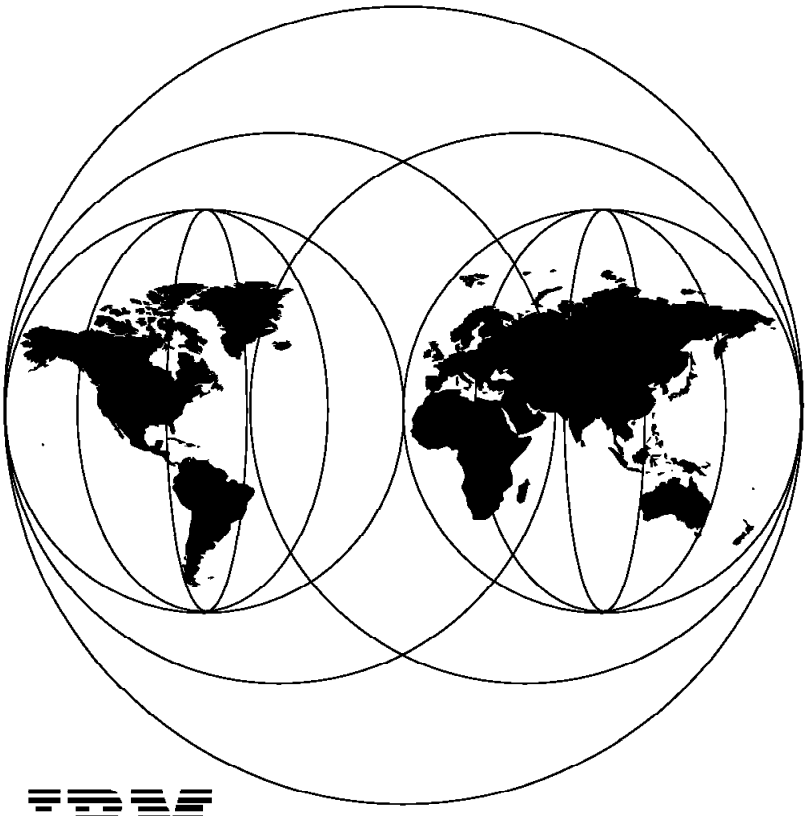
International Technical Support Organization SG24-4730-00

TCP/IP Implementation in an OS/2 Warp Environment

April 1996



IBM

International Technical Support Organization
Raleigh Center

International Technical Support Organization

SG24-4730-00

TCP/IP Implementation in an OS/2 Warp Environment

April 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xix.

First Edition (April 1996)

This edition applies to IBM TCP/IP Version 3.1 for OS/2 as implemented in IBM OS/2 Warp Server Version 4.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Abstract

This redbook is unique in its detailed coverage of TCP/IP for OS/2 and its implementation in the OS/2 Warp environment. It focuses on how TCP/IP for OS/2 can be used in an environment with various operating system platforms, communications media, and protocol stacks.

This redbook was written for the technical specialist who will evaluate and implement TCP/IP for OS/2. The reader is assumed to have a basic knowledge of the TCP/IP protocol suite and to be familiar with the OS/2 Warp environment. To help the less knowledgeable reader, we have included a chapter that covers the basics of TCP/IP.

It is not the intention of this document to replace the original product documentation, or to teach how to develop applications using the Programmer′s Tool Kit and the APIs that it provides.

(430 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help technical specialists to evaluate TCP/IP for OS/2 Warp and to implement its functions. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/2 Warp Server or TCP/IP for OS/2 Warp. See the PUBLICATIONS section of the IBM Programming Announcement for OS/2 Warp Server Version 4 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

AIX	AS/400
BookManager	Common User Access
CUA	Current

GDDM
IBM
Netcomber
OS/2
Presentation Manager
RACF
S/370
Ultimedia
VTAM
WIN-OS/2
Workplace Shell
400

Hummingbird
Library Reader
NetView
OS/400
PS/2
RS/6000
S/390
VisualAge
WebExplorer
Workplace
XGA

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other trademarks are trademarks of their respective companies.

Preface

This redbook describes TCP/IP for OS/2 and its implementation in the OS/2 Warp environment.

In addition, the book focuses on how TCP/IP for OS/2 can be used in an environment with various operating system platforms, communications media, and protocol stacks. It provides information on the installation, configuration and use of TCP/IP for OS/2.

This book is intended for the technical specialist audience who will evaluate the product possibilities and who will implement the product. The reader is assumed to have a basic knowledge of the TCP/IP protocol suite and be familiar with the OS/2 Warp environment. To help the less knowledgeable reader, we have included a chapter that covers the basics of TCP/IP.

It is not the intention of this redbook to replace the original product documentation, or to teach how to develop applications using the Programmer's Tool Kit and the APIs that it provides.

How This Redbook is Organized

The redbook is organized as follows:

- Chapter 1, "Introduction to TCP/IP"

  This chapter provides an introduction to the basics of TCP/IP. Among the areas covered are TCP, IP, ARP, routing and name resolution. did In addition a simple example of using TCP/IP is provided.

- Chapter 2, "Functional Overview of TCP/IP for OS/2"

  This chapter describes the functions and features of TCP/IP for OS/2 that are included in the base and the optional kits.

- Chapter 3, "CID Installation of TCP/IP for OS/2"

  This chapter explains how to perform a CID installation of TCP/IP V3.1 for OS/2. Also described are the CID installations of the PMX and NFS kits. In addition, the new CID keywords supporting dynamic IP in an unattended installation are explained.

- Chapter 4, "Double-Byte Character Set Support"

  This chapter provides some guidance in using DBCS versions of TCP/IP for OS/2 Warp in countries that use ideographic characters such as China, Japan and Korea.

- Chapter 5, "Dynamic IP"

  This chapter details the installation and setup of Dynamic IP in terms of its components Dynamic Host Configuration Protocol (DHCP) and Dynamic Domain Name Services (DDNS).

- Chapter 6, "Electronic Mail"

  This chapter describes the mail services in TCP/IP for OS/2, which are based on the Simple Mail Transfer Protocol (SMTP). The Multipurpose Internet Mail Extensions (MIME) to SMTP is also covered in this chapter. Examples of extending this environment to include Lotus Notes are also provided.

- Chapter 7, “Internet Applications”

  This chapter introduces applications that can be used to access resources on the Internet. Applications covered include NewsReader/2, WebExplorer and InterNotes.

- Chapter 8, “Remote Logon”

  This chapter describes the remote logon and terminal emulation facilities of TCP/IP for OS/2, its interoperability with the 3270 (VM, MVS), 5250 (AS/400), UNIX (AIX), DOS, and OEM TCP/IP services, and their integration into the OS/2 Workplace Shell.

- Chapter 9, “File Transfer”

  This chapter describes the file transfer facilities of TCP/IP for OS/2, its interoperability with the 3270 (VM, MVS), 5250 (AS/400), UNIX (AIX) and DOS TCP/IP services, and its integration into the OS/2 Workplace Shell.

- Chapter 10, “Remote Printing”

  This chapter describes the remote printing capabilities of TCP/IP for OS/2 using LPR, LPD, LPRPORTD, and their integration into the OS/2 Workplace Shell.

- Chapter 11, “Remote Execution of Commands”

  This chapter describes remote command execution using REXEC and RSH.

- Chapter 12, “DOS/Windows Access”

  This chapter describes the DOS/Windows Access interface of TCP/IP for OS/2.

- Chapter 13, “X Window System Server (PMX)”

  This chapter describes the X Window System Server function that is available with TCP/IP for OS/2.

- Chapter 14, “X Window System Client and OSF/Motif”

  This chapter describes the new X Window System Client and OSF Motif functions that are available with TCP/IP for OS/2.

- Chapter 15, “Remote File Systems”

  This chapter describes how to share and access remote file systems using the NFS functions provided with TCP/IP for OS/2, and shows interoperability with the VM, MVS, AIX, DOS, and OEM NFS services.

- Chapter 16, “Extended Networking”

  This chapter describes the extended networking capabilities of TCP/IP for OS/2 to connect to wide area networks using X.25 and SNALINK.

- Chapter 17, “Application Programming Interfaces”

  This chapter describes the application programming interfaces of TCP/IP for OS/2.

- Chapter 18, “Problem Determination”

  This chapter describes the problem determination facilities provided with TCP/IP for OS/2.

- Chapter 19, “Network Management”

  This chapter discusses the network management functions and tools available with TCP/IP for OS/2 including SNMP.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

TCP/IP for OS/2 product documentation:

- *IBM TCP/IP for OS/2: Network File System Guide*, SC31-7069
- *IBM TCP/IP for OS/2: X Window System Server Guide*, SC31-7070
- *IBM TCP/IP for OS/2: Extended Networking Guide*, SC31-7071

Documentations on other IBM TCP/IP products:

- *IBM TCP/IP V2.1.1 for DOS: User's Guide*, SC31-7045
- *IBM TCP/IP V2.1.1 for DOS: Programmer's Reference*, SC31-7046
- *IBM TCP/IP V2.1.1 for DOS: Command Reference*, SX75-0083
- *IBM TCP/IP for MVS: User's Guide*, SC31-7136
- *IBM TCP/IP for MVS: Customization and Administration Guide*, SC31-7134
- *IBM TCP/IP for MVS: Planning and Migration Guide*, SC31-7189
- *IBM TCP/IP for MVS: Programmer's Reference*, SC31-7135
- *IBM TCP/IP for MVS: Network Print Facility*, SC31-8074

Valuable TCP/IP documentations:

- *Stevens, W. Richard, TCP/IP Illustrated, Volume 1: The Protocols, ISBN 0-201-63346-9* IBM order no. SR28-5586-00
- *Stevens, W. Richard, TCP/IP Illustrated, Volume 2: The Implementation, ISBN 0-201-63354-X* IBM order no. SR28-5630-00

International Technical Support Organization Publications

- *TCP/IP Tutorial and Technical Overview, GG24-3376*
- *IBM TCP/IP Version 2 Release 2 for VM: Installation and Interoperability, GG24-3624*
- *IBM TCP/IP V3R1 for MVS Implementation Guide, GG24-3687*
- *TCP/IP for MVS, VM, OS/2, and DOS: Troubleshooting Guide, GG24-3852*
- *TCP/IP for MVS, VM, OS/2, and DOS: X Window System Guide, GG24-3911*
- *Using NFS in a Multivendor Environment, GG24-4087*
- *TCP/IP for DOS/Windows Interoperability and Coexistence, GG24-4374*
- *The Basics of IP Network Design, SG24-2580*
- *Accessing the Internet, SG24-2597*

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

How Customers Can Get Redbooks and Other ITSO Deliverables

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **IBMLINK**

  Registered customers have access to PUBORDER to order hardcopy, to REDPRINT to obtain BookManager BOOKs

- **IBM Bookshop** — send orders to:

  usib6fpl@ibmmail.com (United States)
  bookshop@dk.ibm.com (Outside United States)

- **Telephone orders**

1-800-879-2755	Toll free, United States only
(45) 4810-1500	Long-distance charge to Denmark, answered in English
(45) 4810-1200	Long-distance charge to Denmark, answered in French
(45) 4810-1000	Long-distance charge to Denmark, answered in German
(45) 4810-1600	Long-distance charge to Denmark, answered in Italian
(45) 4810-1100	Long-distance charge to Denmark, answered in Spanish

- **Mail Orders** — send orders to:

IBM Publications	IBM Direct Services
P.O. Box 9046	Sortemosevej 21
Boulder, CO 80301-9191	DK-3450 Allerød
USA	Denmark

- **Fax** — send orders to:

1-800-445-9269	Toll-free, United States only
45-4814-2207	Long distance to Denmark

- **1-800-IBM-4FAX (United States only)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services**

  Send note to softwareshop@vnet.ibm.com

- **Redbooks Home Page on the World Wide Web**

  http://www.redbooks.ibm.com/redbooks

- **E-mail (Internet)**

  Send note to redbook@vnet.ibm.com

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How IBM Employees Can Get Redbooks and ITSO Deliverables

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet**

  Type GOPHER
  Select IBM GOPHER SERVERS
  Select ITSO GOPHER SERVER for Redbooks

- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET GG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET GG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  `http://w3.itso.ibm.com/redbooks/redbooks.html`

  IBM employees may obtain LIST3820s of redbooks from this page.

