multiple servers, do not attempt to start an individual server by using a specific server command, such as FTPD.

For example, if you have used INETD to start FTPD, do not attempt to start FTPD with the FTPD command also.

Setting Up the Environment

Before using the INETD command, you must create or modify the INETD.LST file, and the file must reside in the ETC directory, or the directory specified by the ETC environment variable.

INETD.LST File

The INETD.LST file is used by the INETD server to define the servers that are to be activated.

The INETD.LST file contains one or more entries. The following is an example of the content of an INETD.LST file containing multiple entries. For the file format, see Appendix A, "Optional Files."

```
telnet    tcp telnetd
exec      tcp rexecd
ftp       tcp ftpdc
printer   tcp lpd
tftp      udp tftpd
shell     tcp rshd
```

Where:

telnet is the application, tcp is the protocol, as defined in the services file, and telnetd is the server to be activated.

exec is the application, tcp is the protocol, as defined in the services file, and rexecd is the server to be activated.

Note: If the TFTP server is started without using INETD, its file access path can be set as a parameter to the command. No such parameter can be included in the INETD.LST file, but the file access path can be specified by the environmental variable TFTPDPATH. For example:

```
SET TFTDPATH=C:\TEMP\
```

Comment characters and blank lines are not allowed in the INETD.LST file. INETD will not start if it discovers any blank lines.

Setting Up the INETD Server

A server is required on one of the hosts involved in the transfer of data.

To start the server on your local host, type INETD at an OS/2 command prompt, and press the Enter key.

INETD starts the INETD.EXE program. INETD.EXE runs as a task until you shut down the server.

FTP

This section describes how to set up the environment and server for FTP.

Setting Up the Environment

Several different files can be used by FTP to enable or automate various functions. These files are:

- TRUSERS
- NETRC

These files are created by the user and must reside in the ETC directory, or the directory specified by the ETC environment variable.

TRUSERS File

The TRUSERS file is used by the FTP server to define access authorization to users on the foreign host. You can define users with read access to particular directories and write access to particular directories.

The following is an example of the content of a TRUSERS file containing multiple entries. For the file format, see Appendix A, "Optional Files."

```
user:  chris boz
rd:    d:\ c:\
wr:    d:\tmp c:\tmp

user:  anonymous
rd:    c:\anonymous
wr:

user:  diane green
wr^:   c:\etc
```

Where:

`chris` is the FTP user, `boz` is the password for chris, `rd: d:\ c:\` gives chris access to read files and subdirectories in the c:\ and d:\ subdirectories, and `wr: d:\tmp c:\tmp` gives chris access to write to files and subdirectores only in the c:\tmp or d:\tmp directories.

`anonymous` is defined as the user with no password. This user name has special meaning because you are not required to define a password. This is the only user name you can define without a password. This user can read files and directories in c:\anonymous but cannot write to any files or directories.

`diane` is the user, `green` is the password for diane, and `wr^: c:\etc` gives diane access to read or write to any file or directory except c:\etc. This also gives diane access to read or write to any other drives, floppy and mounted.

Warning: Use discretion in giving write access to other users. A remote user with write access can destroy files and directories on your PC.

NETRC File

The NETRC file is used by the FTP (and REXEC) clients as a source for *user* and *password* values. Create the NETRC file in the ETC directory. For the file format, see Appendix A, "Optional Files."

The following is an example of the content of a NETRC file containing multiple entries.

```
machine raleigh login kent password baseball
machine boston login bruce password september macdef mymacro
bell
hash
prompt
binary
cd c:\mydir
get myfile.bin

machine york login jane password workday account payday
```

In this example, when using the FTP command on the local host to connect to the `raleigh` host, the user `kent` and the password `baseball` are automatically sent to the FTP server on the foreign host. To allow the user `kent` access to the FTP server, the value of the password `baseball` must also be the password specified for the user `kent` on the foreign host running the FTP server.

If a user uses FTP to open a connection to `machine boston`, the user name `bruce` and the password `september` are automatically passed to the FTP server on the other end of the connection. Also, the macro called `mymacro` is defined from the line following the `macdef mymacro`, until a null line is encountered. To execute the macro, `mymacro`, type $mymacro at the FTP command shell.

Warning: If you have a Telnet, REXEC, TFTP, RSH, or FTP server running on your machine, be aware that a NETRC file provides users with *user* and *password* information that may allow them access to other users' files.

Setting Up the Server

A server is required on one of the hosts involved in the transfer of data.

To start the server on your local host, type FTPD at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the FTPD command.

FTPD

There are no parameters for the FTPD command.

As an alternative, you can start this server using INETD. INETD allows you to start multiple servers from an OS/2 session.

FTPD starts the FTPD.EXE program. FTPDC.EXE runs as a task until you shut down the server.

LPD

This section describes how to set up the LPD server.

Setting Up the Environment

All of the Line Printer commands require the host name of the server providing the print spooling services, as well as the printer name, and in some cases a user name.

You can set default values for all of these by defining the following environment variables. You can also override the values for printer name and host name by using the -p and -s parameters. See the format of the specific Line Printer command you want to use.

The environment variables and their associated parameters used to set default values for the Line Printer commands are:

Environment Variable	Parameter
LPR_PRINTER	*printer*
LPR_SERVER	*server*
USER	*username*

The following is a description of the parameters.

Parameter	Description
printer	Name of the printer that provides the output. When connecting to an OS/2 LPD server, the printer name corresponds to a queue defined in the Print Manager of the server machine. The user can also specify a device name. The LPD server tries to determine the associated queue as defined in the Print Manager.
server	Internet address or name of the host that provides the print spooling service.
username	Character string passed to the host that provides the print spooling services as an identifier of who created the print job.

Setting Up the Server

The Line Printer commands use the LPD server for remote printing.

Before using a Line Printer command, the LPD server must be running on the foreign host that provides the print spooling service.

To start the server on your local host, type LPD at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the LPD command.

```
LPD -? | {[-c] [-b] [-s]}
```

The parameters of the LPD command are:

Parameter	Description
-?	Displays help information.
-c	Prevents printing of the control file.
-b	Prevents printing of the banner page.
-s	Causes LPD to validate client requests based on the port addresses. According to RFC 1179, all line printer requests should come from clients on a port within the range of 721 to 731. However, because some clients do not support this, the default does not verify that the client is connecting from a valid port within this range.

As an alternative, you can start LPD using INETD. INETD allows you to start multiple servers from an OS/2 session.

LPD.EXE runs as a task until you shut down the server.

Entering the LPRMON Command

LPRMON is a Parallel Device Monitor that allows you to set up your PC to automatically send data to a remote LPR server. This allows you to print to an LPR server without an application using the Line Printer protocol directly.

LPRMON can work with the OS/2 Print Manager. If you defined a queue associated with a parallel port for which LPRMON is going to monitor, then the Print Manager first queues the request, at which time you can manage the queue using the Print Manager interface before sending the request over a TCP/IP network to a remote LPR server. If a printer is not set up in the Print Manager for the parallel port that LPRMON is going to monitor, then the data is sent directly to the LPR server without going through the Print Manager.

If you wish to use the Print Manager and LPRMON together, configure the Print Manager to be consistent with the type of printer on the remote LPD server where the printing actually occurs. This is done by installing the appropriate driver connected to the parallel port that LPRMON is going to monitor.

For example, if the remote printer is an IBM 4019 Laser Printer in PostScript** mode and LPRMON is started to monitor LPT3, the IBM 4019 printer should be installed in the Print Manager as connected to LPT3 and the PostScript printer driver should be loaded for the IBM 4019 printer.

If there is no equivalent driver for the remote printer available for installation in the Print Manager, you should configure the port as being connected to the IBMNULL printer and driver.

Note: You should use the -b option unless the remote LPD printer is strictly a text printer (the printer does not support embedded binary control characters).

The following example shows the format of the LPRMON command.

```
LPRMON -? | {[-b] [-f] [-n] [-r n] [-q n] [-p printer] [-s server] devicename}
```

The parameters of the LPRMON command are:

Parameters	Description
-?	Displays help information
-b	Specifies that the data is interpreted as binary by the LPR server LPD.
-f	When the print server is running on a UNIX system, the -f parameter formats the file using the UNIX PR command. When the print server is running under OS/2, LPD passes the file through unaltered.
-n	Disables the beep that occurs when there is an error.
-r *n*	Sets the number of retries. The default is 3 tries.
-q *n*	Sets retry delay in seconds. The default is 10 seconds
-p *printer*	Specifies the name of the printer to which the file is sent. If you omit the -p parameter, LPRMON looks at the environment variable, LPR_PRINTER, for the corresponding value.
-s *server*	Specifies the name or internet address of a network host with print spooling capabilities. If a print server is not specified on the command line, LPRMON looks at the environment variable, LPR_SERVER, for the corresponding value and uses that value as the print server. If a print server is not specified with the LPRMON command or defined in the environment variable, LPRMON displays an error message and terminates.
devicename	Specifies the parallel port for LPRMON to monitor. Data sent to this port is then redirected to a remote LPR server. This should be specified as LPD *n*, where *n* is a number between 1 and 3.

PMX

This section describes how to start the X Window System Server (X server).

Starting the X Server

Start the X server from a PM group menu, from an OS/2 command line, or from a batch (CMD) file.

The following example shows the format of the PMX command.

```
PMX [parameters...]
```

The parameters for the PMX command are:

Parameter	Description
-a *n*	Sets mouse acceleration (*n* is the number of pixels).
-co *filename*	Sets color database file name. The default filename is X11\RGB.TXT.
-fc *fontname*	Sets cursor font. The default is the font named `cursor`.
-fn *fontname*	Sets default font. The default is the font named `fixed`.
-fp *pathname*	Sets default font path. The default is X11\MISC,X11\75DPI
-help	Displays help information (will not start the X server).
-lc	Doubles the dimensions of any cursor, unless the cursor would become too large for a PM cursor.
-nocopyright	Does not display initial copyright window when starting the server.
-r	Turns off the keyboard auto-repeat function.
r	Turns on the keyboard auto-repeat function. This is the default.
-t *n*	Sets mouse threshold (*n* is the number of pixels).
-to *n*	Sets connection time-out (*n* is the number of seconds).
`-I`	Ignores all remaining arguments.

In the case of the font path (-fp *pathname*), multiple directories are separated by commas, rather than blanks. For example, to start the server with both the miscellaneous and the 75 dot per inch (dpi) font directories, as well as your own personal fonts directory C:\myfonts, specify the following command parameters:

```
pmx -fp d:tcpip\X11\misc,d:tcpip\X11\75dpi,c:\myfonts
```

This example assumes that you installed TCP/IP on drive d: in the \tcpip directory.

For more information about X fonts, see "X Font Support" on page 148. For more information about the color database and font files, see "The Color Database (RGB.TXT)" on page 147 and "How the OS/2 X Server Accesses Fonts" on page 149.

Portmap

This section describes how to set up the Portmap server.

Setting Up the Server

The Portmapper program maps client programs to the port numbers of server programs. Portmap is used with Remote Procedure Call (RPC) programs. See *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for additional information about Portmap and RPC.

To start the PORTMAP server on your local host, type PORTMAP at an OS/2 command prompt, and press the Enter key. PORTMAP.EXE runs as a task until you shut down the server.

REXEC

This section describes how to set up the environment and server for REXEC.

Setting Up the Environment

Before you activate the REXEC server, set the environment variables USER and PASSWD. These environment variables define the user's ID and password, which a remote user specifies to log on to your PC. The values for these environment variables are case sensitive.

Examples of the USER and PASSWD environment variables are:

```
SET USER=user
SET PASSWD=password
```

These environment variables can be set in your CONFIG.SYS file, or they can be typed in a command shell before starting the REXEC server. Using a command shell has the advantage that only the command shell and any functions running in it have knowledge of the USER and PASSWD variables.

NETRC File

The NETRC file is used by the REXEC (and FTP) client as a source for *user* and *password* values. Create the NETRC file in the ETC directory. For the file format, see Appendix A, "Optional Files."

The following is an example of the content of a NETRC file containing multiple entries.

```
machine raleigh user kent password baseball
machine boston user bruce password september
machine 251.1.11.3 user jane password workday
```

Note: The account and macdef parameters are not used by the REXEC server.

In this example, when you issue the REXEC command to machine `raleigh`, the user name `kent` and password `baseball` are automatically sent to the foreign host. The foreign host called `raleigh` has a REXEC server running and has the user `kent` with the password `baseball` defined.

Warning: If you have a Telnet, REXEC, TFTP, RSH, or FTP server running on your machine, be aware that a NETRC file provides users with *user* and *password* information that may allow them access to other users' files.

Setting Up the Server

To use the REXEC command, the REXECD server must be running on the foreign host.

To start the server on your local host, type REXECD at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the REXECD command.

REXECD

There are no parameters for the REXECD command.

As an alternative, you can start REXECD using INETD. INETD allows you to start multiple servers from an OS/2 session.

REXECD starts the REXECD.EXE program and runs as a task until you shut down the server.

Security can be an issue when the REXEC server is running. If a remote user learns the name of the user and password on your system, that remote user can execute commands on your PC.

Warning: Use discretion in allowing remote users to learn your USER and PASSWD environment variables.

RSH

This section describes how to set up the environment and server for RSH.

Setting Up the Environment

Before you activate the RSH server, create the RHOSTS file in the ETC directory.

RHOSTS File

The RHOSTS file is used by the RSH server to verify the authorization of remote hosts. You must specify the full domain name of the remote hosts. The RHOSTS file must be in the ETC directory, as defined by the ETC environmental variable.

The following is an example of the content of an RHOSTS file containing multiple entries.

```
shofert.raleigh.ibm.com
agusta.raleigh.ibm.com
yazzo.watson.ibm.com
```

Warning: If you have a Telnet, REXEC, TFTP, RSH, or FTP server running on your machine, be aware that a NETRC file provides users with *user* and *password* information that may allow them access to other users' files.

Setting Up the Server

To use the RSH command, the RSHD server must be running on the foreign host.

To start the server on your local host, type RSHD at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the RSHD command.

```
RSHD
```

There are no parameters for the RSHD command.

As an alternative, you can start RSH using INETD. INETD allows you to start multiple servers from an OS/2 session.

RSHD starts the RSHD.EXE program and runs as a task until you shut down the server.

Security can be an issue when the RSH server is running. If a remote user learns the host names in the RHOSTS file, that remote user can execute commands on your PC.

ROUTED

This section describes how to set up the environment and server for ROUTED.

Setting up the Environment

The ROUTED server queries the network and dynamically builds routing tables from routing information transmitted by other hosts that are directly connected to the network. The ROUTED server implements the Routing Information Protocol (RIP). See RFC 1058 for more information. The gateways file is used to configure the ROUTED server, and must reside in the ETC directory.

An `active` entry is used to add routes to RIP routers that cannot be reached by normal IP broadcasts. A `passive` entry in the gateways file is used to add a route to a part of the network that does not support RIP. An `external` entry in the gateways file indicates a route that should never be added to the routing tables. If another RIP server offers this route to your host, the route is discarded and not added to the routing tables.

The following example shows the line format for the gateways file.

```
{net | host} name1 gateway name2 metric value {active | passive | external}
```

The following is a description of each element in a gateways file entry.

Element	Description
net	Specifies that a network is to be added or deleted.
host	Specifies that a host destination is to be added or deleted.
name1	Can be either a symbolic name or the internet address of the destination network or host.
gateway	Specifies a gateway or router. The parameters that follow this keyword identify the gateway or router for this destination.
name2	Can be either a symbolic name or the internet address of the gateway or router for this destination.
metric	Specifies the number of hops to the destination host or network. The value that follows this keyword is the metric hop count.

value — Specifies the hop count to the destination. The number is an integer in the range of 0 through 16, where 16 (infinity) indicates the network cannot be reached.

active — Specifies the type of gateway. An active gateway is treated like a network interface. It is expected to exchange routing information, and if it does not do so for a period of time, the route associated with this gateway is deleted. Information about the active gateway is maintained in the routing tables indefinitely and is included in any routing information that is transmitted.

passive — Specifies the type of gateway. A passive gateway does not exchange routing information. Information about the passive gateway is maintained in the local routing tables indefinitely and is only local to this ROUTED server. Passive gateway entries are not included in any routing information that is transmitted.

external — Specifies the type of gateway. An external gateway option indicates that entries for this destination should never be added to the routing table. The ROUTED server discards any routes for this destination that it receives from other ROUTED servers. Only the destination field is significant; the gateway and metric fields are ignored.

Note: The keywords in the gateways file are case-sensitive and must be entered in lowercase.

The following example shows the contents of a gateways file containing multiple entries:

```
host joespc        gateway  192.9.201.5       metric 4  active
net  acmenet       gateway  gateway.acme.com  metric 5  passive
host vm3.ibm.com   gateway  9.67.43.126       metric 6  passive
host bad.host      gateway  xxx               metric 1  external
```

In the first entry, the route identified goes to a specific host, joespc, through a gateway 192.9.201.5. The hop count metric to joespc is 4, and the gateway is treated as active.

In the second entry, the route indicates that acmenet can be reached through the gateway gateway.acme.com, and that it is 5 hop counts away.

In the third entry, the route indicates that vm3.ibm.com can be reached through the gateway 9.67.43.126, and that it is 6 hop counts away.

In the fourth entry, the external gateway option indicates that routes for the host bad.host should not be added to the routing tables, and that routes received from other ROUTED servers for bad.host should not be accepted.

Setting Up the Server

The following steps describe how to set up the ROUTED server.

To start the server on your local host, type ROUTED, with the desired parameters specified, at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the ROUTED command.

```
ROUTED [-d] [-g] [-s] [-q] [[-t] | [-t -t] | [-t -t -t] | [-t -t -t -t]]
```

The parameters of the ROUTED command are:

Parameters	Description
-d	Enables additional debugging information to be logged, such as bad packets received.
-g	Offers a route to the default destination. This is typically used on a gateway to an internet, or on a gateway that uses another routing protocol, whose routes are not reported to other local gateways.
-s	Forces the ROUTED command to supply routing information regardless of whether it is acting as an internetwork router. This is the default if multiple network interfaces are present, or if a point-to-point link is in use.
-q	Suppresses broadcasting of routing information.
-t	Starts the packet tracing process.
-t -t	Starts the packet tracing process and traces all packets sent or received on the standard output.
-t -t -t	Starts the packet tracing process, traces all packets sent or received on the standard output, and starts the history tracing.
-t -t -t -t	Starts the packet tracing process, traces all packets sent or received on the standard output, starts the history tracing, and starts tracing the packet contents.

Notes:

1. The parameters for the ROUTED command are case sensitive and must be entered in lowercase.
2. You must enter a space between each -t parameter when entering multiple -t parameters.

ROUTED starts the ROUTED.EXE program and runs as a task until you shut down the server.

Sendmail

This section describes how to set up the Sendmail environment.

Setting Up the Sendmail Environment

The following steps describe how to set up the Sendmail environment. If you use the Installation and Configuration Automation Tool (ICAT) program, the Sendmail environment is set up for you.

1. SENDMAIL.CF is the configuration file for Sendmail. The SENDMAIL.CF file must reside in the ETC directory or the directory specified by the ETC environment variable.

   The SENDMAIL.CF file is updated with your specific installation requirements during the ICAT installation process for TCP/IP for OS/2.

   Make the following changes to the SENDMAIL.CF file:

   - Change the Dw and Cw parameters to reflect your host name.
   - Change the DD parameter to reflect your domain.

     Note: Delete the line concerning the DD parameter if you do not have a domain name.

2. The following subdirectories must reside in the ETC directory or the directory specified by the ETC environment variable.

MAIL	The MAIL subdirectory; incoming mail is stored in this directory.
MQUEUE	The MQUEUE subdirectory; outgoing mail and temporary files are stored in this directory.

   Ensure that the MAIL and MQUEUE subdirectories have been created.

The following example shows the format of the SENDMAIL command.

```
SENDMAIL -bd -qtime [-d | -d1.1]
```

The parameters of the SENDMAIL command are:

Parameter	Description
-bd	Starts Sendmail as a server.
-q*time*	Specifies how often the mail queue should be processed. *time* should be entered in the format of *number*, *letter*. The letter s = seconds, m = minutes, h = hours, d = days, and w = weeks. For example: `-q30m` specifies every 30 minutes `-q1h30m` specifies every hour and 30 minutes. **Note:** A recommended time value is 30 minutes (-q30m).

-d Causes detailed debug information to be written to the Sendmail console, and creates a SENDMAIL.LOG file in the ETC directory that contains the SMTP transactions between the Sendmail server and the foreign SMTP server.

-d1.1 Causes only the SENDMAIL.LOG file to be created in the ETC directory.

For example, to start the Sendmail server with the detailed debug information parameter, use:

```
[C:/]sendmail -bd -q30m -d
```

If you attempt to send mail to a host that is not up and running, Sendmail stores (queues) the mail in the MQUEUE subdirectory and continuously resends the mail after the specified time interval, until the mail is successfully sent.

After the SENDMAIL program starts, a status message is displayed confirming that the program started correctly.

SENDMAIL.EXE should be run continuously to allow you to send and receive mail. To run Sendmail continuously, initiate SENDMAIL.EXE when you bring up the system.

Note: If you stop the Sendmail program while it is sending or receiving mail, files can be stranded in the MQUEUE directory. Periodically check the MQUEUE directory, and delete old files.

Using MX Records

MX records direct the SMTP server to deliver mail to alternative hosts. MX records are obtained from the Domain Name Server. If SMTP is not using a name server, then MX records are not used.

For example, if SMTP wants to send mail to USER@HOST, it checks the name server for MX records and finds the following:

```
HOST    MX    0    HOST
HOST    MX    5    HOST-BACKUP1
HOST    MX    10   HOST-BACKUP2
```

SMTP delivers the mail to the record (host) that has the lowest count, in this example, directly to HOST. If HOST is unable to receive the mail, SMTP then tries to deliver it to HOST-BACKUP1. If HOST-BACKUP1 cannot receive the mail, it tries HOST-BACKUP2. If none of the hosts can receive the mail, SMTP stores the mail and queues it for later delivery, at which time the process is repeated.

If SMTP does not find MX records for a host, it delivers mail only to the primary host.

For more information about MX records, see RFC 974.

Talk

This section describes how to set up the environment and server for Talk.

Setting Up the Environment

To use the TALK command, the originating machine:

- Must have the HOSTNAME environment variable defined.
- The host name and the destination host must be defined on an accessible name server or in the HOSTS file.

In addition, the originating host name must also exist in the destination host's name server or HOSTS file. If the destination host cannot resolve the originating host name, the following message is generated to the originating host:

```
Target machine does not recognize us.
```

Setting Up the Server

To use the TALK command, the TALKD server must be running on ***both*** the local and foreign hosts, to exchange TALK messages.

To start the server on your local host, type TALKD at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the TALKD command.

```
TALKD
```

There are no parameters for the TALKD command.

TALKD starts the TALK.EXE program and runs as a task until you shut down the server.

Telnet

This section describes how to set up the environment and server for Telnet.

Setting Up the Environment

The Telnet server uses Dynamic Link Library (DLL) files to implement the supported terminal types. You must specify the path where the DLL files that are used with Telnet reside. This path is specified using the LIBPATH statement in your CONFIG.SYS file.

The DLL files are:

- VT100.DLL
- ANSI.DLL
- DUMB.DLL

The TELNET.PASSWORD.ID environment variable contains the required password for the Telnet server. The TELNET.PASSWORD.ID environment variable can be set in your CONFIG.SYS file or from an OS/2 command prompt. During login by a

Telnet client, the client user is prompted for a password, which is the value of TELNET.PASSWORD.ID. The client user must type the value of TELNET.PASSWORD.ID as the password to access the Telnet server machine.

Note: If you make changes to your CONFIG.SYS file, you must reboot your PC to activate the changes.

Setting Up the Server

To allow Telnet logins to your PC using TCP/IP for OS/2, you must have a Telnet server running on your PC. You must also set the environment variable TELNET.PASSWORD.ID to a value that connecting Telnet clients are required to enter during login. For more information about TELNET.PASSWORD.ID, see "Setting Up the Environment" on page 70.

To start the server on your local host, type TELNETD at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the TELNETD command.

```
TELNETD
```

There are no parameters for the TELNETD command.

As an alternative, you can start TELNETD using INETD. INETD allows you to start multiple servers from an OS/2 session.

TELNETD starts the TELENETD.EXE program and runs as a task until you shut down the server.

A separate task, Telnet Session, is displayed in the "Task List" window for each client that establishes a connection with the Telnet server.

Note: TELNETD uses functions of OS/2 that support full-screen sessions only. As a result, the remote logon client must only run full-screen applications. Presentation Manager applications cannot be executed remotely.

TFTP

This section describes how to set up the TFTP server. Restricting access to files is also described. The TCP/IP for OS/2 program is implemented with both client and server support for TFTP.

To use TFTP, a server must be running on the foreign host.

Setting Up the Server

To start the server on your local host, type TFTPD at an OS/2 command prompt, and press the Enter key.

The following example shows the format of the TFTPD command.

```
TFTPD
```

There are no parameters for the TFTPD command.

As an alternative, you can start TFTPD using INETD. INETD allows you to start multiple servers from an OS/2 session.

TFTPD starts the TFTPD.EXE program and runs as a task until you shut down the server.

Restricting Access to Files or Directories

The following example shows the format of the TFTPD command to specify access to a particular path.

```
TFTPD [path]
```

The only parameter of the TFTPD command is:

Parameter	Description
path	Specifies the path for which you are granting access to the TFTP client.

The following example shows the format of the TFTPD command specifying access for TFTP clients to the \TEMP directory on the C drive.

```
TFTPD C:\TEMP\
```

The value of the parameter is used as a prefix for all file names specified by the PUT and GET subcommands. If you request a GET or PUT operation, specifying a file name of xxx.aaa, the resulting GET or PUT is for the file xxx.aaa in the \TEMP directory on the C drive.

If you specify the TFTPD command, as shown in the following example, and request a GET or PUT operation for the file xxx.aaa, the resulting GET or PUT would be for the file TEMPxxx.aaa on the C drive.

```
TFTPD C:\TEMP
```

If you start the TFTPD server with the TFTPD *path* parameter, all TFTP clients are restricted to the specified path. The TFTP clients do not have access to files on any other path.

Chapter 7. Installing and Customizing Network Management Components

Chapter 7. Installing and Customizing Network Management Components

This chapter describes how to install and customize the components of TCP/IP for OS/2 that are used to manage your TCP/IP network.

Overview of Network Management Functions

TCP/IP for OS/2 provides the following functions to assist you in managing your TCP/IP Network.

- Simple Network Management Protocol (SNMP) client is the network manager.
- SNMP Agent is the network element being managed.
- The Network Monitor (PMPING), which continuously verifies that a list of hosts can be reached using echo requests (PING).

A complete SNMP configuration involves both the OS/2 SNMP client and the OS/2 SNMP agent. In addition, other IBM and non-IBM SNMP clients and agents may need to be configured. For an overview of the SNMP and PMPING, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide.*

Summary of Commands

The commands that are available for each function are:

- SNMP Client
  - SNMPGRP to retrieve a group of management information variables
  - SNMP GET to retrieve a single management information variable
  - SNMP NEXT to retrieve a single management information variable from a table
  - SNMPTRAP to receive unsolicited notification of network events from SNMP agents
- SNMP Agent
  - SNMPD to start the SNMP agent
- Network Monitoring
  - PMPING to display the status of a group of user defined hosts.

For a description of these commands and information about their usage, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide.*

Other network management functions that are provided with OS/2 TCP/IP include ARP, FINGER, NETSTAT, and PING. These functions do not require any customization. For a description of these functions and their command usage, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide.*

Setting Up SNMP (Simple Network Management Protocol) - Summary

The following is a summary of the steps needed to configure SNMP.

OS/2 SNMP Client

- Modify the MIB2.TBL file to include any vendor specific MIB variables.

  To manage other hosts using SNMP, an SNMP agent must be installed and operational. Specifically:

  - To retrieve management information from SNMP agents, you must know the host name (or IP address) and community name of the agent.
  - To receive TRAPs from SNMP agents, you must configure the agent to forward these TRAPs to the OS/2 SNMP client.

OS/2 SNMP Agent

- Modify the CONFIG.SYS file for the MIB variables SYSCONTACT and SYSLOCATION.

  Configure the Community Name(s) for the Agent

- Create the PW.SRC file that contains the agent's community name(s).
- Scramble the PW.SRC file using MAKE_PW to produce the SNMP.PW file.
- Move the scrambled file to the directory defined by the ETC environment variable.

  Configure the Destination(s) to which TRAPs are sent

- Create the SNMPTRAP.DST file which contains a list of the SNMP client(s) to which TRAPs will be sent.

Other SNMP Agents

To manage agents other than OS/2 SNMP agents, those agents must be configured with a community name and set up to send TRAPs to your OS/2 SNMP client. This procedure depends on the particular SNMP agent you are managing. See the documentation accompanying your SNMP agent. For example, to send TRAPs from the IBM VM SNMP agent to your OS/2 SNMP client:

- Create the file SNMPTRAP DEST on the VM host.
- Specify your OS/2 SNMP client host name or IP address in this file.

Other SNMP Clients

To manage your OS/2 SNMP agent from a client other than an OS/2 SNMP client, configure the OS/2 SNMP agent as described in "Installing and Configuring the OS/2 SNMP Agent" on page 82. See the documentation accompanying your SNMP client for setup information.

Setting Up the Network Monitor (PMPING) - Summary

The following is a summary of the steps needed to configure PMPING.

- Modify the PINGHOST.LST file to contain the IP address and information to be displayed for each host you want to monitor.

Installing and Configuring the SNMP Client

This section describes how to set up the environment for the SNMP client.

Management Information Base (MIB)

The Management Information Base (MIB) defines management information obtained from SNMP agents. The MIB defines objects such as the description of the system being managed, packets received in error, and the status of an interface.

MIB objects can be described in two ways:

- Using an English-like textual notation. For example - sysDescr (system description)
- Using Abstract Syntax Notation.1 (ASN.1). For example - 1.3.6.1.2.1.1.1.0 is the ASN.1 equivalent of sysDescr.

Requests to obtain the value of a MIB object are sent to an SNMP agent using ASN.1 notation.

Some of these MIB objects are members of a table. For example, the Interfaces Table is a two dimensional array of MIB objects related to the interfaces installed. This array contains information such as the description of each interface and the speed of each interface. There can be several instances of a particular object within the table. For example, there would be a description of interface number 1, interface number 2, and so on.

Some MIB objects are scalars. That is, there is only one instance of that particular object. For example, there is only one system description.

Logically related MIB objects are placed into groups. A group can contain both scalars and tables. For example, the Interfaces Group contains a scalar (the object ifNumber, which is the number of interfaces present) and a table (the Interfaces Table).

For more information about MIB, see Appendix D, "Management Information Base (MIB) Objects."

Setting Up the Environment

The MIB2.TBL File

During installation, the MIB2.TBL file is placed in the directory defined by the ETC environment variable in your CONFIG.SYS file.

Several of the SNMP commands use the MIB2.TBL file. This file defines the mapping between an object's ASN.1 notation and an object's textual notation. When you issue an SNMP GET command, and specify the object name using the textual description, for example sysDescr, the OS/2 SNMP client looks for that object in the MIB2.TBL file and uses the corresponding ASN.1 notation in the SNMP request to an SNMP agent.

The MIB2.TBL file for the OS/2 SNMP client contains the textual names as defined in RFC 1213. A copy of this file is contained in Appendix E, "MIB2.TBL File: MIB-II Objects." Each line in this file has the following format:

textual_name asn.1_name syntax

For example:

```
sysDescr        1.3.6.1.2.1.1.1.0       display
```

Each field in the line must be separated by one or more spaces. The fields *textual_name* and *asn.1_name* correspond to the English-like textual notation and ASN.1 notation previously described.

The *syntax* field is the data type of the MIB object and has the value of display, object, number, counter, ticks, gauge, string, or internet. This syntax is defined in RFC 1155.

You can add entries to the MIB2.TBL file. Entries do not have to be in a specific sequence, but each entry must start on a new line.

Modify this file as needed for vendor-specific MIB objects, and save the new file in the directory defined by the ETC environment variable in your CONFIG.SYS file.

Note: You must modify the MIB2.TBL file only if you are managing SNMP agents that have objects that are not defined in RFC 1213.

Customizing SNMP for Your Network

This section describes information about how to modify the MIB2.TBL file. The MIB2.TBL file, which was installed in the directory specified by your ETC environment variable, contains all of the MIB objects as defined in RFC 1213.

Before adding MIB objects to the MIB2.TBL file, you should have the following information:

- The textual name for each object (for example, sysDescr)
- The ASN.1 notation for each object (for example, 1.3.6.1.2.1.1.1.0)
- The syntax (data type) of each object (for example, display).

See Appendix E, "MIB2.TBL File: MIB-II Objects" for a copy of this file.

The format of each line in the MIB2.TBL file is:

```
textual_name    asn.1_name      syntax
```

The Textual Name Field

The textual name field contains the name of the MIB object that is entered by the end user.

You can modify the textual names of the MIB objects in your MIB2.TBL file and still have the agent respond properly. For example, you could change `sysDescr` to `systemDescription`. As long as the asn.1_name field is correct, the value requested is correct.

Note: To remain consistent with the conventions in the MIB-II RFC, do not change these names.

The ASN.1_Name Field

This field represents the object identifier, in ASN.1 notation, of the MIB object. This value is sent to the SNMP agent during a SNMP GET or SNMP NEXT request.

For those MIB objects that are scalars (unique), the value in this field must end with **.0**. The following is the entry for the scalar MIB object sysDescr in the MIB2.TBL file.

```
sysDescr  1.3.6.1.2.1.1.1.0  display
```

If a scalar MIB object does not have the .0 suffix present, many SNMP agents respond with `no such name` when a request is made to obtain the value of that object.

For those MIB objects which are members of a table, the value in the asn.1_name field must end with **x.**, where *x* is a digit. For example, the entry for the MIB object ifIndex (from the Interfaces Table) in the MIB2.TBL file is:

```
ifIndex   1.3.6.1.2.1.2.2.1.1.  integer
```

If the trailing . (period) is left off, the request sent to the agent is not formatted properly and can result in a response of `no such name`.

The Syntax Field

The syntax of an object defines the data type of the object. It identifies the structure corresponding to object types. For a complete description of all data types and their meanings, see RFC 1155.

There are eight main data types:

- Integer

  A 32 bit numeric value.

- Octet String

  A string of octets. Each byte in an octet string can take any value from 0 to 255.

- Object Identifier

  An authoritative identification of an object. This is the ASN.1 notation.

- NetworkAddress

  This data type represents an address from a protcol family. The Internet family is the only protocol currently supported. Because of this, NetworkAddress is equivalent to IpAddress.

- IpAddress

  The IpAddress is a 32-bit internet address, represented as a string of eight components (octets) each with a length four bytes, in network byte order.

- Counter

  This data type represents a nonnegative integer that increases by one until it reaches a maximum value, at which time it resets to zero and starts increasing again.

- Gauge

  This data type represents a nonnegative integer that can increase or decrease, but which latches at a maximum value.

- TimeTicks

  This data type represents a nonnegative integer that counts the time in hundreths of a second since some event.

In addtion to the data types listed previously, the MIB-II RFC defines two additional data types:

- DisplayString

This data type is the same as an octet string, but is limited to the ASCII character set. It contains *human readable* characters.

- PhysAddress

  This data type is an octet string used for hardware addresses.

Data Types Used in the MIB2.TBL File

This section describes the correspondence between data types, as defined in the RFCs and the types supported by the MIB2.TBL file (and used by the SNMP GET and SNMP NEXT commands). When adding new objects to the MIB2.TBL, these are the values that should be used in the syntax field:

- Integer

  For integer objects, use **number** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `number` in the syntax field is:

  ```
  ifNumber  1.3.6.1.2.1.2.1.0  number
  ```

- Octet String and PhysAddress

  For Octet String objects and PhysAddress objects, use **string** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `string` in the syntax field is:

  ```
  ifPhysaddress  1.3.6.1.2.1.2.2.1.6  string
  ```

- Object Identifier

  For Object Identifier objects, use **object** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `object` in the syntax field is:

  ```
  sysObjectID  1.3.6.1.2.1.1.2.0  object
  ```

- NetworkAddress and IpAddress

  For NetworkAddress and IpAddress objects, use **internet** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `internet` in the syntax field is:

  ```
  ipRouteDest  1.3.6.1.2.1.4.21.1.1  internet
  ```

- Counter

  For Counter objects use **counter** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `counter` in the syntax field is:

  ```
  ifInOctets 1.3.6.1.2.1.2.2.1.10  counter
  ```

- Gauge

  For Gauge objects use **gauge** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `gauge` in the syntax field is:

  ```
  iSpeed   1.3.6.1.2.1.2.2.1.5  gauge
  ```

- TimeTicks

  For TimeTicks objects use **ticks** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `ticks` in the syntax field is:

  ```
  sysUptime   1.3.6.1.2.1.1.3.0  ticks
  ```

- DisplayString

  For DisplayString objects use **display** in the MIB2.TBL syntax field. An example of a MIB2.TBL entry with `display` in the syntax field is:

  ```
  sysDescr  1.3.6.1.2.1.1.1.0  display
  ```

Your CONFIG.SYS File

Verify that your CONFIG.SYS file contains a PATH entry for the SNMP executable (EXE) files. During installation the SNMP executables and other TCP/IP executables are installed in the directory of your choice.

Verify that your CONFIG.SYS file contains a LIBPATH entry for the ISODEDLL.DLL file.

Verify that your CONFIG.SYS file contains an ETC entry. If there is no ETC entry, C:\ETC is the default.

Starting SNMPREQD

You must start the SNMPREQD program before invoking any of the SNMP commands. This can be done in several ways.