- **ITSO4USA category on INEWS**
- **IBM Bookshop** — send orders to:

  USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to `announce@webster.ibmlink.ibm.com` with the keyword `subscribe` in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

Acknowledgments

This project was designed and managed by:

Eamon Murphy
International Technical Support Organization, Raleigh Center

The authors of this redbook are:

Walter Grode
IBM Germany

S.H. Lee
IBM Taiwan

Klaus Wichmann
IBM Germany

This publication is the result of a residency conducted at the International Technical Support Organization, Raleigh Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this redbook:

Martin Murhammer
IBM Austria

Alfred Christensen
David Boone
International Technical Support Organization, Raleigh Center

Chapter 1. Introduction to TCP/IP

In this chapter we cover the basic TCP/IP and Internet protocols.

The Internet network protocols have been based on different hardware and software packages communicating with one another. The Internet has grown out of networking based on the TCP/IP protocol. We cover some of the basics of the TCP/IP protocol.

1.1 TCP/IP Basics

TCP/IP is a layered networking protocol. The whole TCP/IP suite is comprised of a combination of protocols at different layers, as shown in the following four-layer reference model.

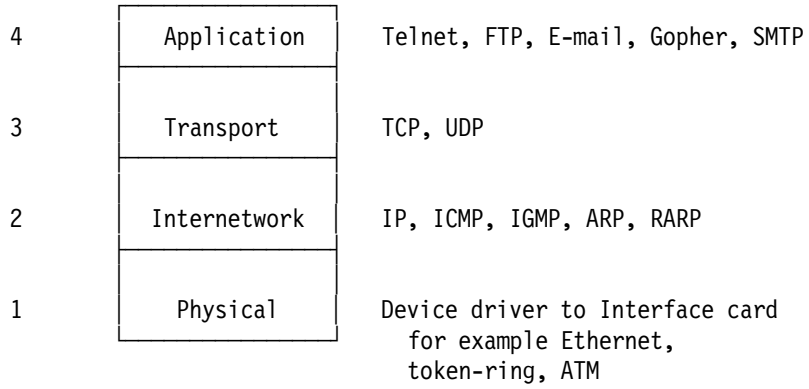


Figure 1. Four-Layer Network Model

Each layer has a different network responsibility. If you have two hosts on a network, each host layer will communicate on a logical basis with its equivalent layer on the other host, as indicated in Figure 2.

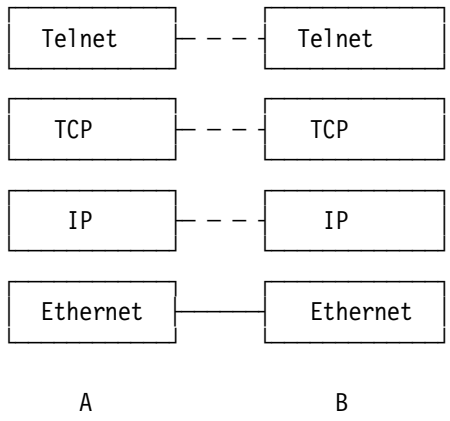


Figure 2. Inter-Layer Communication Model

For example, the IP internetwork layer on host A will communicate logically with the IP internetwork layer on host B across the Internet. The physical communication of data occurs on the physical link layer.

Although the commonly used name for the entire suite of protocols is TCP/IP, TCP and IP are only two of many protocols used on the TCP/IP Internet implementation. Some applications do not have to use all four layers to communicate. For example, the ping command is based on the ICMP protocol (level 2), and only uses levels 1 and 2 to communicate with another host.

The following sections cover the major components of each layer in more detail.

1.1.1 IP Protocol

The Internet is comprised of both physical wire and software connections. When you try to imagine what the Internet is and how it operates, it is natural to think of a chaotic unmanaged network. How does a single network request know where to go? This is where an Internet address or IP address is used. The IP address is based on a hexadecimal numbering system. The clever part of IP addressing is that the numbers are chosen to make the network and routing more efficient. Specifically, an IP address encodes the identification of the network to which an end user is attached within the IP address specified, all at the IP network layer.

Every interface on the Internet must have a unique IP address. This book does not go into the complexities involved in designing an IP network. However, to be able to understand the domain concept that will be introduced later, some of the basics of IP addressing need to be understood. Each host attached to the Internet has an assigned unique 32-bit universal identifier, or IP address. Conceptually, each IP address is made up of a pair of numbers: network ID (netid) and host ID (hostid). In practice, this pairing can take one of five classes, as follows:

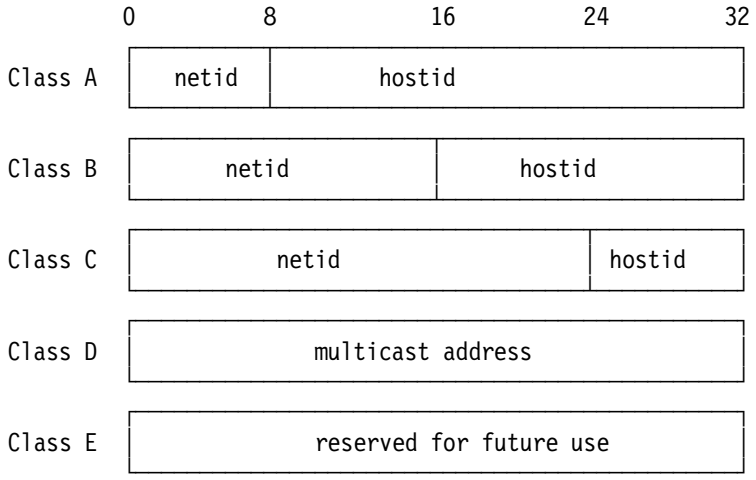


Figure 3. IP Address Classes

Each network class will allow different network and host possibility combinations, as shown in Table 1 on page 3.

Table 1. Class versus Network and Hosts		
Class	**Number of Network**	**Number of Hosts**
A	less than 256	greater than 65536
B	256 to 65536	256 to 65536
C	greater than 65536	less than 256

For ease of communicating, IP addresses are written as four decimal integers separated by decimal points, where each integer is given the value of one octet of the IP address. Thus a 32-bit address is written as xx.xx.xx.xx. For example:

The binary network address

10000000 00001010 00000010 00011110

is written
128 10 2 30

or 128.10.2.30

The classes are distinguished by the high-order bits. Given the addresses, one can distinguish the three primary classes as shown in Table 2.

Table 2. Class versus Dotted Decimal Notation	
Class	**Dotted Decimal Range**
A	0.0.0.0 to 127.255.255.255
B	128.0.0.0 to 191.255.255.255
C	192.0.0.0 to 223.255.255.255
D	224.0.0.0 to 239.255.255.255
E	240.0.0.0 to 247.255.255.255

Since every host on the Internet must have a unique IP address, there must be some central authority for allocating these addresses for networks and hosts. This authority is the Internet Network Information Center (InterNIC). InterNIC is responsible for network and domain registration. End users do not get their IP address from InterNIC. InterNIC normally assigns a range of IP addresses to service providers. To get an IP address, you must approach your service provider who, depending on your connection type, will assign you an IP number from a range of IP addresses they have been allotted. If you do not want to connect through a service provider, and intend to connect to the Internet directly, you must apply to InterNIC for a domain address and an IP network ID. To apply directly to the InterNIC, you must be either a service provider or a very large global corporation. The assignment of host IDs is then up to the system administrator on your site. InterNIC does not readily provide a direct service, and will, in almost every case, redirect queries through to a service provider. Some service providers operate at a regional level, and are responsible for a wider range of top-level IP addresses. This is covered in more detail in RFC 1466.

The current practice is to assign globally unique addresses to all hosts that use TCP/IP. There is a growing concern that the finite IP address space might become exhausted. Therefore, the guidelines for assigning IP address space have been tightened in recent years. These rules are often more conservative than enterprises would like, in order to implement and operate their networks.

Hosts within enterprises that use IP can be partitioned into the following three categories:

- Hosts that do not require access to hosts in other enterprises or the Internet at large
- Hosts that need access to a limited set of outside services (for example, E-mail, FTP, USENET news, remote login) that can be handled by application-layer gateways and firewalls
- Hosts that need network layer access outside the enterprise (provided via IP connectivity)

For many hosts in the second category, an unrestricted external access (provided via IP connectivity) may be unnecessary and even undesirable for privacy/security reasons. Just like hosts within the first category, such hosts may use IP addresses that are unambiguous within an enterprise, but may be ambiguous between enterprises. Only hosts in the last category require IP addresses that are globally unambiguous.

Many applications require connectivity only within one enterprise and do not even need external connectivity for the majority of internal hosts. In larger enterprises, it is often easy to identify a substantial number of hosts using TCP/IP that do not need network layer connectivity outside the enterprise.

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private networks:

```
10.0.0.0        –   10.255.255.255
172.16.0.0      –   172.31.255.255
192.168.0.0     –   192.168.255.255
```

An enterprise that decides to use IP addresses out of the address space defined in this document can do so without any coordination with IANA or an Internet registry. The address space can thus be used by many enterprises. Addresses within this private address space will only be unique within the enterprise.

As before, any enterprise that needs globally unique address space is required to obtain such addresses from an Internet registry. An enterprise that requests IP addresses for its external connectivity will never be assigned addresses from the blocks defined above.

In order to use private address space, an enterprise needs to determine which hosts do not need to have network layer connectivity outside the enterprise in the foreseeable future. Such hosts will be called private hosts, and will use the private address space defined above. Private hosts can communicate with all other hosts inside the enterprise, both public and private. However, they cannot have IP connectivity to any external host, while not having external network layer

connectivity private hosts can still have access to external services via application layer relays.

All other hosts will be called public and will use globally unique address space assigned by an Internet registry. Public hosts can communicate with other hosts inside the enterprise, both public and private, and can have IP connectivity to external public hosts. Public hosts do not have connectivity to private hosts of other enterprises.

Moving a host from private network to public network or vice versa will likely involve a change of IP address.

Because private addresses have no global meaning, routing information about private networks should not be propagated on inter-enterprise links, and packets with private source or destination addresses should not be forwarded across such links. Routers in networks not using private address space, especially those of Internet service providers, are expected to be configured to reject (filter out) routing information about private networks. If such a router receives such information, the rejection shall not be treated as a routing protocol error.