- From the OS/2 command prompt type `START SNMPREQD` and press the Enter key. The OS/2 START command starts a program in another session. See the *OS/2 Commands Reference* for more information about the START command.
- Modify or create the file STARTUP.CMD to contain the line

  ```
  START "SNMPREQD" SNMPREQD
  ```

  STARTUP.CMD is a batch file that is the first session started by OS/2.

Display the OS/2 Task List and verify that there is an entry SNMPREQD. The OS/2 Task List can be displayed by pressing the Ctrl and Esc keys, or by pressing the right mouse button on an unused area of the screen.

SNMPREQD runs as a task until explicitly stopped.

Verifying Your Setup—Invoking the OS/2 SNMP Client Commands

You can invoke the OS/2 SNMP client commands after the environment has been established and SNMPREQD is running.

For a detailed description of the OS/2 SNMP commands' syntax and usage, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

The following examples are shown to allow a quick verification that SNMP was installed properly. The examples assume that an SNMP agent is installed and operational with the following parameters:

- SNMP agent host name is `quicktest`
- SNMP agent community name is `green`

The examples also assume that the SNMP agent supports the MIB objects `sysDescr` and `ifIndex`, and has been properly configured to send the authentication failure TRAP to your OS/2 SNMP client.

If an SNMP agent is not available, stop, and install the OS/2 SNMP agent on your PC. See "Installing and Configuring the OS/2 SNMP Agent" on page 82.

The following is a list of OS/2 SNMP client commands:

- SNMPGRP

  The SNMPGRP command is used to retrieve an entire group of MIB objects. Using the example information, at the OS/2 command prompt, type

  ```
  snmpgrp quicktest green sys
  ```

and press Enter . The SNMP agent should respond with all of the supported objects in the System group.

- SNMP GET

  The SNMP GET command is used to retrieve an individual MIB object. Using the example information, at the OS/2 command prompt, type

  ```
  snmp get quicktest green sysDescr
  ```

  and press Enter . The SNMP agent should respond with the value of sysDescr, the system description.

- SNMP NEXT

  The SNMP NEXT command is used to retrieve an individual MIB object from a table. Using the example information, at the OS/2 command prompt, type

  ```
  snmp next quicktest green ifIndex.0
  ```

  and press Enter . The SNMP agent should respond with the value for the first instance of ifIndex, which is 1.

- SNMPTRAP

  The SNMPTRAP command is used to display unsolicited TRAPs from an SNMP agent. Using the example information, at the OS/2 command prompt, type

  ```
  start snmptrap
  ```

  and press Enter . After the SNMPTRAP program window appears, select *Get_Traps* from the menu bar, and then select *Start*. At an OS/2 command prompt, type

  ```
  snmp get quicktest red sysDescr
  ```

  and press Enter . Because red is an incorrect community name, the SNMP agent should respond with an authentication failure TRAP, which is displayed on the SNMPTRAP window.

What to Do if there Is no Response from the SNMP Agent

If the agent does not respond to queries from the OS/2 SNMP client, check the following:

- Can you reach the host? Try to ping the host using PING.
- Is the agent operational?
- Is the community name you used correct?
- Does the agent support the MIB objects you used?

If the authentication failure TRAP was not displayed, check the following:

- Is the agent configured to send TRAPs to your OS/2 SNMP client?
- Does the agent support the authentication failure TRAP?
- Is the community name you used incorrect? To generate this TRAP, you should use an incorrect community name.

Installing and Configuring the OS/2 SNMP Agent

This section describes how to install and configure the OS/2 SNMP agent.

Setting Up the Environment

To set up the environment for the SNMP agent, you must do the following:

- Define the values for two MIB objects by modifying your CONFIG.SYS file.
- Define a community name for this agent by creating and scrambling the PW.SRC file.
- Define a list of SNMP clients that are sent TRAPs generated by the OS/2 SNMP agent, by creating the SNMPTRAP.DST file.

Modifying Your CONFIG.SYS File

To modify your CONFIG.SYS file, you must define values for the following two MIB objects.

- SYSCONTACT
- SYSLOCATION

You define these MIB objects by using the OS/2 SET command to set the value of two environment variables, SYSCONTACT and SYSLOCATION.

The following example shows the format of the SET command.

SET *environment_variable=value*

SYSCONTACT contains information about a contact person for this managed node, along with information about how to contact the person. For example, if Bill Smith at telephone extension 389 is the contact person for this node, the CONFIG.SYS contains the entry:

```
SET SYSCONTACT=B. Smith, Extension 389
```

SYSLOCATION contains the physical location of this node. For example, if the node is physically located in Raleigh, Building 503, Room A145, on the floor tile K-19, the CONFIG.SYS contains the entry:

```
SET SYSLOCATION=Raleigh, Bldg. 503, Room A145, Tile K-19
```

After modifying your CONFIG.SYS, you must reboot your PC for the changes to take effect.

TRAPs

TRAPs are unsolicited notifications of network-significant events that are sent from an SNMP agent to an SNMP client. For a description of TRAPs, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

The SNMPTRAP.DST file determines the client(s) that are to be sent the TRAPs generated by the OS/2 SNMP agent. The following describes how to set up the SNMPTRAP.DST file.

1. Create a file called SNMPTRAP.DST in the directory defined by the ETC environment variable in your CONFIG.SYS file.
2. The SNMPTRAP.DST file has a list of clients who are sent TRAPs, and identifies User Datagram Protocol (UDP) as the transport protocol used to send TRAPs.

The format of each line in the SNMPTRAP.DST file is: *hostname* UDP

The following is an example of the contents of an SNMPTRAP.DST file containing multiple entries.

```
124.34.216.1    UDP
Manager2        UDP
```

Creating a Community Name File

SNMP agents respond to requests for information from remote SNMP clients or network management stations. The community name is the authentication mechanism used by the SNMP to verify that a request for information is valid. A community name is similar to a password. Each request sent to the OS/2 SNMP agent must be accompanied by the correct community name. If a request is received with an incorrect community name, an authentication failure TRAP is sent to the SNMP clients listed in the SNMPTRAP.DST file.

In addition, the OS/2 SNMP agent has a protection mechanism to hide the information in the community name file so that it cannot be viewed. The information contained in the community name file, PW.SRC, is scrambled to prevent someone with access to the community name file from obtaining the actual community names.

The actual community names should reside in an unscrambled format in a master file on a secure host.

The following describes how to create a protected community name file.

1. Verify that there is an entry in your CONFIG.SYS to SET the HOSTNAME environment variable, SET HOSTNAME=*hostname*. Add this line if it is not present.
2. Create the PW.SRC file in the directory where the SNMP executable (EXE) files are installed by ICAT.
3. The format of each line in the PW.SRC file is:

   community_name desired_network snmp_mask

   The following is an example of the contents of a PW.SRC file containing multiple entries.

   ```
   passwd1   9.0.0.0       255.0.0.0
   passwd2   129.34.81.22 255.255.255.255
   ```

   When a request is received from an SNMP client, the community name received is checked against the entries in the PW.SRC file. If the community name received does not match any of the entries listed in the PW.SRC file, an AUTHENTICATION_FAILURE TRAP is sent, provided that a destination entry exists in the SNMPTRAP.DST file. If the community name received matches an entry in the PW.SRC file, the originating IP address of the incoming SNMP request is logically ANDed with the *snmp_mask*. The result of the logical ANDing process is compared with the *desired_network*, and if they are equal, the request is accepted. This allows the agent to accept requests only from certain clients which can have different community names.

   Using the previously described PW.SRC file as an example: assume a request from an SNMP client at IP address 9.34.22.122 is received and the correct community name of passwd1 was used by the manager. The IP address 9.34.22.122 is ANDed with 255.0.0.0. The result is 9.0.0.0, which equals the specified *desired_network* and the request is valid.

Looking at the second entry of the example PW.SRC file, a request that contains the community name passwd2 is only accepted from the SNMP client at host 129.34.81.22. If a request is received from any other client with the passwd2 community name, an authentication Failure TRAP is sent. A *desired_network* and *snmp_mask* of all zeros allows any host with the correct *community_name* to make requests.

4. Execute the MAKE_PW program. At the OS/2 command prompt, type MAKE_PW and press Enter. This program creates the scrambled SNMP.PW file.
5. Copy the SNMP.PW file to the directory defined by the ETC environment variable in your CONFIG.SYS file.

   To protect a remote SNMP agent, SNMP.PW should be created at a secure location and then sent to the remote host for inclusion in the ETC directory.

Starting the SNMP Agent

Before starting the OS/2 SNMP agent, the SNMPREQD program must be running. See "Starting SNMPREQD" on page 81.

SNMPD starts the OS/2 SNMP agent. There are several ways to start the OS/2 SNMP agent (SNMPD):

- From the OS/2 command prompt, type START SNMPD and press the Enter key. The OS/2 START command starts a program in another session. See *IBM OS/2 Commands Reference* for more information.
- Modify or create the file STARTUP.CMD to contain the line:

  ```
  START "SNMPD" SNMPD
  ```

  STARTUP.CMD is a batch file that is the first session started by OS/2.

Note: The entry to start SNMPD must come after the entry to start SNMPREQD in your STARTUP.CMD file.

Display the OS/2 Task List and verify that there is an entry SNMPD. The OS/2 Task List can be displayed by pressing the Ctrl and Esc keys, or by pressing the right mouse button on an unused area of the screen.

SNMPD runs as a task until explicitly stopped.

Verifying your OS/2 SNMP Agent Setup

When the environment is established and SNMPREQD is running, the OS/2 SNMP agent responds to requests for management information and sends TRAPs to SNMP clients. To verify the agent setup, you need an SNMP client to issue these requests and display the results, and which also displays TRAPs. If an SNMP client is not available, stop, and install the OS/2 SNMP client on your PC. When the client installation is done, see "Verifying Your Setup—Invoking the OS/2 SNMP Client Commands" on page 81 to verify that the SNMP agent was set up properly.

Monitoring Your Network Using PMPING

PMPING is a Presentation Manager (PM) program that displays the status of a user defined list of hosts using ICMP echo requests (PING). The PINGHOST.LST file contains the lists of hosts to be monitored.

Setting Up the Environment

During installation, a sample PINGHOST.LST file is placed in the directory defined by the ETC environment variable in your CONFIG.SYS file.

This table defines a list of hosts to be continuously monitored and a description of the host. The following example shows the format for the PINGHOST.LST file.

host_ip_address description

The following is an example of multiple entries in the PINGHOST.LST file.

```
9.67.30.100   **Nameserver-Call_Dan
9.67.22.1       RALVMM_via_3172-Call_IS
# This is a comment line.
```

Each field in the line must be separated by one or more spaces. The *host_ip_address* field is the IP address of the host being monitored. The *description* field is up to forty characters of comments that are displayed. A line with "#" starting in column 1 will be treated as a comment. Spaces are not allowed in the *description* field.

You can list up to 300 hosts (entries) in the PINGHOST.LST file. Each entry must start on a new line, and the entries do not have to be in a specific sequence.

Modify the sample file for your network and save the new file in the directory defined by the ETC environment variable in your CONFIG.SYS file.

Your CONFIG.SYS File

Verify that your CONFIG.SYS contains a PATH entry for the PMPING executable (EXE) file. During installation, the PMPING executable and other TCP/IP executables are installed in the directory of your choice.

Verify that your CONFIG.SYS contains a ETC entry. If there is no ETC entry, C:\ETC is the default.

Starting PMPING—Verifying Your Setup

When the environment is established you can start the the PMPING program. This can be done in one of several ways:

- From the OS/2 command prompt, type `START PMPING` and press the Enter key. The OS/2 START command starts a program in another session. For more information about the START command, see *IBM OS/2 Commands Reference*.
- Modify or create the file STARTUP.CMD to contain the line

  ```
  START PMPING
  ```

  STARTUP.CMD is a batch file that is the first session started by OS/2.

Display the OS/2 Task List and verify that there is an entry PMPING. The OS/2 Task List can be displayed by pressing the Ctrl and Esc keys, or by pressing the right mouse button on an unused area of the screen.

PMPING runs as a task until explicitly stopped.

From PMPING window, select Ping_all from the Menu bar, and then select Start. The list of hosts you defined appears in the PMPING window with their status.

To verify correct operation, make one of the hosts unreachable from your PC and ensure that the line corresponding to the unreachable host turns red in the PMPING

window. Reestablish connectivity and ensure that the host turns back to black in the PMPING window.

For a description of the other options available on the Menu bar, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

Chapter 8. Installing the Network File System Client

Chapter 8. Installing the Network File System Client

This chapter describes how to install the Network File System (NFS) client, and provides information about how to mount a remote NFS server for OS/2, AIX, VM, and MVS operating systems.

Setting Up Your Local Host

Because the OS/2 NFS client is built on the OS/2 Installable File System (IFS) base, an IFS statement must exist in the CONFIG.SYS file of the machine on which the client runs. If you allow ICAT to modify your CONFIG.SYS file, ICAT adds the statement for you.

If you do not allow ICAT to modify your CONFIG.SYS file, then you must add the IFS statement to your CONFIG.SYS file. For example, if you are installing TCP/IP for OS/2 in the C:\TCPIP directory, the IFS statement is:

```
IFS=C:\TCPIP\BIN\NFS.IFS
```

Setting Up the Environment

You can add SET commands to your CONFIG.SYS file when you add the IFS statement, to set defaults for the UNIX.UID, UNIX.GID, NFS.PERMISSION.BITS, NFS.PERMISSION.DBITS, and HOSTNAME environment variables.

Note: The environment variables depend on the type of access, and the type of server to which the client is communicating.

The environment variables are:

Environment Variable	Description
UNIX.UID	MOUNT can use the UNIX.UID environment variable to identify the client's user ID on UNIX systems. The NFS server uses the user ID value and the group ID value for permission checking. If these values are not set, MOUNT may prompt you to enter them.
UNIX.GID	MOUNT can use the UNIX.GID environment variable to identify the client's group ID on UNIX systems. The NFS server uses the group ID value and the user ID value for permission checking. If these values are not set, MOUNT may prompt you to enter them. **Note:** For security reasons, you should not set UNIX.UID and UNIX.GID in the CONFIG.SYS file, and instead let mount prompt you for them.
NFS.PERMISSION.BITS	The NFS Control Program (NFSCTL) uses the NFS.PERMISSION.BITS variable as the value, in octal, for UNIX permission bits when creating a file. The NFS.PERMISSION.BITS variable can also apply to non-UNIX servers, depending on the server implementation. For more information, see the documentation for the server you are using.

NFS.PERMISSION.DBITS	The NFS Control Program (NFSCTL) uses the NFS.PERMISSION.DBITS variable as the value, in octal, for UNIX permission bits when creating a directory. If this variable is not set, NFSCTL uses the value for NFS.PERMISSION.BITS when creating a directory. The NFS.PERMISSION.DBITS variable can also apply to non-UNIX servers, depending on the server implementation.
HOSTNAME	The NFS Control Program (NFSCTL) uses the HOSTNAME variable to identify the client requesting access to the server.
TZ	The NFS Control Program (NFSCTL) uses the TZ environment variable to determine the correct date and time associated with a file. For more information about the TZ environment variable, see "The TZ Environment Variable."

You are ready to start the NFS client after you complete the following steps:

- The IFS statement has been added to your CONFIG.SYS file.
- The applicable environment variables have been set.
- You have rebooted the machine, so that the CONFIG.SYS file changes take effect.

The TZ Environment Variable

You must set the TZ environment variable, which is used to determine the correct date and time for clients located in different time zones. The TZ environment variable contains a string that has the abbreviation for your time zone, and the number of hours your time zone differs from Greenwich Mean Time (GMT).

The following example shows the format for the TZ environment variable.

```
SET TZ=xxxnyyy
```

The parameters of the TZ environment variable are:

Parameter	Description
xxx	The three-character label for your time zone.
n	The number of hours your time zone differs from GMT. If you are east of GMT, put a minus sign (-) before the number. If you are west of GMT, you can optionally put a plus sign (+) before the number. The *n* value ranges from -12 to +12.
yyy	The three-character label for your time zone when in daylight savings time. If you do not observe daylight savings time, leave this parameter blank.

Note: The default for the TZ environment variable is Eastern Standard Time (EST).

The following are examples of the TZ environment variable for different time zones.

Example	Description
EST5EDT	For the eastern coast of the United States. The time zone label is EST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is EDT. During standard time there is a five hour difference between local time and GMT.
CST6CDT	For the central time zone of the United States. The time zone label is CST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is CDT. During standard time there is a six hour difference between local time and GMT.
MST7	For the mountain time zone of the United States. The time zone label is MST. Daylight savings time is not observed. During standard time there is a seven hour difference between local time and GMT.
PST8PDT	For the western coast of the United States. The time zone label is PST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is PDT. During standard time there is an eight hour difference between local time and GMT.

The TZ environment variable should be included in your CONFIG.SYS file. You can let ICAT make this modification for you during configuration. If you change the TZ environment variable, reboot your PC so that the change takes effect.

Starting the NFSCTL Program

The NFS Control Program (NFSCTL) is an application that communicates with the NFS IFS driver. The NFS Control Program (NFSCTL) must be running to mount a remote file system as a local drive.

To start the control program, a REXX file called NFSC.CMD is supplied with TCP/IP for OS/2. The parameters of the REXX file are identical to the parameters for NFSCTL. You can invoke the REXX file by typing NFSC followed by the parameters at an OS/2 command prompt. When you require special configuration of the client control program, you can either use the NFSCTL program directly, or modify the NFSC.CMD file.

Using NFSSTART

NFSSTART is a command file, written in REXX, which starts the NFS client and automatically mounts entries that are in the FSTAB file. You can use NFSSTART with the FSTAB file when you have a set of servers that you have to mount every day to avoid mounting each one manually.

The following example shows the format of the NFSSTART command.

NFSSTART [*etc_dir* [*nfsctl_parameters*]]

The parameters of the NFSSTART command are:

Parameter	Description
etc_dir	The base directory for the FSTAB file. If not specified, the value of the ETC environment variable is used as the *etc_dir*.
nfsctl_parameters	The parameters to be passed to the NFS Control Program (NFSCTL).

If FSTAB does not exist, then using the NFSSTART.CMD file is the same as using the NFSC.CMD file.

Interfacing with ICAT

ICAT edits the STARTUP.CMD file for you and adds a call to the TCPSTART.CMD file, which starts the services (and other programs) that you selected during the configuration phase of ICAT. The entry added for NFS is a call to NFSSTART with any parameters you specified on the ICAT services screen.

The FSTAB File

The FSTAB file specifies the NFS servers that are mounted automatically when your machine is booted. The FSTAB file contains two types of commands: MOUNT and MVSLOGIN. For more information about the MOUNT and MVSLOGIN commands, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

To enter comments into the FSTAB file put a pound sign (#) in front of the comment to be added. For example:

```
MOUNT -u -g o: OS2:d:\                          #mount an OS/2 machine
MOUNT -v v vm1:diska.191,userid=myid,ro         #mount a vm disk
# mount a Risc System/6000 machine
u: aix1:/u/shawnna
```

Note: When you enter a MOUNT command in the FSTAB file, you can omit the word "MOUNT", as in the previous example.

Stopping the NFSCTL Program

Because the NFSCTL program runs at a low level of the operating system, you must use the following method to stop the NFSCTL program.

1. While in the NFSCTL window, press the Ctrl key and C key simultaneously. If there are no NFS drives currently mounted, the NFS client Control Program stops after a few minutes.
2. If any NFS drives are mounted, the control program notifies you, and does not stop. To stop the control program, unmount any mounted drives, using UMOUNT, and press the Ctrl key and C key again.

See *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for information on how to determine the drives that are still mounted using the QMOUNT command.

If you cannot unmount all mounted drives, the preferred alternative method to stop the NFSCTL program is:

1. From the *Task List*, highlight the NFS Control Program entry, and select *End Task*. A dialog box appears, informing you that the session selected still contains an active program, and asks if you want to close the session anyway.
2. Select *Yes* to stop the NFS Control Program.

Note: Stopping the control program without unmounting all NFS drives can cause unpredictable results in applications accessing a mounted drive.

Mounting a Remote NFS Server

This section provides guidelines about how to mount a remote NFS server for common operating systems (for example, UNIX, VM, and MVS). For more information, see the documentation for the NFS server that you are using.

Note: The MOUNT command varies, depending on the NFS server that you are attempting to mount. When you invoke the MOUNT command, the client passes the last portion to the server without any modifications. The server interprets the requested action of the client. See the specific NFS server documentation to determine the information that the server expects from the client on a mount request.

OS/2 NFS Servers

To mount a directory on an OS/2 NFS server, you must export the OS/2 file system by adding an entry to the ETC\EXPORTS file on the server machine. The entry must specify the mount point, which is the directory to be exported.

The OS/2 NFS server allows you to specify hosts that can have access and specify the type of access. Your user name, password, and the HOSTNAME environment variable, which are specified on the client machine, are sent to the server to determine if a client can mount a particular directory and, if so, with what type of access.

The following steps describe how to mount a directory on an OS/2 machine.

Note: In the following steps:

- The client machine is an OS/2 machine, with the host name `andrew`
- The server machine is a OS/2 machine, with the host name `christy`
- The mount directory is `e:\ftppm`

1. Set your HOSTNAME at the client, as shown in the following example:

   ```
   os/2>SET HOSTNAME=ANDREW
   ```

2. Start the NFS Control Program (NFSCTL), if it is not running, with the following command:

   ```
   os/2>NFSC
   ```

3. Determine if the `e:\ftppm` directory has been exported for you to mount by using the show exports (SHOWEXP) command, as shown in the following example:

   ```
   os/2>SHOWEXP CHRISTY
   export list for CHRISTY:
   e:\ftppm            andrew
   c:\tools
   ```

   If the SHOWEXP CHRISTY command responds with:

   ```
   e:\ftppm            andrew
   ```

   or:

   ```
   e:\ftppm            everyone
   ```

   you are ready to mount the `e:\ftppm` directory. If the SHOWEXP CHRISTY command does not show `e:\ftppm` being exported to your machine, you must add it to the ETC\EXPORTS file on CHRISTY. For more information about the EXPORTS file, see "The EXPORTS File" on page 104.

4. Mount the remote directory as an unused drive, as shown in the following example:

```
os/2>MOUNT -u -g z: CHRISTY:E:\FTPPM
mount: CHRISTY:E:\FTPPM
```

AIX NFS Servers

To mount a directory on a RISC System/6000* machine with AIX version 3.0, you must export the file system to your OS/2 machine. To do this, add an entry in the /etc/exports file on the server machine. The entry must specify a mount point, which is the directory to be exported, and can optionally specify a set of hosts that have access.

The HOSTNAME environment variable, which is set at the client, and your UID and GID are sent to the server to determine if a client can mount a particular directory, and with what type of access.

You can determine the mapping between user logon IDs and UNIX UIDs and GIDs by checking the /etc/passwd file on the server machine.

The following steps describe how to mount a directory on a RISC System/6000 machine.

Note: In the following steps:

- The user logon ID is `jr`
- The client machine is an OS/2 machine, with the host name `ziggy`
- The server machine is a RISC System/6000, with the hostname `RS6000`
- The mount directory is `tools`

1. Determine whether PCNFSD is running on the server. If it is, proceed to step 3. If not, proceed to step 2.
2. Determine your UID and GID on the RS6000.

   Enter the following command:

   ```
   RS6000>grep jr /etc/passwd

   jr::300:30::/usrs/jr:/bin/csh
   ```

   Where:

   `300` is the UID and `30` is the GID.
3. Set your HOSTNAME at the client, as shown in the following example.

   ```
   os/2>SET HOSTNAME=ziggy
   ```
4. Start NFSCTL with the following command if it is not already running.

   ```
   os/2>NFSC
   ```
5. Determine if the `/tools` directory has been exported for you to mount by using the show exports (SHOWEXP) command, as shown in the following example.

   ```
   os/2>SHOWEXP RS6000
   export list for RS6000:
   /usrs/jr      ziggy
   /usrs/ps      ps
   ```

   If the RS6000 response lists:

   ```
   /tools        ziggy
   ```

   or:

```
/tools          everyone
```

you are ready to mount the /tools directory.

If the SHOWEXP RS6000 command does not show /tools being exported to your machine, you must add it to the /etc/exports file on RS6000. You must also have the NFS server read the file again.

To add the /tools directory to the /etc/exports file on RS6000, do the following steps.

a. Edit the /etc/exports file, and add the line

```
/tools
```

b. At the RS6000 prompt, enter the following:

```
RS6000>exportfs -av
```

c. Verify that the correct directory has been exported. For example:

```
os/2>SHOWEXP RS6000
export list for RS6000:
/usrs/jr      ziggy
/usrs/ps      ps
/tools        everyone
```

Note: If you added the line to the exports file as /tools -access=ziggy, then only the host ziggy can mount the /tools directory.

6. Mount the remote directory as an unused drive, as shown in the following example.

```
os/2>MOUNT z: RS6000:/tools
```

If PCNFSD is running on the remote server, MOUNT prompts you for your user ID and password. If PCNFSD is not running and you have not set the UNIX.UID and UNIX.GID environmental variables, then MOUNT prompts you to enter your UID and GID, as determined in step 2.

VM NFS Servers

To mount an NFS minidisk on a VM machine running TCP/IP Version 2.0 as a local drive, you must enter your user ID, password, and other options with the MOUNT command.

The following steps describe how to mount a directory on a VM machine.

Note: In the following steps:

- The user logon ID is jr
- The user password is pass
- The server machine is a VM machine with the host name VM1
- The mount minidisk is jr.191

1. Start NFSCTL with the following command if its not already running.

```
os/2>NFSC
```

2. Give the NFS service MULTIPLE access to your minidisk.

Note: This step can vary, depending on your installation.

3. Mount the remote minidisk as an available drive.

```
os/2>MOUNT -v z: vm1:jr.191,rw,user=jr,record=nl,names=fold
mount: vm1:jr.191,rw,user=jr,record=nl,names=fold
Enter password: pass
```

Note: Your password does not appear on the screen when you type it.

MVS NFS Servers

When using an MVS NFS server, the client needs to authorize access to the mounted directory by using the MVSLOGIN program provided with TCP/IP for OS/2. See *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for the format of the MVSLOGIN command.

To access files from the MVS NFS machine, invoke the MVSLOGIN program only once to a particular MVS NFS server. For example, you can mount a particular MVS NFS server multiple times, but use the MVSLOGIN program with that server only once.

To do this, mount the remote data set, and use the MVSLOGIN program to get access to the mounted files.

Note: An MVS system can periodically log off a user. If you are logged off by the MVS system, reissue the MVSLOGIN command to regain access. You do not have to reissue the MOUNT command.

When you finish accessing the MVS NFS server, issue the MVSLOGUT command to log off your user ID.

The following steps describe how to mount a directory on an MVS machine.

Note: In the following steps:

- The user logon ID is `jr`
- The server machine is an MVS machine with the host name `MVS1`
- The mount data set is `jrose`

1. Start NFSCTL with the following command if its not already running.

   ```
   os/2>NFSC
   ```

2. Add an entry to the MVSNFS.CNTL (EXPORTS) data set to export the `jrose` data set.

3. Restart the MVSNFS server.

4. Authorize the MVSNFS server to have access to the exported directory.

   Note: This step varies, depending on your particular installation.

5. Verify that the correct directory has been exported. Enter the following command:

   ```
   os/2>SHOWEXP MVS1
   export list for MVS1:
   SHIFERT       rose shifert
   JROSE         everyone
   DMBARDON      everyone
   BSTOW         everyone
   ```

6. Mount the remote directory as an unused drive. Enter the following command:

   ```
   os/2>MOUNT -u -g z: mvs1:jrose,text
   mount: mvs1:jrose,text
   ```

   Note: In the example, you are mounting the data set with the `text` attribute. See *MVS/DFP Version 3 Release 3: Using the Network File System Server* for information on this and other attributes.

7. Authorize your access to the mounted drive. If access is not authorized, you will get an authentication failure when you try to access a file. Enter the following:

```
os/2>MVSLOGIN -p mvs1 jrose

Enter MVS password:
```

When you have successfully logged on using MVSLOGIN, you can access files.

Note: Some systems automatically log off users after a system-defined time period. If authentication errors occur, you may have to reissue the MVSLOGIN command.

When you finish accessing files, or have unmounted an MVS mounted NFS drive, log off using the MVSLOGUT command, as shown in the following example.

```
os/2>MVSLOGUT mvs1
```

Chapter 9. Installing the Network File System Server

Chapter 9. Installing the Network File System Server

The OS/2 Network File System (NFS) server allows you to transparently share files between your OS/2 machine and any other machine equipped with an NFS client. The server allows files on your machine to be available to people on other machines. To use files from their machines on your machine, you must use the NFS client. For more information about the NFS client, see Chapter 8, "Installing the Network File System Client."

Setting Up the Network File System Server

The NFS server requires a special file, called the EXPORTS file, to exist in your ETC directory. This file contains a list of directories on your machine and the remote machines that can access these directories. ICAT does not create this file for you. See "The EXPORTS File" on page 104 for a description of the format of this file.

When you run the NFS server, the PORTMAP program must also be running.

The TZ Environment Variable

You must set the TZ environment variable, which is used to determine the correct date and time for clients located in different time zones. The TZ environment variable contains a string that has the abbreviation for your time zone, and the number of hours your time zone differs from Greenwich Mean Time (GMT).

The following example shows the format for the TZ environment variable.

```
SET TZ=xxxnyyy
```

The parameters of the TZ environment variable are:

Parameter	Description
xxx	The three-character label for your time zone.
n	The number of hours your time zone differs from GMT. If you are east of GMT, put a minus sign (-) before the number. If you are west of GMT, you can optionally put a plus sign (+) before the number. The *n* value ranges from -12 to +12.
yyy	The three-character label for your time zone when in daylight savings time. If you do not observe daylight savings time, leave this parameter blank.

Note: The default for the TZ environment variable is Eastern Standard Time (EST).

The following are examples of the TZ environment variable for different time zones.

Example	Description
EST5EDT	For the eastern coast of the United States. The time zone label is EST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is EDT. During standard time there is a five hour difference between local time and GMT.

CST6CDT	For the central time zone of the United States. The time zone label is CST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is CDT. During standard time there is a six hour difference between local time and GMT.
MST7	For the mountain time zone of the United States. The time zone label is MST. Daylight savings time is not observed. During standard time there is a seven hour difference between local time and GMT.
PST8PDT	For the western coast of the United States. The time zone label is PST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is PDT. During standard time there is an eight hour difference between local time and GMT.

The TZ environment variable should be included in your CONFIG.SYS file. You can let ICAT make this modification for you during configuration. If you change the TZ environment variable, reboot your PC so that the change takes effect.

Starting the NFS Server

Verify that PORTMAP is running before starting the NFS server. If PORTMAP is not running, type PORTMAP at an OS/2 command prompt to start it. Then type NFSD in a different OS/2 window to start the NFS server.

The following example shows the format of the NFSD command.

```
NFSD
```

There are no parameters for the NFSD command.

Stopping the NFS Server

You can stop the NFS server at any time by pressing the Ctrl key and the C key simultaneously.

The EXPORTS File

The EXPORTS file is located in your ETC directory and contains an entry for each directory that can be exported to NFS clients. The EXPORTS file is read only during NFSD startup.

The following example shows the format of the EXPORTS file.

```
directory option [,option ] ...
```

These elements are defined as follows:

Element	Description
directory	Specifies the path name of the directory.
option	Specifies a client that can mount this directory or an option.

The only option for the elements in the EXPORTS file is:

Option	Description
-ro	Exports the directory with read-only permission. If not specified, the directory is exported with read-write permission.

A pound sign (#) anywhere in the file indicates a comment that extends to the end of the line.

The following shows examples of entries in the EXPORTS file:

- Export read/write to the world:

  ```
  C:\usr\local
  ```

- Export read/write to only the specified machines:

  ```
  C:\usr2 hermes,zip,tutorial
  ```

- Export read-only to everyone:

  ```
  D:\usr\bin -ro
  ```

- Export read-only to roger and vinnie:

  ```
  F:\usr\net -ro roger vinnie
  ```

- Export read-only to jack and trish:

  ```
  R:\usrq jack -ro trish
  ```

- Export read/write to Jack and Jill:

  ```
  C:\hill jack jill
  ```

Notes on the NFS Server:

1. The NFS server does not keep all of the attributes for each file, as defined in the RFC for NFS. Instead, the attributes that map directly to an attribute in OS/2 are stored, and the others are either faked or permanently set to some value.

   In particular, the NFS server keeps the user-write bit in the read/write bit used by OS/2. The user, group, and other read bits are permanently set to 1. The group and other write bits are set to the same value as the user-write bit. The user, group, and other execute bits are set depending on the file's extension. If the file ends with .exe, .EXE, .com, .COM, .cmd, .CMD, .out, .OUT, .nfs, or has no extension, the three execute bits are set.

2. The UID and GID associated with a file are faked by the NFS server. When the NFS server is started, it checks the environment variables UNIX.UID and UNIX.GID. These values are the UID and GID associated with every file on the server.

3. The NFS server does not support hard links or soft links.

4. Because of a limitation in OS/2, the date associated with a file cannot be before January 1, 1980.

5. If you use the NFS server on a FAT drive, file names are restricted to 8.3 format, and must be in uppercase.

6. If you use the NFS server on an HPFS drive, file names can be up to 254 characters in length. Files are stored with the same name you specified during creation (including upper and lower case).

   Note: HPFS considers file names that differ only in case to be the same name (for example, MyFile, MYFILE, and myfile all refer to the same file).

7. For best performance, NFS clients should use no more than a 4KB buffer size when performing writes with BIOD's enabled. The OS/2 NFS Client defaults to this configuration.

Chapter 10. Setting Up an X.25 Interface

Chapter 10. Setting Up an X.25 Interface

An X.25 interface with TCP/IP for OS/2 allows you to connect to another TCP/IP network over an X.25 network. You can use most of the same TCP/IP applications over an X.25 network as if your host is connected to a LAN.

OS/2 allows multiple X.25 adapters. TCP/IP for OS/2 allows you to assign only one IP address to the X.25 interface. This assignment is done with the IFCONFIG statement. This chapter contains the information necessary to install, configure, and use an X.25 interface with TCP/IP for OS/2.

Overview of Installation

This section describes the system and preinstallation requirements for installing an X.25 interface.

System Requirements

You must have OS/2 EE, Version 1.3 with Communications Manager and an IBM X.25 Interface Coprocessor/2 adapter installed. This adapter requires a minimum of a PS/2 Model 50 (micro channel), or higher. For additional information about the required hardware and software environments, see Chapter 2, "Introducing TCP/IP for Your OS/2 Environment."

The following steps describe how to install an X.25 interface for TCP/IP. In the following steps, IPX25.CFG is used as an example.

1. Install the TCP/IP for OS/2 X.25 product using ICAT. This assumes that the TCP/IP for OS/2 Base product is already installed.

   To install the TCP/IP for OS/2 X.25 product:

   - Enter ICAT
   - Select *INSTALL*
   - Select the X25 Interface Software
   - Insert the specified diskette
   - Press Enter .

2. Create the Communications Manager X.25 configuration file (for example, IPX25.CFG). For information about how to create the Communications Manager X.25 configuration file, see "Creating a Communications Manager X.25 Configuration File" on page 110.

3. OS/2 installs the X.25 Communications Manager support and puts the device statement (DEVICE=C:\CMLIB\ICARICIO.SYS) in the CONFIG.SYS file, if you type:

   ```
   reinst
   ```

   a. OS/2 EE prompts you to put a specified diskette into drive A. Insert the diskette and press Enter .

   b. Select *Install Communications Manager* and press Enter .

   c. Select *User Configuration Files and Features* and press Enter .

   d. Enter `C:\CMLIB` as the "Source drive and directory" and press Enter .

   e. Select the configuration file (for example, `IPX25`) and press Enter .

f. Press F3 to install the X.25 support and Exit. OS/2 EE prompts you for several diskettes. Insert the specified diskettes and press Enter as requested.

For more information about `reinst`, see the *IBM Operating System/2: System Administrator's Guide for Communications*.

4. Copy the ICAAIM.COM file from the X.25 option diskette, provided with your X.25 coprocessor adapter, to the CMLIB directory by inserting the diskette in drive A and typing:

```
COPY A:ICAAIM.COM C:\CMLIB
```

Creating a Communications Manager X.25 Configuration File

The following steps describe an example of how to create the Communications Manager X.25 configuration file used for configuring an X.25 interface. For more information about creating the Communications Manager configuration file, see the *IBM Operating System/2: System Administrator's Guide for Communications*. The following steps describe how to create the example configuration file IPX25.CFG. The configuration file XIPSAMP.CFG was created following these steps and has been included in the ETC directory.

Note: You must have certain information available to create your configuration file. Review the following steps to determine the information that you need. The following is only an example. You may want to contact your system administrator for information specific to your network.

1. Select the *OS/2 Full Screen* command prompt from the "Group-Main" window.
2. Copy the base configuration file C:\CMLIB\ACSFG.CFG (non-US version), or C:\CMLIB\ACSFGUS.CFG (US version) to C:\CMLIB\IPX25.CFG.
3. Start Communications Manager by selecting *Communications Manager* from the "Group-Main" window. The "Communications Manager Main Menu" is displayed.
4. Select *Advanced* from the action bar.
5. Select *Configuration*.
6. Specify the configuration file name `IPX25`
7. Press Enter. The "Configuration Menu" is displayed.
8. Select *X.25 feature profiles*. The "X.25 Feature Configuration" window is displayed.
9. Select *Adapter profiles*. The "X.25 Adapter Profile Configuration" window is displayed.
10. Select *Operations* from the action bar.
11. Select *Create*. The "Create/Change X.25 Adapter Profile" window is displayed.
12. Specify the adapter name `ADAPTER1`
13. Press Enter.
14. Optionally specify a comment, such as: `Adapter 1 in slot 3`.
15. Specify the slot number of the IBM X.25 Interface Co-processor/2 adapter.
16. Press Enter. The "X.25 Adapter Profile Configuration" window is displayed with the message: *The profile has been saved.*
17. Press the Escape Esc key. The "X.25 Feature Configuration" window is displayed.

18. Select *Link profiles*.
19. Select *Operations* from the action bar.
20. Select *Create*.
21. Specify the link profile name `LINK1`
22. Press Enter. The "Create/Change X.25 Link Profile" window is displayed.
23. Select *Base link parameters*. The "Create/Change X.25 Base Link Parameters" window is displayed.
24. Optionally specify comment, such as: `LINK1 on Adapter 1.`
25. Specify the Adapter profile name `ADAPTER1`
26. Specify the network type appropriate to your network. If you are using two workstations connected using a modem eliminator or two modems and a telephone line, specify network type 1, otherwise press F1 for help and a list of networks. Choose the network type for your network and note the local CCITT compliance and the link setup mode. You may want to contact your system administrator if you are unsure of your network address. For more information, see the *IBM Operating System/2: System Administrator's Guide for Communications*.
27. Specify the local DTE address. Contact your system administrator if you are uncertain of the local DTE address.
28. Select an option for the local CCITT compliance. If the adapters are connected using a modem eliminator or two modems and a telephone line, select the default the local CCTITT compliance of 1980, otherwise select the local CCITT compliance noted in step 26.
29. Select *Connect* for the initial mode of link.
30. Select an option for the link setup mode. If the adapters are connected using a modem eliminator or two modems and a telephone line, select the default link setup mode of `initiate from DTE`, otherwise select the link setup mode noted in step 26.
31. Press Enter. The "Create/Change X.25 Link Profile" window is displayed with the message: *The profile has been saved.*
32. Select *Virtual circuit ranges*. The "Create/Change X.25 Virtual Circuit Ranges" window is displayed.
33. Specify logical channel numbers and ranges that match the network subscription for the link. Contact your system administrator if you are uncertain of the logical channel numbers and ranges.
34. Press Enter. The "Create/Change X.25 Link Profile" window is displayed with the message: *The profile has been saved.*
35. Select *Virtual circuit parameters*. The "Create/Change X.25 Virtual Circuit Parameters" window is displayed.
36. Specify the SVC default incoming packet size `1024`
37. Specify the SVC maximum incoming packet size `1024`
38. Specify the SVC default outgoing packet size `1024`
39. Specify the SVC maximum outgoing packet size `1024`
40. Press Enter. The "Create/Change X.25 Link Profile" window is displayed with the message: *The profile has been saved.*
41. Press F3. The "X.25 Link Profile Configuration" window is displayed.

42. Press Esc. The "X.25 Feature Configuration" window is displayed.
43. Select *Directory*. The "Directory Configuration" window is displayed.
44. Select Entry Name *M2*.
45. Select *Operations* from the action bar.
46. Select *Create*.
47. Specify the directory entry name `LOCAL1`
48. Press Enter.
49. Specify the link profile name `LINK1`
50. Specify the local DTE address. Contact your system administrator if you are uncertain of the local DTE address.
51. Press Enter. The "X.25 Directory Configuration" window is displayed with the message: *The profile has been saved.*
52. Press Esc. The "X.25 Feature Configuration" window is displayed.
53. Select *Directory*. The "Directory Configuration" window is displayed.
54. Select Entry Name *M6*.
55. Select *Operations* from the action bar.
56. Select *Create*.
57. Specify the directory entry name `REMOTE1`
58. Press Enter.
59. Specify the link profile name `LINK1`
60. Specify the remote DTE address. Contact your system administrator if you are uncertain of the remote DTE address.
61. Press Enter. The "X.25 Directory Configuration" window is displayed with the message: *The profile has been saved.*
62. Press Esc. The "X.25 Feature Configuration" window is displayed.
63. Select *Routing Table*.
64. Select *M7*.
65. Select *Operations* from the action bar.
66. Select *Create*. The "X.25 Routing Table Configuration" window is displayed.
67. Specify the routing table entry name `ROUTE1`
68. Press Enter. The "Create/Change X.25 Routing Table Entry" window is displayed.
69. Optionally specify a comment, such as: `TCP/IP X.25 Route`.
70. Specify the link profile name `LINK1`
71. Specify the Call User Data field as: `CC` (required for TCP/IP).
72. Press Enter.
73. Select *Top of table*. The "X.25 Routing Table Configuration" window is displayed with the message: *The profile has been saved.*
74. Press Esc. The "X.25 Feature Configuration" window is displayed.
75. Press Esc. The "Communication Configuration Menu" is displayed.

76. Select *Verify* from the action bar.
77. Select *Run Verify*.
78. When verification is complete, press Enter.
79. Select *Exit* from the action bar.
80. Select *Exit communication configuration*. The "Communications Manager Main Menu" is displayed.
81. Select *Exit* from the action bar.
82. Select *Exit immediately*.
83. Select *Yes* to confirm exit from the Communications Manager.

Configuring the X.25 Interface

The following steps describe how to configure an X.25 interface for TCP/IP:

1. Create the X25IP file in the ETC directory. This file contains the local directory that identifies a local DTE address with a link. The following is an example of the contents of the X25IP file. It is based on the Communications Manager file IPX25.CFG that you created in "Creating a Communications Manager X.25 Configuration File" on page 110.

   ```
   LOCAL1
   ```

   See Appendix A, "Optional Files" for file format. This example X25IP file has been installed by ICAT in the ETC directory. If you want to use this file, you must copy it to your CMLIB directory.

2. Create the X25RTE file in the ETC directory. This file contains the X.25 routing table and decides which incoming X.25 calls are to be routed to the TCP/IP application. The following is an example of the contents of the X25RTE file. It is based on the Communications Manager example configuration file IPX25.CFG that you created in "Creating a Communications Manager X.25 Configuration File" on page 110.

   ```
   ROUTE1
   ```

   All incoming X.25 call requests that match the fields in the IPX25.CFG file with the routing table entry name ROUTE1 are associated with the TCP/IP application. See Appendix A, "Optional Files" on page 167 for the specific file format. This example X25RTE file has been installed by ICAT in the ETC directory. If you want to use this file, you must copy it to your CMLIB directory.

3. Create the X25DIR file in the ETC directory. This file contains the TCP/IP X.25 directory name that associates a remote host's DTE address and IP address. The following is an example of the contents of the X25DIR file. It is based on the Communications Manager example configuration file IPX25.CFG that you created in "Creating a Communications Manager X.25 Configuration File" on page 110.

   ```
   ipaddress REMOTE1
   ```

   All outgoing IP packets with the destination of the specified remote IP address are sent over the X.25 interface to the remote DTE address defined in the IPX25.CFG file with the directory name REMOTE1. If an X.25 call is not established, an X.25 call request is made. See Appendix A, "Optional Files" for the specific file format. An example X25DIR file with an ipaddress of default has been installed by ICAT in the ETC directory. If you want to use this file, you must copy it to your CMLIB directory.

4. When using TCP/IP for OS/2 over an X.25 network, switched virtual circuit (SVC) may be established to transfer TCP/IP data. An SVC connection is establish for each unique destination IP address. An SVC connection closes after a specified period of inactivity. You can specify the time-out period in seconds with the environment variable IPX25.SVC. The default value is 1800 seconds (30 minutes). The following example changes the time-out period to 5 minutes if you start the X25IO program from the same OS/2 session:

   ```
   SET IPX25.SVC=300
   ```

   The legal range of values are from 0 (SVC connection will remain indefinitely) to 3600 (60 minutes).

Starting an X.25 Interface

The following steps describe how to start an X.25 interface with TCP/IP for OS/2.

1. Start the Communications Manager by selecting *Communications Manager* from the Group-Main window. The Communications Manager X.25 configuration file contains the X.25 configuration information.

   The X.25 link is established. If the X.25 link cannot be established or does not stay established then you must stop and restart Communications Manager. Contact your system administrator if you cannot establish the X.25 link.

2. Start the X25IO program. You can do this by entering `X25IO` at the OS/2 prompt. There is also an XIOWAIT program in the BIN directory so that you can use the start X25IO from a CMD file. ICAT does this in X25.CMD.

   Note: After the X25IO program is running, it cannot be stopped and restarted. You must reboot your machine to do this.

3. Start the X.25 interface by using IFCONFIG, ICAT will put this in the X25.CMD file or you can enter:

   ```
   ifconfig x25 ipaddress mtu 576
   ```

   Where:

   ipaddress is the IP address of the X.25 interface on your local host.

   The mtu size 576 is the default mtu size on X.25 networks that support TCP/IP. The IFCONFIG command with x25 specified as the interface will have no effect unless X25IO is running.

X.25 Limitations

1. X.25 Permanent Virtual Circuits (PVCs) are not separated.
2. A maximum of 16 active SVCs are supported. An SVC will become inactive after a specified period of inactivity. This is discussed in step 4 under "Configuring the X.25 Interface."
3. The MTU size is not negotiated with the remote host.
4. The Department of Defence Network (DDN) algorithm is not used to convert IP addresses to DTE addresses. The conversion of IP addresses to DTE addresses is defined in the X25DIR file.

Chapter 11. Setting Up a SLIP Line

Chapter 11. Setting Up a SLIP Line

The Serial Line Internet Protocol (SLIP) allows you to connect to another TCP/IP network over a serial line. A serial line is used to set up a point-to-point link between a local host to a foreign host. You can use most of the the same TCP/IP applications over a serial line as if your host is connected to a LAN.

Note: You should be aware that SLIP performance depends on the speed of the modems. The performance of some applications with SLIP can be affected.

SLIP allows an OS/2 host to be connected to a remote host over a telephone line using a modem or over a serial line using a null-modem cable. The SLIP connection allows you to have access to the network on which the remote host resides.

SLIP allows you to use any valid OS/2 serial communication port (COM port) for a SLIP connection. However, with TCP/IP for OS/2, you can only use one COM port at a time. Therefore, if you have an active SLIP connection, you cannot originate or accept another SLIP connection.

SLIP Prerequisites

The following elements are required to use TCP/IP for OS/2 SLIP over a telephone line.

- An analog telephone line.
- Two Hayes AT-compatible modems are required and both must support the same speed, from one of the following speeds:
  - 1200 bps
  - 2400 bps
  - 4800 bps
  - 9600 bps.
  - 19 200 bps

  Note: All line speeds are given in bits per second.

Setting Up the Environment

You must have the DEVICE statement for the communication port in the CONFIG.SYS file, and the SLIP.COM environment variable must be defined.

If you allow ICAT to modify your CONFIG.SYS file, ICAT adds the environment variable for you. When you installed OS/2, the DEVICE statement for the synchronous communications port was added to your CONFIG.SYS file in the default case.

The DEVICE statement and the environment variable are shown in the following example.

```
DEVICE=C:\OS2\COM02.SYS
SET SLIP.COM=COM1
```

For a SLIP.COM environment variable, any valid OS/2 communication port can be used.

Note: COM02.SYS is the device driver on a microchannel PS/2 running OS/2 1.3. COM01.SYS is the device driver on a non-microchannel PS/2 running OS/2 1.3. COM.SYS is the device driver on a PS/2 running OS/2 2.0.

Starting a SLIP Interface

The following steps describe how to start a SLIP interface.

1. Start the SLIO program. You can do this by entering `SLIO.EXE` at an OS/2 command prompt. The SLIOWAIT program in the BIN directory allows you to use the start slio program from a CMD file. If you use ICAT to configure SLIP this is your TCPSTART.CMD file by default, which is called from STARTUP.CMD.

   Note: After the SLIO program is running, it can not be stopped and restarted. You must reboot your machine to restart the program.

2. Start the SLIP interface by entering `ifconfig sl` *`ipaddress ipaddress`*

   Where:

   The first *ipaddress* is the IP address of the SLIP interface on your local host and the second *ipaddress* is the IP address of the SLIP interface on the remote host. The IFCONFIG statement defines a point-to-point link. If you use ICAT to configure SLIP this IFCONFIG statement is in your SETUP.CMD file. The IFCONFIG command with `sl` specified as the interface will have no effect unless SLIO is running.

Using the SLIPCALL Command

The SLIPCALL command communicates with a Hayes AT-compatible modem.

Note: If you are using a serial line (with a null-modem cable) you must use the OS/2 MODE command and set the characteristics to 8,n,1. For example:

```
mode com1:9600,8,n,1
```

If you need ICAT to configure SLIP, then the SLIPCALL command is invoked from the SLIP.CMD file. You can also invoke the SLIPCALL command from an OS/2 command prompt; however, it may require that you set the SLIP environment variables. The SLIP.CMD file and the SLIP environment variables are described in "Originating a SLIP Connection" on page 119.

The following example shows the format of the SLIPCALL command.

```
SLIPCALL [-?] [-r] [-a] [-d] [-s]
```

The parameters of the SLIPCALL command are:

Parameter	Description
-?	Displays help information.
-r	Resets the modem to its default settings, and sets Disable Echo (E0) and Terse Result Codes (V0), which are SLIPCALL requirements. The communication port speed is set to the value defined in SLIP.BPS, and the data bits, parity, and stop bits are set to 8, N, and 1, respectively. The -r parameter terminates an existing SLIP connection. This option also turns auto-answer off.

-a Places the modem in auto-answer mode.

-d Dials the modem by invoking the Hayes AT command that is defined by the SLIP.DIAL environment variable. The modem uses the value defined in SLIP.DELAY for the length of pause caused by a comma in SLIP.DIAL.

-s Shows, or displays:

- SLIP environment variable definitions for the current OS/2 session
- SLIP line characteristics
- Current modem signals
- Auto-answer status.

Originating a SLIP Connection

This section describes how to originate a SLIP connection. These steps assume that a remote host supporting SLIP is configured and ready to accept a SLIP connection. This also assumes that SLIO is running.

1. Enter the IFCONFIG statement and ROUTE statements necessary for a point-to-point link. You can enter this information by using ICAT or by editing the BIN\SETUP.CMD file. You must execute the BIN\SETUP.CMD file for the editing changes to take place. See Chapter 5, "Manually Modifying Your TCP/IP Configuration" for more information.

2. Enter the SLIP environment variable definitions by using ICAT or editing the BIN\SLIP.CMD file. The SLIP.CMD file contains definitions for the following environment variables:

Environment Variable	Description
SLIP.BPS	Specifies the baud rate for the modems being used.
SLIP.DELAY	Specifies the modem pause time defined for commas that appear in the dial string. The default pause time is 2 seconds.
SLIP.DIAL	Sends Hayes AT commands to the modem. Commands should include the telephone number. **Warning:** You can use the SLIP.DIAL environment variable to send any Hayes AT command to the modem. Do not use SLIP.DIAL to send the Hayes AT commands Enable Echo (E1) or Verbose Result Codes (V1), because SLIPCALL.EXE does work.

   Note: Running SLIP.CMD in an OS/2 session defines the SLIP environment variables for that specific OS/2 session. The SLIP environment variables are not stored for use in other OS/2 sessions.

   For an example of a SLIP.CMD file, see Appendix B, "Sample SLIP.CMD File."

3. Invoke the SLIP.CMD file to originate a SLIP connection. The SLIP.CMD file also calls SLIPCALL -rd. Enter the following command:

   ```
   SLIP
   ```

   The OS/2 host attempts to make a SLIP connection. The OS/2 host is restricted to one SLIP connection at a time.

Accepting a SLIP Connection

The following steps describe how to accept a SLIP connection.

1. Enter the IFCONFIG statement and ROUTE statements neccessary for a point-to-point link. You can enter this information by using ICAT, or by editing the BIN\SETUP.CMD file. See Chapter 5, "Manually Modifying Your TCP/IP Configuration" for more information.
2. Invoke the SLIPCALL command to accept a SLIP connection. Enter the following command:

   ```
   SLIPCALL -ra
   ```

   Your OS/2 host is ready to accept a SLIP connection. The OS/2 host is restricted to one SLIP connection at a time.

Ending a SLIP Connection

You can end a SLIP connection by entering the following command.

```
SLIPCALL -r
```

This procedure resets the modem, and allows you to originate another SLIP connection.

Chapter 12. Setting Up Your Kerberos System

Chapter 12. Setting Up Your Kerberos System

This chapter describes how to manually configure, customize, and verify the Kerberos Authentication System for TCP/IP for OS/2.

The Kerberos system in TCP/IP for OS/2 consists of the following functions:

- Kerberos database
- Authentication server
- Administration server
- Services application
- Client application
- Remote System Administrator.

Additional background information can be found in "Bibliography" and in *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference*. Additional commands and descriptions of the Kerberos functions can be found in *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

Setting Up the Environment

The Installation and Configuration Automation Tool (ICAT) program does not configure the Kerberos Authentication System. You must manually configure and customize the Kerberos system.

Before you set up the Kerberos system, the Kerberos services must be defined in the SERVICES file in the ETC directory. The Kerberos services are assigned to port 750 for TCP and UDP. You must create the following directories:

- KERBEROS
- TMP

Note: If you have used ICAT, the TMP directory may have already been created.

The KERBEROS directory must reside on the host that contains the Kerberos database.

The TMP directory must reside on the hosts that run client applications. The client's ticket files are stored in the TMP directory.

Environment Variables

If you do not create the KERBEROS and TMP directories in the root directory, you must set the following environment variables.

Environment Variable	Description
KERBEROS	Specifies the drive and directory of the KERBEROS directory. KERBEROS is an optional environment variable, except on the host that is running the Kerberos database.
TMP	Specifies the drive and directory of the TMP directory.

The Kerberos environment variables, KERBEROS and TMP, operate in the same way as the ETC environment variable when you created the ETC directory. For more information about environment variables, see Chapter 3, "Installing TCP/IP for OS/2."

For example, if you create your KERBEROS directory on the C drive from the TCPIP directory, you must add the following statement to your CONFIG.SYS file.

```
SET KERBEROS=C:\TCPIP\KERBEROS
```

If you create your TMP directory on the C drive from the TCPIP directory, you must add the following statement to your CONFIG.SYS file.

```
SET TMP=C:\TCPIP\TMP
```

The default for the KERBEROS directory is C:\KERBEROS. The default for the TMP directory is C:\TMP.

The following environment variables are also used in the Kerberos environment.

Environment Variable	Description
HOSTNAME	Specifies the name of the machine on which your host is running. The HOSTNAME must be able to be resolved; therefore, the same value (for the environment variable) must be defined on an accessible name server or in the HOSTS file. HOSTNAME is required.
TZ	The NFS control program (NFSCTL) uses the TZ environment variable to determine the correct date and time associated with a file.

The TZ Environment Variable

You must set the TZ environment variable, which is used to determine the correct date and time for clients located in different time zones. The TZ environment variable contains a string that has the abbreviation for your time zone, and the number of hours your time zone differs from Greenwich Mean Time (GMT).

The following example shows the format for the TZ environment variable.

SET TZ = *xxxnyyy*

The parameters of the TZ environment variable are:

Parameter	Description
xxx	The three-character label for your time zone.
n	The number of hours your time zone differs from GMT. If you are east of GMT, put a minus sign (-) before the number. If you are west of GMT, you can optionally put a plus sign (+) before the number. The *n* value ranges from -12 to +12.
yyy	The three-character label for your time zone when in daylight savings time. If you do not observe daylight savings time, leave this parameter blank.

Note: The default for the TZ environment variable is Eastern Standard Time (EST).

The following are examples of the TZ environment variable for different time zones.

Example	Description
EST5EDT	For the eastern coast of the United States. The time zone label is EST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is EDT. During standard time there is a five hour difference between local time and GMT.
CST6CDT	For the central time zone of the United States. The time zone label is CST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is CDT. During standard time there is a six hour difference between local time and GMT.
MST7	For the mountain time zone of the United States. The time zone label is MST. Daylight savings time is not observed. During standard time there is a seven hour difference between local time and GMT.
PST8PDT	For the western coast of the United States. The time zone label is PST when daylight savings time is not in effect. When daylight savings time is in effect, the time zone is PDT. During standard time there is an eight hour difference between local time and GMT.

The TZ environment variable should be included in your CONFIG.SYS file. You can let ICAT make this modification for you during configuration. If you change the TZ environment variable, reboot your computer so that the change takes effect.

KERBEROS Directory Files

To specify the Kerberos name of the remote system administrator, you must create the files in Table 1 on the host running the Kerberos database. The Kerberos remote system administrator can add, retrieve, and modify entries in the Kerberos database.

Table 1. KERBEROS Directory Files

Name of File	Contents of File	Sample of File
ADM_ACL.ADD	*administrator's_principal_name.instance@realm*	krbadm.admin@univ.educ.chem
ADM_ACL.GET	*administrator's_principal_name.instance@realm*	krbget.admin@univ.educ.bio
ADM_ACL.MOD	*administrator's_principal_name.instance@realm*	krbmod.admin@univ.educ.math

Note: When you create the contents of these files, you must define *instance* as `admin`. *Realm* is usually the domain name.

These KERBEROS directory files can contain multiple entries in the same format.

TMP Directory Files

You do not need to create any files in the TMP directory. Ticket files are written to the TMP directory when you use Kerberos.

All hosts running client applications must create the TMP directory and set the TMP environment variable to the path where the TMP directory resides.

ETC Directory Files

Table 2 contains the ETC directory files used with Kerberos. You must create the KRB.CNF file. The KRB.RLM file is optional. The ETC directory files used with Kerberos must reside in your ETC directory or in the directory specified by the ETC environment variable.

Table 2. ETC Directory Files

Name of File	Contents of File	Sample of File
KRB.CNF	*realm host_name* [admin server]	univ.educ.chem univ.educ.chem chrispc admin server univ.other.dept joanpc
KRB.RLM	*domain_name realm* or *host_name realm*	.educ.chem univ.educ.chem .educ.bio univ.educ.bio chrispc univ.educ.chem joanpc univ.educ.bio

The KRB.CNF file identifies the hosts that are running the Kerberos authentication server. The first line defines the local *realm* to which that host belongs. Each of the following lines specify the *realm* and *host_name* where the Kerberos server is running. `admin server` indicates that the host provides an administration database server. The *host_name* that you specify must also be defined in the name server or in the HOSTS file in your ETC directory.

The KRB.RLM file indicates the *realms* to which the hosts belong. To specify the *realm* for host names specified in fully qualified domain style, for example, `chrispc.univ.educ.chem`, you should use the format with domain specified in the first column of the KRB.RLM file, as shown in Table 2. To specify the *realm* for hosts with no recognizable domain, for example, `chrispc`, you should use the format with *host_name* in the first column of the KRB.RLM file.

If the KRB.RLM file does not exist and the host name is specified in fully qualified domain style, the *realm* is the host name's domain. For example, `univ.educ.chem`.

If the KRB.RLM file does not exist and the host name is of no recognizable domain, the Kerberos *realm* for that host is the local *realm* that is specified in the first line of the KRB.CNF file.

You are now ready to use the Kerberos commands to build and use your Kerberos database.

Building the Kerberos Database

The following Kerberos commands allow you to create and use the Kerberos database. Enter the Kerberos commands at an OS/2 command prompt to create and use the Kerberos database.

Command	Function
KDB_INIT	Used to build and format the Kerberos database.
KDB_UTIL	Used to load or dump the Kerberos database.
KDB_EDIT	Used to register users to the Kerberos database.
KADMIN	Used to add, retrieve or modify the Kerberos database. remotely.
EXT_SRTB	Used to generate key files for specified instances.

The following sections contain descriptions and examples of the Kerberos commands that are used to build the Kerberos database.

Creating the Kerberos Database—KDB_INIT

The following example shows the format of the KDB_INIT command.

KDB_INIT

There are no parameters for the KDB_INIT command.

The following steps describe how to create the Kerberos database:

1. Set the environment variable KERBEROS to the path on which your KERBEROS directory resides, on the host on which you want the KERBEROS server and Kerberos database to reside. For example:

   ```
   SET KERBEROS=C:\TCPIP\KERBEROS
   ```

   After the Kerberos database is created, the Kerberos database files are stored in the KERBEROS directory.

2. Type KDB_INIT at an OS/2 command prompt, and press the Enter key to create and initialize the Kerberos database files.

   The system prompts you for the local *realm*.

3. Type the name of the *realm* on which the Kerberos database resides at the realm prompt, and press the Enter key. YOUR_KRB.RE.ALM is the default.

   The system prompts you for the master password.

4. Type the master password at the password prompt, and press the Enter key.

   For verification purposes, the system prompts you again for the master password.

5. Retype the master password at the password prompt, and press the Enter key for verification.

   You need the master password to manage the Kerberos database. If the Kerberos database file already exists, the system indicates that the file already exists. After the Kerberos database files are initialized, an OS/2 command prompt is displayed.

When you create a Kerberos database, the following files are automatically created in the KERBEROS directory. Verify that these files reside in your KERBEROS directory.

- PRINC.DAT
- PRINC.IDX
- PRINC.OK

Loading and Dumping Your Kerberos Database—KDB_UTIL

The following example shows the format of the KDB_UTIL command.

KDB_UTIL *operation filename*

The parameters of the KDB_UTIL command are:

Parameter	Description
operation	Specifies `dump` or `load`.
filename	Specifies the name of the text file into which the database is dumped or loaded.

If you specify `dump`, the contents of the Kerberos database are dumped to the specified file name. Edit this dumped file with your System Editor to modify the contents of the database.

If you specify `load`, the contents of the file name are loaded to the Kerberos database.

Read locking of the database is done during the dump operation. No locking is done during a load operation. Loads should be performed with other processes shutdown.

The following is an example of using the KDB_UTIL command to examine the contents of the database. First dump the database to a file, ABC, by typing the KDB_UTIL DUMP ABC command; then display the file ABC by typing the command, TYPE ABC.

```
K M 255 1 1 0 6881 2e63 200001010459 199001082204 db_creation *
changepw os2 255 1 1 0 71d0 a1f7 200001010459 199001082204 db_creation *
default *255 1 1 0 0 0 200001010459 199001082204 db_creation *
krbtgt YOUR_KRB.RE.ALM 255 1 1 0 65f8 7291 200001010459 199001082204 db_creation *
```

Each line is a registered entry.

If the operation is successful, an OS/2 command prompt is displayed. If the operation is unsuccessful, an error message is displayed.

Registering a Kerberos User Locally—KDB_EDIT

The following example shows the format of the KDB_EDIT command.

```
KDB_EDIT
```

There are no parameters for the KDB_EDIT command.

KDB_EDIT is used to register users and services to the Kerberos database.

A user's *instance* and a service's *instance* are usually specified by the following rules:

- A user's *instance* is optional. Users with privileges, for example, the remote system administrator, should register with an *instance* of `admin`. Users without privileges have an *instance* of `null`.
- A service's *instance* is usually the host name where the service is running.
- An *instance* should not exceed 8 characters in length.

The system prompts you for the Kerberos master password and for information about the user and service that you want to add. The following is an example of running KDB_EDIT.

```
Opening database...
password: <krb_pw>

Verifying, please re-enter
Enter Kerberos master: password: <krb._pw>

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in 'brackets'
enter return to leave the same, or new value.

Principal name: <username>
Instance: <enter>
<Not found>, Create 'y' ? <y>
Principal: username, Instance: , kdc_key_ver: 1
password: <userpassword>

Verifying, please re-enter
New Password: <userpassword>


Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) '1999-12-31' ? <enter>
Max ticket lifetime (*5 minutes) '255 - ? <enter>
Attributes '0 - ? <enter>
Edit O.K.
Principal name: <enter>
```

The following steps describe how to register a Kerberos user or service locally.

1. Type KDB_EDIT at an OS/2 command prompt, and press the Enter key.

   The system prompts you for the Kerberos master password.

2. Type the master password at the password prompt, and press the Enter key.

   If the master password is valid, KDB_EDIT allows you to register a user or a service.

   The system prompts you for the *principal name*.

3. Type the user name or service name at the principal name prompt, and press the Enter key.

   Note: The services that you register must also be defined in the SERVICES file in your ETC directory.

   The system prompts you for the *instance*.

4. If you are registering a user, press the Enter key for a null *instance*. If you are registering a remote administrator, type `admin`, and press the Enter key. If you are registering a service, at the instance prompt, type the name of the host where the service resides, and press the Enter key.

   - If the user or the service already exists in the database, the system asks you if you want to change the password.
   - If the user or the service does not exist in the database, the system prompts you for the expiration date, ticket lifetime for the user or services, and the attribute. Defaults are provided for these prompts. If you want to use the default values, press the Enter key. Otherwise, specify the desired value and press the Enter key.

The following message is displayed if the user or service is registered:

```
Edit O.K.
```

The system prompts you for the next entry.

5. You can press the Enter key at the principal name prompt to exit the registration process or repeat steps 3–5 to register the next Kerberos user name or Kerberos service name that you want to establish.

To list the entries in your Kerberos database, use the KDB_UTIL DUMP command. See "Loading and Dumping Your Kerberos Database—KDB_UTIL" on page 127 for more information about listing the entries in your Kerberos database.

You must inform the user of the Kerberos name and password that you assign to them.

Registering a Kerberos User Remotely — KADMIN

To modify the Kerberos database remotely, you must use the KADMIN command and have the remote administration server, ADM_SERV, running on the machine that contains the Kerberos database. See "Kerberos Administration Server" on page 135 for information about setting up the Kerberos administration server.

You can add, retrieve, or modify the database using the KADMIN command only if your *instance* is specified to have `admin` authority in the ADM_ACL.ADD, ADM_ACL.GET, and ADM_ACL.MOD files that reside in the KERBEROS directory, and if you are registered in the Kerberos database with *instance* as `admin`. The KADMIN command can only be used to add, retrieve, or modify a Kerberos user with *instance* as `null`. The only exception is that you, as a remote administrator with *instance* as `admin`, can change your own password.

The following example shows the format of the KADMIN command.

```
KADMIN
```

There are no parameters for the KADMIN command.

Type KADMIN at an OS/2 command prompt, and press the Enter key to start the KADMIN utility.

At the "Your userid" prompt, type the administrator's *principal name*.

The following message is displayed.

```
Welcome to the Kerberos Administration Program, Version X
Type "help" if you need it.
admin:
```

At the admin prompt, type ? to list the available subcommands. To obtain instructions on how to use these subcommands, type `HELP` followed by the subcommand you want to use, and press the Enter key.

The subcommands of the KADMIN command are:

Subcommand	Function
ADD_NEW_KEY	Registers a new *principal name* with the Kerberos database. Requires one argument, the *principal name*. The shorter term ANK can also be used.
CHANGE_PASSWORD	Changes the Kerberos password for the *principal name*. Requires one argument, the *principal name*. The shorter term CPW can also be used.
CHANGE_ADMIN_PASSWORD	Changes your ADMIN *instance* password. Requires no arguments. The shorter term CAP can also be used.
LIST_REQUESTS	Displays a list of possible subcommands. The shorter term LR or ? can also be used.
GET_ENTRY	Gets you an entry into the Kerberos database for review. At the password prompt, type your administrator password, and press the Enter key. The shorter term GET can also be used.
HELP *command name*	Displays help messages for KADMIN. If you enter this subcommand without an argument, a general help message is displayed.
QUIT	Terminates the Kerberos program. The term EXIT can also be used.

To manage the Kerberos database remotely, you must first complete the following steps on the host on which the Kerberos database resides.

1. Register the remote administrator with the Kerberos database.

   The administrator can choose the *principal name* and password. However, the *instance* must be `admin`. The remote administrator's password is requested when you use the KADMIN utilities.

   You should also add the system administrator's full Kerberos name to the ADM_ACL.ADD, ADM_ACL.GET and ADM_ACL.MOD files in the KERBEROS directory on the host that contains the Kerberos database. See "KERBEROS Directory Files" on page 125 for the line format and sample contents of the KERBEROS directory files.

2. Type ADM_SERV at the password prompt, and press the Enter key to start the remote administration server.

   The system prompts you for the Kerberos master password.

3. Type the master password at the password prompt, and press the Enter key.

   If the remote administration server starts successfully, the following message is displayed:

```
Current Kerberos master key version is 1.

Master Key entered.  BEWARE!
```

Note: The remote administration server is one of the services that the Kerberos database provides. The remote administration server has already been registered as `changepw` when you started the Kerberos database. Therefore, you do not have to register the remote administration server.

On the remote host on which you want the remote administration utility, KADMIN, to run:

1. Type KINIT -irv at an OS/2 command prompt, and press the Enter key to get the initial ticket.

   The system prompts you for the Kerberos *principal name*.

2. Type the *principal name* that you registered locally in the Kerberos database at the principal name prompt, and press the Enter key.

   The system prompts you for the *instance*.

3. Type the *instance* `admin` that you registered locally in the Kerberos database at the instance prompt, and press the Enter key.

   The system prompts you for the *realm* name.

4. Type the name of the *realm* where the Kerberos database resides at the realm prompt, and press the Enter key.

   See *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for more information about the KINIT command.

5. Type KADMIN at an OS/2 command prompt to start the KADMIN utility, and press the Enter key.

Generating the Key File for an Instance—EXT_SRTB

The following example shows the format of the EXT_SRTB command.

EXT_SRTB *instance* [*instance*]

The EXT_SRTB command is used by the host running the Kerberos database to generate key files for hosts providing services. The instance is the *instance* whose key file is to be generated. Multiple *instances* can be specified on the same command line.

The following steps describe how to generate the key files for *instances*.

1. Type EXT_SRTB, followed by at least one *instance* at an OS/2 command prompt, on the host running the Kerberos database, and press the Enter key.

   The system prompts you for a password.

2. Type the Kerberos master password at the password prompt, and press the Enter key.

   For verification purposes, the system prompts you again for the master password.

3. Retype the master password at the password prompt, and press the Enter key for verification.

   If the password is correct, the system searches through the Kerberos database entries for each of the specified *instances*. For example, if you type EXT_SRTB `inst1`, the system searches through the Kerberos database, entries for the speci-

fied *instance* of `inst1`. When the system finds `inst1`, a key file INST1.STB is generated.

The following is an example of a key file INST1.STB.

```
[C:\] ext_srtb inst1 inst2
Password: <krb_pw>
Re-enter Verifying:
Enter master password: <krb_pw>

Master Key entered. BEWARE!

generating inst1.stb
generating inst2.stb
[C:\]
```

Copy each *instance*.STB key file to the ETC directory of the corresponding instance's host and rename it to SRVTAB.

4. Use the KLIST -srvtab command to see the contents of this key file. See *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for the usage of the KLIST command.

Setting Up the Kerberos Servers

This section describes how to set up the following Kerberos servers:

- Authentication Server
- Administration Server.

See *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for additional information about the authentication server, which includes the ticket-granting server.

Kerberos Authentication Server

You must start the Kerberos authentication server before you can use the Kerberos system. The authentication server provides a way for authenticated users to prove their identity to other servers across a network. The authentication server reads the Kerberos database to verify that the client making the request is the client named in the request.

Note: The authentication server must run on the same host as the Kerberos database.

The following files must reside in your ETC directory or the directory specified by the ETC environment variable.

- KRB.CNF
- SERVICES
- KRB.RLM (optional)

The local *realm* specified in the first line of the KRB.CNF file must be the same as the *realm* name that is specified when running the KDB_INIT command. The service `kerberos` must be defined in the SERVICES file.

To start the Kerberos authentication server, use the KERBEROS command.

The following example shows the format of the KERBEROS command.

```
KERBEROS -m
```

The only parameter of the KERBEROS command is:

Parameter	Description
-m	Prompts you to manually enter the Kerberos master password.

To start the authentication server, type KERBEROS -m at an OS/2 command prompt, and press the Enter key.

The following screen is an example of running the KERBEROS command:

```
Kerberos server starting
 Sleep forever on error
 Master key will be entered manually
 Log file is \kerberos\kerberos.log
password: <krb_pw>

Verifying, please re-enter
Enter Kerberos master password:  <krb_pw>

Current Kerberos master key version is 1.

Master key entered.  BEWARE!

Current Kerberos master key version is 1
Local realm:  univ.educ.bio
```

The system prompts you for a password.

Type your Kerberos master password at the password prompt, and press the Enter key.

For verification purposes, the system prompts you again for the master password.

Retype the master password at the password prompt, and press the Enter key for verification.

KERBEROS.EXE starts and runs until explicitly stopped.

A log file called KERBEROS.LOG is created in the KERBEROS directory. All transactions done by the Kerberos authentication server are recorded in the KERBEROS.LOG file.

Note: Because Kerberos continuously appends, you should periodically check the size of the KERBEROS.LOG file.

You can now use the Kerberos system. See *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for a description of the Kerberos commands.

Kerberos Administration Server

The Kerberos administration server allows you to modify the Kerberos database remotely.

Note: The administration server must run on the same host as the Kerberos database.

To start the administration server, type ADM_SERV at an OS/2 command prompt, and press the Enter key.

The password prompt is displayed.

Type your Kerberos master password at the password prompt, and press the Enter key.

ADM_SERV starts and runs until explicitly stopped.

A log file called ADM_SRV.LOG is created in the KERBEROS directory. All transactions done by the ADM_SERV are recorded in the ADM_SRV.LOG file.

Note: Because Kerberos continuously appends, you may want to periodically check the size of the ADM_SRV.LOG file.

You can now run the KADMIN utility on a remote host. See "Registering a Kerberos User Remotely — KADMIN" on page 130 for further information about the KADMIN command.

Setting Up a Service and Client Application

The services and clients in the application programs in the Kerberos system must be able to handle the authentication procedure. You can develop your Kerberos services and client's applications using the SAMPLE_S and SAMPLE_C programs, which are provided in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference*.

Setting Up a Service Application

The following steps describe how to set up a service application.

1. Add the service name to the file SERVICES in the ETC directory.

   The services provided in the Kerberos system should be able to support the authentication procedure. Hosts running the client application and the service application must have these entries in their SERVICES file.