Indirect references to such addresses should be contained within the enterprise. Prominent examples of such references are DNS Resource Records and other information referring to internal private addresses. In particular, Internet service providers should take measures to prevent such leakage.

The obvious advantage of using private address space for the Internet at large is to conserve the globally unique address space by not using it where global uniqueness is not required.

Enterprises themselves also enjoy a number of benefits from their usage of private address space. They gain a lot of flexibility in network design by having more address space at their disposal than they could obtain from the globally unique pool. This enables operationally and administratively convenient addressing schemes as well as easier growth paths.

For a variety of reasons the Internet has already encountered situations where an enterprise that has not been connected to the Internet had used IP address space for its hosts without getting this space assigned from the IANA. In some cases this address space had been already assigned to other enterprises. When such an enterprise later connects to the Internet, it could potentially create very serious problems, as IP routing cannot provide correct operations in the presence of ambiguous addressing. Using private address space provides a safe choice for such enterprises, avoiding clashes once outside connectivity is needed.

One could argue that the potential need for renumbering represents a significant drawback of using the addresses out of the block allocated for private networks. However, we need to observe that the need is only potential, since many hosts may never move into the third category, and an enterprise may never decide to interconnect (at the IP level) with another enterprise.

But even if renumbering has to happen, we have to observe that with Classless Inter-Domain Routing (CIDR) an enterprise that is connected to the Internet may be encouraged to renumber its public hosts, as it changes its network service provider. Thus, renumbering is likely to happen more often in the future, regardless of whether an enterprise does or does not use the addresses out of

the block allocated for private networks. Tools to facilitate renumbering (DHCP for example) would certainly make it less of a concern.

Also, observe that the clear division of public and private hosts and the resulting need to renumber makes uncontrolled outside connectivity more difficult, so to some extent the need to renumber could be viewed as an advantage.

A recommended strategy is to design the private part of the network first and use private address space for all internal links. Then plan public subnets at the locations needed and design the external connectivity.

This design is not fixed permanently. If a number of hosts need to change status later this can be accomplished by renumbering only the hosts involved and installing another physical subnet if required.

If a suitable subnetting scheme can be designed and is supported by the equipment concerned, it is advisable to use the 24-bit block of private address space and make an addressing plan with a good growth path. If subnetting is a problem, the 16-bit class C block, which consists of 255 contiguous class C network numbers, can be used.

Using multiple IP (sub)nets on the same physical medium has many pitfalls. We recommend avoiding it unless the operational problems are well understood and it is proven that all equipment supports this properly.

Moving a single host between private and public status will involve a change of address and in most cases physical connectivity. In locations where such changes can be foreseen (machine rooms, etc.) it may be advisable to configure separate physical media for public and private subnets to facilitate such changes.

Changing the status of all hosts on a whole (sub)network can be done easily and without disruption for the enterprise network as a whole. Consequently it is advisable to group hosts whose connectivity needs might undergo similar changes in the future on their own subnets.

It is strongly recommended that routers which connect enterprises to external networks are set up with appropriate packet and routing filters at both ends of the link in order to prevent packet and routing information leakage. An enterprise should also filter any private networks from inbound routing information in order to protect itself from ambiguous routing situations which can occur if routes to the private address space point outside the enterprise.

Groups of organizations that foresee a big need for mutual communication can consider forming an enterprise by designing a common addressing plan supported by the necessary organizational arrangements, such as a registry.

If two sites of the same enterprise need to be connected using an external service provider, they can consider using an IP tunnel to prevent packet leaks form the private network.

A possible approach to avoid leaking of DNS is to run two name servers, one external server authoritative for all globally unique IP addresses of the enterprise and one internal name server authoritative for all IP addresses of the enterprise, both public and private. In order to ensure consistency, both these

servers should be configured from the same data of which the external name server only receives a filtered version.

The resolvers on all internal hosts, both public and private, query only the internal name server. The external server resolves queries from resolvers outside the enterprise and is linked into the global DNS. The internal server forwards all queries for information outside the enterprise to the external name server so all internal hosts can access the global DNS. This ensures that information about private hosts does not reach resolvers and name servers outside the enterprise.

While using private address space can improve security, it is not a substitute for dedicated security measures. This is covered in more detail in RFC 1597.

The IP address information is used for every data packet transmitted. When sending data on the Internet, the data is not transmitted as one continuous stream of data, but rather as a collection of packets. Each packet is sent, and when an acknowledgement from the destination is received, the next packet is sent. Together with the acknowledgement, a checksum size is sent with the packet. If the datagram size of the acknowledgement message (TCP layer) is not the same, the datagram packet is resent. The datagram will also be resent if no acknowledgement is received (TCP layer). The format of each IP datagram header is indicated in Figure 4 on page 8.

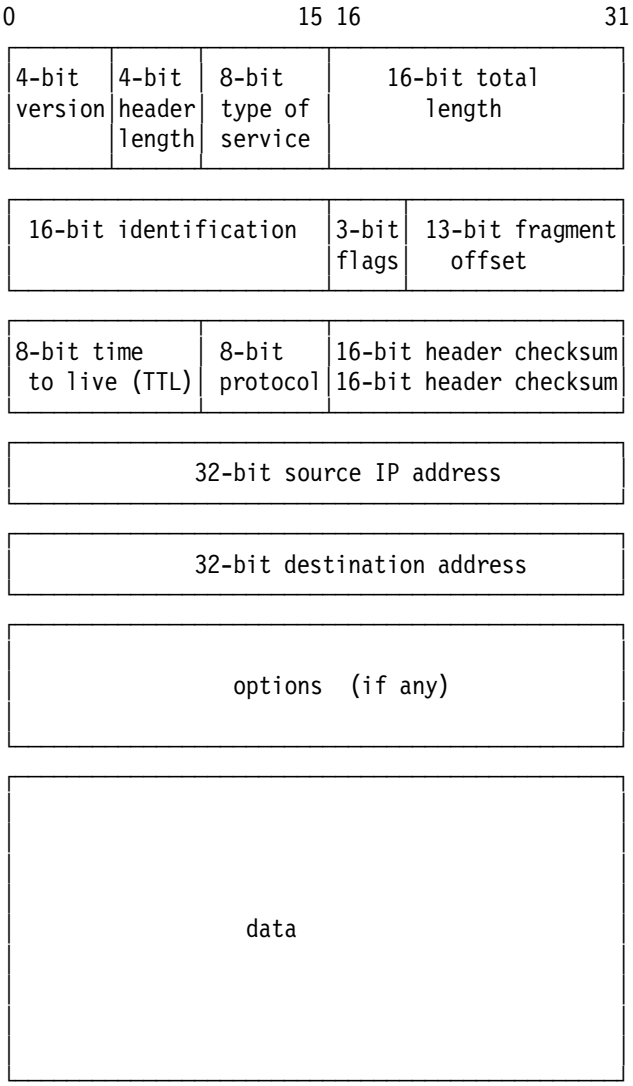


Figure 4. IP Header. IP header encapsulation includes both IP source and destination as well as the checksum data.

A general user might now be asking, what does IP addressing have to do with me? IP addressing is very important when connecting to a service provider. As you have seen, one cannot simply go and choose one's own IP address at random since it may well belong to someone else already and will most likely not work within the network class into which you are connecting. If it does work, you will most likely cause someone on the Internet to have an IP address that no longer works.

Depending on the protocol used, many service providers use dynamic IP addressing when allocating an IP address to a SLIP or PPP user. To do this the user will set up his initial IP address as either 1.1.1.1 or 0.0.0.0 (depending on the software used). When logging into a service provider with the initial IP address, you will be assigned a dynamic IP address out of a library of available IP addresses. This will most likely be different every time you sign on to the system. Your service provider will have been allocated an IP range which can be allocated to users when they sign on to the system. These IP addresses are

released when the user signs off, and can be allocated to new users signing on to the system.

As a result of the phenomenal growth of the Internet over the last few years, the following problems exist with IP:

- Most class B addresses have already been allocated. Although efforts have been made to conserve addresses, it is still likely that the class B address space will be exhausted in the early part of the next century.
- 32-bit IP addresses, in general, are inadequate for the predicted long-term growth of the Internet.

1.1.2 Subnets

This section discusses subnets of Internet networks, which are logically visible subsections of a single Internet network. For administrative or technical reasons, many organizations have chosen to divide one Internet network into several subnets, instead of acquiring a set of Internet network numbers. Background information for a subnetting standard are provided in RFC 940 and RFC 950.

The original view of the Internet universe was a two-level hierarchy: the top level the Internet as a whole, and the level below it individual networks, each with its own network number. The Internet does not have a hierarchical topology; rather, the interpretation of addresses is hierarchical. In this two-level model, each host sees its network as a single entity; that is, the network may be treated as a black box to which a set of hosts is connected.

While this view has proved simple and powerful, a number of organizations have found it inadequate and have added a third level to the interpretation of Internet addresses. In this view, a given Internet network is divided into a collection of subnets. Instead of considering an IP address as just a network ID and host ID, as mentioned earlier, the host portion can be divided into a subnet ID and a host ID. This makes sense because class A and class B addresses have a large number of bits allocated for the host ID (see Table 2 on page 3).

The three-level model is useful in networks belonging to moderately large organizations where it is often necessary to use more than one physical LAN to cover a local area. Each LAN may then be treated as a subnet.

The following are several reasons why an organization might use more than one physical LAN:

- Different technologies: There may be more than one kind of LAN in use. For example, an organization may have some equipment that supports Ethernet, and some that supports a token-ring network.
- Limits of technologies: Most LAN technologies impose limits, based on electrical parameters, on the number of hosts connected and on the total length of the cable. It is easy to exceed these limits, especially those on cable length.
- Network congestion: It is possible for a small subset of the hosts on a LAN to monopolize most of the bandwidth. A common solution to this problem is to divide the hosts into cliques of high mutual communication, and put these cliques on separate cables.

- Point-to-point links: Sometimes a local area is split into two locations too far apart to connect using the preferred LAN technology. In this case, high-speed point-to-point links might connect several LANs.

Normally, one does not attach many hosts to a single network. After obtaining the IP network ID, it is up to the systems administrator to decide how many bits to allocate to the subnet ID and host ID. For example, if we were dealing with a class B address with 16 bits allocated to the network ID and 16 bits allocated to the host ID, we could split the host ID into 8 subnet ID bits and 8 host ID bits. This division will allow 256-2 subnets and 256-2 hosts per subnet (we subtract 2 from 256 because host IDs of all zero or all one bits are invalid).

A host needs to know how many bits are to be used for the subnet ID and how many bits for the host ID. This is specified in a subnet mask. This mask is a 32-bit value containing network ID and subnet ID and zero bits for the host ID.

For example, a class B network ID of 141.191 can be subnetted as shown in Figure 5.

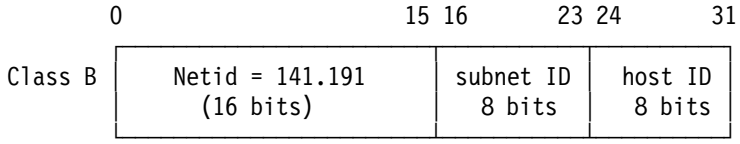


Figure 5. Subnetted Class B Address

Given its own IP address and subnet mask, a host can determine if an IP datagram is destined for the following:

1. A host on its own subnet
2. A host on a different subnet
3. A host on a different network

1.1.3 TCP Transmission Control Protocol

The IP protocol layer provides an unreliable service. That is, it does its best at getting a packet to its final source, but with no guarantees. TCP, on the other hand, provides a reliable transport layer using the unreliable IP layer. To perform this service, TCP performs timeout and retransmission with end-to-end message acknowledgements. Using both the IP layer and the transport layer, an end-user message will be sent through the network with the confidence that it will reach its destination. TCP adopts the role of a datagram manager. Each datagram is managed until acknowledgement of a successful data transmission has been received.