2. On the host where the Kerberos database resides, use KDB_EDIT to register the service name with the Kerberos database locally. Use the service name as the *principal* and the host name where the service is running as the *instance*. For example, `ftp` is the *principal* and the host name where the service is running, `chrispc` is used as the *instance*. Also provide a password for the service you register. The password is converted to the key for the service.

   After you register all the services provided by the same host `chrispc`, at an OS/2 command prompt, type EXT_SRTB instance. For example, EXT_SRTB chrispc generates the file CHRISPC.STB. CHRISPC.STB is the file in which the server's keys are stored. The EXT_SRTB program resides in the host running the Kerberos database.

3. Transfer the key file to the host providing the services (for example, chrispc). Rename the key file to SRVTAB and put it in the ETC directory.
4. You can now start the service on the host providing the services (for example, chrispc).

Setting Up a Client Application

The following steps describe how to set up a client application.

1. Register the client with the Kerberos database locally using KDB_EDIT or remotely using the KADMIN function.
2. Verify that the services you want are in the SERVICES file in the ETC directory. Include the server host name in the name server or in the HOSTS file in your ETC directory.
3. Edit the KRB.CNF file in your ETC directory to show the hosts on which the Kerberos authentication servers are running.
4. At an OS/2 command prompt, type KINIT to logon to the Kerberos authentication server. See *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for more information on the KINIT command.

   The KINIT command logs you onto the KERBEROS server and gets the initial ticket in the TKT0 ticket file in the TMP directory on your host.
5. You can now start your Kerberos client application. If the ticket for the service you want is not in the ticket file, the client should obtain a service ticket for the service you want from the Kerberos authentication server by showing the initial ticket using the programming interface routines. See *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for more information.
6. Use the KLIST command to see the list of the tickets in the client's ticket file.
7. Use the KDESTROY command to delete the ticket file.

   See *IBM TCP/IP Version 1.2 for OS/2: User's Guide* for the format and description of the parameters of the KLIST and KDESTROY commands.

Example of Verifying the Kerberos Configuration

TCP/IP for OS/2 provides a sample application client program and a sample application server program that can be used to verify your Kerberos installation and configuration.

This section contains examples of passwords and the following tasks that are used to set up, run, and verify your Kerberos system.

- Setting up the environment
- Creating the Kerberos database
- Starting the Kerberos authentication server
- Registering the sample service and the user
- Generating the key file for the sample service
- Transferring the service key file to the server
- Starting the sample server
- Getting the initial ticket
- Running the sample client program.

Additional background information about the Kerberos functions is specified in the following sections.

Step 1: Setting Up the Environment

The Kerberos database and authentication server must reside on the same host. The SAMPLE_S and SAMPLE_C programs can run on the same host as the Kerberos database and the authentication server, or the programs can run on other hosts.

In this example, the hosts are assigned the following names:

Host Name	Description
Service	Specifies the name of the host on which the SAMPLE_S server is running.
Client	Specifies the name of the host on which the SAMPLE_C program is running.
Master	Specifies the name of the host on which the authentication server and the Kerberos database are running.

You should verify that the following files reside on the specified host:

File	Host
SAMPLE_C.EXE	Client
SAMPLE_S.EXE	Service
KERBEROS.EXE	Master
KDB_INIT.EXE	Master
KDB_EDIT.EXE	Master
EXT_SRTB.EXE	Master

In the following steps, replace the word service with the name of the appropriate host on which you are verifying the configuration.

See "Setting Up the Environment" on page 123 for additional environment requirements.

Step 2: Creating the Kerberos Database—KDB_INIT

The following steps describe how to create the Kerberos database:

1. On the master host, type KDB_INIT at an OS/2 command prompt, and press the Enter key to create and initialize the Kerberos database files.

   The system prompts you for the local *realm*.

2. Type the name of the local *realm* that you defined in your KRB.CNF file at the realm prompt, and press the Enter key.

   The system prompts you for the master password.

3. Type krbpass at the password prompt, and press the Enter key.

   For verification purposes, the system prompts you again for the master password.

4. Retype krbpass at the password prompt, and press the Enter key for verification.

   After the Kerberos database files are initialized, an OS/2 command prompt is displayed.

See "Creating the Kerberos Database—KDB_INIT" on page 127 for additional information about the Kerberos database.

Step 3: Starting the Kerberos Authentication Server

To start the authentication server, on the master host, type KERBEROS -m at an OS/2 command prompt, and press the Enter key.

The system prompts you for a password.

Type krbpass at the password prompt, and press the Enter key.

For verification purposes, the system prompts you again for the master password.

Retype krbpass at the password prompt, and press the Enter key for verification.

KERBEROS.EXE runs until explicitly stopped.

See "Kerberos Authentication Server" on page 133 for additional information about the authentication server.

Step 4: Registering the Sample Service and the User

The following steps describe how to register a sample service and user locally.

1. On the master host, type KDB_EDIT at an OS/2 command prompt, and press the Enter key.

   The system prompts you for the Kerberos master password.

2. Type krbpass at the password prompt, and press the Enter key.

   For verification purposes, the system prompts you again for the master password.

3. Retype krbpass at the password prompt, and press the Enter key for verification.

   If the master password is valid, KDB_EDIT allows you to register a service.

   The system prompts you for the *principal name*.

4. Type `sample` at the principal name prompt, and press the Enter key.

   Note: The services that you register must also be defined in the SERVICES file in your ETC directory.

   The system prompts you for the *instance*.

5. Type `service` at the instance prompt, and press the Enter key.

   The message `not found` is displayed and the system prompts you for the Create `'y'`.

6. Press the Enter key at the Create `'y'` prompt to use the default value.

   The system prompts you for the password.

7. Type `sam` at the password prompt, and press the Enter key.

   For verification purposes, the system prompts you again for the password.

8. Retype `sam` at the password prompt, and press the Enter key for verification.

9. Press the Enter key to use the default values for the remaining prompts.

   The following message is displayed if service, `sample`, is registered:

   ```
   Edit O.K.
   ```

   The system prompts you for the *principal name*

10. Type `user1` at the principal name prompt, and press the Enter key.

    The system prompts you for the *instance*.

11. Press the Enter key for a null *instance* at the instance prompt.

    The message `not found` is displayed and the system prompts you for the Create `'y'`.

12. Press the Enter key at the Create `'y'` prompt to use the default value.

    The system prompts you for the password.

13. Type `use` at the password prompt, and press the Enter key.

    For verification purposes, the system prompts you again for the password.

14. Retype `use` at the new password prompt, and press the Enter key for verification.

15. Press the Enter key to use the default values for the remaining prompts.

    The following message is displayed if user, `user1` is registered:

    ```
    Edit O.K.
    ```

    The system prompts you for the *principal name*

16. Press the Enter key at the principal name prompt to exit the registration process.

See "Registering a Kerberos User Locally—KDB_EDIT" on page 128 for more information about the KDB_EDIT command.

Step 5: Generating the Key File for the Sample Service

The following steps describe how to generate the key file, service.STB for the sample service.

1. On the master host, type EXT_SRTB service at an OS/2 command prompt, and press the Enter key.

   The system prompts you for a password.

2. Type krbpass at the password prompt, and press the Enter key.

   For verification purposes, the system prompts you again for the master password.

3. Retype krbpass at the password prompt, and press the Enter key for verification.

   The system searches through the Kerberos database entries for the specified *instance* of service. When the system finds service, a key file SERVICE.STB is generated.

   An OS/2 command prompt is displayed.

   See "Generating the Key File for an Instance—EXT_SRTB" on page 132 for further information about the EXT_SRTB command.

Step 6: Transferring the SERVICE.STB Key File to the Server

The following steps describe how to transfer the SERVICE.STB key file to the server host.

1. Use the FTP program to transfer the SERVICE.STB key file from the master host to the service host as a binary transfer.
2. Rename the key file to SRVTAB and put it in the ETC directory.

Step 7: Starting the Sample Server

The following steps describe how to start the sample server, SAMPLE_S.

On the service host, type SAMPLE_S at an OS/2 command prompt, and press the Enter key to start the sample Kerberos service application.

The cursor should move to the next line to wait for requests from the sample client program.

SAMPLE_S.EXE runs as a task until you shut down the server.

Step 8: Getting the Initial Ticket

The following steps describe how the user gets the initial ticket.

1. On the client host, type KINIT at an OS/2 command prompt, and press the Enter key to get the initial ticket.

   The system prompts you for the Kerberos name.

2. Type user1 at the Kerberos name prompt, and press the Enter key.

   The system prompts you for a password.

3. Type use at the password prompt, and press the Enter key.

   If the KLIST command is successful, an OS/2 command prompt is displayed and the TKT0 ticket file is generated in the TMP directory on the client host.

Step 9: Running the Sample Client Program

To start the sample Kerberos client application program, type SAMPLE_C service 100 at an OS/2 command prompt on the client host and press the Enter key.

If all parts of your Kerberos system (environment, Kerberos database, Kerberos authentication server, and client and service applications) are set up correctly, a message is displayed as a return from the SAMPLE_S program. The following is an example of the message that might be displayed:

```
The server says:
You are user1.@UNIV.DEPT.BIO (local name user1),
at address 9.67.43.74, version VERSION X, cksum 100.
```

Chapter 13. Setting Up the X Window System Server

Chapter 13. Setting Up the X Window System Server

The X Window System is a distributed, window-based graphics system developed at Massachusetts Institute of Technology. TCP/IP Version 1.2 for OS/2 supports Version 11 Release 4 (X11R4) of the X Window System server (X server) function.

The OS/2 X server enables users to display and control X Window System client (X client) application programs in an OS/2 Presentation Manager (PM) windowed session. These client application programs can reside in one or more IBM or other computing systems that support the X client function. They are connected to the OS/2 X server host through a TCP/IP network. IBM systems that currently have X client capability include VM, MVS, and AIX (RISC/6000, PS/2, and S/370).

The X server function uses OS/2 PM as the X window manager and supports all of the keyboard, display, and pointer devices that are supported by OS/2 PM. Using PM as the X window manager enables OS/2 PM windowed applications and X client applications to share the same screen. As a result, another window manager (for example, aixwm or mwm) cannot act as the window manager for the OS/2 X server.

Setting up the X Server Support

The TCP/IP V1.2 for OS/2 X Window System server support is composed of the following components:

- OS/2 X Server Software (PMX.EXE)
- X Font Support
- X Clients and Utilities.

All the components are installed when you select *X Window System Server* from the ICAT install menu. ICAT installs the required files into the TCP/IP for OS/2 product directory structure by default. Many X server database files required by the X server software for initialization and operation are installed in a subdirectory named X11, by default. If you allow ICAT to update to your CONFIG.SYS file, ICAT sets an environment variable called XFILES to the X11 subdirectory path. The X server software, PMX.EXE, references the XFILES environment variable to locate the database files. If XFILES is not set properly, PMX.EXE fails to execute.

Note: The X11 subdirectory tree can be located wherever you prefer, as long as the corresponding environment variable, XFILES, is set accordingly. In this book, the location of the X11 subdirectory is represented as X11. For a particular configuration, X11 represents the directory to which the XFILES environment variable points.

After installing the X server support using ICAT, you can choose to do one or more of the following to customize the X Window System server to your requirements:

- Configure options from ICAT:
  - Automatically start the X server when the OS/2 system starts
  - Create or modify the X0HOSTS file
  - Set the DISPLAY environment variable for the X client utilities.
- Install Additional X-Fonts
- Modify the Color Database.

The following sections describe the procedures listed above. For information about starting the server, see "Starting the X Server" on page 147. For information about the X0HOSTS file, see "The X Client Host Authorization File (X0HOSTS)" on

page 146. For information about how to install additional X fonts, see "Installing Additional X Fonts" on page 150. For information about how to modify the color database, see "The Color Database (RGB.TXT)" on page 147.

Note: All of these actions are optional, and you can go back and perform any of these steps at a later time. If you do perform any of these customization procedures later, restart the X server software to activate the changes.

Once you have completed customization, reboot the machine to activate changes that ICAT makes to your CONFIG.SYS. If you chose to automatically start the server from the ICAT menu, PMX will start when the system restarts. If PMX does not start, see "Starting the X Server" on page 147.

The X Server (PMX.EXE)

The following section describes the X server files and gives information about the OS/2 X server PMX.EXE.

Files Used by the X Server

The X server accesses a number of files during initialization and operation. Some of these files are used to define screen colors, authorize client host access, and create font images. ICAT installs these files into the TCP/IP for OS/2 product directory structure in a subdirectory named X11 by default and sets an OS/2 environment variable, XFILES, to point to it. The PMX.EXE server software references the value of XFILES to locate the files. (An exception to this is the X0HOSTS file, which is installed into the directory pointed to by the ETC environment variable.)

By convention in this book, X11 represents the directory path set by the XFILES environment variable. This is normally the directory TCPIP\X11 on the drive where the TCP/IP for OS/2 software is installed. If the XFILES environment variable is not set correctly, PMX.EXE fails to execute.

The X Client Host Authorization File (X0HOSTS)

PMX uses the X0HOSTS file to identify the X client hosts on the network that are authorized to connect to the X server. This file is not supplied with the server, but you can create it using ICAT or a text editor. The X0HOSTS file for PMX corresponds to the X0.hosts file on a UNIX host running the X server.

In the X0HOSTS file, you should have only one host name for each line. The names in X0HOSTS should be either names recognized by your TCP/IP network domain name server or names included in your ETC\HOSTS file. Use only names, no internet addresses. The X0HOSTS file does not have the same format as the TCPIP\ETC\HOSTS file. The following is an example of the X0HOSTS file:

```
cambridg
cambvm3
als-ps2
als-at
```

The file X0HOSTS can reside in the directory named in the ETC environment variable. This environment variable normally points to TCPIP\ETC on the drive on which TCP/IP resides. If the X0HOSTS file cannot be found in the directory named by the ETC environment variable, then a client cannot access the server, unless the XHOST utility is used to permit access.

The Color Database (RGB.TXT)

By default, the color database is found in the file X11\RGB.TXT. On a UNIX Window System server, this file is used as the source of a small relational database. PMX uses this file directly, reading it into memory and keeping it sorted by color name. The X11\RGB.TXT file shows a mapping between a large number of color names and some RGB (Red-Green-Blue) values. There is more than one example color file included with the server. You can customize the color database by editing RGB.TXT with a text editor and changing the RGB values associated with a particular color name. You can also add color names and associated RGB values to create new colors in the database. PMX uses information from the Presentation Manager to map RGB values into indexes to Presentation Manager color tables.

X Font Files

X font files are used by the X server to create text images for X client applications. The OS/2 X server program includes X fonts supplied from the MIT X11R4 distribution, as well as IBM AIX X Windows products. These files have a file extension of .SNF, for Server Natural Format, and reside in the X11\MISC and X11\75DPI subdirectories.

By default, PMX.EXE searches for the X font files in X11\MISC, then in X11/75DPI. You can change the default font directories and search order by using the -fp option of PMX.EXE. You can also dynamically override the defaults by using the fp parameter of the XSET command.

For more information about X font support for the OS/2 X server, see "X Font Support" on page 148.

Starting the X Server

Start the X server from a PM group menu, from an OS/2 command line, or from a batch (CMD) file.

The following example shows the format of the PMX command.

PMX [*parameters...*]

The parameters for the PMX command are:

Parameter	Description
-a *n*	Sets mouse acceleration (*n* is the number of pixels).
-co *filename*	Sets color database file name. The default filename is X11\RGB.TXT.
-fc *fontname*	Sets cursor font. The default is the font named `cursor`.
-fn *fontname*	Sets default font. The default is the font named `fixed`.
-fp *pathname*	Sets default font path. The default is X11\MISC,X11\75DPI
-help	Displays help information (will not start the X server).
-lc	Doubles the dimensions of any cursor, unless the cursor would become too large for a PM cursor.
-nocopyright	Does not display initial copyright window when starting the server.

-r	Turns off the keyboard auto-repeat function.
r	Turns on the keyboard auto-repeat function. This is the default.
-t *n*	Sets mouse threshold (*n* is the number of pixels).
-to *n*	Sets connection time-out (*n* is the number of seconds).
-I	Ignores all remaining arguments.

In the case of the font path (-fp *pathname*), multiple directories are separated by commas, rather than blanks. For example, to start the server with both the miscellaneous and the 75 dot per inch (dpi) font directories, as well as your own personal fonts directory C:\myfonts, specify the following command parameters:

```
pmx -fp d:tcpip\X11\misc,d:tcpip\X11\75dpi,c:\myfonts
```

This example assumes that you installed TCP/IP on drive d: in the \tcpip directory.

For more information about X fonts, see "X Font Support" on page 148. For more information about the color database and font files, see "The Color Database (RGB.TXT)" on page 147 and "How the OS/2 X Server Accesses Fonts" on page 149.

Implementation Notes

The READ.ME file on the installation diskettes provides information about PMX implementation, enhancements, and restrictions. The following describes additional information about the X Window System server.

- The X server is ported from the MIT X Consortium X11R4 distribution.
- Because PM is the window manager, you cannot use other window managers.
- For OS/2 machines that have a pointing device with only two buttons, the middle button is simulated by pressing both the right and left buttons simultaneously.
- Save unders and backing store are currently not supported.
- No X extensions are currently supported in the OS/2 X server, including the shape extension included with the MIT Consortium X11R4 distribution.
- PMX supports only StaticColor and StaticGray visuals, so the color tables have only read-only entries.

X Font Support

The following sections describe the X font support supplied with the OS/2 X server.

Fonts Included with the OS/2 X Server

The X font files included with the OS/2 X Server support are located in the X11\MISC and X11\75DPI subdirectories. The fonts in these directories were supplied from IBM AIX X Windows products, as well as from the MIT X Consortium X11R4 distribution.

X fonts that your client applications need, but are not included with OS/2 can be imported from other systems for use with the OS/2 X server. For information about enhancing the OS/2 X server font support with X fonts from other systems, see "Installing Additional X Fonts" on page 150.

How the OS/2 X Server Accesses Fonts

X font files are files with a file name extension of .SNF. The X Server locates and accesses font files by using the FONTS.DIR and FONTS.ALI files of the directories in the font path. There must be exactly one FONTS.DIR file and zero or one FONTS.ALI file in each directory in the font path. The FONTS.DIR and FONTS.ALI files are used to map X font names, as specified to the X server by requesting X client applications, to X font file names known to the local X server. The mapping of X font names to file names removes the need for the X clients to know about the local file system characteristics and naming conventions.

The Font Search Path

Font information is found along a search path defined to the server. The font path can be changed dynamically as the server runs by using the XSET utility or appropriate X Window System protocol messages from clients. The server searches for a file named FONTS.DIR and a file named FONTS.ALI in each directory in the font path. These files are used to map font names into font files in that directory.

Additional directories with FONTS.DIR and FONTS.ALI files can be specified. For example, to start the X server with both the miscellaneous and the 75 dpi fonts, as well as your own personal fonts in a directory c:\myfonts, specify the following command:

```
pmx -fp d:tcpip\X11\misc,d:tcpip\X11\75dpi,c:\myfonts
```

The FONTS.DIR Files

FONTS.DIR files are created using the MKFONTDR program after you compile the fonts. For more information on the MKFONTDR program, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*. The FONTS.DIR files contain a list of the font files in a directory, and the font names, that are read from the font files. Recent fonts created with conventions that have become standard for the X Window System since Release 3 have names that are long and contain information about the font. Wild card characters can be used to simplify the search for a font when a client program specifies a font. You can use either upper or lower case in your font patterns because the case of letters in font names is ignored.

The first line of the FONTS.DIR file contains an integer specifying the number of fonts. The rest of the lines in the file contain two pieces of information, separated by one or more blanks. The first item on each line is the OS/2 file name. An example file name would be 9X15.SNF. The second item on each line is the font name, which is known to the X server. An example font name would be 9x15. The font name is extracted from information in the font file, and is not derived from the OS/2 file name.

The FONTS.ALI Files

The file FONTS.ALI (in UNIX this would be FONTS.ALIAS) that can be put in any directory of the font path, is used to map new names to existing fonts, and should be edited by an editor. Each line in the file normally contains two items, separated by one or more blanks. The first item is the alias you use for a font. The second contains a font-name pattern.

To embed white space in either the alias or the font name pattern, enclose either of them in double-quote marks; to embed a double-quote mark (or any other special character), precede the character with a backslash.

When a font alias is specified by a client, the name it references is searched for by looking through each font directory (FONTS.DIR file). This means that the aliases do not have to reference fonts in the same directory as the FONTS.ALI file.

If the string FILE_NAMES_ALIASES stands alone on a line in the FONTS.ALI file, each filename in the directory (stripped of its .SNF extension) is used as an alias for that font.

Each directory in the font path can have a FONTS.ALI file. When PMX is installed, a FONTS.DIR and FONTS.ALI file will be installed in each font directory.

Installing Additional X Fonts

You can use X fonts, which are not included with the TCP/IP for OS/2 X server support. For example, you can have an X client application which requires a special font that is available on a UNIX or AIX machine but is not included with the OS/2 support. In this case, you can install the needed fonts on the OS/2 machine for use by the OS/2 X server software.

To install additional X fonts, locate the X font source file for the font you wish to install. Typically, X font source files have a file extension of BDF (such as COURB08.BDF). Once you have located the font source file, copy it to a drive on your OS/2 machine.

Next, compile the font source file using the BDFTOSNF X font compiler utility. BDFTOSNF takes the .BDF font source file as input and creates a new file in server natural format. This new file has a file extension of SNF (such as COURB08.SNF) and contains information that can be used by the OS/2 X server to create character images for display in X windows. All fonts that are used with the OS/2 X Server must be compiled using the BDFTOSNF utility. Fonts compiled with the BDFTOSNF X utility on other operating systems will not create a .SNF file which is usable by the OS/2 X server. For more information about compiling X fonts and the BDFTOSNF X utility program, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

After the font has been compiled, the resulting .SNF file must be placed in a directory where the OS/2 X server can find it. You can choose to put the file in a default X font directory (for example, TCPIP\X11\MISC) or in another directory. If you choose to put the .SNF file in a directory other than one in the default font path, you need to identify that directory to the X server by using the font path (-fp) option when you start PMX. See "Starting the X Server" on page 147 or "How the OS/2 X Server Accesses Fonts" on page 149 for more information on specifying font paths.

Finally, after you have compiled the X font and moved the file to the chosen directory, you must run the MKFONTDR utility on that directory. MKFONTDR creates or updates the FONTS.DIR file in the directory. The FONTS.DIR file is used by the OS/2 X server to map a particular font name, as requested by an X client application, to a font file in the same directory.

The installation of the additional font is now complete. To activate these changes, you can either restart the X server or use the rehash option (fp rehash) of the XSET utility. Either of these actions will cause the server to re-read the FONTS.DIR files in the directories of the current font paths.

The following list summarizes the steps needed to load additional fonts for use by the OS/2 X server:

1. Locate the X font source file and copy the file to your OS/2 machine.
2. Run BDFTOSNF on the source file to create the server natural format (SNF) font file useable by the X server.
3. Copy the resulting .SNF file to a default font directory or a directory known to the server (by using the -fp option of PMX.EXE).
4. Run the MKFONTDR utility on that directory to update the FONTS.DIR file.
5. Restart the X server or issue "XSET fp rehash" to activate changes to the FONTS.DIR file.

X Clients and Utilities

The following X client and utility programs are included with the OS/2 X Server software to help in administering and customizing the PMX server. The X clients are based on the MIT Consortium X11R2 distribution, and are intended for use with the OS/2 X server only. All the programs listed below that start with an X are X clients.

Program	Description
BDFTOSNF	Font compiler
MKFONTDR	Font directory builder
SHOWSNF	Font file display utility
XFD	Font display utility
XHOST	Client host access control utility
XLSFONTS	Font listing utility
XMODMAP	Keyboard modifier utility
XSET	Set user preference utility
XWININFO	Window information utility.

Note: All the X client utilities rely on the user-defined OS/2 environment variable DISPLAY to determine which X server to use by default. ICAT optionally inserts a statement into your CONFIG.SYS that sets the value of DISPLAY when the OS/2 system boots. You should specify the value for DISPLAY in the ICAT Configure Services menu.

The value for DISPLAY has the format `<Host>:<Dpy>`. Host must be either an internet host address or a host name that can be resolved to an internet address. A host name can be resolved to an internet address through queries to the network domain nameserver, if one is present, or through reference to an entry in your ETC\HOSTS file.

For best results, consider setting DISPLAY to the value of `<your_host_internet_addr>:0`. For example, if your host internet address is 9.67.30.44, you can set the DISPLAY field in the ICAT Configure Services menu to the value 9.67.30.44:0. By using your host address you will avoid having to perform host name to address translation. If the internet address of your host machine changes, you must also change the value of the DISPLAY environment variable, either by editing your CONFIG.SYS directly or by changing the DISPLAY variable field in the ICAT Configure Services menu.

For more information about the X clients and utilities, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

Sample X Server Setup Procedure

The following steps describe how to install and configure your TCP/IP for OS/2 X Window System Server software.

Step 1: Install the X Server Software from Diskettes: The following steps describe how to install the X server software from diskettes into the TCP/IP for OS/2 directory tree using the ICAT program.

1. From an OS/2 command prompt, type ICAT. When the main ICAT menu screen appears, select the INSTALL button to enter the TCP/IP for OS/2 option installation menu.
2. Select the box adjacent to the "X Window System Server" feature to install the OS/2 X server. If you do not have a mouse, you may tab through the features until the box you wish to select is highlighted, and then press the space bar to make the selection.

   Note: By default, the OS/2 X server software is installed on the drive and in the directory where ICAT has determined that you installed your TCP/IP for OS/2 base software. If you wish to change the install path, edit the text in the path field.
3. Select the INSTALL button at the bottom of the menu to begin the installation process for the X server. Follow the directions for inserting the X server program diskettes until ICAT indicates that the installation is complete.

Step 2: Configure the Options to Start the X Server Automatically: The following steps describe how to configure the options to start the X server automatically when you bring up the system.

1. Select CONFIGURE from the ICAT main menu to enter the TCP/IP Configuration Tool window.
2. Select the "Automatic Starting of Services" menu, and page forward until the "Enable this machine to function as an X server" option appears.
3. Select the box adjacent to the "Enable this machine to function as an X server" option to automatically start the X server. The -nocopyright parameter of PMX is included by default.

Step 3: Configure the X Server: The following steps describe how to configure the X server options from the ICAT "Configure Services" window.

1. Exit from the "Automatic Starting of Services" menu and return to the TCP/IP Configuration Tool window. Select the "Configure Services" option to enter the "Configure Services" window.
2. Enter default X client host authorization for trusted machines by entering the host name in the X0HOSTS window box. Enter the host names by using the edit options at the top of the menu.

   Note: You can use host names only in the X0HOSTS window box; no host addresses are allowed. Your TCP/IP for OS/2 software must be able to resolve these names into internet addresses, either by querying your TCP/IP domain name server or by referencing your ETC\HOSTS file.
3. Enter the default DISPLAY environment variable used by the X client utilities to indicate the X server to which to direct their requests. You should enter your own host's name or internet address immediately followed by :0. This enables the X client commands to reference the (local) OS/2 X server by default.

Step 4: Exit ICAT and Reboot the Machine: The following steps describe how to activate changes made to your CONFIG.SYS and start your X server automatically.

1. The ICAT program added the XFILES and DISPLAY environment variables to your CONFIG.SYS. Rebooting the machine and restarting the OS/2 system causes these changes to take effect.
2. Your X server software executable, PMX.EXE is now started automatically each time your OS/2 system is started.

Testing the X Server with the Hello World Program

The TCP/IP for OS/2 product includes a *hello world* program, which is an X client application, that runs on OS/2 and can be used to verify that the X server has been set up correctly. The *hello world* program accepts one optional argument, which is the name of a font in usual X Window System font name form. Before starting the *hello world* program, you should make sure the X server is running. For more information about starting the X server, see "Starting the X Server" on page 147.

You can start the *hello world* program by typing the following:

```
xhello
```

A small PM X window should appear with the message "hello world." If the message appears, you are ready to use the PMX server with the X client utilities included with the OS/2 X Server software or with other X client applications on other systems.

If an error message is displayed, which indicates that the XHELLO program could not open the display, the DISPLAY environment variable is either not initialized properly or not initialized at all. The DISPLAY environment variable is used by X client utilities to specify the default X server at which to direct requests. Possible sources of the problem are:

- You did not initialize the DISPLAY variable from the ICAT Configure Services menu
- You have not rebooted your OS/2 machine after correctly initializing the DISPLAY variable while in ICAT. This caused the DISPLAY environment variable, which is set in your CONFIG.SYS, not to be recognized.
- You initialized the DISPLAY variable while in ICAT, but the program was unable to use the value of DISPLAY to access the designated X server.

To correct any of these cases, you should go back into an ICAT session and set the DISPLAY variable properly and then reboot the machine. For best results, DISPLAY should be set to the value `your_host_internet_addr:0`. For example, if your internet address is 9.67.30.44, you should set the DISPLAY field in the ICAT Configure Services menu to 9.67.30.44:0

As an alternative to rebooting, you may set the DISPLAY variable at an OS/2 command prompt by issuing SET DISPLAY=`your_host_internet_addr:0`. This will allow you to use X clients from within this OS/2 session (only) without having to reboot the machine.

Chapter 14. Boot Protocol (BOOTP)

Chapter 14. Boot Protocol (BOOTP)

This section describes how to set up BOOTPD (Server) and BOOTP (Client). The Boot Protocol (BOOTP) enables a client to get an IP address and other information from a server.

Setting up the Server

The BOOTPD server must have a BOOTPTAB file, which is a correspondence file for hardware addresses and IP addresses. A sample BOOTPTAB file is included in the TCPIP\ETC directory. You must change this file to contain hardware addresses, IP addresses, gateway addresses, and name server addresses for the local network. Use the sample BOOTPTAB file in Appendix G, "Sample BOOTPTAB File" on page 233 as a pattern, and replace addresses and names.

To find the hardware addresses of installed terminals, use the NETSTAT command, with the -n parameter, on the installed terminal. For more information about the NETSTAT command, see *IBM TCP/IP for OS/2 Version 1.2: User's Guide*.

You must verify that the SERVICES file in the ETC directory contains the following two lines:

```
sbootp  67/udp  #bootp server
cbootp  68/udp  #bootp client
```

Use the BOOTPD command to start the BOOTPD server.

The following example shows the format of the BOOTPD command.

```
BOOTPD [-d -d -d -d -d]
```

The parameters of the BOOTPD command are:

Parameter	Description
-d	Causes debug information, such as hardware address, to be displayed on the server terminal. You must use multiple -ds to display any debug information. Each additional -d increases the amount of debug information.

BOOTPD.EXE runs as a task until you shut down the server.

Note: BOOTP broadcast packets are used for transfer and only reside on the local network. They do not pass through a router to another network.

The following is an example of a bootpd response:

```
[E:\TCPIP\SRC\SCRBIN]bootpd -d -d -d -d -d
bootpd: reading "E:\TCPIP\ETC\BOOTPTAB"
bootpd: read 7 entries from E:\TCPIP\ETC\BOOTPTAB
bootpd: request from hardware address 10005A2B1EF0
bootpd: vendor magic field is 99.130.83.99
bootpd: sending RFC1048-style reply
```

Setting up the Client

You must verify that the TCPIP\ETC\SERVICES file contains the following two lines:

```
sbootp  67/udp  #bootp server
cbootp  68/udp  #bootp client
```

After you set up the TCPIP\ETC\SERVICES file, use the BOOTP command to start the BOOTP client.

The following example shows the format of the BOOTP command.

```
BOOTP [-?] | [lan0] | [lan1] | [lan2] | [lan3]
```

The parameters of the BOOTP command are:

Parameter	Description
?	Displays the list of parameters and their meaning.
lan0	Broadcasts to the lan0 network. This is the default.
lan1	Broadcasts to the lan1 network.
lan2	Broadcasts to the lan2 network.
lan3	Broadcasts to the lan3 network.

Note: lan*x* is the network of the BOOTPD server.

The client executes and outputs the response from the server to the client's terminal screen as quickly as the connection has been made.

The following is an example of BOOTP response.

```
[C:  tcpip\src\bootp]bootp
ifconfig lan0 9.67.30.70 netmask 255.255.224.0
route add default 9.67.22.2 1
add net default: gateway 9.67.22.2
Name server: 9.67.30.100
Name server: 9.67.30.99
Host name pete.tcp.raleigh.ibm.com
```

Note: The numbers shown are only examples. Your response numbers and names will be different.

In this example, the IFCONFIG and ROUTE commands are executed, the TCPIP\ETC\RESOLV file is checked, and a RESOLV file is created. A RESOLV file uses the domain name server information in the received packet from the server. If the RESOLV file already exists, it is updated.

BOOTP.EXE exits after the output is printed on the client's terminal screen.

Note: If BOOTP is executed and is not successful, SETUP.CMD should be run to restore your address configuration before PING and other commands can be run.

Chapter 15. Security Issues

Chapter 15. Security Issues

This chapter contains information concerning security issues in TCP/IP for OS/2. Security is a concern in all TCP/IP implementations, and the issues are not limited to the OS/2 implementation. Therefore, the information in this chapter is not intended to solve general TCP/IP security issues, but to indicate specific instances in the OS/2 TCP/IP implementation where users can make their hosts more secure.

File System

The OS/2 operating system was designed to be a multitasking, but single-user operating system. Therefore, the file system is not partitioned into separate user space for each user.

Most other TCP/IP implementations reside on multi-user operating systems having built-in features that can restrict individual users from viewing other users' directories and files. By using TCP/IP for OS/2, the OS/2 operating system can effectively become a multi-user system, but the OS/2 file system is still a single-user file system. When an OS/2 TCP/IP server, such as Telnet, gives access to a TCP/IP Telnet client, any file on the OS/2 host can be accessed.

Environment Variables

The TCP/IP for OS/2 implementation uses OS/2 environment variables to store the Telnet server and REXEC server passwords. Environment variables that are defined in the CONFIG.SYS file are defined for every session that is started. Environment variables that are defined in an OS/2 session are valid only for that session.

The OS/2 Installation and Configuration Automation Tool (ICAT) was designed to make the TCP/IP installation, configuration, and startup easy for the user. The environment variables are defined in the CONFIG.SYS file.

If you are concerned with security, do not put the Telnet and REXEC environment variables in the CONFIG.SYS file. Type them immediately before starting the respective server in the same session. This method prevents the values of the environment variables from being available in another OS/2 session or having them stored in the CONFIG.SYS file, which may be accessible from another TCP/IP client.

Accesses Defined by the FTP Server

The TCP/IP for OS/2 FTP server can define *read-only* access for an FTP client. In the OS/2 file system, *read-only* means that you can read a file on the FTP server host. This may include the CONFIG.SYS file, which can define Telnet or REXEC authorization. This may also allow you to read SMTP mail messages.

The FTP server allows you to restrict directory path access to subdirectories on a per-client basis. Users can access only directories and subdirectories defined to them by the FTP server.

NETRC File

The NETRC file is an optional file that is used by FTP and REXEC clients to automatically log on to foreign hosts without being prompted for user IDs and passwords. The NETRC file contains security information about other hosts. You do not have to name the NETRC file NETRC, nor does it have to reside in the ETC directory. You can define the name with the NETRC environment variable.

Warning: If you have a Telnet, REXEC, TFTP, RSH, or FTP server running on your machine, you should not create a NETRC file, because it provides users with *user* and *password* information that may allow them access to other users' files.

Note: Do not assume that because the file is named something other than NETRC, and the NETRC environment variable is not defined in the CONFIG.SYS file, that the file is safe. Software utilities can be written (such as UNIX's grep command) that can search for keywords in files and identify the file.

TFTP Server without a Limited Directory Path

The TFTP server does not require that the TFTP client supply any type of user authentication. However, the TFTP server can be started with a restricted directory path. Starting the TFTP server without a limited directory path allows any one to destroy files existing on their system. For more information about the TFTP path parameter, see "TFTP" on page 71.

SLIP Network Interface

Running TCP/IP with the SLIP interface, with the modem in auto-answer mode (SLIPCALL -a), could be a security issue on your network. To avoid placing the modem in auto-answer mode, do not issue the SLIPCALL -a command. As an alternative method, turn auto-answer off by invoking the SLIPCALL -r command. Some modems have a hardware switch setting for auto-answer; therefore, the SLIPCALL -r can not turn off auto-answer.

Writing Applications that Use Kerberos

The Kerberos Authentication System is a security feature that provides interfaces that can be used to write applications that use Kerberos with TCP/IP.

Chapter 12, "Setting Up Your Kerberos System" describes how to set up your Kerberos system. *IBM TCP/IP Version 1.2 for OS/2: User's Guide* provides an overview of the Kerberos Authentication System. *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* describes the Kerberos Authentication System and the Kerberos routines you can use to write applications.

Authorizing X Client Hosts

The OS/2 X server provides basic X client host authorization by two mechanisms, the ETC\X0HOSTS file and the XHOST authorization utility. The X0HOSTS file is read during X server initialization and defines the network hosts that can access the services of the OS/2 X server. The XHOST utility is used to dynamically change or override the list of authorized hosts.

For more information about the X0HOSTS file and the XHOST utility, see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

Appendixes

Appendix A. Optional Files

This appendix contains the files that can be created to be used by certain applications in TCP/IP for OS/2.