The following pertinent Internet and TCP/IP applications use the TCP protocol:

- Telnet
- FTP
- Finger
- NNTP
- POP
- Whois
- SMTP
- WWW
- Gopher

UDP is an alternative to TCP for applications that prefer to use the datagram service directly, and which do not require the TCP datagram management facilities.

The following TCP/IP and Internet applications use the UDP protocol:

- SNMP
- TFTP
- Route
- Domain
- Name Server

Some client/server applications, such as DNS, consist of very simple exchanges: one query followed by one reply. Managing these connections as TCP would do is wasteful, so the direct approach of UDP is offered. SNMP uses UDP so that it can manage the network depending on the routing conditions it detects, rather than relying on TCP, which would not reroute a datagram if an error were detected.

TCP relies on the final destination to send data acknowledgement and not individual routers, so if data is not received by the final destination, the data has to be resent over the entire route again. This is different from some other protocols, such as IBM SNA, which handles acknowledgement at each router node of the transfer, and only has to resend data from node to node, and not over the entire network. An advantage that TCP has over SNA is that it need not be concerned about the nature of the intermediate routers in the end-to-end connection.

1.1.4 Routers

Now that you understand that you can have numerous different networks with different IP address ranges, you might ask, how do these different networks communicate with one another? The easiest way is to connect the networks together with a router. Routers come in different forms, covering different protocols, including IP. The most common form of router you will encounter on the Internet is the IP router. IP routers will for the rest of this document be referred to simply as routers. Historically, routers have also been called gateways. A gateway connects different protocols together, for example, a TCP/IP and SNA gateway.

When sending an IP datagram, the IP layer will search an ARP table for an entry that matches a datagram′s complete destination address. If found, IP will send the datagram to the indicated interface. If not found, IP sends the datagram to the router function, which will handle the next hop in the route. Using this methodology, the IP layer will attempt to deliver the datagram packet to its final destination, as shown in Figure 6 on page 12.

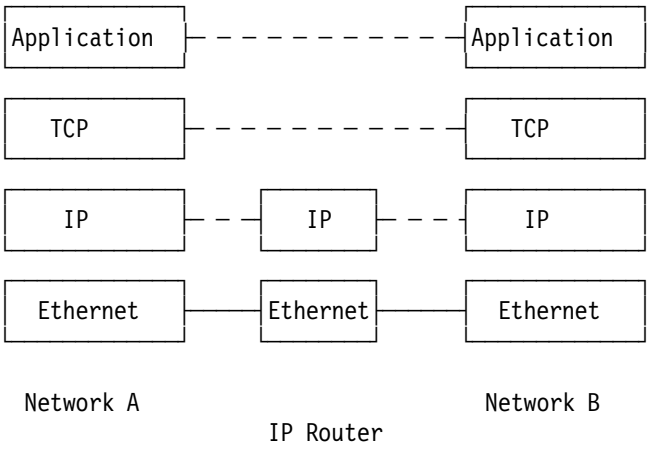


Figure 6. Internet Routing

Two interchangeable routing methodologies exist, static and dynamic routing. In static routing, the route to any IP address is fixed and pre-mapped in routing tables. If one of the predefined routes were to fail, the whole route would fail. The advantage of static routing is that the routing of traffic is controllable, although the actual route itself might not be the most efficient. Dynamic routing involves dynamically determining the shortest route to a host each time routing is required. If one of the routes had collapsed, dynamic routing would always determine another route to that host if one were available. The shortest route is not necessarily the fastest route. The shortest route might be based on the least number of hops from router to router. As will be discussed later, this may not take into account the bandwidth of the line, as shown in the Figure 7.

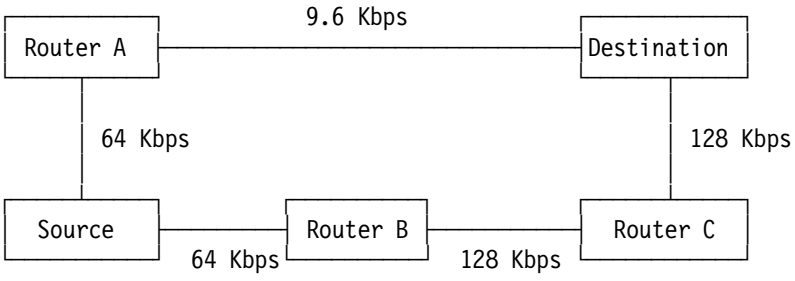


Figure 7. Dynamic Routing. If the hop count were used to determine the best route, the best dynamic route between source and destination would be through Router A. However, this is not the fastest route, which would be through Router B and Router C.

To understand routing on the Internet, one has to understand the concept of interior and exterior routing. The Internet is comprised of a collection of smaller autonomous networks. Each of these individual smaller networks would be viewed from the inside the network as an autonomous network. Such a network would use interior routers. A collection of routers with the same interior gateway protocol (IGP) number are said to belong to one IGP system. The Internet, as a network of networks, requires routers which would function as routers outside the IGP networks, using an exterior routing protocol. One example is exterior gateway protocol (EGP) routers. Exterior routers require an autonomous system

number. This autonomous system number is assigned by a regional service provider. The EGP protocol is being replaced by the border gateway protocol (BGP). BGP Version 3 is defined in RFC 1267. BGP is not widely implemented yet.

Software dynamic routing protocols are implemented using one of the following mutually exclusive daemon types:

- **RouteD** is a basic interior routing daemon supplied with the majority of TCP/IP applications. RouteD uses a RIP routing protocol. It is an IGP protocol only.
- **GateD** is a more sophisticated daemon used for both interior and exterior routing. It can support multiple routing protocols.

The following are most widely implemented IGP routing protocols:

- RIP

  This is a simple routing protocol, based on the use of a vector distance algorithm or, to put it simply, a hop count. It determines the most efficient route based on the minimum number of hops to a destination, even if this is not the fastest transmission route. The routing table is updated every 30 seconds. The official specification for RIP is RFC 1058. There are a large number of IGP routers on the Internet based on the RIP technology. Each passage through a router is termed a hop. RIP allows for a maximum of 15 hops and is not suitable for large exterior networks that compromise the whole Internet. If a hop count is greater than 15 before the destination can be found, RIP considers the network unreachable and flags the route as such. Two versions of RIP exist, RIP V1 and RIP V2. RIP V1 is supported by IBM TCP/IP for OS/2, AIX V3.2 and V4.1, and IBM TCPIP for DOS V2.1.1. RIP is implemented in both the RouteD and GateD daemons.

- OSPF

  This is a more complex routing methodology that determines the quickest route based on the actual transmission time of IP packets. The calculation used to determine the quickest route is based upon the link state algorithm. Being more complex than RIP, it places a higher workload on both the router and the network bandwidth. OSPF is a TCP/IP routing protocol. IBM supports OSPF in its dedicated routers, such as the 6611. AIX V4.1 supports OSPF using the GateD daemon. OSPF is not supported by TCP/IP for OS/2 and TCP/IP for DOS. OSPF Version 2 is described in more detail in RFC 1247.

- Hello

  This was used in LSI/11 microcomputers, which were widely used in Internet experimentation. It is not an Internet standard, but was used by the NSFNET as their second backbone IGP. IBM supports Hello in its dedicated routers, such as the IBM 6611. AIX V4.1 supports Hello using the GateD daemon. Hello has very little usage outside NSFNET. NSFNET has replaced the Hello protocol and has upgraded to a T-1 backbone using the IS-IS protocol.

- IS-IS

  This is an intermediate system-to-intermediate system protocol similar to OSPF. It uses a link state algorithm to determine the optimum route. Unlike OSPF, IS-IS is an OSI (open systems interconnection) protocol. As with OSPF, it is better suited to larger networks with a large number of routers. It is not widely implemented yet. Unlike OSPF, IS-IS allows mixing of the OSI and IP protocols.

Based on experiences gained from the EGP protocol, the border gateway protocol (BGP), was developed. Unlike IGP and EGP, which are connectionless protocols dependent on origin-destination communication for error detection, BGP protocols are connection-oriented so that one would only have to resend a datagram from BGP to BGP in the case of a datagram error. As such, BGP routers operate at the TCP layer. BGP treats the Internet as a single set of autonomous systems arbitrarily connected together.

ARPAnet is based on EGP and IGP routing. Unlike ARPAnet, which requires routers to be connected to the ARPAnet core system, NSFNET′s T-3 backbone usage of BGP and IGP removes NSFNET from acting as a core routing system.

The BGP protocol is implemented in the GateD daemon. This is supported by AIX V3 and V4. The IBM 6611 also implements BGP.

1.1.5 ARP

Address resolution protocol is used to dynamically map IP addresses to physical hardware addresses on an IEEE 802 network. IEEE network cards have a unique physical hardware address. These physical addresses are mapped to the IP address using this address resolution protocol. IEEE 802 networks cover Ethernet and token-ring network types. ARP is not used in serial-line connections. More details on ARP are included in RFC 826. If an application sends out data to a certain IP address, the IP routing mechanism first determines the IP address of the next hop of the packet to be used. It then needs to map the known IP address to a physical hardware address. IP first checks a local ARP cache to see if it can map the required IP address to any known physical connections. If it can, it sends the IP packet to the hardware address specified in the ARP table. If it cannot map the IP address to a hardware address, ARP sends out a network broadcast request of an ARP request. Each host receiving the ARP broadcast request then returns its IP and physical hardware address to the broadcaster. The application that requested the IP information then updates its local cache for later use and sends out the now mapped IP packet.

Another concept used in determining hardware address mapping is Proxy ARP. In the case of the single LAN above, every host on the LAN can see every other host on the LAN; a simple ARP broadcast will yield all the hardware addresses on this LAN. If, on the other hand, a router is used to connect two LANs together, sending out an ARP broadcast will only return the physical addresses of those on one of the LANs. To get the hardware address of a LAN card on the other LAN, Proxy ARP is used, where the router can see both LANs, and can send a proxy ARP broadcast to the other LAN. The router can then return a proxy ARP hardware address the LAN card on the other LAN. Proxy ARP is covered in more detail in RFC 1027.

1.1.6 Domain Name System

Although the network interfaces on a host, and therefore the host itself is known by IP addresses, humans tend to be more comfortable when using the name of a host. In the TCP/IP world the *domain name system* (DNS) is a distributed database system that provides the mapping between IP addresses and hostnames. We use the term *distributed* because no single site on the Internet knows all the information. Each site will maintain its own database and runs a database or name server that other systems across the Internet can query. The DNS provides a protocol that allows clients and servers to communicate with each other.

The DNS name space is hierarchical, as shown in Figure 8 on page 15.