Table 3. Usage of Optional Files for TCP/IP for OS/2

Name of File	Used by	Purpose of File
EXPORTS	NFS server	Specifies directories that clients can mount.
FSTAB	NFS client	Specifies a list of mount commands to be automatically performed during startup.
GATEWAYS	ROUTED Server	Identifies gateways.
HOSTS	Any client or server	Resolves host names if a name server is unavailable.
INETD.LST	Selected servers	Defines servers that are to be activated.
KRB.CNF	Kerberos clients	Specifies the host that is running the Kerberos Authentication Server.
KRB.RLM	Kerberos clients	Specifies the realm name.
MIB2.TBL	Several SNMP commands	Defines the mapping between an object's ASN.1 notation and an object's textual notation.
NETRC	FTP and REXEC clients	Alternative source for user and password.
PINGHOST.LST	PMPING	Specifies a list of hosts to be monitored.
PW.SRC	SNMP Agent	Plain text list of community name(s) for the SNMP agent.
RESOLV	Any client or server	Provides domain name and name server address.
RHOSTS	RSH Server	Specifies the hosts that are authorized to use the RSH Server.
SNMP.PW	SNMP Agent	The scrambled version of the PW.SRC file which is used by the agent.
SNMPTRAP.DST	SNMP Agent	Specifies destination hosts that receive TRAP messages.
TRUSERS	FTP Server	Verifies user and password.
X0HOSTS	X Server (PMX)	Authorizes hosts to connect to OS/2 X Server.
X25DIR	X.25 Interface	Specifies the X.25 remote directory name.
X25IP	X.25 Interface	Specifies the X.25 local directory name.
X25RTE	X.25 Interface	Specifies the X.25 route name.

These files, with the specified names, must reside in the ETC directory or in the directory specified by the ETC environment variable. The required format of these files is listed in the following table.

Table 4 (Page 1 of 3). Contents of Optional Files for TCP/IP for OS/2

Name of File	Contents of File	Sample of File
EXPORTS	directory *client* [,*client*]...	c:\ ashlin f:\appsource build dev.com
FSTAB	mount command line or mvslogin command line	MOUNT -u32 -g5 x wolly1:/home/andy MOUNT -landrew x wolly1:/home/andrew MOUNT -v v: ralvmx:nibmaaa.191,user=nibmaaa,rw MOUNT -u -g z:mvs1:myid,text MVSLOGIN -p mvs1 myid
GATEWAYS	{net | host} *name1* gateway *name2* metric *value* {active | passive | external}	net net2 gateway host4 metric 4 passive host host3 gateway host4 metric 4 passive host host10 gateway 192.9.201.5 metric 9 active host host10 gateway 192.8.201.5 metric 8 external
HOSTS	*internet_address host_name* [*alias(es)*] [*# comment*] <*carriage return*>	124.34.216.1 Host1 joansPC PC1 educdept 124.34.216.3 PC3 edsPC Host3 # This is Ed's PC 124.34.216.5 PC5 janetsPC
INETD.LST	*application protocol server*	telnet tcp telnetd exec tcp rexecd ftp tcp ftpd printer tcp lpd tftp udp tftpd shell tcp rshd Where: telnet is the service, tcp is the protocol, and telnetd is the server to be activated. exec is the service, tcp is the protocol, and rexecd is the server to be activated.
KRB.CNF	*realm host_name* [admin server]	univ.educ.chem chrispc admin server
KRB.RLM	*domain_name realm* or *host_name realm*	.educ.chem univ.educ.chem .educ.bio univ.educ.bio chrispc univ.educ.chem joanpc univ.educ.bio
MIB2.TBL	*textual_name asn.1_name syntax*	sysDescr 1.3.6.1.2.1.1.1.0 display
NETRC	machine *host_name* [login *user_id*] [password *password*] [account *account*] [macdef *macroname*]	machine raleigh login kent password baseball machine boston login chris password boz macdef mymacro bell hash prompt binary cd c:\mydir get myfile.bin machine phoenix login writer password account payday
PINGHOST.LST	*host_ip_address description*	9.67.30.100 **Nameserver-Call_Dan 9.67.22.1 RALVMM_via_3172-Call_IS

Table 4 (Page 2 of 3). Contents of Optional Files for TCP/IP for OS/2

Name of File	Contents of File	Sample of File
PW.SRC	*community_name desired_network snmp_mask*	passwd1 9.0.0.0 255.0.0.0 passwd2 129.34.81.22 255.255.255.255
RESOLV	domain *domain_name* nameserver *internet_address*	domain eng.mit.edu nameserver 129.34.128.245 nameserver 129.34.128.246
RHOSTS	*host_name.domain_name* [*user*]	kant.watson.ibm.com Scott jorge.raleigh.ibm.com Where: Scott is the only user on kant.watson.ibm.com that is served and all users on jorge.raleigh.ibm.com are served.
SNMPTRAP.DST	*host_name UDP*	124.34.216.1 UDP Manager2 UDP
TRUSERS	user: *user_name* [*password*] rd[^]: [[*path*] *path*...] wr[^]: [[*path*] *path*...]	user: chris boz rd: d:\ c:\ wr: d:\tmp c:\tmp Where: chris is the user, boz is the password for chris, rd: d:\ c:\ gives chris access to read files and subdirectories in the c:\ and d:\ directories and wr: d:\tmp c:\tmp gives chris access to write to files and subdirectories only in the c:\tmp or c:\tmp directories.
X0HOSTS	*host_name*	mpw.tcpip.rtp.ibm.com glenns_mod80 als_mod95
X25DIR	*ipaddress directory_name* default *directory_name*	9.67.22.1 REMOTE1 9.67.22.2 REMOTE2 default REMOTE3 Where: REMOTE1, REMOTE2, and REMOTE3 are remote directory entry names defined in a Communications Manager X.25 configuration file. IP packets with a destination address of 9.67.22.1 generate an X.25 call request to the DTE address defined in REMOTE1. IP packets with a destination address of 9.67.22.2 generate an X.25 call request to the DTE address defined in REMOTE2. All other IP packets generate an X.25 call request to the DTE address defined in REMOTE3.
X25IP	directory_name	LOCAL1 Where: LOCAL1 is a local directory entry defined in a Communications Manager X.25 configuration file. This file contains the local directory entry that associates the local DTE address with a link. Only one directory name can be listed.

Table 4 (Page 3 of 3). Contents of Optional Files for TCP/IP for OS/2

Name of File	Contents of File	Sample of File
X25RTE	*route_name*	ROUTE1 ROUTE2 Where: ROUTE1 and ROUTE2 are routing table entry names defined in a Communications Manager X.25 configuration file. Multiple route names can be listed.

Appendix B. Sample SLIP.CMD File

This appendix contains a sample SLIP.CMD file that illustrates the elements contained in a SLIP.CMD file.

```
rem Set the Baud rate of the modems being used.
set slip.bps=2400
rem
rem Set the Modem pause time (in seconds) for a comma.
set slip.delay=2
rem
rem Set the dialing string for the modem.  The phone number is 555-1234.
rem set slip.dial=ATDT555,1234
rem
rem Execute the code that dials the modem.
slipcall -rd
```

Appendix C. Sample OS/2 TCP/IP Default Directory Structure

This appendix outlines the default directory structure used when ICAT installs the TCP/IP for OS/2 product.

```
TCPIP
  BIN                 (contains all executable modules)
       *.exe
       *.ex
       *.cmd
       *.ico
       *.sys
  INCLUDE             (contains all header files)
       *.h
       ARPA
           *.h
       IDL             (contains ncs header files)
           *.IDL
           C
               *.h
       NETINET
           *.h
       NET
           *.h
       PROTOCOL
           *.h
       RPC             (contains rpc header files)
           *.h
       SYS
           *.h
       DPI             (contains snmp header files)
           *.h
       KRB             (contains kerberos header files)
           *.h
  HELP
       *.hlp
  LIB
       tcpip.lib       (contains the tcpip library)
       tcpipmt.lib
       tcpipdll.lib
       sunrpc.lib      (contains the rpc library)
       rpcdll.lib
       nckmt.lib
       krb.lib         (contains kerberos libraries)
       dpi.lib         (contains the snmp library)
       ftpapi.lib      (contains the ftpapi library)
       ftpapimt.lib
```

```
ETC
     bootptab
     rpc
     protocol
     services
     pinghost.lst
     mib2.tbl
     sendmail.cf
     sendmail.hf
     x25dir
     x25ip
     x25rte
     x25samp.cfg
SAMPLES
     NSTAT
     RN
     NAMED
     TRACERT
     DPI
     NCS
     RPC
     RPCGEN
     KRB
     SOCKET
     RCOPY
X11
     misc            (fonts)
     75dpi           (fonts)
DLL
     *.DLL
DOC
```

Appendix D. Management Information Base (MIB) Objects

This appendix lists the objects for the following groups defined by the Management Information Base (MIB)-II.

- System
- Interfaces
- Address Translation
- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- Exterior Gateway Protocol (EGP)
- Simple Network Management Protocol (SNMP).

The object types are defined using the following fields:

Object	A textual string (referred to as the OBJECT DESCRIPTOR) and the administratively assigned name (referred to as the Object Identifier).
ASN.1 Notation	The Abstract Syntax Notation which represents the object identifier.
Syntax	The data type of the MIB object.
Definition	A description of the MIB object.
Access	One of read-only, read-write, write-only, or not-accessible.

System Group

Table 5 lists the objects in the system group. The system objects identify the type of system with a text description and the vendor-assigned object-id as an identification to the type of SNMP agent.

Table 5 (Page 1 of 2). Implementation of the System Group

Object and ASN.1 Notation	Syntax	Definition	Access
SYSTEM GROUP 1.3.6.1.2.1.1			
sysDescr { system 1 } 1.3.6.1.2.1.1.1.0	DisplayString	A description of the entry. This value should include the full name and version identification of the system's hardware type, software operating system, and networking software. This description must only contain printable ASCII characters.	read-only
sysObjectID { system 2 } 1.3.6.1.2.1.1.2.0	Object Identifier	The vendor's authorization identification of the network management subsystem contained in the entry. This value is allocated within the Structure for Management Information (SMI) enterprise's subtree (1.3.6.1.4.1) and provides an easy and clear means for determining *what kind of box* is being managed. For example, if vendor *Stones, Inc.* was assigned the subtree 1.3.6.1.4.1.42, it could assign the identifier 1.3.6.1.4.1.42.1.1 to the router *Fred Router*.	read-only
sysUpTime { system 3 } 1.3.6.1.2.1.1.3.0	TimeTicks	The time (in hundredths of a second) since the network management portion of the system was last started.	read-only
sysContact { system 4 } 1.3.6.1.2.1.1.4.0	DisplayString	The textual identification of the contact person for this managed node, together with information on how to contact this person.	read-write
sysName { system 5 } 1.3.6.1.2.1.1.5.0	DisplayString	An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name.	read-write
sysLocation { system 6 } 1.3.6.1.2.1.1.6.0	DisplayString	The physical location of this node (for example, telephone closet, 3rd floor).	read-write

Table 5 (Page 2 of 2). Implementation of the System Group

Object and ASN.1 Notation	Syntax	Definition	Access
sysServices { system 7 } 1.3.6.1.2.1.1.7.0	Integer	A value that indicates the set of services that this entity primarily offers. The value is a sum. This sum initially takes the value zero, then, for each layer, L, in the range 1 through 7, for which this node performs transactions, 2 raised to (L - 1) is added to the sum. For example, a node which performs primarily routing functions would have a value of 4 (2**(3-1)). In contrast, a node that is a host offering application services would have a value of 72 (2**(4-1) + 2**(7-1)). In the context of the Internet suite of protocols, values should be calculated accordingly: **layer** **functionality** 1 physical (for example, repeaters) 2 datalink/subnetwork (for example, bridges) 3 internet (for example, IP gateways) 4 end-to-end (for example, IP hosts) 7 applications (for example, mail relays) For systems including OSI protocols, layers 5 and 6 may also be counted.	read-only

Interfaces Group

Table 6 on page 179 lists the objects in the interfaces group. The interfaces objects are a set of entries for each network interface below the IP layer that can send and receive datagrams.

Table 6 (Page 1 of 5). Implementation of the Interfaces Group

Object and ASN.1 Notation	Syntax	Definition	Access
INTERFACES GROUP 1.3.6.1.2.1.2			
ifNumber { interfaces 1 } 1.3.6.1.2.1.2.1.	Integer	The number of network interfaces (regardless of their current state) present on this system.	read-only
ifTable { interfaces 2 } 1.3.6.1.2.1.2.2	SEQUENCE of IfEntry	A list of interface entries. The number of entries is given by the value of ifNumber.	not-accessible
ifEntry { ifTable 1 } 1.3.6.1.2.1.2.2.1	IfEntry ::= SEQUENCE ifIndex INTEGER, ifDescr DisplayString, ifType INTEGER, ifMtu INTEGER, ifSpeed Gauge, ifPhysAddress PhysAddress, ifAdminStatus INTEGER, ifOperStatus INTEGER, ifLastChange TimeTicks, ifInOctets Counter, ifInUcastPkts Counter, ifInNUcastPkts Counter, ifInDiscards Counter, ifInErrors Counter, ifInUnkownProtos Counter,	An interface entry that contains objects at the subnetwork layer and below for a particular interface.	not-accessible

Table 6 (Page 2 of 5). Implementation of the Interfaces Group

Object and ASN.1 Notation	Syntax	Definition	Access
ifEntry (Cont.)	ifOutOctets Counter, ifOutUcastPkts Counter, ifOutNUcastPkts Counter, ifOutDiscards Counter, ifOutErrors Counter, ifOutQLen Gauge ifSpecific Object Identifier,		
ifIndex { ifEntry 1 } 1.3.6.1.2.1.2.2.1.1.	Integer	A unique value for each interface. Values range between 1 and the value of ifNumber. The value for each interface must remain constant for at least one start of the systems network management system to the next start.	read-only
ifDescr { ifEntry 2 } 1.3.6.1.2.1.2.2.1.2.	DisplayString	A text string containing information about the interface. This string should include the name of the manufacturer, the product name, and the version of the hardware interface.	read-only

Table 6 (Page 3 of 5). Implementation of the Interfaces Group

Object and ASN.1 Notation	Syntax	Definition	Access
ifType { ifEntry 3 } 1.3.6.1.2.1.2.2.1.3.	Integer other (1), regular 1822 (2), hdh1822 (3), ddn-x25 (4), rfc877-x25 (5), ethernet-csmacd (6), iso88023-csmacd (7), iso88024-tokenBus (8), iso88025-tokenRing (9), iso88026-kman (10), starLan (11), proteon-10Mbit (12), proteon-80Mbit (13), hyperchannel (14), fddi (15), lapb (16), sdlc (17), tl-carrier (18), cept (19), basicISDN (20), primaryISDN (21), propPointToPointSerial (22), terminalServer-asypcPort (23), softwareLoopback (24), eon (25), ethernet-3Mbit (26), nsip (27), slip (28) ultra (29), ds3 (30), sip (31), frame-relay (32),	The type of interface, distinguished according to the physical/link/network protocol(s) immediately *below* IP in the protocol stack.	read-only
ifMtu { ifEntry 4 } 1.3.6.1.2.1.2.2.1.4.	Integer	The size of the largest datagram that can be sent or received on the interface, specified in octets. For interfaces that are used for transmitting IP datagrams, this is the size of the largest IP datagram that can be sent on the interface.	read-only
ifSpeed { ifEntry 5 } 1.3.6.1.2.1.2.2.1.5.	Gauge	An estimate of the interface's current bandwidth in bits per second. For interfaces that do not vary in bandwidth or for those where no accurate estimate can be made, this object should contain the nominal bandwidth.	read-only
ifPhysAddress { ifEntry 6 } 1.3.6.1.2.1.2.2.1.6.	PhysAddress	The interface's address at the protocol layer immediately *below* IP in the protocol stack. For interfaces that do not have such an address (for example, a serial line), this object should contain an octet string of length zero.	read-only

Table 6 (Page 4 of 5). Implementation of the Interfaces Group

Object and ASN.1 Notation	Syntax	Definition	Access
ifAdminStatus { ifEntry 7 } 1.3.6.1.2.1.2.2.1.7.	Integer up (1), down (2), testing (3)	The desired state of the interface. The testing (3) state indicates that operational packets cannot be passed.	read-write
ifOperStatus { ifEntry 8 } 1.3.6.1.2.1.2.2.1.8.	INTEGER up (1), down (2), testing (3)	The current operational state of the interface. The testing (3) state indicates that operational packets cannot be passed.	read-only
ifLastChange { ifEntry 9 } 1.3.6.1.2.1.2.2.1.9.	TimeTicks	The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered before the last start-up of the local network management subsystem, then this object contains a value of zero.	read-only
ifInOctets { ifEntry 10 } 1.3.6.1.2.1.2.2.1.10.	Counter	The total number of octets received on the interface, including framing characters.	read-only
ifInUcastPkts { ifEntry 11 } 1.3.6.1.2.1.2.2.1.11.	Counter	The number of subnetwork-unicast packets delivered to a higher-layer protocol.	read-only
IfInNUcastPkts { ifEntry 12 } 1.3.6.1.2.1.2.2.1.12.	Counter	The number of non-unicast (for example, subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.	read-only
ifInDiscards { ifEntry 13 } 1.3.6.1.2.1.2.2.1.13.	Counter	The number of inbound packets that were chosen to be discarded even though errors had not been detected to prevent their delivery to a higher-layer protocol. One possible reason for discarding such a packet could be to free buffer space.	read-only
ifInErrors { ifEntry 14 } 1.3.6.1.2.1.2.2.1.14.	Counter	The number of inbound packets that contain errors that prevent delivery to a higher-layer protocol.	read-only
ifInUnknownProtos { ifEntry 15 } 1.3.6.1.2.1.2.2.1.15.	Counter	The number of packets received through the interface that were discarded because of an unknown or unsupported protocol.	read-only
ifOutOctets { ifEntry 16 } 1.3.6.1.2.1.2.2.1.16.	Counter	The total number of octets transmitted out of the interface, including framing characters.	read-only
ifOutUcastPkts { ifEntry 17 } 1.3.6.1.2.1.2.2.1.17.	Counter	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.	read-only

Table 6 (Page 5 of 5). Implementation of the Interfaces Group

Object and ASN.1 Notation	Syntax	Definition	Access
ifOutNUcastPkts { ifEntry 18 } 1.3.6.1.2.1.2.2.1.18.	Counter	The total number of packets that higher-level protocols request to be transmitted to a non-unicast (for example, a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent.	read-only
ifOutDiscards { ifEntry 19 } 1.3.6.1.2.1.2.2.1.19.	Counter	The number of outbound packets that were chosen to be discarded even though errors had not been detected to prevent their being transmitted. One reason for discarding such a packet could be to free buffer space.	read-only
ifOutErrors { ifEntry 20 } 1.3.6.1.2.1.2.2.1.20.	Counter	The number of outbound packets that could not be transmitted because of errors.	read-only
ifOutQLen { ifEntry 21 } 1.3.6.1.2.1.2.2.1.21.	Gauge	The length of the output packet queue (in packets).	read-only
ifSpecific { ifEntry 22 } 1.3.6.1.2.1.2.2.1.22.	Object identifier	A reference to MIB definitions specific to the particular media being used to realize the interface. For example, if the interface is realized by an ethernet, then the value of this object refers to a document defining objects specific to ethernet. If this information is not present, its value should be set to the OBJECT IDENTIFIER (0 0), which is a syntactically valid object identifier, and any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.	read-only

Address Translation Group

Table 7 lists the objects in the address translation group. The address translation group consists of one table, which shows the mapping between IP addresses and physical addresses.

Table 7. Implementation of the Address Translation Group

Object and ASN.1 Notation	Syntax	Definition	Access
ADDRESS TRANSLATION GROUP 1.3.6.1.2.1.3			
atTable { at 1 } 1.3.6.1.2.1.3.1	Sequence of AtEntry	The Address Translation tables contain the NetworkAddress to *physical* address equivalences. Some interfaces do not use translation tables to determine address equivalences (for example, DDN-X.25 has an algorithmic method). If all interfaces are of this type, then the Address Translation table is empty, it has zero entries.	not-accessible
atEntry { atTable 1 } 1.3.6.1.2.1.3.1.1	AtEntry ::= SEQUENCE atIfIndex INTEGER, atPhysAddress PhysAddress, atNetAddress NetworkAddress	Each entry contains one NetworkAddress to the physical address equivalent.	not-accessible
atIfIndex { atEntry 1 } 1.3.6.1.2.1.3.1.1.1.	Integer	The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface that is identified by the same value of ifIndex.	read-write
atPhysAddress { atEntry 2 } 1.3.6.1.2.1.3.1.1.2.	PhysAddress	The media-dependent *physical* address.	read-write
atNetAddress { atEntry 3 } 1.3.6.1.2.1.3.1.1.3.	NetworkAddress	The NetworkAddress (for example, the IP address) corresponding to the media-dependent *physical* address.	read-write

IP Group

Table 8 lists the objects in the IP group. The IP objects are the statistics and gateway routing tables for the IP layer.

Table 8 (Page 1 of 7). Implementation of the IP Group

Object and ASN.1 Notation	Syntax	Definition	Access
IP GROUP 1.3.6.1.2.1.4			
ipForwarding { ip 1 } 1.3.6.1.2.1.4.1.	Integer gateway (1), -- entry forwards datagrams host (2) -- entry does NOT forward datagrams	Indicates if this entry is acting as an IP gateway for the forwarding of datagrams received by, but not addressed to, this entry. IP gateways forward datagrams; hosts do not, except those source-routed through the host.	read-write
ipDefaultTTL { ip 2 } 1.3.6.1.2.1.4.2.	Integer	When a TTL value is not supplied by the transport layer protocol, the default value inserts into the time-to-live field of the IP header of datagrams that originate at this entry.	read-write
ipInReceives { ip 3 } 1.3.6.1.2.1.4.3.	Counter	The number of input datagrams received from interfaces, including those received in error.	read-only
ipInHdrErrors { ip 4 } 1.3.6.1.2.1.4.4.	Counter	The number of input datagrams discarded because of errors in their IP headers. For example, bad checksums, mismatched version number, format errors, time-to-live exceeded, and processing errors in IP options.	read-only
ipInAddrErrors { ip 5 } 1.3.6.1.2.1.4.5.	Counter	The number of input datagrams discarded because the IP address in their IP header's destination field was not a valid address to be received at this entry. This count includes invalid addresses (for example, 0.0.0.0), addresses of unsupported classes (for example, Class E), and destination addresses that was not a local address (for example, IP gateways).	read-only
ipForwDatagrams { ip 6 } 1.3.6.1.2.1.4.6.	Counter	The number of input datagrams for which this entry is not their final IP destination. As a result, an attempt is made to find a route to their final destination. For entries that do not act as IP gateways, this count includes only those packets that are source-routed successfully through this entry.	read-only
ipInUnknownProtos { ip 7 } 1.3.6.1.2.1.4.7.	Counter	The number of locally-addressed datagrams received successfully, but discarded because of an unknown or unsupported protocol.	read-only

Table 8 (Page 2 of 7). Implementation of the IP Group

Object and ASN.1 Notation	Syntax	Definition	Access
ipInDiscards { ip 8 } 1.3.6.1.2.1.4.8.	Counter	The number of input IP datagrams that are processed without problems, but are discarded. For example, for lack of buffer space. This count does not include any datagrams discarded while awaiting reassembly.	read-only
ipInDelivers { ip 9 } 1.3.6.1.2.1.4.9.	Counter	The number of input datagrams successfully delivered to IP user-protocols including ICMP.	read-only
ipOutRequests { ip 10 } 1.3.6.1.2.1.4.10.	Counter	The number of IP datagrams that are supplied to IP and ICMP in requests for transmission. This count does not include datagrams counted in ipForwDatagrams.	read-only
ipOutDiscards { ip 11 } 1.3.6.1.2.1.4.11.	Counter	The number of output IP datagrams that transmit without problems, but are discarded. For example, for lack of buffer space. This count includes datagrams in ipForwDatagrams that meet this discard criterion.	read-only
ipOutNoRoutes { ip 12 } 1.3.6.1.2.1.4.12.	Counter	The number of IP datagrams discarded because no route can transmit them to their destination. This count includes packets in ipForwDatagrams that meet this no-route criterion.	read-only
ipReasmTimeout { ip 13 } 1.3.6.1.2.1.4.13.	Integer	The maximum number of seconds that received fragments are held while awaiting reassembly at this entry.	read-only
ipReasmReqds { ip 14 } 1.3.6.1.2.1.4.14.	Counter	The number of IP fragments that are received and need to be reassembled at this entry.	read-only
ipReasmOKs { ip 15 } 1.3.6.1.2.1.4.15.	Counter	The number of IP datagrams reassembled without problems.	read-only
ipReasmFails { ip 16 } 1.3.6.1.2.1.4.16.	Counter	The number of failures detected by the IP reassembly algorithm. This is not a count of discarded IP fragments because some algorithms can lose track of the number of fragments by combining them as they are received.	read-only
ipFragOKs { ip 17 } 1.3.6.1.2.1.4.17.	Counter	The number of IP datagrams that have fragmented at this entry without problems.	read-only
ipFragFails { ip 18 } 1.3.6.1.2.1.4.18.	Counter	The number of IP datagrams that should have been fragmented at this entry, but were not because their *Don't Fragment* flag was set.	read-only

Table 8 (Page 3 of 7). Implementation of the IP Group

Object and ASN.1 Notation	Syntax	Definition	Access
ipFragCreates { ip 19 } 1.3.6.1.2.1.4.19.	Counter	The number of IP datagram fragments that have been generated, because of fragmentation at this entry.	read-only
ipAddrTable { ip 20 } 1.3.6.1.2.1.4.20	SEQUENCE OF IpAddrEntry	A table that contains addressing information relevant to this entry's IP addresses.	not-accessible
ipAddrEntry { ipAddrTable 1 } 1.3.6.1.2.1.4.20.1	IpAddrEntry ::= SEQUENCE ipAdEntAddr IpAddress, ipAdEntIfIndex INTEGER, ipAdEntNetMask IpAddress, ipAdEntBcastAddr INTEGER ipAdEntReasmMaxSize INTEGER (0..65535)	The addressing information for one of this entry's IP addresses.	not-accessible
ipAdEntAddr { ipAddrEntry 1 } 1.3.6.1.2.1.4.20.1.1.	IpAddress	The IP address pertaining to this entry's addressing information.	read-only
ipAdEntIfIndex { ipAddrEntry 2 } 1.3.6.1.2.1.4.20.1.2.	Integer	The index value that uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface that is identified by the same value of ifIndex.	read-only
ipAdEntNetMask { ipAddrEntry 3 } 1.3.6.1.2.1.4.20.1.3.	IpAddress	The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.	read-only
ipAdEntBcastAddr { ipAddrEntry 4 } 1.3.6.1.2.1.4.20.1.4.	Integer	The value of the least-significant bit in the IP broadcast address used for sending datagrams on the (logical) interface associated with the IP address of this entry. For example, when the internet standard all-ones broadcast address is used, the value is 1.	read-only
ipAdEntReasmMaxSize { ipAddrEntry 5 } 1.3.6.1.2.1.4.20.1.5	Integer (0..65535)	The size of the largest IP datagram which this entity can reassemble from incoming IP fragmented datagrams received on this interface.	read-only
ipRoutingTable { ip 21 } 1.3.6.1.2.1.4.21	SEQUENCE OF IpRouteEntry	This entry's IP routing table.	not-accessible

Table 8 (Page 4 of 7). Implementation of the IP Group

Object and ASN.1 Notation	Syntax	Definition	Access
ipRouteEntry { ipRoutingTable 1 } 1.3.6.1.2.1.4.21.1	IpRouteEntry ::= SEQUENCE ipRouteDest IpAddress, ipRouteIfIndex INTEGER, ipRouteMetric1 INTEGER, ipRouteMetric2 INTEGER, ipRouteMetric3 INTEGER, ipRouteMetric4 INTEGER, ipRouteNextHop IpAddress, ipRouteType INTEGER, ipRouteProto INTEGER, ipRouteAge INTEGER ipRouteMask IpAddress ipRouteMetric5 Integer ipRouteInfo ObjectIdentifier	A route to a particular destination.	
ipRouteDest { ipRouteEntry 1 } 1.3.6.1.2.1.4.21.1.1.	IpAddress	The destination IP address of this route. An entry with a value of 0.0.0.0 is considered a default route. Multiple default routes can appear in the table, but access to these multiple entries is dependent on the table-access mechanisms defined by the network management protocol in use.	read-write
ipRouteIfIndex { ipRouteEntry 2 } 1.3.6.1.2.1.4.21.1.2.	Integer	The index value that uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface that is identified by the same value of ifIndex.	read-write
ipRouteMetric1 { ipRouteEntry 3 } 1.3.6.1.2.1.4.21.1.3.	Integer	The primary routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.	read-write
ipRouteMetric2 { ipRouteEntry 4 } 1.3.6.1.2.1.4.21.1.4.	Integer	An alternative routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.	read-write

Table 8 (Page 5 of 7). Implementation of the IP Group

Object and ASN.1 Notation	Syntax	Definition	Access
ipRouteMetric3 { ipRouteEntry 5 } 1.3.6.1.2.1.4.21.1.5.	Integer	An alternative routing metric for this route. The semantics of this metric are determined by the routing protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.	read-write
ipRouteMetric4 { ipRouteEntry 6 } 1.3.6.1.2.1.4.21.1.6.	Integer	An alternative routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.	read-write
ipRouteNextHop { ipRouteEntry 7 } 1.3.6.1.2.1.4.21.1.7.	IpAddress	The IP address of the next hop of this route.	read-write
ipRouteType { ipRouteEntry 8 } 1.3.6.1.2.1.4.21.1.8.	Integer other (1), invalid (2), direct (3), remote (4)	The type of route.	read-write
ipRouteProto { ipRouteEntry 9 } 1.3.6.1.2.1.4.21.1.9.	Integer other (1), local (2), netmgmt (3), icmp (4), egp (5), ggp (6), hello (7), rip (8), is-is (9), es-is (10), ciscolgrp (11), bbnSpflgp (12), ospf (13)	The routing mechanism by which this route was learned. Inclusion of values for gateway routing protocols is not intended to imply that hosts should support those protocols.	read-only
ipRouteAge { ipRouteEntry 10 } 1.3.6.1.2.1.4.21.1.10.	Integer	The number of seconds since this route was last updated or otherwise determined to be correct. Note semantics of *too old* cannot be implied, except through knowledge of the routing protocol by which the route was learned.	read-write

Table 8 (Page 6 of 7). Implementation of the IP Group

Object and ASN.1 Notation	Syntax	Definition	Access
ipRouteMask { ipRouteEntry 11 } 1.3.6.1.2.1.4.21.1.11.	IpAddress	Indicate the mask to be logical-ANDed with the destination address before being compared to the value in the ipRouteDest field. For those systems that do not support arbitrary subnet masks, an agent constructs the value of the ipRouteMask by determining whether the value of the correspondent ipRouteDest field belong to a class-A, B, or C network, and then using one of: **mask** **network** 255.0.0.0 Class-A 255.255.0.0 Class-B 255.255.255.0 Class-A If the value of the ipRouteDest is 0.0.0.0 (default route), then the mask value is also 0.0.0.0. All IP routing subsystems implicitly use this mechanism.	read-write
ipRouteMetric5 { ipRouteEntry 12 } 1.3.6.1.2.1.4.21.1.12.	Integer	An alternate routing metric for this route. The semantics of this metric are determined by the routing-protocol specified in the route's ipRouteProto value. If this metric is not used, its value should be set to -1.	read-write
ipRouteInfo { ipRouteEntry 13 } 1.3.6.1.2.1.4.21.1.13.	Object Identifier	A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's ipRouteProto value. If this information is not present, its value should be set to the OBJECT IDENTIFIER (0 0), which is a syntactically valid object identifier, and any conformant implementation of ASN.1 and BER must be able to generate and recognize this value.	read-only
IP Address Translation Table 1.3.6.1.2.1.4.22		The IP address translation table contains the ipAddress to *physical* address equivalences. Some interfaces do not use translation tables for determining address equivalences (for example, DDN-X.25 has an algorithmic method); if all interfaces are of this type, then the Address Translation Table is empty, that is, has zero entries.	not-accessible
ipNetToMediaTable { ip 22 } 1.3.6.1.2.1.4.22.1	SEQUENCE OF IpNetToMediaEntry	The IP Address Translation table used for mapping from IP addresses to physical addresses.	not-accessible

Table 8 (Page 7 of 7). Implementation of the IP Group

Object and ASN.1 Notation	Syntax	Definition	Access
ipNetToMediaEntry	ipNetToMedialfIndex INTEGER, ipNetToMediaPhysAddress PhysAddress, ipNetToMediaNetAddress IpAddress, ipNetToMediaType INTEGER	Each entry contains one ipAddress to physical address equivalence.	not-accessible
ipNetToMedialfIndex { ipNetToMediaEntry 1 } 1.3.6.1.2.1.4.22.1.1	Integer	The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.	read-write
ipNetToMediaPhysAddress { ipNetToMediaEntry 2 } 1.3.6.1.2.1.4.22.1.2	Octet string	The media-dependent *physical* address.	read-write
ipNetToMediaNetAddress { ipNetToMediaEntry 3 } 1.3.6.1.2.1.4.22.1.3	ipAddress	The IpAddress corresponding to the media-dependent 'physical' address.	read-write
ipNetToMediaType { ipNetToMediaEntry 4 } 1.3.6.1.2.1.4.22.1.4	Integer	The type of mapping. Setting this object to the value invalid(2) has the effect of invalidating the corresponding entry in the ipNetToMediaTable. That is, it effectively disassociates the interface identified with said entry from the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Proper interpretation of such entries requires examination of the relevant ipNetToMediaType object.	read-write

ICMP Group

Table 9 lists the objects in the ICMP group. The ICMP objects are the input and output error and control message statistics for the IP layer.

Table 9 (Page 1 of 3). Implementation of the ICMP Group

Object and ASN.1 Notation	Syntax	Definition	Access
ICMP GROUP 1.3.6.1.2.1.5			
icmpInMsgs { icmp 1 } 1.3.6.1.2.1.5.1.0	Counter	The number of ICMP messages that the entity received. This counter includes all those counted by icmpInErrors.	read-only
icmpInErrors { icmp 2 } 1.3.6.1.2.1.5.2.0	Counter	The number of ICMP messages that the entity received. and determines ICMP specific errors (bad ICMP checksums, bad length).	read-only
icmpInDestUnreachs { icmp 3 } 1.3.6.1.2.1.5.3.0	Counter	The number of ICMP destination Unreachable messages received.	read-only
icmpInTimeExcds { icmp 4 } 1.3.6.1.2.1.5.4.0	Counter	The number of ICMP Time Exceeded messages received.	read-only
icmpInParmProbs { icmp 5 } 1.3.6.1.2.1.5.5.0	Counter	The number of ICMP Parameter Problem messages received.	read-only
icmpInSrcQuenchs { icmp 6 } 1.3.6.1.2.1.5.6.0	Counter	The number of ICMP Source Quench messages received.	read-only
icmpInRedirects { icmp 7 } 1.3.6.1.2.1.5.7.0	Counter	The number of ICMP Redirect messages received.	read-only
icmpInEchos { icmp 8 } 1.3.6.1.2.1.5.8.0	Counter	The number of ICMP Echo (request) messages received.	read-only
icmpInEchoReps { icmp 9 } 1.3.6.1.2.1.5.9.0	Counter	The number of ICMP Echo Reply messages received.	read-only
icmpInTimestamps { icmp 10 } 1.3.6.1.2.1.5.10.0	Counter	The number of ICMP Timestamp (request) messages received.	read-only
icmpInTimestampReps { icmp 11 } 1.3.6.1.2.1.5.11.0	Counter	The number of ICMP Timestamp Reply messages received.	read-only

Table 9 (Page 2 of 3). Implementation of the ICMP Group

Object and ASN.1 Notation	Syntax	Definition	Access
icmpInAddrMasks { icmp 12 } 1.3.6.1.2.1.5.12.0	Counter	The number of ICMP Address Mask Request messages received.	read-only
icmpInAddrMaskReps { icmp 13 } 1.3.6.1.2.1.5.13.0	Counter	The number of ICMP Address Mask Reply messages received.	read-only
icmpOutMsgs { icmp 14 } 1.3.6.1.2.1.5.14.0	Counter	The number of ICMP messages sent. This counter includes icmpOutErrors.	read-only
icmpOutErrors { icmp 15 } 1.3.6.1.2.1.5.15.0	Counter	The number of ICMP messages that this entity did not send because of problems within ICMP. For example, no buffers. This value should not include errors outside the ICMP layer. For example, the inability of IP to route the resulting datagram. In some implementations, there may not be error types that contribute to the counter's value.	read-only
icmpOutDestUnreachs { icmp 16 } 1.3.6.1.2.1.5.16.0	Counter	The number of ICMP Destination Unreachable messages sent.	read-only
icmpOutTimeExcds { icmp 17 } 1.3.6.1.2.1.5.17.0	Counter	The number of ICMP Time Exceeded messages sent.	read-only
icmpOutParmProbs { icmp 18 } 1.3.6.1.2.1.5.18.0	Counter	The number of ICMP Parameter Problem messages sent.	read-only
icmpOutSrcQuenches { icmp 19 } 1.3.6.1.2.1.5.19.0	Counter	The number of ICMP Source Quench messages sent.	read-only
icmpOutRedirects { icmp 20 } 1.3.6.1.2.1.5.20.0	Counter	The number of ICMP Redirect messages sent. For a host, this object is always zero, hosts do not send redirects.	read-only
icmpOutEchos { icmp 21 } 1.3.6.1.2.1.5.21.0	Counter	The number of ICMP Echo (request) messages sent.	read-only
icmpOutEchoReps { icmp 22 } 1.3.6.1.2.1.5.22.0	Counter	The number of ICMP Echo Reply messages sent.	read-only
icmpOutTimestamps { icmp 23 } 1.3.6.1.2.1.5.23.0	Counter	The number of ICMP Timestamp (request) messages sent.	read-only

Table 9 (Page 3 of 3). Implementation of the ICMP Group

Object and ASN.1 Notation	Syntax	Definition	Access
icmpOutTimestampReps { icmp 24 } 1.3.6.1.2.1.5.24.0	Counter	The number of ICMP Timestamp Reply messages sent.	read-only
icmpOutAddrMasks { icmp 25 } 1.3.6.1.2.1.5.25.0	Counter	The number of ICMP Address Mask Request messages sent.	read-only
icmpOutAddrMasksReps { icmp 26 } 1.3.6.1.2.1.5.26.0	Counter	The number of ICMP Address Mask Reply messages sent.	read-only

TCP Group

Table 10 lists the objects in the TCP group. The TCP objects are the data transmission statistics and connection data for the TCP layer.