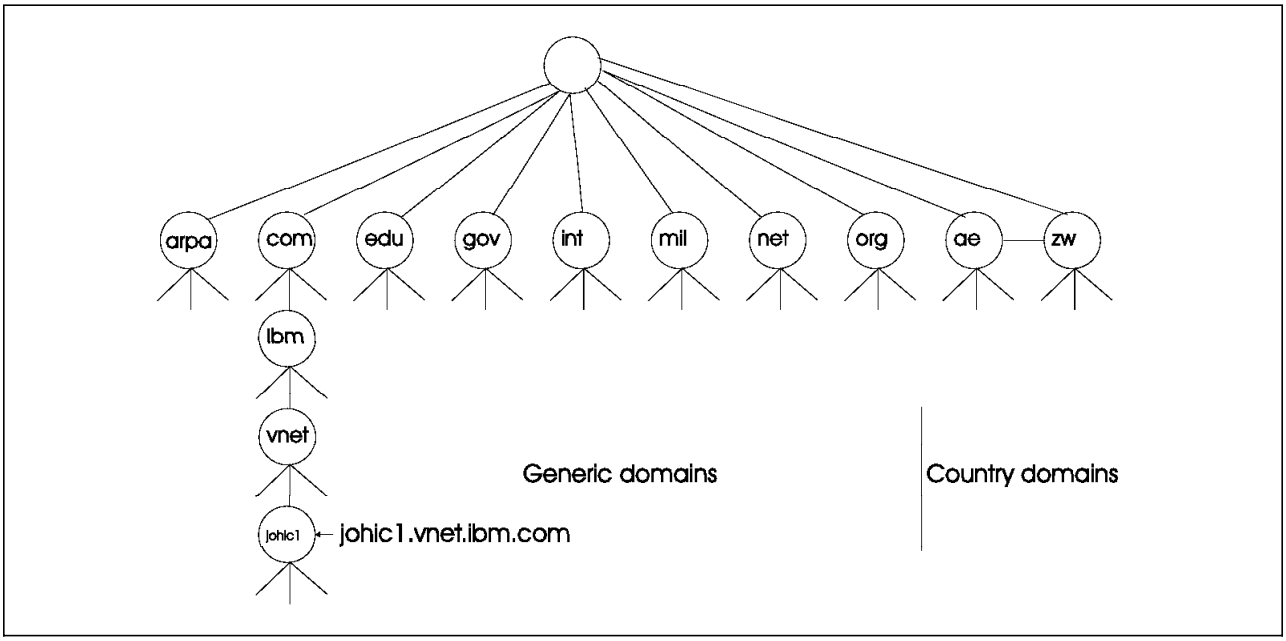


Figure 8. Domain Name Space Hierarchy

The top-level domains are divided into the following three categories:

1. ARPA, which is a special domain used for address-to-name mappings
2. Seven three-character domains, called the generic domains
3. Two-character country domain codes

Table 3 lists the normal classification of the seven generic domains.

Table 3. The Three-Character Generic Domains

Domain	Description
com	Commercial Organizations
edu	Educational Institutions
gov	Other U.S. Governmental Organizations
int	International Organizations
mil	U.S. Military
net	Network Provider
org	Other Organizations

Many countries form second-level domains beneath their two-character country code similar to the top-level generic codes. An example of such a combination of country and generic domains is .co.za where a top-level country code of za is used and a commercial organization of .co is used. The InterNIC maintains the top level domains, and delegates responsibility to others for third and fourth sublevels.

Using this type of domain naming, we might end up with a domain name of:

```
johic1.vnet.ibm.com
```

where:

- `johic1` is the system name and fourth level domain code.
- `vnet` is the domain name and is the third level domain code
- `ibm` is the second level domain code.
- `com` is the top level generic domain code.

For example, using this domain resolution, an example user ID would be classified as guyd@johic1.vnet.ibm.com.

1.1.7 SLIP

SLIP (Serial Line Interface Protocol) is a simple form of encapsulation for IP datagrams on serial lines. SLIP has become popular for connecting home computers to the Internet through the RS-232 serial port. SLIP is a simple framing method. The IP datagram is terminated by an END (0xc0) character. To prevent noise from being interpreted as part of the datagram, the END statement is also transmitted at the beginning of the datagram. If a byte of data in the datagram equals the END character, a two-byte sequence 0xdb, 0xdc is transmitted instead. This simple framing allows datagram packets to be transmitted over a simple serial line link.

SLIP has the following deficiencies:

- Each end must know the other′s IP address.
- A serial line cannot be shared between SLIP and another protocol.
- There is no checksum added to the SLIP datagram. SLIP relies on error-checking modems or on the higher network layers to detect errors.

The IBM global network uses SLIP as its serial connection standard. Most UNIX systems include SLIP support in their base operating software. Another form of SLIP is compressed SLIP, which is better at handling small datagram packets.

1.1.8 PPP

PPP (Point-to-point protocol) is a serial-line protocol similar to SLIP. PPP corrects the deficiencies in SLIP. PPP provides the following advantages over SLIP:

- Support for multiple protocols across a single serial line, not just IP datagrams
- A cyclic redundancy check on each datagram frame
- TCP and IP header compression similar to CSLIP, making it potentially faster than SLIP

PPP provides distinct advantages over SLIP and is recommended when starting up a serial-line based IP network from scratch. When connecting into a service provider, you need to know which protocol they support. You cannot connect into a SLIP line using PPP and vice versa.

Many service providers provide dynamic SLIP and PPP IP address resolution. That is, you enter your default address as 1.1.1.1 or 0.0.0.0, depending on the software, connection and service provider you are using, and the provider will then allocate you an IP address from a pool of addresses that are available at that time. Connecting into the IBM Global Network using a SLIP protocol connection, you need to specify your default IP address as 1.1.1.1.

The GateD daemon is required to support SLIP in a dynamic routing environment. Routed does not properly support SLIP. If given the option, PPP is a better protocol than SLIP, although it is not as widely implemented.

1.1.9 Example of TCP/IP Usage

We have included an example of how some of the TCP/IP protocols are used when you connect to another host on the Internet.

We typed the command:

```
ftp tollbooth3.cwp.ibm.com
```

In simple terms, the following steps took place:

1. The application FTP called a function to convert the hostname to an IP address at the application layer, in this case, tollbooth3.cwp.ibm.com.
2. The local host could not resolve the IP address, so the host then looked at its defined DNS (9.19.141.242). This DNS attempted to resolve the name to an IP address. The DNS looked at its resolve tables and resolved tollbooth3.cwp.ibm.com to the IP address of 198.74.69.100 through router 9.24.104.1.
3. Once the application layer resolved the IP address, the application layer communicated with the TCP layer. The TCP layer encapsulated the data with a TCP header record and passed the data onto the IP layer.
4. The IP layer compared the datagram size with the MTU (Maximum Transmission Unit) and performed data fragmentation as necessary. The IP layer encapsulated each MTU size packet of data with the IP header data.
5. The IP layer passed MTU size datagram down to the link layer. The link layer encapsulated the MTU size unit with a token-ring header and trailer record.
6. Using the ARP function, the IP layer function has determined the physical hardware connection to which the datagram must be sent. This physical connection is the router hardware address for the segment of the Internet.
7. The hardware link layer then sends the datagram via the LAN to the router hardware address specified by the ARP function.
8. The router then looked at the IP record in the header layer and then resent the data to its final destination at 198.74.69.100.
9. The datagram then passes back up the stack to the TCP layer, which sent a data acknowledgement packet by the reverse route back to the original sender.
10. The original sender received the acknowledged data and continued to send more data packets.
11. If the datagram had not been received, the TCP layer would have resent the datagram packets until acknowledgement was received.
12. Once the acknowledgement was received, the remote system responded at the application layer with the FTP login sequence.

1.1.10 WinSock

Microsoft Windows network software was built around NetBIOS network protocols. It does not support native TCP/IP calls internally. There is an interface specification standard for communicating with TCP/IP via Windows that is termed WinSock (Windows Sockets). This standard is based on TCP/IP sockets and specifies a set of calls that Windows applications can make to the TCP/IP software. WinSock has been developed as an external library call to communicate between Windows and TCP/IP, as shown in Figure 9 on page 18.

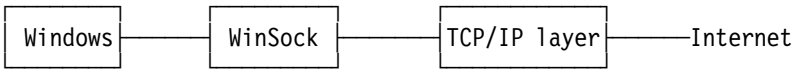


Figure 9. WinSock Operation

The IBM Internet Connection for Windows relies on WinSock. There are a number of WinSock applications on the market for connecting Windows applications to the Internet. An application must support the WinSock protocol to be able to use WinSock libraries correctly.

The WinSock library can be obtained via anonymous FTP from:

`ftp.cica.indiana.edu/pub/pc/win3/winsock`

Chapter 2. Functional Overview of TCP/IP for OS/2

This chapter describes the functions and features of TCP/IP for OS/2 that are included in the base and optional kits.

2.1 OS/2 Warp and TCP/IP

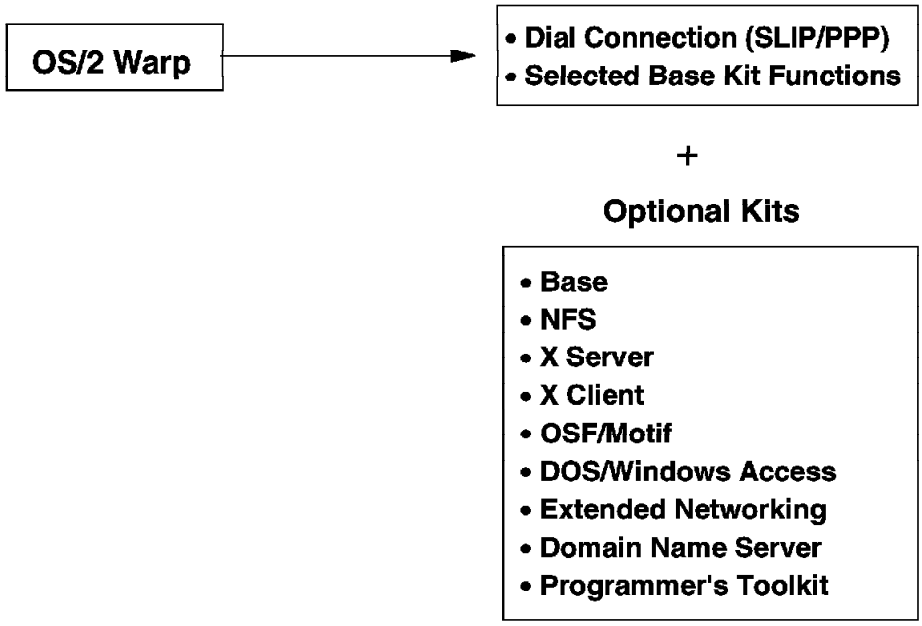


Figure 10. OS/2 Warp - TCP/IP Support

Before introducing the functions and features of the OS/2 Warp TCP/IP implementation, it will be helpful to understand some background information. As indicated in Figure 10, the first version of OS/2 Warp implemented a dial-only TCP/IP stack. There was a requirement to install TCP/IP V2.0 for OS/2 to obtain TCP/IP LAN connectivity and to provide other application functions.

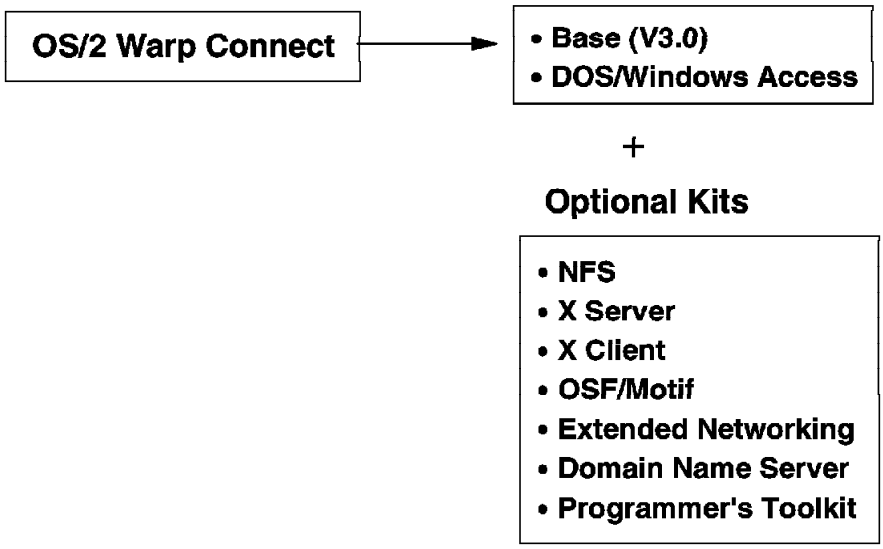


Figure 11. OS/2 Warp Connect - TCP/IP Support

As indicated in Figure 11, Warp Connect currently includes TCP/IP V3.0 for OS/2, which provides all base kit functions. In addition, the DOS/Windows Access kit has been bundled with Warp Connect. It is also possible to install additional kits (NFS and X Server for example) as long as appropriate CSD maintenance is applied to them.

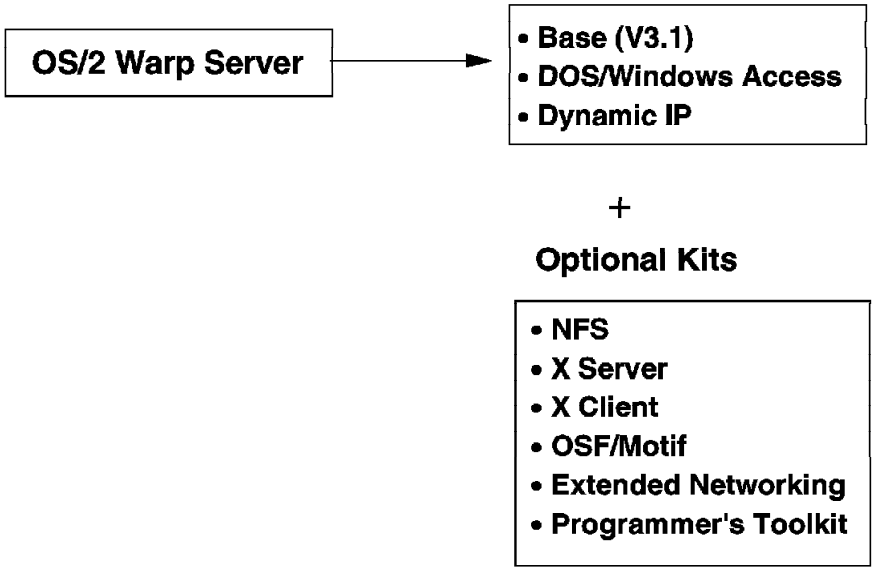


Figure 12. OS/2 Warp Server - TCP/IP Support

Finally, as described in Figure 12, Warp Server includes TCP/IP V3.1 for OS/2, which features support for Dynamic IP. Dynamic IP includes DHCP and DDNS

server/client functions; these are described in greater detail in Chapter 5, “Dynamic IP” on page 47.

Again, it is possible to install additional kits as long as appropriate CSD maintenance is applied to them.

In this publication, all references to TCP/IP for OS/2 should be interpreted as pertaining to V3.1 as provided with OS/2 Warp Server unless we explicitly state otherwise.

2.2 Overview

TCP/IP for OS/2 gives the user at an OS/2 workstation attached to a TCP/IP network the following components:

1. TCP/IP clients that can be run in OS/2 full-screen, text windows or from the OS/2 Workplace Shell
2. TCP/IP servers that can be started in OS/2 full-screen sessions, te xt windows, or from the OS/2 Workplace Shell
3. Network status commands to query the status of local and remote workstations
4. Device drivers and configuration programs to access a LAN or a serial line
5. A methodology to install many workstations remotely from a single server providing the necessary product files
6. Tools for developing network client/server applications
7. Online publications and a utility program that displays them

The TCP/IP for OS/2 base is supplemented by functional packages called kits. These can be obtained in addition to the base that is shipped with Warp Server or Warp Connect. The following table gives an overview of the provided functions.

Table 4. TCP/IP for OS/2 Functions

Package	**Functions**
TCP/IP for OS/2 V3.1 (Warp Server)	Base TCP/IP functions: • Telnet • FTP • TFTP • REXEC • RSH • Remote Printing • SMTP • WEB Explorer • UltiMail Lite • NewsReader/2 • TALK • SNMP • Routing • ARP • PING • BOOTP • Dynamic IP Support (DHCP and DDNS) • DOS/Windows Access • DBCS Support Multi Protocol Transport Support (MPTS) Redirected Installation Method (CID) Configuration Notebook Serial Line Internet Protocol (SLIP) Point to Point Protocol (PPP) Online Documentation
Network File System Kit	NFS Server NFS Client Redirected Installation Method (CID) Configuration Notebook Online Documentation
TCP/IP for OS/2 Programmer's Toolkit	32-bit APIs REXX Support SNMP DPI Samples Redirected Installation Method (CID) Online Documentation
X Window System Server Kit	X Server Version 11 Release 5 Utilities Redirected Installation Method (CID) Configuration Notebook Online Documentation
X Window System Client Kit	X Client APIs Samples Redirected Installation Method (CID) Online Documentation
OSF/Motif Kit	OSF/Motif V1.2 Client APIs Samples
Extended Networking Kit	X.25 SNALINK Redirected Installation Method (CID) Configuration Notebook Online Documentation

2.3 Servers and Clients

This section describes the servers and clients provided by TCP/IP for OS/2.

2.3.1 Servers

With the TCP/IP for OS/2 product, you can set up the following servers:

TELNETD: TELNETD supports ANSI and VT100 terminals in OS/2 full-screen mode.

FTPD: FTPD supports file exchange using the File Transfer Protocol with other TCP/IP hosts.

TFTPD: TFTPD supports file exchange using the Trivial File Transfer Protocol with other TCP/IP hosts. You can restrict access for remote clients to files in a single directory.

REXECD: REXECD executes OS/2 commands issued from other TCP/IP hosts.

RSHD: RSHD executes OS/2 commands issued from other TCP/IP hosts.

LPD: LPD interfaces to the OS/2 print spooler and makes it possible to share an OS/2 printer with other TCP/IP workstations.

INETD: This server incorporates all servers mentioned so far by starting them within a single task. This reduces system overhead, but you cannot specify parameters to a server if it is started by INETD.

LPRPORTD: This server provides the capability to print using LPR port icons.

SENDMAIL: SENDMAIL implements SMTP to transmit mail to and receive mail from other TCP/IP hosts.

UltiMail Lite: UltiMail Lite implements SMTP and the MIME extension of 822 RFC to exchange multimedia mail between TCP/IP hosts.

TALKD: TALKD together with the TALK client supports interactive conversation with a user at another TCP/IP host.

ROUTED: An OS/2 workstation connected to two LANs works as an Internet gateway. ROUTED expands the basic routing capabilities of the Internet Protocol (IP) by using the Routing Information Protocol (RIP) to maintain its routing table.

SNMPD: The Simple Network Management Protocol agent (SNMPD) can communicate with an SNMP network management host (an SNMP Monitor) and with SNMP subagents to provide network management support in a TCP/IP environment.

NFSD: The NFS server enables you to export directories of an OS/2 system so that users running an NFS client can mount them and access them as local disk drives. You can restrict access to certain directories to selected systems, and you can export directories read-only.

Portmapper: The portmapper program maps RPC program and version numbers to transport-specific port numbers. This program makes dynamic binding of remote programs possible. This is desirable because the range of

reserved port numbers is very small and the potential number of remote programs is very large.

PMX: PMX is the Presentation Manager X Window System Server. It allows the user to run remote X client applications and display them locally.

NAMED: The dynamic name server (DDNS) does the Internet-to-name translation on one centralized system. It also implements dynamic updates of its name tables in order to communicate with dynamic IP components. Remote systems can ask to get the internet address for a certain name.

DHCPSD: DHCPSD is the dynamic host configuration server. It provides all required configuration parameters for DHCP clients.

BOOTPD: BOOTPD sends an internet address, subnet mask, default IP router address, hostname, and domain name to a BOOTP client on the local LAN that broadcasts its network adapter address using the BOOTP protocol.

DOS/Windows Servers: The DOS/Windows Access enables DOS and Windows applications that were originally written using the APIs provided by IBM TCP/IP V2.1.1 for DOS, to run as a server in an OS/2 Virtual DOS Machine or WIN-OS/2 session, using the TCP/IP protocol stack provided by TCP/IP for OS/2.

2.3.2 Clients

TCP/IP for OS/2 has the following clients implemented:

Telnet: Telnet in TCP/IP for OS/2 supports ASCII (ANSI, VT100, VT220, HFT, and NVT) and EBCDIC (3270, 5250) terminals for logging on to other TCP/IP hosts. Telnet is also integrated in the OS/2 Workplace Shell. 3270 Telnet is implemented as a Presentation Manager (Telnet3270), and as an OS/2 full-screen or window application (TN3270). 5250 Telnet is implemented as a Presentation Manager application (TN5250).

Telnet3270: Telnet3270 is a 3270 terminal emulator that supports all 3270 terminal types, extended colors, and a copy to PM clipboard function. A number of different fonts can be selected to control the size of the window. It runs as a Presentation Manager task and is integrated in the OS/2 Workplace Shell.

FTP: File Transfer Protocol supports file transfer to/from other TCP/IP hosts.

FTPPM: FTPPM is the Presentation Manager version of the FTP client and is integrated in the OS/2 Workplace Shell. The application is based on selection lists.

TFTP: Trivial File Transfer Protocol is an alternative to transferring files with FTP. It does not provide all the features available in FTP.

REXEC: REXEC enables remote command execution on other TCP/IP hosts.

RSH: RSH enables remote command execution on other TCP/IP hosts.

LPR: Remote Printing Protocol allows remote printing on other TCP/IP hosts. The line printer client (LPR) sends the file to be printed to a specified print server host and to a specific printer.

LPRMON: LPRMON is a Parallel Device Monitor that allows you to set up your PC to automatically send data to a remote LPR server.

LPR Ports: LPR ports provide output to a remote TCP/IP printer from within a printer object of the OS/2 Workplace Shell.

DHCPCD: DHCPCD is the dynamic host configuration client. It will get all required configuration parameters from a DHCP server during workstation startup. Users do not have to worry about TCP/IP configuration anymore.

NFS: The NFS client enables you to mount remote file systems and access them as local disk drives.

Portmapper: When a client wishes to access an RPC service, such as NFS, it first sends an enquiry including the program, procedure and version numbers together with the type of the desired protocol to the portmapper of the target computer. It receives the port number of the service by return and can then send the request directly to the server.