Note: Objects that represent information about a particular TCP connection are transient; the objects exist only as long as the specified connection is in use.

Table 10 (Page 1 of 3). Implementation of the TCP Group

Object and ASN.1 Notation	Syntax	Definition	Access
TCP GROUP 1.3.6.1.2.1.6			
tcpRtoAlgorithm { tcp 1 } 1.3.6.1.2.1.6.1.	Integer other (1) -- none of the following constant (2) -- a contant rto rsre (3) --MIL-STD-1778 vanj (4) -- Van Jacobson's algorithm	The algorithm used to determine the time-out value used for retransmitting unacknowledged octets.	read-only
tcpRtoMin { tcp 2 } 1.3.6.1.2.1.6.2.	Integer	The minimum value allowed by a TCP implementation for the retransmission time-out, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission time-out. For example, when the time-out algorithm is rsre (3), an object of this type has the semantics of the LBOUND quantity described in RFC 793.	read-only
tcpRtoMax { tcp 3 } 1.3.6.1.2.1.6.3.	Integer	The maximum value allowed by a TCP implementation for the retransmission time-out, measured in milliseconds. More refined semantics for objects of this type depend upon the algorithm used to determine the retransmission time-out. For example, when the time-out algorithm is rsre (3), an object of this type has the semantics of the UBOUND quantity described in RFC 793.	read-only
tcpMaxConn { tcp 4 } 1.3.6.1.2.1.6.4.	Integer	The limit on the number of TCP connections the entry can support. In entities where the maximum number of connections is dynamic, this object should be -1.	read-only
tcpActiveOpens { tcp 5 } 1.3.6.1.2.1.6.5.	Counter	The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.	read-only
tcpPassiveOpens { tcp 6 } 1.3.6.1.2.1.6.6.	Counter	The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.	read-only

Table 10 (Page 2 of 3). Implementation of the TCP Group

Object and ASN.1 Notation	Syntax	Definition	Access
tcpAttemptFails { tcp 7 } 1.3.6.1.2.1.6.7.	Counter	The number of times TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.	read-only
tcpEstabResets { tcp 8 } 1.3.6.1.2.1.6.8.	Counter	The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED or CLOSE-WAIT.	read-only
tcpCurrEstab { tcp 9 } 1.3.6.1.2.1.6.9.	Gauge	The number of TCP connections of the current state that are either ESTABLISHED or CLOSE-WAIT.	read-only
tcpInSegs { tcp 10 } 1.3.6.1.2.1.6.10.	Counter	The total number of segments including those received in error. This count includes segments received on currently established connections.	read-only
tcpOutSegs { tcp 11 } 1.3.6.1.2.1.6.11.	Counter	The total number of segments sent including those on currently established connections, but excluding those containing only retransmitted octets.	read-only
tcpRetransSegs { tcp 12 } 1.3.6.1.2.1.6.12.	Counter	The total number of segments retransmitted that contain one or more previously transmitted octets.	read-only
tcpConnTable { tcp 13 } 1.3.6.1.2.1.6.13	SEQUENCE OF TcpConnEntry	A table that contains TCP connnection-specific information	not-accessible
tcpConnEntry { tcpConnTable 1 } 1.3.6.1.2.1.6.13.1	TcpConnEntry :: =SEQUENCE tcpConnState INTEGER, tcpConnLocalAddress IpAddress, tcpConnLocalPort INTEGER (0..65535), tcpConnRemAddress IpAddress, tcpConnRemPort INTEGER (0..65535)	Information about a certain current TCP connection. An object of this type is transient. It does not exist when (or soon after) the connection makes the transition to the CLOSED state.	not-accessible

Table 10 (Page 3 of 3). Implementation of the TCP Group

Object and ASN.1 Notation	Syntax	Definition	Access
tcpConnState { tcpConnEntry 1 } 1.3.6.1.2.1.6.13.1.1.	Integer closed(1), listen(2), synSent(3), synReceived(4), established(5), finWait1(6), finWait2(7), closeWait(8), lastAck(9), closing(10), timeWait(11) deleteTCB(12)	The TCP connection status.	read-write
tcpConnLocalAddress { tcpConnEntry 2 } 1.3.6.1.2.1.6.13.1.2.	IpAddress	The local IP address for this TCP connection. In the case of a connection in the listen state which is willing to accept connections for any IP interface associated with the node, the value 0.0.0.0 is used.	read-only
tcpConnLocalPort { tcpConnEntry 3 } 1.3.6.1.2.1.6.13.1.3.	Integer (0..65535)	The local port number of this TCP connection.	read-only
tcpConnRemAddress { tcpConnEntry 4 } 1.3.6.1.2.1.6.13.1.4.	IpAddress	The remote IP address of this TCP connection.	read-only
tcpConnRemPort { tcpConnEntry 5 } 1.3.6.1.2.1.6.13.1.5.	Integer (0..65535)	The remote port number of this TCP connection.	read-only
tcpInErrs { tcp 14 } 1.3.6.1.2.1.6.14.0	Counter	The total number of segments received in error (for example, bad TCP checksums).	read-only
tcpOutRsts { tcp 15 } 1.3.6.1.2.1.6.15.0	Counter	The number of TCP segments sent containing the RST flag.	read-only

UDP Group

Table 11 lists the objects in the UDP group. The UDP objects are the datagram statistics of the UDP layer.

Table 11. Implementation of the UDP Group

Object and ASN.1 Notation	Syntax	Definition	Access
UDP GROUP 1.3.6.1.2.1.7			
udpInDatagrams { udp 1 } 1.3.6.1.2.1.7.1.0	Counter	The number of UDP datagrams delivered to UDP users.	read-only
udpNoPorts { udp 2 } 1.3.6.1.2.1.7.2.0	Counter	The number of UDP datagrams received where there was no application at the destination port.	read-only
udpInErrors { udp 3 } 1.3.6.1.2.1.7.3.0	Counter	The number of UDP datagrams received that could not be delivered for reasons other than the lack of an application at the destination port.	read-only
udpOutDatagrams { udp 4 } 1.3.6.1.2.1.7.4.0	Counter	The number of UDP datagrams sent from this entry.	read-only
UDP Listener Table 1.3.6.1.2.1.7.5		The UDP listener table contains information about this entity's UDP end-points on which a local application is currently accepting datagrams.	not-accessible
udpTable 1.3.6.1.2.1.7.5.1	SEQUENCE OF UdpEntry	A table containing UDP listener information.	not-accessible
udpEntry	udpEntry ::= SEQUENCE udpLocalAddress IpAddress, udpLocalPort INTEGER (0..65535)	Information about a particular current UDP listener.	not-accessible
udpLocalAddress 1.3.6.1.2.1.7.5.1.1	IpAddress	The local IP address for this listener. In the case of a UDP listener that can accept datagrams for any IP interface associated with the node, the value 0.0.0.0 is used.	read-only
udpLocalPort 1.3.6.1.2.1.7.5.1.2	Integer	The local port number for this UDP listener.	read-only

EGP Group

Table 12 lists the objects in the EGP group.

Table 12 (Page 1 of 3). Implementation of the EGP Group

Object and ASN.1 Notation	Syntax	Definition	Access
EGP GROUP 1.3.6.1.2.1.8			
egpInMsgs 1.3.6.1.2.1.8.1.0	Counter	The number of EGP messages received without error.	read-only
egpInErrors 1.3.6.1.2.1.8.2.0	Counter	The number of EGP messages received that proved to be in error.	read-only
egpOutMsgs 1.3.6.1.2.1.8.3.0	Counter	The total number of locally generated EGP messages.	read-only
egpOutErrors 1.3.6.1.2.1.8.4.0	Counter	The number of locally generated EGP messages not sent because of resource limitations within an EGP entity.	read-only
egpNeighTable 1.3.6.1.2.1.8.5	SEQUENCE OF EgpNeighEntry	Information about this entity's relationship with a particular EGP neighbor.	not-accessible
egpNeighEntry 1.3.6.1.2.1.8.5.1	EgpNeighEntry ::= SEQUENCE egpNeighState INTEGER, egpNeighAddr IpAddress, egpNeighAs INTEGER, egpNeighInMsgs Counter, egpNeighInErrs Counter, egpNeighOutMsgs Counter, egpNeighOutErrs Counter, egpNeighInErrMsgs Counter, egpNeighOutErrMsgs Counter, egpNeighStateUps Counter, egpNeighStateDowns Counter, egpNeighIntervalHello INTEGER, egpNeighIntervalPoll INTEGER, egpNeighMode INTEGER, egpNeighEventTrigger INTEGER	Information about this entity's relationship with a particular EGP neighbor.	not-accessible

Table 12 (Page 2 of 3). Implementation of the EGP Group

Object and ASN.1 Notation	Syntax	Definition	Access
egpNeighState { egpNeighEntry 1 } 1.3.6.1.2.1.8.5.1.1	Integer	the EGP state of the local system with respect to this entry's EGP neighbor. Each EGP state is represented by a value that is one greater than the numerical value associated with said state in RFC 904.	read-only
egpNeighAddr { egpNeighEntry 2 } 1.3.6.1.2.1.8.5.1.2	IpAddress	The IP address of this entry's EGP neighbor.	read-only
egpNeighAs { egpNeighEntry 3 } 1.3.6.1.2.1.8.5.1.3	Integer	The autonomous system of this EGP peer. Zero should be specified if the autonomous system number of the neighbor is not yet known.	read-only
egpNeighInMsgs { egpNeighEntry 4 } 1.3.6.1.2.1.8.5.1.4	Counter	The number of EGP messages received without error from this EGP peer.	read-only
egpNeighInErrs { egpNeighEntry 5 } 1.3.6.1.2.1.8.5.1.5	Counter	The number of EGP messages received from this EGP peer that proved to be in error (for example, bad EGP checksum).	read-only
egpNeighOutMsgs { egpNeighEntry 6 } 1.3.6.1.2.1.8.5.1.6	Counter	The number of locally generated EGP messages to this EGP peer.	read-only
egpNeighOutErrs { egpNeighEntry 7 } 1.3.6.1.2.1.8.5.1.7	Counter	The number of locally generated EGP messages not sent to this EGP peer because of resource limitations within an EGP entity.	read-only
egpNeighInErrMsgs { egpNeighEntry 8 } 1.3.6.1.2.1.8.5.1.8	Counter	The number of EGP-defined error messages received from this EGP peer.	read-only
egpNeighOutErrMsgs { egpNeighEntry 9 } 1.3.6.1.2.1.8.5.1.9	Counter	The number of EGP-defined error messages sent to this EGP peer.	read-only
egpNeighStateUps { egpNeighEntry 10 } 1.3.6.1.2.1.8.5.1.10	Counter	The number of EGP state transitions to the UP state with this EGP peer.	read-only
egpNeighStateDowns { egpNeighEntry 11 } 1.3.6.1.2.1.8.5.1.11	Counter	The number of EGP state transitions from the UP state to any other state with this EGP peer.	read-only
egpNeighIntervalHello { egpNeighEntry 12 } 1.3.6.1.2.1.8.5.1.12	Integer	The interval between EGP HELLO command retransmissions (in hundredths of a second). This represents the t1 timer as defined in RFC 904.	read-only

Table 12 (Page 3 of 3). Implementation of the EGP Group

Object and ASN.1 Notation	Syntax	Definition	Access
egpNeighIntervalPoll { egpNeighEntry 13 } 1.3.6.1.2.1.8.5.1.13	Integer	The interval between EGP POLL command retransmissions (in hundredths of a second). This represents the t3 timer as defined in RFC 904.	read-only
egpNeighMode { egpNeighEntry 14 } 1.3.6.1.2.1.8.5.1.14	Integer	The polling mode of this EGP entity, either passive or active.	read-only
egpNeighEventTrigger { egpNeighEntry 15 } 1.3.6.1.2.1.8.5.1.15	Integer	A control variable used to trigger operator-initiated Start and Stop events. When read, this variable always returns the most recent value to which egpNeighEventTrigger was set. If it has not been set since the last initialization of the network management subsystem on the node, it returns a value of 'stop'. When set, this variable causes a Start or Stop event on the specified neighbor, as specified in RFC 904. A Start event causes an Idle peer to begin neighbor acquisition and a non-Idle peer to reinitiate neighbor acquisition. A Stop event causes an non-Idle peer to return to the Idle state until a Start event occurs, either by egpNeighEventTrigger or otherwise.	read-write
egpAs 1.3.6.1.2.1.8.6	Integer	The autonomous system number of this EGP entity.	read-only

SNMP GROUP

Table 13 lists the objects in the SNMP group.

Table 13 (Page 1 of 3). Implementation of the SNMP Group

Object and ASN.1 Notation	Syntax	Definition	Access
SNMP GROUP 1.3.6.1.2.1.11			
snmpInPkts { snmp 1 } 1.3.6.1.2.1.11.1.0	Counter	The total number of Messages delivered to the SNMP entity from the transport service.	read-only
snmpOutPkts { snmp 2 } 1.3.6.1.2.1.11.2.0	Counter	The total number of SNMP Messages that were passed from the SNMP protocol entity to the transport service.	read-only
snmpInBadVersions { snmp 3 } 1.3.6.1.2.1.11.3.0	Counter	The total number of SNMP Messages that were delivered to the SNMP protocol entity and were for an unsupported SNMP version.	read-only
snmpInBadCommunityNames { snmp 4 } 1.3.6.1.2.1.11.4.0	Counter	The total number of SNMP Messages delivered to the SNMP protocol entity that used a SNMP community name not known to said entity.	read-only
snmpInBadCommunityUses { snmp 5 } 1.3.6.1.2.1.11.5.0	Counter	The total number of SNMP Messages delivered to the SNMP protocol entity that represented an SNMP operation that was not allowed by the SNMP community named in the Message.	read-only
snmpInASNParseErrs { snmp 6 } 1.3.6.1.2.1.11.6.0	Counter	The total number of ASN.1 or BER errors encountered by the SNMP protocol entity when decoding received SNMP Messages.	read-only
snmpInTooBigs { snmp 8 } 1.3.6.1.2.1.11.8.0	Counter	The total number of SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is 'tooBig'.	read-only
snmpInNoSuchNames { snmp 9 } 1.3.6.1.2.1.11.9.0	Counter	The total number of SNMP PDUs which were delivered to the SNMP protocol entity and for which the value of the error-status field is 'noSuchName'.	read-only
snmpInBadValues { snmp 10 } 1.3.6.1.2.1.11.10.0	Counter	The total number of SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is 'badValue'.	read-only

Table 13 (Page 2 of 3). Implementation of the SNMP Group

Object and ASN.1 Notation	Syntax	Definition	Access
snmpInReadOnlys { snmp 11 } 1.3.6.1.2.1.11.11.0	Counter	The total number of valid SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is 'readOnly'. It is a protocol error to generate an SNMP PDU that contains the value 'readOnly' in the error-status field, as such this object is provided as a means of detecting incorrect implementation of the SNMP.	read-only
snmpInGenErrs { snmp 12 } 1.3.6.1.2.1.11.12.0	Counter	The total number of SNMP PDUs that were delivered to the SNMP protocol entity and for which the value of the error-status field is 'genErr'.	read-only
snmpInTotalReqVars { snmp 13 } 1.3.6.1.2.1.11.13.0	Counter	The total number of MIB objects that have been retrieved successfully by the SNMP protocol entity as the result of receiving valid SNMP Get-Request and Get-Next PDUs.	read-only
snmpInTotalSetVars { snmp 14 } 1.3.6.1.2.1.11.14.0	Counter	The total number of MIB objects that have been altered successfully by the SNMP protocol entity as the result of receiving valid SNMP Set-Request PDUs.	read-only
snmpInGetRequests { snmp 15 } 1.3.6.1.2.1.11.15.0	Counter	The total number of SNMP Get-Request PDUs thath have been accepted and processed by the SNMP protocol entity.	read-only
snmpInGetNexts { snmp 16 } 1.3.6.1.2.1.11.16.0	Counter	The total number of SNMP Get-Next PDUs that have been accepted and processed by the SNMP protocol entity.	read-only
snmpInGetSetRequests { snmp 17 } 1.3.6.1.2.1.11.17.0	Counter		read-only
snmpInGetResponses { snmp 18 } 1.3.6.1.2.1.11.18.0	Counter	The total number of SNMP Get-Response PDUs which have been accepted and processed by the SNMP protocol entity.	read-only
snmpInTraps { snmp 19 } 1.3.6.1.2.1.11.19.0	Counter	The total number of SNMP Trap PDUs that have been accepted and processed by the SNMP protocol entity.	read-only
snmpOutTooBigs { snmp 20 } 1.3.6.1.2.1.11.20.0	Counter	The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status field is 'tooBig'.	read-only

Table 13 (Page 3 of 3). Implementation of the SNMP Group

Object and ASN.1 Notation	Syntax	Definition	Access
snmpOutNoSuchNames { snmp 21 } 1.3.6.1.2.1.11.21.0	Counter	The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status is 'noSuchName'.	read-only
snmpOutBadValues { snmp 22 } 1.3.6.1.2.1.11.22.0	Counter	The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status field is 'badValue'.	read-only
snmpOutReadOnlys { snmp 23 } 1.3.6.1.2.1.11.23.0	Counter		read-only
snmpOutGenErrs { snmp 24 } 1.3.6.1.2.1.11.24.0	Counter	The total number of SNMP PDUs that were generated by the SNMP protocol entity and for which the value of the error-status field is 'genErr'.	read-only
snmpOutGetRequests { snmp 25 } 1.3.6.1.2.1.11.25.0	Counter	The total number of SNMP Get-Request PDUs thath have been generated by the SNMP protocol entity.	read-only
snmpOutGetNexts { snmp 26 } 1.3.6.1.2.1.11.26.0	Counter	The total number of SNMP Get-Next PDUs that have been generated by the SNMP protocol entity.	read-only
snmpOutSetRequests { snmp 27 } 1.3.6.1.2.1.11.27.0	Counter	The total number of SNMP Set-Request PDUs that have been generated by the SNMP protocol entity.	read-only
snmpOutGetResponses { snmp 28 } 1.3.6.1.2.1.11.28.0	Counter	The total number of SNMP Get-Response PDUs that have been generated by the SNMP protocol entity.	read-only
snmpOutTraps { snmp 29 } 1.3.6.1.2.1.11.29.0	Counter	The total number of SNMP Trap PDUs thath have been generated by the SNMP protocol entity.	read-only
snmpEnableAuthTraps { snmp 30 } 1.3.6.1.2.1.11.30.0	Integer	Indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authentication-failure traps can be disabled. This object should be stored in nonvolatile memory so that it remains constant between reinitializations of the network management system.	read-write

Appendix E. MIB2.TBL File: MIB-II Objects

This appendix contains the MIB2.TBL file as installed on your PC.

```
sysDescr                1.3.6.1.2.1.1.1.0          display
sysObjectID             1.3.6.1.2.1.1.2.0          object
sysUpTime               1.3.6.1.2.1.1.3.0          ticks
sysContact              1.3.6.1.2.1.1.4.0          display
sysName                 1.3.6.1.2.1.1.5.0          display
sysLocation             1.3.6.1.2.1.1.6.0          display
sysServices             1.3.6.1.2.1.1.7.0          number
ifNumber                1.3.6.1.2.1.2.1.0          number
ifIndex                 1.3.6.1.2.1.2.2.1.1.       number
ifDescr                 1.3.6.1.2.1.2.2.1.2.       display
ifType                  1.3.6.1.2.1.2.2.1.3.       number
ifMtu                   1.3.6.1.2.1.2.2.1.4.       number
ifSpeed                 1.3.6.1.2.1.2.2.1.5.       gauge
ifPhysAddress           1.3.6.1.2.1.2.2.1.6.       string
ifAdminStatus           1.3.6.1.2.1.2.2.1.7.       number
ifOperStatus            1.3.6.1.2.1.2.2.1.8.       number
ifLastChange            1.3.6.1.2.1.2.2.1.9.       ticks
ifInOctets              1.3.6.1.2.1.2.2.1.10.      counter
ifInUcastPkts           1.3.6.1.2.1.2.2.1.11.      counter
ifInNUcastPkts          1.3.6.1.2.1.2.2.1.12.      counter
ifInDiscards            1.3.6.1.2.1.2.2.1.13.      counter
ifInErrors              1.3.6.1.2.1.2.2.1.14.      counter
ifInUnknownProtos       1.3.6.1.2.1.2.2.1.15.      counter
ifOutOctets             1.3.6.1.2.1.2.2.1.16.      counter
ifOutUcastPkts          1.3.6.1.2.1.2.2.1.17.      counter
ifOutNUcastPkts         1.3.6.1.2.1.2.2.1.18.      counter
ifOutDiscards           1.3.6.1.2.1.2.2.1.19.      counter
ifOutErrors             1.3.6.1.2.1.2.2.1.20.      counter
ifOutQLen               1.3.6.1.2.1.2.2.1.21.      gauge
ifSpecific              1.3.6.1.2.1.2.2.1.22.      object
atIfIndex               1.3.6.1.2.1.3.1.1.1.       number
atPhysAddress           1.3.6.1.2.1.3.1.1.2.       string
atNetAddress            1.3.6.1.2.1.3.1.1.3.       internet
ipForwarding            1.3.6.1.2.1.4.1.0          number
ipDefaultTTL            1.3.6.1.2.1.4.2.0          number
ipInReceives            1.3.6.1.2.1.4.3.0          counter
ipInHdrErrors           1.3.6.1.2.1.4.4.0          counter
ipInAddrErrors          1.3.6.1.2.1.4.5.0          counter
ipForwDatagrams         1.3.6.1.2.1.4.6.0          counter
ipInUnknownProtos       1.3.6.1.2.1.4.7.0          counter
ipInDiscards            1.3.6.1.2.1.4.8.0          counter
ipInDelivers            1.3.6.1.2.1.4.9.0          counter
ipOutRequests           1.3.6.1.2.1.4.10.0         counter
ipOutDiscards           1.3.6.1.2.1.4.11.0         counter
ipOutNoRoutes           1.3.6.1.2.1.4.12.0         counter
ipReasmTimeout          1.3.6.1.2.1.4.13.0         number
ipReasmReqds            1.3.6.1.2.1.4.14.0         counter
ipReasmOKs              1.3.6.1.2.1.4.15.0         counter
ipReasmFails            1.3.6.1.2.1.4.16.0         counter
ipFragOKs               1.3.6.1.2.1.4.17.0         counter
ipFragFails             1.3.6.1.2.1.4.18.0         counter
ipFragCreates           1.3.6.1.2.1.4.19.0         counter
ipAdEntAddr             1.3.6.1.2.1.4.20.1.1.      internet
ipAdEntIfIndex          1.3.6.1.2.1.4.20.1.2.      number
```

ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3.	internet
ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4.	number
ipAdEntReasmMaxSize	1.3.6.1.2.1.4.20.1.5.	number
ipRouteDest	1.3.6.1.2.1.4.21.1.1.	internet
ipRouteIfIndex	1.3.6.1.2.1.4.21.1.2.	number
ipRouteMetric1	1.3.6.1.2.1.4.21.1.3.	number
ipRouteMetric2	1.3.6.1.2.1.4.21.1.4.	number
ipRouteMetric3	1.3.6.1.2.1.4.21.1.5.	number
ipRouteMetric4	1.3.6.1.2.1.4.21.1.6.	number
ipRouteNextHop	1.3.6.1.2.1.4.21.1.7.	internet
ipRouteType	1.3.6.1.2.1.4.21.1.8.	number
ipRouteProto	1.3.6.1.2.1.4.21.1.9.	number
ipRouteAge	1.3.6.1.2.1.4.21.1.10.	number
ipRouteMask	1.3.6.1.2.1.4.21.1.11.	internet
ipNetToMediaIfIndex	1.3.6.1.2.1.4.22.1.1.	number
ipNetToMediaPhysAddress	1.3.6.1.2.1.4.22.1.2.	string
ipNetToMediaNetAddress	1.3.6.1.2.1.4.22.1.3.	internet
ipNetToMediaType	1.3.6.1.2.1.4.22.1.4.	number
icmpInMsgs	1.3.6.1.2.1.5.1.0	counter
icmpInErrors	1.3.6.1.2.1.5.2.0	counter
icmpInDestUnreachs	1.3.6.1.2.1.5.3.0	counter
icmpInTimeExcds	1.3.6.1.2.1.5.4.0	counter
icmpInParmProbs	1.3.6.1.2.1.5.5.0	counter
icmpInSrcQuenchs	1.3.6.1.2.1.5.6.0	counter
icmpInRedirects	1.3.6.1.2.1.5.7.0	counter
icmpInEchos	1.3.6.1.2.1.5.8.0	counter
icmpInEchoReps	1.3.6.1.2.1.5.9.0	counter
icmpInTimestamps	1.3.6.1.2.1.5.10.0	counter
icmpInTimestampReps	1.3.6.1.2.1.5.11.0	counter
icmpInAddrMasks	1.3.6.1.2.1.5.12.0	counter
icmpInAddrMaskReps	1.3.6.1.2.1.5.13.0	counter
icmpOutMsgs	1.3.6.1.2.1.5.14.0	counter
icmpOutErrors	1.3.6.1.2.1.5.15.0	counter
icmpOutDestUnreachs	1.3.6.1.2.1.5.16.0	counter
icmpOutTimeExcds	1.3.6.1.2.1.5.17.0	counter
icmpOutParmProbs	1.3.6.1.2.1.5.18.0	counter
icmpOutSrcQuenchs	1.3.6.1.2.1.5.19.0	counter
icmpOutRedirects	1.3.6.1.2.1.5.20.0	counter
icmpOutEchos	1.3.6.1.2.1.5.21.0	counter
icmpOutEchoReps	1.3.6.1.2.1.5.22.0	counter
icmpOutTimestamps	1.3.6.1.2.1.5.23.0	counter
icmpOutTimestampReps	1.3.6.1.2.1.5.24.0	counter
icmpOutAddrMasks	1.3.6.1.2.1.5.25.0	counter
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26.0	counter
tcpRtoAlgorithm	1.3.6.1.2.1.6.1.0	number
tcpRtoMin	1.3.6.1.2.1.6.2.0	number
tcpRtoMax	1.3.6.1.2.1.6.3.0	number
tcpMaxConn	1.3.6.1.2.1.6.4.0	number
tcpActiveOpens	1.3.6.1.2.1.6.5.0	counter
tcpPassiveOpens	1.3.6.1.2.1.6.6.0	counter
tcpAttemptFails	1.3.6.1.2.1.6.7.0	counter
tcpEstabResets	1.3.6.1.2.1.6.8.0	counter
tcpCurrEstab	1.3.6.1.2.1.6.9.0	gauge
tcpInSegs	1.3.6.1.2.1.6.10.0	counter
tcpOutSegs	1.3.6.1.2.1.6.11.0	counter
tcpRetransSegs	1.3.6.1.2.1.6.12.0	counter
tcpConnState	1.3.6.1.2.1.6.13.1.1.	number
tcpConnLocalAddress	1.3.6.1.2.1.6.13.1.2.	internet
tcpConnLocalPort	1.3.6.1.2.1.6.13.1.3.	number
tcpConnRemAddress	1.3.6.1.2.1.6.13.1.4.	internet

tcpConnRemPort	1.3.6.1.2.1.6.13.1.5.	number
tcpInErrs	1.3.6.1.2.1.6.14.0	counter
tcpOutRsts	1.3.6.1.2.1.6.15.0	counter
udpInDatagrams	1.3.6.1.2.1.7.1.0	counter
udpNoPorts	1.3.6.1.2.1.7.2.0	counter
udpInErrors	1.3.6.1.2.1.7.3.0	counter
udpOutDatagrams	1.3.6.1.2.1.7.4.0	counter
udpLocalAddress	1.3.6.1.2.1.7.5.1.1.	internet
udpLocalPort	1.3.6.1.2.1.7.5.1.2.	number
egpInMsgs	1.3.6.1.2.1.8.1.0	counter
egpInErrors	1.3.6.1.2.1.8.2.0	counter
egpOutMsgs	1.3.6.1.2.1.8.3.0	counter
egpOutErrors	1.3.6.1.2.1.8.4.0	counter
egpNeighState	1.3.6.1.2.1.8.5.1.1.	number
egpNeighAddr	1.3.6.1.2.1.8.5.1.2.	internet
egpNeighAs	1.3.6.1.2.1.8.5.1.3.	number
egpNeighInMsgs	1.3.6.1.2.1.8.5.1.4.	counter
egpNeighInErrs	1.3.6.1.2.1.8.5.1.5.	counter
egpNeighOutMsgs	1.3.6.1.2.1.8.5.1.6.	counter
egpNeighOutErrs	1.3.6.1.2.1.8.5.1.7.	counter
egpNeighInErrMsgs	1.3.6.1.2.1.8.5.1.8.	counter
egpNeighOutErrMsgs	1.3.6.1.2.1.8.5.1.9.	counter
egpNeighStateUps	1.3.6.1.2.1.8.5.1.10.	counter
egpNeighStateDowns	1.3.6.1.2.1.8.5.1.11.	counter
egpNeighIntervalHello	1.3.6.1.2.1.8.5.1.12.	number
egpNeighIntervalPoll	1.3.6.1.2.1.8.5.1.13.	number
egpNeighMode	1.3.6.1.2.1.8.5.1.14.	number
egpNeighEventTrigger	1.3.6.1.2.1.8.5.1.15.	number
egpAs	1.3.6.1.2.1.8.6.0	number
snmpInPkts	1.3.6.1.2.1.11.1.0	counter
snmpOutPkts	1.3.6.1.2.1.11.2.0	counter
snmpInBadVersions	1.3.6.1.2.1.11.3.0	counter
snmpInBadCommunityNames	1.3.6.1.2.1.11.4.0	counter
snmpInBadCommunityUses	1.3.6.1.2.1.11.5.0	counter
snmpInASNParseErrs	1.3.6.1.2.1.11.6.0	counter
snmpInBadTypes	1.3.6.1.2.1.11.7.0	counter
snmpInTooBigs	1.3.6.1.2.1.11.8.0	counter
snmpInNoSuchNames	1.3.6.1.2.1.11.9.0	counter
snmpInBadValues	1.3.6.1.2.1.11.10.0	counter
snmpInReadOnlys	1.3.6.1.2.1.11.11.0	counter
snmpInGenErrs	1.3.6.1.2.1.11.12.0	counter
snmpInTotalReqVars	1.3.6.1.2.1.11.13.0	counter
snmpInTotalSetVars	1.3.6.1.2.1.11.14.0	counter
snmpInGetRequests	1.3.6.1.2.1.11.15.0	counter
snmpInGetNexts	1.3.6.1.2.1.11.16.0	counter
snmpInSetRequests	1.3.6.1.2.1.11.17.0	counter
snmpInGetResponses	1.3.6.1.2.1.11.18.0	counter
snmpInTraps	1.3.6.1.2.1.11.19.0	counter
snmpOutTooBigs	1.3.6.1.2.1.11.20.0	counter
snmpOutNoSuchNames	1.3.6.1.2.1.11.21.0	counter
snmpOutBadValues	1.3.6.1.2.1.11.22.0	counter
snmpOutReadOnlys	1.3.6.1.2.1.11.23.0	counter
snmpOutGenErrs	1.3.6.1.2.1.11.24.0	counter
snmpOutGetRequests	1.3.6.1.2.1.11.25.0	counter
snmpOutGetNexts	1.3.6.1.2.1.11.26.0	counter
snmpOutSetRequests	1.3.6.1.2.1.11.27.0	counter
snmpOutGetResponses	1.3.6.1.2.1.11.28.0	counter
snmpOutTraps	1.3.6.1.2.1.11.29.0	counter
snmpEnableAuthTraps	1.3.6.1.2.1.11.30.0	number
DPI_port	1.3.6.1.4.1.2.2.1.1.0	number

Appendix F. Messages and Codes

This appendix describes the messages and exit codes that are displayed in TCP/IP for OS/2. The messages are arranged alphabetically, grouped by command.

FINGER

Table 14. FINGER Messages and Codes

Exit Code	Message	Explanation
N/A	Unable to connect to *host*	The foreign host can be reached, but the finger server is not running. **Action:** Start the finger server on the foreign host.

FTP

Table 15. FTP Messages and Codes

Exit Code	Message	Explanation
N/A	Error: 2	This message covers many error situations. The most likely reason is that a file name in the subcommand does not exist. **Action:** Check the directories on the local and remote hosts, using dir, and verify that the file names are correct.
1	Could not create a ftpds semaphore	System problem. **Action:** Reboot the system. If the problem persists, contact IBM service.

FTP Server FTPDC—Exit Messages

These messages are printed by the FTPDC program. This program is started by the FTP server to handle client requests. The program exits with the code listed.

Table 16 (Page 1 of 2). FTP Server FTPDC Exit Messages

Exit Code	Message	Explanation
0	Repeated login failures from *host*.	A user on another host is trying to log on to the FTP server and has been unsuccessful. **Action:** Verify that the user attempting to log on knows the correct user name and password.
1	ftpds:ioctl (trying to set socket to nonblocking)	System error. This should not occur. Contact the System Administrator.

Table 16 (Page 2 of 2). FTP Server FTPDC Exit Messages

Exit Code	Message	Explanation
1	panic: all enough memory	System error. Cannot allocate enough memory to read the TRUSERS file. This may mean that the TRUSERS file is too large, or too many programs are running. **Action:** Check the size of the TRUSERS file. Unless it is several megabytes, this should not be the problem. Reduce the number of programs running, reboot the system, and restart TCP/IP. If the problem persists, contact IBM Service.
1	FTPDS.EXE is not running (when trying to get shared segment with FTPDS)	FTPDC.EXE was started with the FTP server not running, or the FTP server died during startup of FTPDC.EXE. **Action:** Verify that the FTP server is running. If it is not running, restart it.
1	Could not open attn semaphore Could not open a mail semaphore	No message, but this problem could occur, because no socket exists. System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
2	Could not get pid	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
10	panic: FTPDS did not respond within 30 secs	The FTP server is not operating. **Action:** Restart the FTP server and FTPDC.EXE.

FTP Server FTPDC—Nonexit Messages

These messages are printed by the FTPDC.EXE. The program does not exit when these errors occur.

Table 17. FTP Server FTPDC Nonexit Messages

Exit Code	Message	Explanation
N/A	getpeername *program_name* trying to get peer information on connection	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
N/A	getsockname *program_name* trying to get socket name information	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
N/A	local disk full. aborted	The local disk is full. **Action:** Clean up the disk space, and retry the action that failed.

FTP Server FTPDS—Exit Messages

These messages are printed by the FTP server. The server exits with the code listed.

Table 18. FTP Server FTPDS Exit Messages

Exit Code	Message	Explanation
1	Could not create a ftpds semaphore	System problem. **Action:** Reboot the system. If the problem persists, contact IBM service.
1	ftpd:tcp/ftp: unknown service	FTP is not in the SERVICES file, or the SERVICES file does not exist. **Action:** Add FTP to the SERVICES file, or create a SERVICES file, if the file does not exist.
1	ftpd:socket	A socket could not be allocated by TCP. Too many applications are running. **Action:** Reduce the number of programs running, and restart the FTP server.
1	ftpds:listen	System problem. **Action:** Reboot the system, and restart TCP/IP. If the problem persists, contact IBM Service.
1	ftpds:ioctl	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
1	ftpds:accept	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
2	Could not get pid	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
8	No message printed, but...DOSGETENV failed	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.
9	Could not execute FTPDC.EXE	The FTP server cannot execute the FTPDC.EXE program. Either the program does not exist on the machine, or it is not on the path of executables for the FTP server, or the program has been corrupted. **Action:** Verify that: • The program exists. • It is in a subdirectory in the PATH environment variable. • It has not been corrupted.

FTP Server FTPDS—Nonexit Messages

These messages are printed by the FTP server. The server does not exit when these errors occur.

Table 19. FTP Server FTPDS Nonexit Messages

Exit Code	Message	Explanation
N/A	DosOpenSem failed on *semaphore*	System problem. **Action:** Reboot the system and restart TCP/IP. If the problem persists, contact IBM Service.

IFCONFIG

Table 20. IFCONFIG Messages and Codes

Exit Code	Message	Explanation
N/A	*illegal parameters*: bad value	The syntax of the IFCONFIG command is incorrect. **Action:** Type `ifconfig` for help or see *IBM TCP/IP Version 1.2 for OS/2: User's Guide*.

Kerberos Authentication System

Table 21 (Page 1 of 7). KERBEROS Messages and Codes

Exit Code	Message	Explanation
	OK	The Kerberos operation is successful. **Action:** No action to be taken.
	Principal expired	User registration with the Kerberos database has expired. **Action:** Contact your Kerberos administrator, and register with the Kerberos database again. Use the KDB_EDIT or the KADMIN utility.
	Service expired	The requested service registered with the Kerberos database has expired. **Action:** The service provider should register the service in the database again.
	Authentication expired	User's authentication has expired. **Action:** Get new tickets and authentication.
	Unknown protocol version number	The protocol version number does not match that of the TCP/IP product. **Action:** Install versions of Kerberos that use compatible protocol version numbers.