SENDMAIL: The OS/2 mail server that uses SMTP to route mail from one host to another host on the network. A file can be piped to the SENDMAIL server by invoking SENDMAIL in foreground mode. Normally, the OS/2 user will use the Presentation Manager UltiMail/Lite application instead.

UltiMail Lite Client: UltiMail Lite is an electronic mail system used to exchange mail with different environments based on the Simple Mail Transfer Protocol (SMTP) and the MIME extension of RFC 822, which allows you to send multimedia mail using the TCP/IP network.

TALK: TALK together with the TALKD server enables interactive conversations with a user at another TCP/IP host.

NewsReader/2: NewsReader/2 (NR/2) is a Presentation Manager application that allows you to access USENET news servers on an Internet. News servers contain a broad range of news groups that can be viewed, subscribed, and appended to by NR/2.

BOOTP: BOOTP is used to obtain a host's internet address, subnet mask, default IP router address, hostname, and domain name from a BOOTPD server on the local LAN by broadcasting its network adapter address using the BOOTP protocol.

X Window System Clients: The X Window System Client Kit provides a few X Client sample programs that are ready to use if the OS/2 X Server is installed.

DOS/Windows Clients: The DOS/Windows Access enables DOS and Windows applications that were originally written using the APIs provided by IBM TCP/IP V2.1.1 for DOS, to run in an OS/2 Virtual DOS Machine or WIN-OS/2 session, using the TCP/IP protocol stack provided by TCP/IP for OS/2.

2.3.3 Tools for Developing Network Applications

TCP/IP for OS/2 and its Programming Kit V3 provide you with the following application programming interfaces:

- Berkeley sockets and Sun RPC for development of cooperative processing client/server applications between an OS/2 workstation and another TCP/IP host supporting these interfaces.
- A distributed programming interface (DPI) to the SNMPD agent in OS/2 that makes it possible to dynamically add objects to the Management Information Base (MIB) and generate alerts (TRAPs) that are forwarded by SNMPD to a SNMP network management host.
- An FTP API that allows the user to execute a file transfer from within an application. All the FTP subcommands used during an FTP session, can be invoked and controlled by a user program.
- Support for developing client applications according to the X Window System is provided by the X Client Kit.
- An REXX FTP API for easy access to the OS/2 TCP/IP FTP API.
- An REXX Socket API for easy access to the OS/2 TCP/IP Socket API.

2.3.4 Online Documentation

TCP/IP for OS/2 and its additional kits provide online documentation that can be viewed through the OS/2 Workplace Shell. Most of the kits provide manuals in IBM BookManager Read/2 format which can be viewed using IBM Library Reader/2. A copy of the IBM Library Reader can be found on the Warp Server CDROM in the BOOKLIBREAD subdirectory.

In addition to these online manuals, there is online information that can be viewed using the OS/2 View command or by selecting an icon from the OS/2 Workplace Shell.

2.4 ITSO Environment

The ITSO environment that was used in providing the product usage examples in this redbook is shown in Figure 13 on page 27.

Note that the ITSO intranet was used (9.24.104.0 network) which is part of the IBM intranet (IBM MPN or MultiProtocol Network) which is connected to the Internet through the Socks server 9.67.43.8.

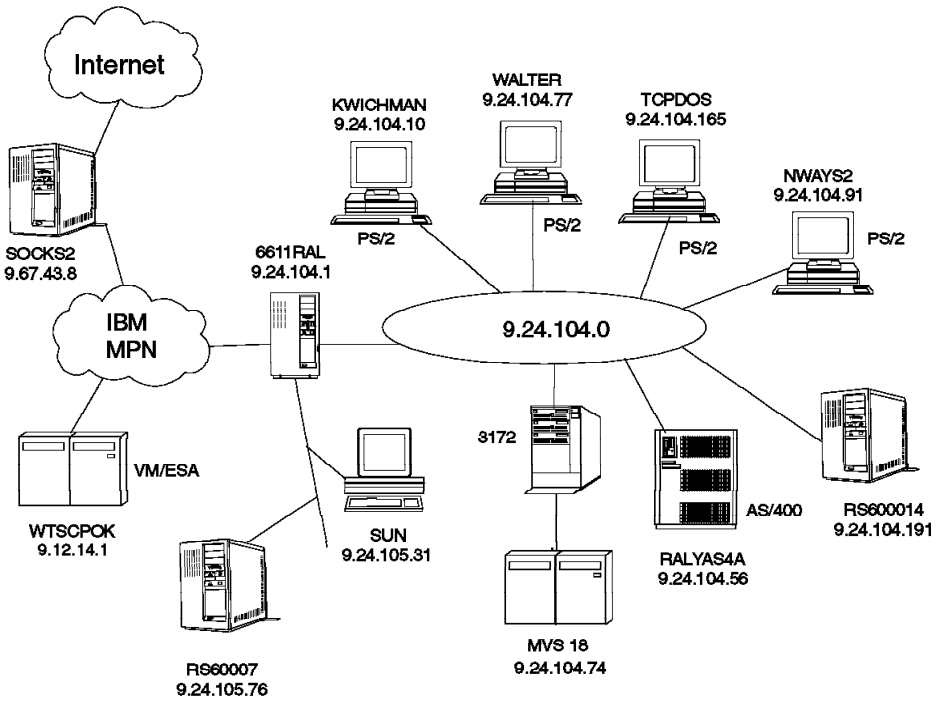


Figure 13. ITSO Environment

Chapter 3. CID Installation of TCP/IP for OS/2

This chapter explains how to perform a CID installation of TCP/IP V3.1 for OS/2. Also described are the CID installations of the PMX and NFS kits. In addition, the new CID keywords supporting dynamic IP in an unattended installation are covered.

You can install TCP/IP V3.1 on any existing OS/2 Warp workstation or replace an older level of TCP/IP. The installation consists of two steps:

1. Install MPTS Version 5 (which contains the TCP/IP stack)
2. Install the TCP/IP applications (Telnet, FTP etc.)

3.1 MPTS Configuration

The MPTS configuration response file is divided into three main parts:

1. Install section
2. Protocol section
3. Sockets MPTS section

We concentrate on the Sockets MPTS section, since it contains the TCP/IP-related information necessary for the CID installation. For more information on the other parts, please refer to the MPTS online information.

The Sockets MPTS section contains the TCP/IP configuration parameters. It generates an MPTCONFIG.INI file with the values specified. During installation/configuration (if there is no TCP/IP for OS/2 component on the workstation), the values specified in the Sockets MPTS section are used to generate the SETUP.CMD file.

The MPTS section contains several segments that are [CONTROL], [IFCONFIG], [ROUTE], [DHCP], and [NETBIOS]. If the workstation currently has TCP/IP installed on it, only the [CONTROL] section needs to be specified (if required) to configure local IPC sockets, NetBIOS sockets, or TCP/IP sockets access. If the workstation does not have TCP/IP installed on the system, then a complete Sockets MPTS section (with all segments properly completed) must be specified to enable TCP/IP sockets access.

You can specify a valid MPTCONFG.INI file within the Sockets MPTS section. The information specified in this section overwrites an existing MPTCONFG.INI file. To update a currently configured MPTCONFG.INI file, use the MPTS update section. The Sockets MPTS section should be placed after any NetBIOS sections.

The new keywords in the [DHCP] section are:

Adapter The network interface that will be configured by DHCP.

ClientID The client identifier that is included in all packets sent to the DHCP server. MAC is the only allowable value.

DDNS When executed, information in this command string updates the Dynamic DNS server with the current IP address for the specified host name.

Numlogfiles The number of desired log files.

LogfileSize The maximum size, in kilobytes, of the log files.

LogfileName The name of the most recent log file.

The keywords SYSERR, OBJERR, PROTERR, WARNING, EVENT, ACTION, INFO, ACNTING, TRACE specify what kind of information will be written to the logs. Please refer to the DHCP online documentation for further information.

The installation program that has to be called for CID installation is simply the MPTS.EXE program with appropriate parameters specified. The correct MPTS.EXE file can be found on the WARP Server CDROM in the CIDSERVERMPTS subdirectory. It uses the following parameters:

/s: Specifies the source directory where the product files reside.

/t: Specifies the target directory.

/r: Specifies the full path and file name of the CID response file.

/l1: Specifies the log file name of the installation process.

A valid response file to install unattended MPTS with DHCP enabled to a target workstation follows:

```
INST_SECTION = (
UPGRADE_LEVEL = NEW
INSTALL = PRODUCT
  )

PROTOCOL = (
[PROT_MAN]

   DRIVERNAME = PROTMAN$

[IBMLXCFG]

   landd_nif = landd.nif
   netbeui_nif = netbeui.nif
   tcpip_nif = tcpip.nif
   IBMTOK_nif = IBMTOK.NIF

[NETBIOS]

   DriverName = netbios$
   ADAPTER0 = netbeui$,0

[landd_nif]

   DriverName = LANDD$
   Bindings = IBMTOK_nif
   ETHERAND_TYPE = "I"
   SYSTEM_KEY = 0x0
   OPEN_OPTIONS = 0x2000
   TRACE = 0x0
   LINKS = 8
   MAX_SAPS = 3
   MAX_G_SAPS = 0
   USERS = 3
   TI_TICK_G1 = 255
```

```
   T1_TICK_G1 = 15
   T2_TICK_G1 = 3
   TI_TICK_G2 = 255
   T1_TICK_G2 = 25
   T2_TICK_G2 = 10
   IPACKETS = 250
   UIPACKETS = 100
   MAXTRANSMITS = 6
   MINTRANSMITS = 2
   TCBS = 64
   GDTS = 30
   ELEMENTS = 800

[netbeui_nif]

   DriverName = netbeui$
   Bindings = IBMTOK_nif
   ETHERAND_TYPE = "I"
   USEADDRREV = "YES"
   OS2TRACEMASK = 0x0
   SESSIONS = 130
   NCBS = 225
   NAMES = 21
   SELECTORS = 15
   USEMAXDATAGRAM = "NO"
   ADAPTRATE = 1000
   WINDOWERRORS = 0
   MAXDATARCV = 4168
   TI = 30000
   T1 = 1000
   T2 = 200
   MAXIN = 1
   MAXOUT = 1
   NETBIOSTIMEOUT = 2000
   NETBIOSRETRIES = 3
   NAMECACHE = 1000
   RNDOPTION = 1
   PIGGYBACKACKS = 1
   DATAGRAMPACKETS = 10
   PACKETS = 350
   LOOPPACKETS = 8
   PIPELINE = 5
   MAXTRANSMITS = 6
   MINTRANSMITS = 2
   DLCRETRIES = 10
   FCPRIORITY = 5
   NETFLAGS = 0x0

[tcpip_nif]

   DriverName = TCPIP$
   Bindings = IBMTOK_nif

[IBMTOK_nif]

   DriverName = IBMTOK$
   MAXTRANSMITS = 6
   RECVBUFS = 2
   RECVBUFSIZE = 256
```

```
   XMITBUFS = 1

  )

MPTS = (

[CONTROL]
  Local_IPC      = YES
  INET_Access    = YES
  NETBIOS_Access = NO
[IFCONFIG]
  Interface      = 0,,,
  Address        = ,,,
  Brdcast        = ,,,
  Dest           = ,,,
  Enable         = DOWN,,,
  Netmask        = ,,,
  Metric         = 0,,,
  Mtu            = 1500,,,
  Trailers       = ,,,
  Arp            = ,,,
  Bridge         = ,,,
  Snap           = ,,,
  Allrs          = ,,,
  802.3          = ,,,
  Icmpred        = ,,,
  Canonical      = ,,,
  EnableDhcp     = YES,,,
[DHCP]
 Adapter        = 0
 ClientID       = MAC
 DDNS           = NO
 NumLogFiles    = 0
 LogFileSize    = 0
 LogFileName    =
 SYSERR         = NO
 OBJERR         = NO
 PROTERR        = NO
 WARNING        = YES
 EVENT          = YES
 ACTION         = NO
 INFO           = YES
 ACNTING        = NO
 TRACE          = YES

)
```

3.2 TCP/IP Configuration

Since most of the stack configuration has already been done in the MPTS part, the TCP/IP configuration response file is very easy to understand. You can find the installation program that has to be called for CID installation on the WARP Server CDROM in the CIDSERVERTCPAPPS subdirectory. It is called install.exe and uses the following parameters:

/s: Specifies the source directory where the product files reside.