Table 21 (Page 2 of 7). KERBEROS Messages and Codes

Exit Code	Message	Explanation
	Principal unknown	The supplied principal name is not registered in the database. **Action:** If the principal name is typed incorrectly, try again with the correct registered principal name. If you are not registered in the database, ask your Kerberos administrator to register your name in the database.
	Principal not unique	There is more than one principal in the same database. **Action:** Report the problem to your system administrator. Nonunique principals should be checked by the KDB_EDIT or KADMIN registration program when registering.
	Principal has null key	The principal has registered with a null key. **Action:** Report the problem to your system administrator. A null key is not accepted when registering.
	Cannot read ticket file (krb_get_cred)	The ticket file does not exist, or the ticket file is corrupted. **Action:** Use the KINIT command to get the initial ticket, and a ticket file is created in TMP directory accordingly.
	Cannot find ticket (krb_get_cred)	The ticket for the requested service is not found in the ticket file. **Action:** Use the KINIT command to get the initial ticket. If mk_req() is used, it will get the desired ticket from the Kerberos ticket granting server automatically.
	Ticket granting ticket expired (krb_mk_req)	The time interval since you issued the KINIT command has exceeded the predefined ticket life. **Action:** Issue the KINIT command to obtain the new initial ticket. You can negotiate with the Kerberos master administrator for the maximum ticket life time (0-255)*5 minutes.
	Cannot decode authenticator (krb_rd_req)	The authenticator sent by the client cannot be decoded. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client.
	Ticket expired (krb_rd_req)	Expired ticket from the client has been detected by the krb_rd_req routine of the server. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client.
	Ticket issue date too far in the future (krb_rd_req)	Invalid ticket issuing date has been detected. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client.
	Ticket for the wrong server (krb_rd_req)	The ticket received by the server is for another server. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client.
	Request is inconsistent (krb_rd_req)	The information in the authenticator is different from that in the ticket. A possible reply is detected. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client.

Table 21 (Page 3 of 7). KERBEROS Messages and Codes

Exit Code	Message	Explanation
	Time is out of bounds (krb_rd_req)	The time interval for a ticket traveling in the network is longer than the predefined CLOCK_SKEW time. **Action:** Reject the service, and inform the client of the error message. **Note:** To change the CLOCK_SKEW time, you should change the CLOCK_SKEW defined in the KRB.H (default is 300 seconds), recompile your KRB library, and relink the server with the new KRB library.
	Incorrect network address (krb_rd_req)	The client's network address in the authenticator is not consistent with the packet received. It can be a reply by an intruder. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client.
	Protocol version mismatch (krb_rd_req)	The protocol version of the ticket from a client is different from the version that the server is using. **Action:** Notify the client that the protocol version defined in the PROT.H does not agree with that in the server. To change the protocol version, modify the protocol version in the PROT.H, recompile the KRB library, and link the client program or server program to the new KRB library.
	Illegal message type (krb_rd_req)	The message type defined in the ticket is illegal. The valid message types are defined in the PROT.H file. The corresponding KRB library function pairs in clients and server are not consistent. rd_req() , rd_safe(), and rd_priv() in the client program do not match their counterparts in the server. The counterparts are mk_req(), mk_safe(), and mk_priv(), respectively. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client. Verify that the function pairs in the client and server are consistent and the version number defined in the PROT.H is the same.
	Message integrity error (krb_rd_req)	The krb_rd_req() function detects an error in the ticket format. The ticket may have been modified. **Action:** Depends on your application. You can reject the request, or take other action, such as sending the proper message back to the client.
	Current password is null (get_pw_tkt)	The principal had a null password in the Kerberos database, which indicates that the principal is known to Kerberos, but does not have a password yet. **Action:** Try to get a ticket for the principal *default.changepw@realm* to use the *changepw.KRB_MASTER* server. Use the password *changepwkrb* rather than *cpw*. Return GT_PW_NULL. If this routine succeeds, a ticket and session key for either the principal *user.instance@realm*, or *default.changepw@realm*, will be in the your ticket file, which allows you to use the password-changing server.

Table 21 (Page 4 of 7). KERBEROS Messages and Codes

Exit Code	Message	Explanation
	The current password incorrect (get_pw_tkt)	The current password was invalid. **Action:** Enter the password again. If the message appears again, see your system administrator.
	Retry count exceeded (send_to_kdc)	Retry count has been exceeded, but you do not get an answer from the Kerberos server. The Kerberos servers listed in your KRB.CNF file cannot be reached. **Action:** Ping the host where the Kerberos server is running to see if the host can be reached. If the host can be reached, check with the system administrator to see if the Kerberos server is running on that host.
	Cannot send request (send_to_kdc)	Cannot get local realm **Action:** Verify that the local realm is defined in your KRB.CNF file in ETC directory
	Password incorrect	The password is incorrect, the message is not correctly decrypted using the current password. **Action:** Provide the correct password.
	Protocol error (get_intkt)	Wrong protocol version. **Action:** Get the updated Kerberos code.
	Generic error (get_intkt)	Other errors detected by krb_get_in_tkt(). **Action:** See the Kerberos administrator.
	Do not have ticket granting ticket (get_ad_tkt)	The initial ticket is not in the ticket file. **Action:** Issue the KINIT command to obtain the initial ticket.
	No ticket file (tf_util)	The ticket file is not created. **Action:** Issue the KINIT command.
	Cannot access ticket file (tf_util)	The ticket file is in the wrong mode. **Action:** Verify that the 'tkt0' in TMP directory can be accessed.
	Cannot lock ticket file; try later (tf_util)	Could not lock the ticket file, even after several tries. **Action:** Try later.
	Bad ticket file format (tf_util)	If the name was null, EOF was encountered, or the name was longer than ANAME_SZ, TKT_FIL_FMT is returned. **Action:** Check the ticket file format. If the format is wrong, get new tickets.
	Read the ticket file before tf_init (tf_util)	tf_init not called first. **Action:** Call tf_init before any calling for any access to the ticket file.
	Bad Kerberos name format (kname_parse)	The Kerberos name is detected as an incorrect Kerberos name. **Action:** For the correct Kerberos name format, see Chapter 12, "Setting Up Your Kerberos System."
	Generic Kerberos error (kfailure)	Other errors are detected by Kerberos. **Action:** See the Kerberos administrator.

Table 21 (Page 5 of 7). KERBEROS Messages and Codes

Exit Code	Message	Explanation
	KADMIN error : Insufficient access to perform requested operation	As a remote administrator, you are not authorized to perform the requested operation. **Action:** Ask the Kerberos master administrator to add your name to the ADM_ACL file on the host where the Kerberos database resides. Error message from EXT_SRTB.
	Error message from EXT_SRTB. Unknown control argument	An argument parameter other than -n is specified in the command line. The format of ext_srtb is: `ext_srtb [-n] inst1 inst2` If -n is specified, then the program tries to read the master key from the file MKEYFILE, which is defined in KDC.H. **Action:** Verify that the MKEYFILE defined in KDC.H exists.
	Could not read the master key	You specified -n option in the EXT_SRTB program, and the MKEYFILE cannot be accessed. **Action:** Drop the -n option, and enter the master key from the keyboard.
	Could not get local realm	The local realm is missing from the first line of the KRB.CNF file. **Action:** Add your local realm to your KRB.CNF file in the ETC directory.
	Could not create file <inst>.stb	The EXT_SRTB program has a problem creating the `inst.stb` file. The instance name may be too long (more than 7 characters), making the file name invalid. **Action:** Do not supply instance names longer than 7 characters when registering a service with the Kerberos database.
	More than 40 entries found	There are more than 40 entries in the database with the specified instance name. **Action:** Decrease the number of database entries.
	Bad instance name:	The instance name is not valid. **Action:** Verify the instance name for validity.
	Error writing the output file. Terminating.	The program has difficulty writing the entries to the `inst.stb` file. **Action:** Check for adequate file space.
	Could not open database	The Kerberos database does not exist or is locked. The ext_srtb must be run on the host from which the Kerberos database can be accessed. **Action:** Make the database accessible to the ext_srtb program. If the database does not exist, run KDB_INIT program to create it. See Chapter 12, "Setting Up Your Kerberos System" for information about KDB_INIT.
	Kerberos : gethostname error	Unable to get the host name environment variable. **Action:** Set the HOSTNAME environment variable properly. You can add the SET HOSTNAME=*host_name* statement in your CONFIG.SYS file, and reboot the machine, or issue the SET command on the command line.

Table 21 (Page 6 of 7). KERBEROS Messages and Codes

Exit Code	Message	Explanation

	Kerberos : udp/Kerberos unknown service	The Kerberos service is not defined in the SERVICES file in the ETC directory.
		
		Action: If you installed the TCP/IP for OS/2 product using ICAT, this file should be installed in your ETC directory.
		
		Note: You can add the following lines in your SERVICES file :
		
		`Kerberos 750/udp kdc # Kerberos authentication--udp`
		`Kerberos 750/tcp kdc # Kerberos authentication--tcp`

	Kerberos cannot bind socket	The Kerberos service ports 750/tcp and 750/udp are bound. The Kerberos server is running, or has been stopped, but the socket port has not been released. **Action:** Reboot the machine.
	Cannot open the database:	The Kerberos server and the ADM_SERV server have trouble opening the Kerberos database. **Action:** Verify that the database is in the KERBEROS directory.
	KINIT : k_gethostname failed	The KINIT program does not know on which host the Kerberos authentication server resides. **Action:** Verify that the KRB.CNF file in ETC directory is properly defined. Verify that the host specified in the KRB.CNF can be reached by using the PING function.
	KINIT : bad Kerberos name/instance/realm format	The specified Kerberos name is not correct. **Action:** See Chapter 12, "Setting Up Your Kerberos System" for the correct Kerberos name format.
	KINIT : krb_get_lrealm failed	KINIT is unable to get the local realm. **Action:** Verify that the local realm is specified in the KRB.CNF file.
	Cannot find local realm	KINIT is unable to get the local realm. **Action:** Verify that the local realm is specified in the KRB.CNF file.
	Cannot fetch local realm	KINIT is unable to get the local realm. **Action:** Verify that the local realm is specified in the KRB.CNF file.
	Bad key supplied	A weak key or semi-weak key is detected. A trivial key pattern is detected if you attempt to break the key. **Action:** Supply the correct key.
	Could not find administrating host	The KRB.CNF file is not edited properly. KADMIN is unable to find the host where the database resides. **Action:** Verify that the host name where the Kerberos database resides is correctly specified in the KRB.CNF file. See Chapter 12, "Setting Up Your Kerberos System" for information about KRB.CNF.
	Administrating host name is unknown	The host name specified in the KRB.CNF cannot be resolved. **Action:** Verify that the name server is running properly or the hosts file contains the administrating host. If you can PING the administrating host, this problem is solved.

Table 21 (Page 7 of 7). KERBEROS Messages and Codes

Exit Code	Message	Explanation
	Entry already exists in database	You are trying to register a user with the KADMIN ADD subcommand, and an entry already exists in the database with the same name. **Action:** Use another name or quit the ADD operation.
	Insufficient access to perform requested operation	Your name is not listed in the ADM_ACL.GET ADM_ACL.MOD, ADM_ACL.ADD file in the KERBEROS directory of the host where the database resides. **Action:** To perform the ADD function, ask the database administrator to add your name to ADM_ACL.ADD. Your name must also be added for using the GET (ADM_ACL.GET) and CPW (ADM_ACL.MOD) operations.
	No such entry in the database	You used GET or CPW to access an nonexistent entry in the database. **Action:** Register the user before attempting the access.

LPD

Table 22. LPD Messages and Codes

Exit Code	Message	Explanation
1	Unknown Option '%c'	The option '%c' specified on the command line is invalid. **Action:** Check the parameters on the command line and respecify.
N/A	Error: receiving command from client: rc = %d	An error occured while reading command data from the client. **Action:** Check the connecting client to verify that their configuration is correct.
N/A	Lpd: error receiving data (errno = %d)	An error occured while reading data from the client. **Action:** Check the connecting client to verify that their configuration is correct.
N/A	Print request aborted by Client!	The client requested that any print job that is currently being created be cancelled.
N/A	Error: Invalid Control file!	The client has not sent a valid control file for the current job. **Action:** Check the connecting client to verify that their configuration is correct.
N/A	Error: Invalid Data file!	The client has not sent a valid data file for the current job. **Action:** Check the connecting client to verify that their configuration is correct.
N/A	Invalid socket specified!	An invalid parameter was passed to lpd on the command line. **Action:** Check the parameters on the command line and respecify.

LPQ

Table 23. LPQ Messages and Codes

Exit Code	Message	Explanation
1	No Printer was specified!	The LPQ command was specified without a printer, nor was there an LPR_PRINTER environment variable set. **Action:** Respecify the LPQ command with the -p *printer* parameter, or set the environment variable LPR_PRINTER.
1	No Server was specified!	The LPQ command was specified without a server, nor was there an LPR_SERVER environment variable set. **Action:** Respecify the LPQ command with the -s *server* parameter, or set the environment variable LPR_SERVER.
N/A	Unknown Option '%c'	The option %c specified on the LPQ command line is invalid. **Action:** Respecify the LPQ command with the correct option specified.
2	LPQ: Unknown server %s!	The server %s is not a valid server. **Action:** Respecify the LPQ command line with the correct server.
1	printer: printer/tcp: unknown service	LPQ was unable to determine the socket number to connect to on the server to reach the Remote Printer Server. **Action:** Check your SERVICES file to verify that an entry exists for ***printer***.
2	Unable to bind socket	The LPQ protocol states that all LPQ requests must come from a port within the range of 721 to 731. Because of a time-out on port numbers, there is a chance to run out of available ports.

LPR

Table 24 (Page 1 of 3). LPR Messages and Codes

Exit Code	Message	Explanation
N/A	Can't open %s:	Unable to open the file '%s' to send to the server. **Action:** Check the file name specified on the LPR command line.
N/A	Early EOF found transmitting file	An imbedded End Of File marker was found in the file. **Action:** Retry the LPR command specifying the -b option for binary files.
N/A	Unable to connect to %s (errno = %d)!	LPR was unable to connect to the server %s. **Action:** Verify that the server specified is correct.
N/A	printer: printer/tcp: unknown service	LPR was unable to determine the socket number to connect to on the server to reach the Remote Printer Server. **Action:** Check your SERVICES file to verify that an entry exists for ***printer***.

Table 24 (Page 2 of 3). LPR Messages and Codes

Exit Code	Message	Explanation
N/A	unknown host	LPR was unable to connect to the server specified. **Action:** Verify that the server specified is correct.
N/A	Unable to bind socket	The LPR protocol states that all LPR requests must come from a port within the range of 721 to 731. **Action:** Because of a time-out on port numbers, there is a chance to run out of available ports when sending lots of files in a short period of time. LPR will retry sending the file if this does occur. If for some reason files do not get sent, you can adjust the number of retries as well as the delay between retries.
N/A	Unable to connect to foreign host	LPR was unable to connect to the specified server because no server was running on that host. **Action:** Verify that an LPD server is running on the remote host specified as the server.
N/A	Unable to receive response:	The specified server is not responding to LPR's requests. **Action:** Check the configuration of the LPD server that is running on the remote host specified as the server.
N/A	Server closed connection prematurely	The specified server closed the connection prematurely. **Action:** Check the configuration of the LPD server that is running on the remote host specified as the server.
N/A	Server Error: Server cannot open or write to printer	An invalid printer name was specified. **Action:** Check the printer name specified on the LPR command line, and respecify if necessary.
N/A	Server Error: Out of storage space	The LPD server was unable to satisfy your request because it was out of storage. **Action:** Try to free up resources at the remote host running the LPD server.
N/A	Server Error: Unknown error:	The LPD server reported an unknown error back to LPR. **Action:** Check the configuration of the LPD server that is running on the remote host specified as the server.
N/A	Unable to send file %s after %d tries!	LPR was unable to print the file %s after %d retries. **Action:** Retry the LPR command, increasing either or both of the -r *retries* or -q *delay* parameters.
N/A	File not found: %s	LPR was unable to find the file %s. **Action:** Verify that the file exists on the local host, and respecify the LPR command if necessary.
1	Cannot specify -f and -b flags together!	The user specified both the -f and -b flags on the LPR command line. These flags are mutually exclusive. **Action:** Respecify the LPR command with only the required flag.
1	Invalid delay specified!	The user specified an out of range value for the -q *delay* parameter. **Action:** Respecify the LPR command with a value for *delay* between 0 and 30.

Table 24 (Page 3 of 3). LPR Messages and Codes

Exit Code	Message	Explanation
1	Invalid Number of retries specified!	The user specified an out of range value for the -r *retries* parameter. **Action:** Respecify the LPR command with a value for *retries* between 0 and 5.
1	No Printer was specified!	The LPR command was specified without a printer, nor was there an LPR_PRINTER environment variable set. **Action:** Respecify the LPR command with the -p *printer* parameter, or set the environment variable LPR_PRINTER.
1	No Server was specified!	The LPR command was specified without a server, nor was there an LPR_SERVER environment variable set. **Action:** Respecify the LPR command with the -s *server* parameter, or set the environment variable LPR_SERVER.
N/A	Warning: Sending unknown options to the server!	The LPR command was specified with unknown options. LPR will send these options to the specified server as part of the control file.

LPRM

Table 25 (Page 1 of 2). LPRM Messages and Codes

Exit Code	Message	Explanation
1	No Printer was specified!	The LPRM command was specified without a printer, nor was there an LPR_PRINTER environment variable set. **Action:** Respecify the LPRM command with the -p *printer* parameter, or set the environment variable LPR_PRINTER.
1	No Server was specified!	The LPRM command was specified without a server, nor was there an LPR_SERVER environment variable set. **Action:** Respecify the LPRM command with the -s *server* parameter, or set the environment variable LPR_SERVER.
1	No User or Agent was specified!	The LPRM command was specified without an agent, nor was there a USER environment variable set. **Action:** Respecify the LPRM command with the -a *agent* parameter, or set the environment variable USER.
N/A	Unknown Option '%c'	The option %c specified on the LPRM command line is invalid. **Action:** Respecify the LPRM command with the correct option specified.
2	LPRM: Unknown server %s!	The server %s is not a valid server. **Action:** Respecify the LPRM command line with the correct server.
1	printer: printer/tcp: unknown service	LPRM was unable to determine the socket number to connect to on the server to reach the Remote Printer Server. **Action:** Check your SERVICES file to verify that an entry exists for printer.

Table 25 (Page 2 of 2). LPRM Messages and Codes

Exit Code	Message	Explanation
2	Unable to bind socket	The LPRM protocol states that all LPRM requests must come from a port within the range of 721 to 731. Because of a time-out on port numbers, there is a chance to run out of available ports.

LPRMON

Table 26. LPRM Messages and Codes

Exit Code	Message	Explanation
1	Cannot specify -f and -b flags together!	The user specified both the -f and -b flags on the LPRMON command line. These flags are mutually exclusive. **Action:** Respecify the LPRMON command with only the required flag.
1	Invalid delay specified!	The user specified an out of range value for the -q *delay* parameter. Respecify the LPRMON command with a value for *delay* between 0 and 30.
1	Invalid Number of retries specified!	The user specified an out of range value for the -r *retries* parameter. **Action:** Respecify the LPRMON command with a value for *retries* between 0 and 5.
1	No Printer was specified!	The LPRMON command was specified without a printer, nor was there an LPR_PRINTER environment variable set. **Action:** Respecify the LPRMON command with the -p *printer* parameter, or set the environment variable LPR_PRINTER.
1	No Server was specified!	The LPRMON command was specified without a server, nor was there an LPR_SERVER environment variable set. **Action:** Respecify the LPRMON command with the -s *server* parameter, or set the environment variable LPR_SERVER.
N/A	Warning: Sending unknown options to the server!	The LPRMON command was specified with unknown options. LPRMON will send these options to the specified server as part of the control file.

NFS Client

Table 27 (Page 1 of 5). NFS Client Messages and Codes

Exit Code	Message	Explanation
N/A	>>>RETRY failure!<<<	The remote server did not respond and NFS had to resend the request. **Action:** Verify that that your NFS server is functioning properly.

Table 27 (Page 2 of 5). NFS Client Messages and Codes

Exit Code	Message	Explanation
N/A	NFSC0001: Drive 'x' is not properly attached and will be detached.	The drive was mounted improperly, possibly because an error might have occurred during the mount. **Action:** Run NFSCLEAN to detach the drive.
N/A	NFSC0002: Drive 'x' is not attached and therefore cannot be queried.	You attempted a QMOUNT on a drive that was not mounted by NFS. **Action:** Mount the drive with the MOUNT command.
N/A	pfsnfs—>pnfsmount malloc failed	**Action:** Note the circumstances under which this occurred, and contact IBM support.
N/A	Semaphore allocation service error.	The NFS control program could not allocate a semaphore for its use. This could be the result of improperly terminating the control program, or attempting to use the same semaphore name in a different program. **Action:** Reboot your computer and try again. If this error persists, make a note of the circumstances and contact IBM support.
N/A	There are mounted drives. Please unmount all NFS drives before ending this program. 'umount *' can be used to unmount all NFS drives.	You attempted to end the NFS control program when NFS drives were mounted. **Action:** Unmount all NFS drives before terminating the control program. Use the UMOUNT command with the asterisk (*) option to unmount all the drives.
N/A	Terminating child processes. Please be patient... killing biod:xxx process number:xxx rc:xxx	This is normal when terminating the NFS control program. **Action:** No action is required.
N/A	There are mounted drives. Please use the program 'nfsclean' to unmount the mounted drives after restarting this program.	You selected the close option from the NFS control program window's system menu, and terminated NFS while drives were mounted. **Action:** When you restart NFS, be sure to run NFSCLEAN.
N/A	The maximum number of mount requests has been exceeded	**Action:** Please make a note of the circumstances and contact IBM support.
N/A	mount: <host> not in hosts database	An error occurred while resolving the host name. **Action:** Verify that that your name server and hosts file are correct and operational.
N/A	ERROR:READ ZERO bytes	The server to which you are connected returned zero bytes when NFS attempted to read a file. **Action:** Verify that that your NFS server is functioning properly.
N/A	biod:read block error.	One of the BIODs encountered an error when attempting to read from an NFS server. **Action:** Verify that that your NFS server is functioning properly.
N/A	biod:write block error.	One of the BIODs encountered an error when attempting to write to an NFS server. **Action:** Verify that that your NFS server is functioning properly.

Table 27 (Page 3 of 5). NFS Client Messages and Codes

Exit Code	Message	Explanation
N/A	SYMBOLIC LINK is: name '<name>' This link seems to be recursive.	The NFS Client believes that the symbolic link is recursive. This can also occur when there are too many links to link. **Action:** Delete the recursive symbolic link or shorten the number of links in the path.
N/A	mount: <host> server not responding	The NFS control program could not contact the NFS server when it was mounting a new file system. **Action:** Verify that that your NFS server is functioning properly.
N/A	mount:Unable to bind socket errno:xxx	**Action:** Please make a note of the circumstances and contact IBM support.
N/A	error opening '\pipe\nfs\0000000x'	The NFS control program could not allocate a named pipe for its use. This could be the result of improperly terminating the control program, or attempting to use the same pipe name in a different program. **Action:** No action is required. The NFS control program attempts to open the pipe several times before giving up. If no additional messages appear, then NFS has successfully opened the pipe.
N/A	The Local Hostname should be set to a value Please add a line of the following form to the CONFIG.SYS file SET hostname=<local-internet-host-name>	You did not set the HOSTNAME environment variable in your CONFIG.SYS file. **Action:** Do one of the following: • Use ICAT to modify your CONFIG.SYS file and add the proper SET HOSTNAME= statement. • Add the following line to your CONFIG.SYS file: SET HOSTNAME=<local-internet-hostname>
N/A	xdr_rrok: FAILED, can't get mbuf	**Action:** Please make a note of the circumstances and contact IBM support.
N/A	Delete failed	LN could not delete the link. **Action:** Verify that that your NFS server is functioning properly and that you have been granted access to the link.
N/A	You do not have ownership of this disk.	You attempted to mount a VM disk that you do not own.
N/A	Password change required by host.	Your MVS password has expired and must be changed. **Action:** Do one of the following: • Use the -n option of the MVSLOGIN command to change your MVS password. • Log on to your MVS account and change your password.
N/A	NFSWAIT: Unable to access NFS shared segment	The NFS control program was not started properly. **Action:** Stop and restart the NFS control program. If this does not correct the error, reboot your computer and try to start the program again.
1	unrecognized option: '-<option>'	You entered an option that is not valid for this command. **Action:** See the *IBM TCP/IP Version 1.2 for OS/2: User's Guide* or *IBM TCP/IP Version 1.2 for OS/2: Quick Reference Guide* for the correct syntax.

Table 27 (Page 4 of 5). NFS Client Messages and Codes

Exit Code	Message	Explanation
1	nfsbiod could not be found in the path	The NFS control program could not find NFSBIOD.EXE in your path. **Action:** Add the directory containing NFSBIOD.EXE to your path.
1	A Biod critical error has occurred Please stop all NFSC programs and start the programs again	The NFS control program could not allocate the pipe designated for its use. This could be the result of improperly terminating the control program or attempting to use the same pipe name in a different program. **Action:** Stop and then restart the NFS control program. If this does not correct the error, reboot the computer.
2	The server could not find the directory specified. ERROR_FILE_NOT_FOUND	The server could not find the file or directory specified. **Action:** Respecify with the correct file or directory name.
2	The NFS Client is already executing (process id:xxx).	You attempted to start the NFS control program more than once.
3	Error: The NFS installed file system is not loaded! Please add 'ifs=nfs.ifs' to the c:\config.sys file.	You did not add the correct IFS statement for NFS in your CONFIG.SYS file. **Action:** Use ICAT to modify your CONFIG.SYS and add the proper IFS statement.
5	ERROR_ACCESS_DENIED	You tried to do a file operation on an invalid file. **Action:** Respecify with the correct file name.
12	ERROR_INVALID_ACCESS	The file you are attempting to access no longer exists. **Action:** Verify that the file exists.
	ERROR_INVALID_ACCESS	You do not have ownership to perform the requested operation. **Action:** Check the permission settings at the server.
19	ERROR_WRITE_PROTECT	You are attempting to write to a read-only file system. **Action:** Use one of the following options. • Remount the server with read-write permission. • Check the export method at the server to verify that it was exported as a read-write file system.
27	A hard error has occurred at the server. ERROR_SECTOR_NOT_FOUND	A hard error occurred. **Action:** Run the server diagnostics.
50	The NFSCTL program is probably not running. ERROR_NOT_SUPPORTED	The NFSCTL program is probably not running. **Action:** Start the NFSCTL program, either directly or using NFSC. -or- The operation you have requested is not supported.
55	Device not found at the server. ERROR_DEV_NOT_EXIST	The device does not exist. **Action:** Verify that the device exists on the server.

Table 27 (Page 5 of 5). NFS Client Messages and Codes

Exit Code	Message	Explanation
60	Server timed out.	The server response timed out. **Action:** Use one of the following options. • Check the NFS server on the remote machine. • Unmount all drives, stop the NFS control program, and restart the program with a longer time out. • Verify that the directory you are mounting is valid, using the SHOWEXP command.
80	ERROR_FILE_EXISTS	The file you specified already exists. **Action:** Respecify with a different file name.
85	Drive *drive* is already attached to the file system.	The drive you are trying to mount is already used. **Action:** Select a different drive. Use the QMOUNT command to determine the drives that are in use.
100	The buffer size requested (xxx) is larger than the supported maximum of (yyy)	You started the NFSCTL program with a buffer size (the -b option) larger than NFS can handle. **Action:** Use a smaller value for the -b option.
100	Only one mount request can be issued at a time	You attempted to execute two MOUNT commands simultaneously. **Action:** Wait for the first mount to finish before issuing another MOUNT command.
112	ERROR_DISK_FULL	The file is too large, or the file has grown beyond the server's limit. **Action:** Contact your system administrator for more disk space.
145	ERROR_DIR_NOT_EMPTY	You are attempting to remove a directory that is not empty. **Action:** Delete all files within the specified directory, and try again.
206	ERROR_FILENAME_EXCED_RANGE	The file name specified is too long. **Action:** Respecify the file, using the correct format and length.
267	ERROR_DIRECTORY	The operation specified an invalid directory name in a directory operation. **Action:** Respecify with a valid directory name.

PORTMAP

Table 28 (Page 1 of 2). PORTMAP Messages and Codes

Exit Code	Message	Explanation
1	Error: Portmap cannot create socket	The sockets socket() procedure did not work in the portmap routine. **Action:** See the socket() call in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for specific errors. The socket() call can fail because of a protocol mismatch. The sockets may all be in use. Verify that TCP or UDP is operating. Verify that memory is available. Domain may be incorrectly specified. Run the NETSTAT command with the sockets options for more information. Sockets remain active for a time-out period after they have been closed.
1	Error: portmap cannot bind	The sockets bind() function did not work in the portmap routine. The portmap may have been stopped with the address still in use. **Action:** Clear the portmap, and reinitiate. The portmap workspace may have to be completely reset. See the bind() call in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for specific errors. Verify that memory is available. The socket descriptor may have been altered by another function. Run the NETSTAT command with the sockets options for more information.
1	Error: could not do tcp_create	The rpc svctcp_create() function did not work in the portmap routine. **Action:** See the svctcp_create() call description in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for information. The socket descriptor may have been altered by another function. Verify that memory is available and allocated. The socket descriptor may not be available. Run the NETSTAT command with the sockets options for more information.
1	Error: could not do udp_create	The rpc svcudp_create() function did not work in the portmap routine. **Action:** See the svcudp_create() call description in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for information. The socket descriptor may have been altered by another function. Verify that memory is available and allocated. The socket descriptor may not be available. Run the NETSTAT command with the sockets options for more information.
N/A	Error: svc_sendreply1 svc_sendreply1 Set a program, version to port mapping	The rpc svc_sendreply() function did not work in the portmap routine when it attempted to return information to the caller. **Action:** See the rpc svc_sendreply() call description in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for information. The transport handle may have been altered by another function. The transport handle may not be available. Verify that memory is available and allocated. An xdr error may have occurred on the transfer. A socket error may have occurred. Test the path with rpcinfo, ping, netstat, make changes, and try again.

Table 28 (Page 2 of 2). PORTMAP Messages and Codes

Exit Code	Message	Explanation
N/A	Error: svc_sendreply2 svc_sendreply2 -Remove a program, version to port mapping	The rpc svc_sendreply() function did not work in the portmap routine when it attempted to return information to the caller. **Action:** See the rpc svc_sendreply() call description in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for information. The transport handle may have been altered by another function. The transport handle may not be available. Verify that memory is available and allocated. An xdr error may have occurred on the transfer. A socket error may have occurred. Test the path with rpcinfo, ping, netstat, make changes, and try again.
N/A	Error: svc_sendreply3 svc_sendreply3 -Lookup the mapping for a program, version	The rpc svc_sendreply() function did not work in the portmap routine when it attempted to return information to the caller. **Action:** See the rpc svc_sendreply() call description in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for information. The transport handle may have been altered by another function. The transport handle may not be available. Verify that memory is available and allocated. An xdr error may have occurred on the transfer. A socket error may have occurred. Test the path with rpcinfo, ping, netstat, make changes, and try again.
N/A	Error: svc_sendreply4 svc_sendreply4 -Return the current set of mapped program, version	The rpc svc_sendreply() function did not work in the portmap routine when it attempted to return information to the caller. **Action:** See the rpc svc_sendreply() call description in the *IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference* for information. The transport handle may have been altered by another function. The transport handle may not be available. Verify that memory is available and allocated. An xdr error may have occurred on the transfer. A socket error may have occurred. Test the path with rpcinfo, ping, netstat, make changes, and try again.

SENDMAIL—SENDMAIL.ERR Errors

The following are errors that can be found in the ETC\SENDMAIL.ERR file.

Table 29 (Page 1 of 2). Sendmail Messages and Codes

Exit Code	Message	Explanation
N/A	Cannot send mail	Warning message. The recipient host's SMTP is down or cannot be reached through the network. **Action:** No action to be taken. Sendmail automatically stores the mail in the MQUEUE and tries the delivery again, based on the time you specified using the `-qtime` parameter. If the problem persists for several days, verify that the note is correct by checking the data files in the MQUEUE directory. You can either make corrections to the date files or delete them.

Table 29 (Page 2 of 2). Sendmail Messages and Codes

Exit Code	Message	Explanation
N/A	Timed out waiting for *hostname*	Warning message. The connection to another SMTP server was lost. **Action:** No action to be taken. Sendmail and other SMTP servers automatically try to deliver the mail again.
N/A	Line *number*: invalid rewrite line	The rewrite rule in the ETC\SENDMAIL.CF file on the specified line is incorrectly formed. **Action:** Edit the ETC\SENDMAIL.CF configuration file and correct the line. The three separate fields must be separated by tabs, not spaces. If you have a large number of these errors in your file, you probably used an editor that converts tabs to spaces, and have introduced a large number of errors into your configuration file. If you are unable to recover the changes, restore an older version of the SENDMAIL.CF file from the backups in the ETC directory or from the product disks.
N/A	Cannot change drive	Sendmail is unable to access either the MAIL or MQUEUE subdirectories. **Action:** Search the ETC\SENDMAIL.CF configuration file to find the MAIL and MQUEUE subdirectories that Sendmail expects to find. These directories are specified by the *OQ* and *Mlocal* parameters, respectfully. Verify that these directories exist and that Sendmail has write access to them.
N/A	getrequests: cannot bind	Sendmail cannot bind to port 25, the port reserved for SMTP. This occurs if Sendmail is already running, or if Sendmail has previously terminated with an error and the port was not freed. **Action:** Verify that Sendmail is not already running. If port 25 cannot be freed, reboot the machine.
N/A	fopen to inbox.ndx failed, inbox.ndx locked	Sendmail cannot open the MAIL\INBOX.NDX file, because it is in use by another application, usually LaMail. **Action:** Normally, the problem corrects itself, as the other application finishes accessing the INBOX.NDX file and allows Sendmail to update it. If the problem persists, the INBOX.NDX file may have become corrupted and must be erased. If the INBOX.NDX file must be erased, manually read the remaining mail in the MAIL directory to prevent the loss of information.
N/A	Sendmail gave up trying to open inbox.ndx, mailfile delivered, inbox not updated	Sendmail is unable to open the MAIL\INBOX.NDX file because it is in use by another application, usually LaMail. **Action:** Normally, the problem corrects itself, as the other application finishes accessing the INBOX.NDX file and allows Sendmail to update it. If the problem persists, the INBOX.NDX file may have become corrupted and needs to be erased. If the INBOX.NDX file needs to be erased, manually read the remaining mail in the MAIL directory to prevent the loss of information.

SENDMAIL—Exit Codes

The following are exit codes returned by Sendmail.

Table 30 (Page 1 of 2). Sendmail Messages and Codes

Exit Code	Message	Explanation
64	Command line usage error	The command was used incorrectly. For example, the command used the wrong number of arguments, an invalid flag, or incorrect syntax. **Action:** Verify the command you entered, correct if necessary, and try again.
65	Data format error	The input data was incorrect. The data you entered should be used only as user's data, not as system files. **Action:** The mail file you are trying to send is in the wrong format. Verify the format, correct if necessary, and try again.
66	Cannot open input	An input mail file does not exist or cannot be read. This error could also include errors, such as "No message" to a mailer. **Action:** Verify the file name of the input file, correct if necessary, and try again.
67	Addressee unknown	The user you specified does not exist. **Action:** Verify the target user ID, correct if necessary, and try again.
68	Host name unknown	The host you specified does not exist. This message occurs for mail addresses or network requests. **Action:** Check the target host to see if it is defined in the HOSTS file or name server.
69	Service unavailable	A service is unavailable. This message occurs if a support program or file does not exist, or it occurs as a general message when some task you wanted to perform does not work. **Action:** Please make a note of the circumstances and contact IBM support.
70	Internal software error	An internal software error has been detected. This message should be limited to non-operating system errors. **Action:** Please make a note of the circumstances and call IBM support.
72	Critical OS file missing	A system file does not exist or cannot be opened. **Action:** Verify that the SENDMAIL.CF and SERVICES files exist in your ETC directory.
73	Can't create (user) output file	An output file you specified cannot be created. For example, you cannot open a file when receiving mail, your disk is full, or a file is read-only. **Action:**
75	Temp failure; user is invited to retry	This is a temporary failure. For example, a mailer could not create a connection. **Action:** You should try to send the mail again.

Table 30 (Page 2 of 2). Sendmail Messages and Codes

Exit Code	Message	Explanation
76	Remote error in protocol	The remote system attempted an impossible task during a protocol exchange. **Action:** Verify that the target host is working correctly.

SNMP

Table 31. SNMP Messages and Codes

Exit Code	Message	Explanation
N/A	Unknown SNMP request type	There are only three SNMP request types: • snmp_get • snmp_getnext • snmp_set. **Action:** Verify that you have entered a valid type.
N/A	arptbl_setup:cannot get memory for arptblp1	In file MIB_AT.C, the c function calloc() allocates storage for the data structure *arp_ent*. **Action:** Run fewer applications, and restart the application that failed.
N/A	iftable_setup:cannot get memory for ifp	In file MIB_INTE.C, the c function calloc() allocates storage for the data structure *if_ent*. **Action:** Run fewer applications, and restart the application that failed.
N/A	if_addrsetup:cannot get memory for ipadr1	In file MIB_IPAD.C, the c function calloc() allocates storage for the data structure *ipadr_ent*. **Action:** Run fewer applications, and restart the application that failed.
N/A	iproute_setup:cannot get memory for iprouteptr	In file MIB_IPRO.C, the c function calloc() allocates storage for the data structure *iproute_ent*. **Action:** Run fewer applications, and restart the application that failed.

TALK

Table 32. TALK Messages and Codes

Exit Code	Message	Explanation
N/A	talk: *hostname*:	Your HOSTNAME environment variable is defined as *hostname*, which cannot be resolved. **Action:** Define *hostname* in your HOSTS file or name server.
N/A	talk: *host*: invalid host name	The host is probably valid, but it cannot be resolved. **Action:** Define the *host* in your HOSTS file or name server.

Telnet Server

Table 33. Telnet Server Messages and Codes

Exit Code	Message	Explanation
	Invalid password.	You entered an invalid password. **Action:** Contact your system administrator for password information.
	Telnetd: panic state = *state*	The Telnet state machine is in an unknown state. **Action:** Verify the communication parameters between client and server. For example, SYNC and FLUSH.
N/A	Unable to start up a Telnet Session to service the client	The server was unable to start up a session to process the connecting clients requests. This is usually because the server host has run out of resources.
N/A	Unable to start shell specified in COMSPEC variable in config.sys!	The telnet server was unable to start up a command processor as specified by the COMSPEC environment variable. **Action:** Check your config.sys to verify the value of this environment variable.
N/A	Login failure: {Hard error abort Trap operation DosKillProcess Unknown Failure}	An unusual error occured during a client login. **Action:** Verify that the correct login.exe in the TCPIP\BIN directory is the login that is being run (verify that this is the first executable with the name 'login' in your PATH).
	Could NOT execute login.exe command!	The server was unable to run the login program to allow clients access. **Action:** Verify that your PATH environment variable is correct.

Appendix G. Sample BOOTPTAB File

This appendix contains a sample BOOTPTAB file.

```
# tcpip\etc\bootptab: database for bootp server BOOTPD
# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
# first field -- hostname
#   (may be full domain name and probably should be)
#
# bf -- bootfile
# ds -- domain name server address list
# gw -- gateway address list
# ha -- host hardware address (follows ht)(hexadecimal)
# hd -- home directory
# hn -- send host name (boolean tag)
# ht -- host hardware type (precedes ha) (ethernet, ether)
# ip -- host IP address
# sm -- subnet mask
# tc -- template host (points to similar host entry)

#
# Be careful about including backslashes where they're needed.
# Strange things can happen when a backslash is
# omitted where one is intended.
#


# First, we define a global entry which specifies the info every
# host uses.  ***Make changes here***

global.dummy:\
        :sm=255.255.224.0:\
        :hd=/bootpd/trypd:bf=null:\
        :ds=9.67.30.100 9.67.30.99:

# Then information specific to a subnet. ***Make changes here***

subnet60.dummy:\
        :tc=global.dummy:gw=9.67.60.129
subnet22.dummy:\
        :tc=global.dummy:gw=9.67.22.2

# Finally, individual host information that needs to be changed.
#       ***Make changes here***

ricky.tcp.raleigh.ibm.com:  tc=subnet60.dummy: ht=ethernet:\
     ha=10005a25095e:  ip=9.67.60.131: hn:
maggie.tcp.raleigh.ibm.com: tc=subnet60.dummy: ht=ethernet:\
     ha=10005a6d1877: ip=9.67.60.131: hn:
pete.tcp.raleigh.ibm.com:  tc=subnet22.dummy: ht=ethernet:\
     ha=10005a2b1ef0:  ip=9.67.30.70: hn:
homer.tcp.raleigh.ibm.com:  tc=subnet22.dummy: ht=ethernet:\
     ha=10005a251419: ip=9.67.60.70: hn:
```

Appendix H. Related Protocol Specifications

IBM is committed to industry standards. The internet protocol suite is still evolving through Requests for Comments (RFC). New protocols are being designed and implemented by researchers, and are brought to the attention of the internet community in the form of RFCs. Some of these are so useful that they become a recommended protocol. That is, all future implementations for TCP/IP are recommended to implement this particular function or protocol. These become the *de facto* standards, on which the TCP/IP protocol suite is built.

Many features of TCP/IP for OS/2 are based on the following RFCs:

RFC	Title and Author
768	*User Datagram Protocol*, J.B. Postel
783	*Trivial File Transfer Protocol*, (Revision 2), K.R. Sollins
791	*Internet Protocol*, J.B. Postel
792	*Internet Control Message Protocol*, J.B. Postel
793	*Transmission Control Protocol*, J.B. Postel
821	*Simple Mail Transfer Protocol*, J.B. Postel
822	*Standard for the Format of ARPA Internet Text Messages*, D. Crocker
823	*DARPA Internet Gateway*, R.M. Hinden, A. Sheltzer
826	*Ethernet Address Resolution Protocol: or Converting Network Protocol Addresses to 48.Bit Ethernet Address for Transmission on Ethernet Hardware*, D.C. Plummer
854	*Telnet Protocol Specification*, J.B. Postel, J.K. Reynolds
856	*Telnet Binary Transmission*, J.B. Postel, J.K. Reynolds
857	*Telnet Echo Option*, J.B. Postel, J.K. Reynolds
877	*Standard for the Transmission of IP Datagrams over Public Data Networks*, J.T. Korb
885	*Telnet End of Record Option*, J.B. Postel
919	*Broadcasting Internet Datagrams*, J.C. Mogul
922	*Broadcasting Internet Datagrams in the Presence of Subnets*, J.C. Mogul
950	*Internet Standard Subnetting Procedure*, J.C. Mogul, J.B. Postel
951	*Bootstrap Protocol*, W.J.Croft, J. Gilmore
952	*DoD Internet Host Table Specification*, K. Harrenstien, M.K. Stahl, E.J. Feinler
959	*File Transfer Protocol*, J.B. Postel, J.K. Reynolds
974	*Mail Routing And The Domain Name System*, C. Partridge
1013	*X Window System Protocol, Version 11: Alpha Update*, R.W. Scheifler
1014	*XDR: External Data Representation Standard*, Sun Microsystems Incorporated
1034	*Domain Names—Concepts and Facilities*, P.V. Mockapetris
1035	*Domain Names—Implementation and Specification*, P.V. Mockapetris

1055 *Nonstandard for Transmission of IP Datagrams Over Serial Lines: SLIP*, J.L. Romkey

1057 *RPC: Remote Procedure Call Protocol Version 2 Specification*, Sun Microsystems Incorporated

1058 *Routing Information Protocol*, C.L. Hedrick

1060 *Assigned Numbers*, J.K. Reynolds, J.B. Postel

1084 *BOOTP vendor information extensions*, J.K. Reynolds

1091 *Telnet Terminal-Type Option*, J. VanBokkelen

1094 *NFS: Network File System Protocol Specification*, Sun Microsystems Incorporated.

1118 *Hitchhikers guide to the Internet*, E. Krol

1122 *Requirements for Internet Hosts—Communication Layers*, R.T. Braden, editor

1123 *Requirements for Internet Hosts—Application and Support*, R.T. Braden, editor

1155 *Structure and Identification of Management Information for TCP/IP-Based Internets*, M.T. Rose, K. McCloghrie

1157 *Simple Network Management Protocol (SNMP)*, J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin

1179 *Line Printer Daemon Protocol*, The Wollongong Group, L. McLaughlin III, editor

1180 *TCP/IP Tutorial*, T.J. Socolofsky, C.J. Kale

1187 *Bulk Table Retrieval with the SNMP.*

1200 *Defense Advanced Research Projects Agency, Internet Activities Board IAB official protocol standards.*

1206 *FYI on Questions and Answers: Answers to commonly asked "new Internet user" questions,*, G.S. Malkin, A.N. Marine

1207 *FYI on Questions and Answers: Answers to commonly asked "experienced Internet user" questions,*, G.S. Malkin, A.N. Marine, J.K. Reynolds

1208 *Glossary of networking terms,*, O.J. Jacobsen, D.C. Lynch

1213 *Management information Base for network management of TCP/IP-based internets:MIB-II*, K. McCloghrie, M.T.Rose, editors

These documents can be obtained from:

SRI International
Network Information Systems Center
Room EJ291
333 Ravenswood Avenue
Menlo Park, CA. 94025

Many RFCs are available online. Hard copies of all RFCs are available from the NIC, either individually or on a subscription basis. Online copies are available using FTP from the NIC at `nic.ddn.mil`. Use FTP to download the files, using the following format:

```
RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS
```

Where:

nnnn	Is the RFC number.
TXT	Is the text format.
PS	Is the PostScript format.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to service@nic.ddn.mil with a subject line of RFC *nnnn* for text versions or a subject line of RFC *nnnn*.PS for PostScript versions. To request a copy of the RFC index, send a message with a subject line of RFC INDEX.

For more information, contact nic@nic.ddn.mil.

Glossary, Bibliography, and Index

Glossary

This glossary describes the most common terms associated with TCP/IP communication in an internet environment, as used in this book.

If you do not find the term you are looking for, see *IBM Dictionary of Computing*, SC20-1699.

This glossary includes some terms from *IBM Dictionary of Computing*.

For abbreviations, the definition usually consists only of the words represented by the letters; for complete definitions, see the entries for the words.

A

ABEND. The abnormal termination of a program or task.

accelerator key. A key or combination of keys that invokes an application-defined function. Also known as a function key.

action bar. The highlighted area at the top of a panel that contains the choices currently available in the application program that a user is running.

active open. The state of a connection that is actively seeking a service. Contrast with *passive open*.

adapter. (1) A piece of hardware that connects a computer and an external device. (2) An auxiliary device or unit used to extend the operation of another system.

address. The unique code assigned to each device or workstation connected to a network. A standard internet address is a 32-bit address field. This field can be broken into two parts. The first part contains the network address; the second part contains the host number.

Address Resolution Protocol (ARP). A protocol used to dynamically bind an internet address to a hardware address. ARP is implemented on a single physical network and is limited to networks that support broadcast addressing.

agent. As defined in the SNMP architecture, an agent, or an SNMP server is responsible for performing the network management functions requested by the network management stations.

American National Standard Code for Information Interchange (ASCII). (1) The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (2) The default file transfer type for FTP, used to transfer files that contain ASCII text characters.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States.

ANSI. American National Standards Institute

application. The use to which an information processing system is put, for example, a payroll application, an airline reservation application, a network application.

argument. A parameter passed between a calling program and a called program.

ARP. Address Resolution Protocol.

ASCII. American National Standard Code for Information Interchange.

asynchronous. Without regular time relationship; unexpected or unpredictable with respect to the execution of program instruction. See *synchronous*.

attribute. A characteristic or property. For example, the color of a line, or the length of a data field.

authentication server. The service that reads a Kerberos database to verify that a client making a request for access to an end-service is the client named in the request. The authentication server provides an authenticated client a ticket as permission to access the ticket-granting server.

authenticator. Information encrypted by a Kerberos authentication server that a client presents along with a ticket to an end-server as permission to access the service.

authorization. The right granted to a user to communicate with, or to make use of, a computer system or service.

B

backbone. (1) In a local area network multiple-bridge ring configuration, a high-speed link to which rings are connected by means of bridges. A backbone can be configured as a bus or as a ring. (2) In a wide area network, a high-speed link to which nodes or data switching exchanges (DSES) are connected.

batch. (1) An accumulation of data to be processed. (2) A group of records or data processing jobs brought together for processing or transmission. (3) Pertaining to activity involving little or no user action. See *interactive*.

block. A string of data elements recorded, processed, or transmitted as a unit. The elements can be characters, words, or physical records.

Boolean. A value of 0 or 1 represented internally in binary notation.

bridge. A router that connects two or more networks and forwards packets among them. The operations carried out by a bridge are done at the physical layer and are transparent to TCP/IP and TCP/IP routing.

broadcast. The simultaneous transmission of data packets to all nodes on a network or subnetwork.

broadcast address. An address that is common to all nodes on a network.

bus topology. A network configuration in which only one path is maintained between stations. Any data transmitted by a station is concurrently available to all other stations on the link.

button. (1) A mechanism on a pointing device, such as a mouse, used to request or initiate an action. (2) A rounded-corner rectangle with text inside, used in graphics applications for actions that occur when the pushbutton is selected.

C

case-sensitive. A condition in which entries for an entry field must conform to a specific lower -, upper -, or mixed-case format in order to be valid.

checksum. The sum of a group of data associated with the group and used for checking purposes.

Class A network. An internet network in which the high-order bit of the address is 0. The host number occupies the three low-order octets.

Class B network. An internet network in which the high-order bit of the address is 1 and the next high-order bit is 0. The host number occupies the two low-order octets.

Class C network. An internet network in which the two high-order bits of the address are 1 and the next high-order bit is 0. The host number occupies the low-order octet.

click. To press and release the select button on a mouse.

client. A function that requests services from a server, and makes them available to the user.

client-server relationship. A device that provides resources or services to other devices on a network is a *server*. A device that employs the resources provided by a server is a *client*.

clipboard. A temporary storage area used for copying and storing data.

CMS. Conversational Monitor System

command. The name and any parameters associated with an action that can be performed by a program. The command is entered by the user; the computer performs the action requested by the command name.

command prompt. A displayed symbol, such as [C:\] that requests input from a user.

Communications Manager. A component of OS/2 that allows a workstation to connect to a host computer and use the host resources as well as the resources of other personal computers to which the workstation is attached, either directly or through a host.

community name. The name of a group of hosts that share SNMP management network information.

compile. (1) To translate a program written in a high-level language into a machine language program. (2) The computer actions required to transform a source file into an executable object file.

Compiler. A program that translates a source program into an executable program (an object program).

CONFIG.SYS. A file that contains the configuration options for an OS/2 personal computer.

configuration file. For the base operating system, the CONFIG.SYS file that describes the devices, system parameters, and resource options of a personal computer.

connection. (1) An association established between functional units for conveying information. (2) The path between two protocol modules that provides reliable stream delivery service. In an internet, a connection extends from a TCP module on one machine to a TCP module on the other.

conversational monitor system (CMS). A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities, and operates only under control of the VM/370 VM control program.

D

daemon. A background process usually started at system initialization that runs continuously and performs a function required by other processes.

datagram. The basic unit of information that is passed across the internet, it consists of one or more data packets.

data set. The major unit of data storage and retrieval in MVS, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. Synonymous with *file* in VM and OS/2.

default. A value, attribute or option that is assumed when none is explicitly specified.

destination node. The node to which a request or data is sent.

dialog box. A movable window, fixed in size, which provides information that is required by an application to continue your request.

directory. A named grouping of files in a file system.

Distributed Program Interface. The SNMP DPI is a programming interface that provides an extension to the functionality provided by the SNMP agents.

DLL. Dynamic Link Library

DNS. Domain Name System

domain. In an internet, a part of the naming hierarchy. Syntactically, a domain name consists of a sequence of names (labels) separated by periods (dots).

Domain Name System. A system in which a *resolver* queries name servers for resource records about a host.

domain naming. A hierarchical system for naming network resources.

dotted-decimal notation. The syntactic representation for a 32-bit integer that consists of four 8-bit numbers, written in base 10 and separated by periods (dots). Many internet application programs accept dotted decimal notations in place of destination machine names.

double-precision. A specification that causes a floating-point value to be stored internally in the long format.

DPI. Distributed Program Interface.

dragging. Moving an object on the display screen as if it were attached to the pointer, or mouse; performed by holding the select button and moving the pointer.

drive. The device used to read and write data on disks or diskettes.

dynamic link library (DLL). A module containing dynamic link routines that is linked at load or run time.

E

EBCDIC. Extended binary-coded decimal interchange code.

encapsulation. A process used by layered protocols in which a lower level protocol accepts a message from a higher level protocol and places it in the data portion of the low level frame.

entry field. A panel element, usually highlighted in some manner and usually with its boundaries indicated, where users type in information.

Ethernet. The name given to a local area packet-switched network technology invented in the early 1970s by Xerox Incorporated. Ethernet uses a Carrier Sense Multiple Access/Collision Detection (CSMA/CD) mechanism to send packets.

extended binary-coded decimal interchange code (EBCDIC). A coded character set consisting of 8-bit coded characters.

eXternal Data Representation (XDR). A standard developed by SUN Microsystems Incorporated for representing data in machine-independent format.

F

file. In VM and OS/2, a named set of records stored or processed as a unit. Synonymous with *data set* in MVS.

File Transfer Protocol (FTP). A TCP/IP protocol used for transferring files to and from foreign hosts. FTP also provides the capability to access directories. Password protection is provided as part of the protocol.

folder. In LaMail, a collection of mail files that share a common attribute, such as userid, location, or subject.

foreign host. Any host on the network including the local host.

foreign network. In an internet, any other network interconnected to the local network by one or more intermediate gateways or routers.

foreign node. See *foreign host.*

FTAM. File Transfer Access and Management.

FTP. File Transfer Protocol.

G

gateway. (1) A functional unit that interconnects a local data network with another network having different protocols. (2) A host that connects a TCP/IP network to a non-TCP/IP network at the application layer. See also *router*.

H

handle. A temporary data representation that identifies a file.

header file. A file that contains constant declarations, type declarations, and variable declarations and assignments. Header files are supplied with all programming interfaces.

High Performance File System (HPFS). An installable file system (IFS) designed to provide better performance than the existing file allocation table (FAT) based file system. HPFS is designed to provide extremely fast access to very large disk volumes.

hop count. The number of hosts through which a packet passes on its way to its destination.

host. A computer connected to a network, which provides an access method to that network. A host provides end-user services.

HPFS. High Performance File System

I

ICAT (Installation Configuration Automation Tool). TCP/IP for OS/2 provides this application for installing and configuring TCP/IP for OS/2.

ICMP. Internet Control Message Protocol.

IEEE. Institute of Electrical and Electronic Engineers.

include file. A file that contains preprocessor text, which is called by a program, using a standard programming call. Synonymous with *header file*.

installation. The process of placing one or more OS/2 components on a personal computer's fixed disk.

instance. One of the three parts of a Kerberos name. Instance specifies the machine on which a service is run.

Institute of Electrical and Electronic Engineers (IEEE). An electronics industry organization.

Integrated Services Digital Network (ISDN). A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

interactive. Pertaining to a program or a system that alternately accepts input and then responds. An interactive system is conversational, that is, a continuous dialog exists between user and system. See *batch*.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

internet or internetwork. A collection of packet switching networks interconnected by gateways, routers, bridges, and hosts to function as a single, coordinated, virtual network.

internet address. The unique 32-bit address identifying each node in an internet. See also *address*.

Internet Control Message Protocol (ICMP). The part of the Internet Protocol layer that handles error messages and control messages.

Internet Protocol (IP). The TCP/IP layer between the higher level host-to-host protocol and the local network protocols. IP uses local area network protocols to carry packets, in the form of datagrams, to the next gateway, router, or destination host.

interoperability. The capability of different hardware and software by different vendors to effectively communicate together.

IP. Internet Protocol.

ISDN. Integrated Services Digital Network.

ISO. International Organization for Standardization.

K

Kerberos Authentication System. An authentication mechanism used to check authorization at the user level.

L

LaMail. The client that communicates with the OS/2 Presentation Manager to manage mail on the network.

LAN. Local area network.

Line printer daemon (LPD). The remote printer server that allows other hosts to print on a printer local to your host.

Line Printer Protocol. A TCP/IP protocol used for printing files on printers attached to remote hosts.

local area network (LAN). A data network located on the user's premises in which serial transmission is used for direct data communication among data stations.

local host. In an internet, the computer to which a user's terminal is directly connected without using the internet.

local network. The portion of a network that is physically connected to the host without intermediate gateways or routers.

Logical ANDing. When the Boolean operator AND is applied to two bits, the result is one when both bits are one; otherwise, the result is zero. When two bytes are ANDed, each pair of bits is handled separately; there is no connection from one bit position to another.

LPD. Line printer daemon.

LPR. A client command that allows the local host to submit a file to be printed on a remote print server.

M

Management Information Base (MIB). A standard used to define SNMP objects, such as packet counts and routing tables, that are in a TCP/IP environment.

mapping. The process of relating internet addresses to physical addresses in the network.

MARK. A Presentation Manager function that marks a section of text to be copied or cut.

marshall. To copy data into an RPC packet. Stubs perform marshalling. See also *unmarshall*.

mask. (1) A pattern of characters used to control retention or elimination of portions of another pattern of characters. (2) To use a pattern of characters to control retention or elimination of another pattern of characters. (3) A pattern of characters that controls the keeping, deleting, or testing of portions of another pattern of characters.

menu. A type of panel that consists of one or more selection fields.

menu item. A selection item on a pull-down menu.

MIB. Management Information Base.

mouse. A device that is used to move a pointer on the screen and select items.

multitasking. A mode of operation that provides for the concurrent performance execution of two or more tasks.

MVS. Multiple Virtual Storage.

N

name server. The server that stores resource records about hosts.

NCP. Network Control Program.

NCS. Network Computing System.

network. An arrangement of nodes and connecting branches. Connections are made between data stations.

network adapter. A physical device, and its associated software, that enables a processor or controller to be connected to a network.

network administrator. The person responsible for the installation, management, control, and configuration of a network.

Network Computing System (NCS). Network Computing System. A set of software components developed by Apollo that conform to the Network Computing Architecture (NCA). NCS is made up of two parts: the nidl Compiler and Network Computing Kernel (NCK).

network control program (NCP). An IBM-licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

network elements. As defined in the SNMP architecture, network elements are gateways, routers, and hosts that contain management agents responsible for performing the network management functions requested by the network management stations.

network file system (NFS). The NFS protocol, which was developed by Sun Microsystems Incorporated, allows computers in a network to access each other's file systems. Once accessed, the file system appears to reside on the local host.

network management stations. As defined in the SNMP architecture, network management stations, or SNMP clients, execute management applications that monitor and control network elements.

NFS. Network file system.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (2) In a network topology, the point at an end of a branch.

O

octet. A byte composed of eight binary elements.

open system. A system with specified standards and that therefore can be readily connected to other systems that comply with the same standards.

Open Systems Interconnection (OSI). (1) The interconnection of open systems in accordance with specific ISO standards. (2) The use of standardized procedures to enable the interconnection of data processing systems.

OS/2. Operating System/2.

OSI. Open Systems Interconnection.

P

packet. A sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole.

parameter. A variable that is given a constant value for a specified application.

parse. To analyze the operands entered with a command.

passive open. The state of a connection that is prepared to provide a service on demand. Contrast with *active open*.

path. The course or route of drives and subdirectories leading from the root directory and drive of an operating system to where files or data information are stored.

PC. Personal computer.

PC Network. A low-cost broadband network that allows attached IBM personal computers, such as IBM 5150 Personal Computers, IBM Computer ATs, IBM PC/XTs, and IBM Portable Personal Computers to communicate and to share resources.

PDU. Protocol Data Units

peer-to-peer. In network architecture, any functional unit that resides in the same layer as another entity.

PING. The command that sends an ICMP Echo Request packet to a gateway, router, or host with the expectation of receiving a reply.

piping. in advanced DOS, a feature that allows the output of a program as it is displayed on the screen to be used as input to another program without reentering the data on the keyboard.

port. (1) An endpoint for communication between devices, generally referring to a logical connection. (2) A 16-bit number identifying a particular Transmission Control Protocol or User Datagram Protocol resource within a given TCP/IP node.

PORTMAP. Synonymous with *Portmapper*.

Portmapper. A program that maps client programs to the port numbers of server programs. Portmapper is used with Remote Procedure Call (RPC) programs.

Presentation Manager. A component of OS/2 that provides a complete graphics-based user interface, with pull-down windows, action bars, and layered menus.

principal name. One of the three parts of a Kerberos name. Principal name specifies the name of a user or service.

process. (1) A unique, finite course of events defined by its purpose or by its effect, achieved under defined conditions. (2) Any operation or combination of operations on data. (3) A function being performed or waiting to be performed. (4) A program in operation; for example, a daemon is a system process that is always running on the system.

protocol. A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication. Protocols can determine low-level details of machine-to-machine interfaces, such as the order in which bits from a byte are sent; they can also determine high-level exchanges between application programs, such as file transfer.

Protocol Data Unit (PDU). A set of commands used by the SNMP agent to request management station data.

protocol suite. A set of protocols that cooperate to handle the transmission tasks for a data communication system.

pull-down. An extension of the action bar that displays a list of choices that are available for a selected action bar choice.

R

RAM. Random Access Memory.

Random Access Memory (RAM). A memory device into which data is entered and from which data is retrieved in a nonsequential manner.

RARP. Reverse Address Resolution Protocol.

realm. One of the three parts of a Kerberos name. Realm specifies the service that provides authentication for the principal name. Realm can also specify the name of an administrative entry that is

responsible for its own database on its own Kerberos machine.

Remote Execution Protocol (REXEC). A protocol that allows the execution of a command or program on a foreign host. The local host receives the results of the command execution. This protocol uses the REXEC command.

remote host. Any *foreign host*, not including the local host.

remote logon. The process by which a terminal user establishes a terminal session with a remote host.

Remote Procedure Call (RPC). A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an eXternal data representation.

Request For Comments (RFC). A series of documents that covers a broad range of topics affecting internetwork communication. Some RFCs are established as internet standards.

resolver. A program or subroutine that obtains information from a name server or local table for use by the calling program.

resource records. Individual records of data used by the Domain Name System. Examples of resource records include the following: a host's Internet Protocol addresses, preferred mail addresses, and aliases.

return code. (1) A code used to influence the execution of succeeding instructions. (2) A value returned to a program to indicate the results of an operation requested by that program.

Reverse Address Resolution Protocol (RARP). A protocol that maintains a database of mappings between physical hardware addresses and IP addresses.

REXEC. Remote Execution Protocol.

RFC. Request For Comments.

RIP. Routing Information Protocol.

router. A device that connects networks at the ISO Network Layer. A router is protocol-dependent and connects only networks operating the same protocol. Routers do more than transmit data; they also select the best transmission paths and optimum sizes for packets. In TCP/IP, routers operate at the Internetwork layer. See also *gateway*.

Routing Information Protocol (RIP). The protocol that maintains routing table entries for gateways, routers, and hosts.

routing table. A list of network numbers and the information needed to route packets to each.

RPC. Remote Procedure Call.

S

Sendmail. The OS/2 mail server that uses Simple Mail Transfer Protocol to route mail from one host to another host on the network.

serial line. A network media that is a de facto standard, not an international standard, commonly used for point-to-point TCP/IP connections. Generally, a serial line consists of an RS-232 connection into a modem and over a telephone line.

server. A function that provides services for users. A machine can run client and server processes at the same time.

Simple Mail Transfer Protocol (SMTP). A TCP/IP application protocol used to transfer mail between users on different systems. SMTP specifies how mail systems interact and the format of control messages they use to transfer mail.

Simple Network Management Protocol (SNMP). A protocol that allows network management by elements, such as gateways, routers, and hosts. This protocol provides a means of communication between network elements regarding network resources.

SMI. Structure for Management Information.

SMTP. Simple Mail Transfer Protocol.

SNMP. Simple Network Management Protocol.

socket. (1) An endpoint for communication between processes or applications. (2) A pair consisting of TCP port and IP address, or UDP port and IP address.

socket interface. An application interface that allows users to write their own applications to supplement those supplied by TCP/IP.

stream. A continuous sequence of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

stubs. (1) A program module that transfers remote procedure calls and responses between a client and a server. Stubs perform marshalling, unmarshalling and data format conversion. Both clients and servers have stubs. The NIDL Compiler generates client and server stub code from an interface definition. (2) Hooking functions used as extensions to the protocol to generate protocol requests for X Window System.

subagent. In the SNMP architecture, a subagent provides an extension to the functionality provided by the SNMP agent.

subdirectory. A directory contained within another directory in a file system hierarchy.

subnet. A networking scheme that divides a single logical network into smaller physical networks to simplify routing.

subnet address. The portion of the host address that identifies a subnetwork.

subnet mask. A mask used in the IP protocol layer to separate the subnet address from the host portion of the address.

subnetwork. Synonymous with *subnet*.

synchronous. (1) Pertaining to two or more processes that depend on the occurrences of a specific event such as common timing signal. (2) Occurring with a regular or predictable time relationship. See *asynchronous*.

T

TALK. An interactive messaging system that sends messages between the local host and a foreign host.

task manager. The OS/2 function that controls the starting and stopping of programs, including shutting down the system.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

Telnet. The Terminal Emulation Protocol, a TCP/IP application protocol for remote connection service. Telnet allows a user at one site to gain access to a foreign host as if the user's terminal were connected directly to that foreign host.

TFTP. Trivial File Transfer Protocol.

ticket. Encrypted information obtained from a Kerberos authentication server or a ticket-granting server. A ticket authenticates a user and, in conjunction with an authenticator, serves as permission to access a service when presented by the authenticated user.

ticket-granting server. Grants Kerberos tickets to authenticated users as permission to access an end-service.

token. In a local network, the symbol of authority passed among data stations to indicate the station temporarily in control of the transmission medium.

token ring network. A ring network that allows unidirectional data transmission between data stations by a token-passing procedure over one transmission medium, so that the transmitted data returns to the transmitting station.

Transmission Control Protocol (TCP). The TCP/IP layer that provides reliable process-to-process data stream delivery between nodes in interconnected computer networks. TCP assumes that IP (Internet Protocol) is the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network.

TRAP. An unsolicited message that is sent by an SNMP agent to an SNMP network management station.

Trivial File Transfer Protocol (TFTP). A TCP/IP application primarily used to transfer files among personal computers. TFTP allows files to be sent and received, but does not provide any password protection or directory capability.

U

UDP. User Datagram Protocol.

unmarshall. To copy data from an RPC packet. Stubs perform unmarshalling. See also *marshall*.

user. A function that utilizes the services provided by a server. A host can be a user and a server at the same time. See *client*.

User Datagram Protocol (UDP). A packet-level protocol built directly on the IP layer. UDP is used for application to application programs between TCP/IP hosts.

V

VM. Virtual Machine.

W

WAN. Wide area network.

well-known port. A port number that has been preassigned for specific use by a specific protocol or application. Clients and servers using the same protocol communicate over the same well-known port.

wide area network (WAN). A network that provides communication services to a geographic area larger than that served by a local area network.

window. An area of the screen with visible boundaries through which a panel or portion of a panel is displayed.

working directory. The directory in which an application program is found. The working directory becomes the current directory when the application is started.

X

XDR. eXternal Data Representation.

Bibliography

TCP/IP for OS/2 Publications

The following paragraphs describe the other books associated with the TCP/IP for OS/2 library. If you want more information about IBM publications, ask your IBM representative, or write to the IBM branch office serving your location.

IBM TCP/IP Version 1.2 for OS/2: Installation and Maintenance, SC31-6075

This book provides system programmers, network administrators, and PC users responsible for installing TCP/IP for OS/2 with the information required to plan and implement the installation of TCP/IP for OS/2. The topics include hardware and software requirements, pre-installation system performance considerations, instructions for installing TCP/IP for OS/2, instructions for customizing the TCP/IP for OS/2 environment and installation examples.

IBM TCP/IP Version 1.2 for OS/2: Programmer's Reference, SC31-6077

This book is written for application and system programmers to aid in writing application programs that use TCP/IP for OS/2 on a PC. Application programmers should know the OS/2 operating system, and have knowledge of multitasking operating system concepts. Application programmers should be knowledgeable in the C programming language.

IBM TCP/IP Version 1.2 for OS/2: User's Guide, SC31-6076

This book is written for people who use a PC with TCP/IP for OS/2, such as end users and system programmers. The people who use this book should be familiar with OS/2 and the PC, and also understand the multitasking operating system concepts.

IBM TCP/IP Version 1.2 for OS/2: Quick Reference Guide, SX75-0070

This book is written as a quick reference guide for people who use a PC with TCP/IP for OS/2. It is intended as an adjunct to the *IBM TCP/IP Version 1.2 for OS/2: User's Guide* and *IBM TCP/IP Version 1.2 for OS/2: Installation and Maintenance*.

Other Related Publications

The following are other related publications.

OS/2 Publications

The following lists shows OS/2 publications.

IBM Operating System/2 User's Guide Volume 1: Base Operating System

IBM Operating System/2 User's Guide Volume 2: Communications Manager and LAN Requester

IBM Operating System/2 System Administrator's Guide for Communications

Operating System/2 Electronic Device Driver Distribution Mechanism

IBM Operating System/2 Command Reference

IBM OS/2 LAN Technical Reference, SC30-3383

TCP/IP Publications

The following list shows selected TCP/IP publications.

Introducing IBM's TCP/IP Products for OS/2, VM, and MVS, GC31-6080

This book introduces managers, system designers, programmers, and other data processing personnel to the basic concepts of IBM's TCP/IP products for OS/2, VM, and MVS. This book also describes the relationship between IBM's TCP/IP implementations and other IBM products, including those based on SNA.

Internetworking With TCP/IP Volume I: Principles, Protocols, and Architecture, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

Internetworking With TCP/IP Volume II: Implementation and Internals, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

IBM TCP/IP Tutorial and Technical Overview, GG24-3376

NetBIOS Version 1 for TCP/IP for OS/2: User's Guide, SC31-6122

This book provides information for using the IBM NetBIOS Version 1 for TCP/IP for OS/2 program. The Network Basic Input/Output System (NetBIOS) program provides a standard interface to the local area network for Operating Systems/2 applications using IBM's

Transport Control Protocol/Internet Protocol (TCP/IP) implementation for OS/2.

The Simple Book: An Introduction to Management of TCP/IP-based Internets, Marshall T. Rose, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

IBM Systems Application Architecture: SystemView and the OS/2 Environment. This document can be ordered from IBM with the number G01F-0281.

"Network Management and the Design of SNMP," J.D. Case, J.R. Davin, M.S. Fedor, M.L. Schoffstall, *ConneXions-The Interoperability Report*, Volume 3, No. 3, March 1989.

"Special Issue: Network Management and Network Security," *ConneXions-The Interoperability Report*, Volume 4, No. 8, August 1990.

"MIB II Extends SNMP Interoperability," C. Vandenberg, *Data Communications*, October 1990.

"Network Management of TCP/IP Networks: Present and Future," A. Ben-Artzi, A. Chandna, V. Warrier, *IEEE Network Magazine*, July 1990.

Programming Publications

The following list shows selected programming publications.

AIX Communications Concepts and Procedures for IBM RISC System/6000, SC23-2203

IBM AIX Operating System Technical Reference: System Calls and Subroutines, SC23-2125

Network Computing System (NCS) Reference (010200, Rev.00), Apollo Computer, Inc.

Networking on the Sun Workstation: Remote Procedure Call Programming Guide (800-1324-03), Sun Microsystems, Inc.

Network Programming (800-1779-10), Sun Microsystems, Inc.

UNIX Programmer's Reference Manual (4.3 Berkeley Software Distribution, Virtual VAX-11 Version). Department of Electrical Engineering and Computer Science. University of California, Berkeley, 1988.

X Protocol Reference Manual, Adrian Nye, ed. O'Reilly & Associates, Inc., 1990.

X Window System User's Guide, Valerie Quercia & Tim O'Reilly., O'Reilly & Associates, Inc., 1990.

The Art of Distributed Application: Programming Techniques for Remote Procedure Calls, John R. Corbin, Springer-Verlog, 1991.

Jennifer G. Steiner, Clifford Neuman, and Jeffrey I. Schiller. "Kerberos: An Authentication Service for Open Networks." Massachusetts Institute of Technology, 12 January 1988.

S.P. Miller et al. "Kerberos Authentication and Authorization System," Project Athena Technical Plan, Section E.2.1. Massachusetts Institute of Technology, 21 December 1987.

MVS NFS User's Guide, SC30-3383

MVS/DFP Version 3 Release 3: Using the Network File System Server, SC26-4732-0

Network Computing System (NCS) Publications

The following list shows NCS publications.

- *Network Computing System Reference Manual*, Mike Kong, (Terence H. Dineen, Paul J. Leach, Elizabeth A. Martin, Nathaniel W. Mishkin, Joseph N. Pato, Geoffrey L. Wyant), Apollo Computer Inc., a subsidiary of Hewlett-Packard Company, Chelmsford, Massachusetts, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1990.

- *Network Computing Architecture*, Lisa Zahn, (Terence H. Dineen, Paul J. Leach, Elizabeth A. Martin, Nathaniel W. Mishkin, Joseph N. Pato, Geoffrey L. Wyant), Apollo Computer Inc., a subsidiary of Hewlett-Packard Company, Chelmsford, Massachusetts, Prentice Hall, Englewood Cliffs, New Jersey 07632, 1990.

- *Network Computing Architecture (NCA) Protocol Specifications* , Apollo Computer Inc.,330 Billerica Road, Chelmsford, MA 01824, (508) 256-6600, 1989. Apollo Order No. 010201-A00

- *Network Computing System (NCS) Reference*, Apollo Computer Inc., 330 Billerica Road, Chelmsford, MA 01824, 1987. Apollo Order No. 010200, Revision 00.

- *Managing the NCS Location Broker*, Apollo Computer Inc., 330 Billerica Road, Chelmsford, MA 01824, 1988. Apollo Order No. 011895-A00.

Index

G

H

I

K

L

M

N

O

P

R

S

T

U

W

X

Readers' Comments

**IBM Transmission Control Protocol/
Internet Protocol Version 1.2 for OS/2:
Installation and Maintenance**

Publication No. SC31-6075-2

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply. If we have questions about your comment, may we call you? If so, please include your phone number.

Name

Address

Company or Organization

Phone No.

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department E15
PO BOX 12195
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-9990

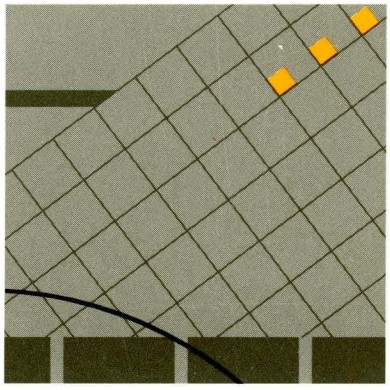
Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

File Number: S370/4300/30XX-50

Printed in USA

10G4229



SC31-6075-02

9010G42290001