/t: Specifies the target directory. This can be overridden by the TARGET= CID keyword in the TCP/IP response file.

/r: Specifies the full path and file name of the CID response file.

/l1: Specifies the log file name of the installation process.

A valid response file to install unattended all TCP/IP applications to a target workstation follows:

```
ATTENDED=N
TARGET_PATH=C:\TCPIP

// Default response file for PRODUCT DISK install

INSTALL_TITLE = TCP/IP for OS/2 Client

INSTALL_NAME = BASE 8.48   1 4 "Client Kit"  Base TCP/IP Applications
INSTALL_NAME = INET 3.59   5 6 "Client Kit"  Feature TCP/IP Applications
INSTALL_NAME = DBOX  1.77  7 7 "Client Kit"  DOS\Windows Access
INSTALL_NAME = UMAIL 4.51  7 8 "Client Kit"  UltiMail Lite

EXEC = BASE call clntxt
EXEC = DBOX call dboxxt
EXEC = UMAIL call umlitext
```

Since there are no new CID keywords, please refer to online documentation for further information.

3.3 Installing Additional Kits

This section describes how to install the PMX and NFS Kits after TCP/IP V3.1 has been previously installed. Although these kits are named as Version 2.0 kits, they are supported to run with TCP/IP V3.1 as long as the latest maintenance CSDs (corrective service diskettes) have been applied.

It is important that this CID installation be done in a different step, since these kits use a different installation routine from the TCP/IP code itself.

3.3.1 PMX Kit (OS/2 X Server)

If you want to install the PMX kit and the latest CSD (UN86625) in one step, you have to prepare the code server first. The following steps are necessary:

- Create a subdirectory for the product files.
- Copy the PMX product files from the diskettes to the code subdirectory.
- Copy the PMX CSD (UN86625) from the diskettes to the code subdirectory. Make sure that you replace older files with the new ones that come with the CSD.

The directory should look like this:

```
The volume label in drive E is E_DRIVE.
The Volume Serial Number is 2632:4C15.
Directory of E:\TCPKITS

 2-09-96  10:15a     <DIR>           0  .
 2-09-96   9:09a     <DIR>           0  ..
 2-09-96   9:22a       428           0  DEFAULT.RSP
 2-09-96  10:15a         0           0  files
 3-23-94  11:13a     15756           0  IBMLANLK.EXE
 3-23-94  11:13a      3988           0  IBMLANLK.SYS
 2-09-96   9:26a     32079           0  log.txt
 3-23-94  11:14a      4550           0  LSI.MSG
 3-23-94  11:14a      6571           0  LSIH.MSG
 8-11-93   2:01p    814847           0  PMX1.ZIP
 8-11-93   2:00p   1177595           0  PMX2.ZIP
 8-12-93   9:05a   1183348           0  PMX3.ZIP
 8-12-93   9:05a   1178290           0  PMX4.ZIP
 8-11-93   2:03p   1177514           0  PMX5.ZIP
 8-12-93   7:41p    282446           0  PMX6.ZIP
 1-19-96   1:30p    759171           0  PMXD1.ZIP
12-13-95   5:41p   1437215           0  PMXD2.ZIP
12-13-95   5:33p   1389256           0  PMXD3.ZIP
12-13-95   5:34p    994791           0  PMXD4.ZIP
12-13-95   5:42p    660791           0  PMXD5.ZIP
12-11-95  10:36a     31232          49  PMXGREHK.EXE
12-19-95   2:22p     69136          49  PMXXT.EXE
10-27-94   3:38p     53904           0  pmxxt.org
 8-12-93   7:29p      5436           0  README
 1-18-96   4:40p     18622           0  README.PMX
12-19-95   2:21p    117642           0  TCPINST.EXE
 4-12-94   5:35p     12663           0  TCPINST.HLP
12-19-95   2:22p    119151          49  TCPINST2.EXE
12-11-95   7:41a     89152          49  UNZIP.DLL
       29 file(s)   11635574 bytes used
                   121719808 bytes free
```

Now you have to create an appropriate response file. The following response file installs the product and the CSD to F:TCPIP in one step:

```
// Default response file for PRODUCT DISK install


TARGET_PATH=F:\TCPIP
ATTENDED=N

INSTALL_NAME = PMX  11.36 1 6 "X Window System Server Kit" PMX Kit
INSTALL_NAME = PMXD 12.63 1 5 "CSD UN86625 X Window System Server Kit" CSD UN86625 PMX Kit

EXEC = PMX call pmxxt BOOT_DRIVE TARGET_PATH HOSTNAME
EXEC = PMXD call pmxxt BOOT_DRIVE TARGET_PATH HOSTNAME
EXEC = PMXD call pmxgrehk BOOT_DRIVE TARGET_PATH HOSTNAME
```

Now you can invoke the CID installation using the tcpinst.exe routine with the following command line:

```
tcpinst /r:e:tcpkitsdefault.rsp /l1:e:tcpkitslog.txt
```

This starts the installation and produces a log file in the given subdirectory. The following shows the beginning and end of a successful installation using the previously described files:

```
========================= TCP/IP 2.0 Installation ======================
Install initiated on 2/9/1996 at 9:23.
Command line: TCPINST
        /r:e:\tcpkits\default.rsp
        /l1:e:\tcpkits\log.txt
Loading UNZIP...
   loading unzip from (E:\TCPKITS\UNZIP.DLL)
Initializing...
Reading response file...
  using response file (e:\tcpkits\default.rsp)
Preparing the target directories
WARNING: ETC subdirectories exist, F:\TCPIP\ETC and F:\MPTN\ETC.
Most configuration files are placed in ETC. If you haven't backed
up configuration files, your changes may be lost.  Do you wish
to continue installing?
Backing up CONFIG.SYS...
Installing Selected Components...
Checking for linked kits....
Checking Dependencies...
Installing (PMX)
Installing PMX Kit
F:\TCPIP\x11\misc\heb8x13.pcf
: : :
: : :
: : :
: : :
Writing CONFIG.TCP...
Running the kit exits...
Running exit for PMX Kit
Running pgm (F:\$cidtmp$\pmxxt.exe) with args ( F F:\TCPIP\ dos3)
Running exit for CSD UN86625 PMX Kit
Running pgm (F:\$cidtmp$\pmxxt.exe) with args ( F F:\TCPIP\ dos3)
Running pgm (F:\$cidtmp$\pmxgrehk.exe) with args ( F F:\TCPIP\ dos3)

------------------------------------------------------------
PMX Graphics Engine Hook currently installed at:
>F:\tcpip\DLL\GREHOOK.DLL<
------------------------------------------------------------
PMX Graphics Engine Hook will be >F:\TCPIP\DLL\GREHOOK.DLL<
===> You MUST REBOOT OS/2 in order to activate GREHOOK.DLL <===

Processing ETC directory...
Creating Desktop...
TCPINST exiting with rc == FE04
```

rc == FE04 shows that the installation was successful and a reboot of the machine is required. The reason why it does not complete with rc == 00 is that there was a warning informing the operator that some files in the ETC subdirectory might be overwritten by the installation without being saved.

3.3.2 NFS Kit (OS/2 Network File System)

The CID installation of the NFS Kit and the latest CSD (UN57064) in one step is very similar to the PMX Kit installation steps. You first have to set up the code server:

- Create a subdirectory for the product files.
- Copy the NFS product files from the diskette to the code subdirectory.
- Copy the NFS CSD (UN57064) from the diskette to the code subdirectory. Make sure that you replace older files with the new ones that come with the CSD.

Now you have to create an appropriate response file. The following response file installs the product and the CSD to C:TCPIP in one step:

```
// Default response file for PRODUCT DISK install

TARGET_PATH=C:\TCPIP
ATTENDED=N

INSTALL_NAME = NFS  1.10 1 1 "Network File System Kit"  NFS Kit
INSTALL_NAME = NFSC 0.99 1 1 "CSD UN57064, NFS Kit"  CSD UN57064 for NFS Kit

EXEC = NFS call nfsxt BOOT_DRIVE TARGET_PATH
EXEC = NFSC call nfscxt TARGET_PATH
```

Now you can invoke the CID installation using the tcpinst.exe routine with the following command line:

```
tcpinst /r:e:tcpkits1default.rsp /l1:e:tcpkits1log.txt
```

This starts the installation and produces a log file in the given subdirectory. The following shows the beginning and end of a successful installation using the previously described files:

```
========================= TCP/IP 2.0 Installation =====================
Install initiated on 2/9/1996 at 13:55.
Command line: TCPINST
        /r:x:\tcpkits1\default.rsp
        /l1:x:\tcpkits1\log.txt
Loading UNZIP...
   loading unzip from (X:\TCPKITS1\UNZIP.DLL)
Initializing...
Reading response file...
  using response file (x:\tcpkits1\default.rsp)
Preparing the target directories
: : :
: : :
: : :
: : :
C:\TCPIP\bin\dsknfs.txt

Writing CONFIG.TCP...
Running the kit exits...
Running exit for NFS Kit
Running pgm (C:\$cidtmp$\nfsxt.exe) with args ( C C:\TCPIP\)
```

```
Running exit for CSD UN57064 for NFS Kit
Running pgm (C:\$cidtmp$\nfscxt.exe) with args ( C:\TCPIP\)

Processing ETC directory...
Creating Desktop...
TCPINST exiting with rc == FE04
```

Again, rc == FE04 shows that the installation was successful and a reboot of the machine is required.

In order to fix a mount problem running the NFS Kit in conjunction with TCP/IP 3.0 or higher, APAR Fix PN69745 (NFSD.EXE) has to be applied after installing the complete kit.

Chapter 4. Double-Byte Character Set Support

In countries that use ideographic characters such as China, Japan, and Korea, it is a fundamental requirement to handle the double-byte character set (DBCS) as well as the single-byte character set (SBCS) in application systems. In TCP/IP V2.0 for OS/2, it was necessary to use the Asia Pacific DBCS (AP/DBCS) kit instead of the TCP/IP V2.0 for OS/2 Base kit to support TCP/IP applications in a DBCS OS/2 environment. TCP/IP V3.x for OS/2 Warp has been enabled with DBCS support in four DBCS versions of OS/2 Warp Connect and OS/2 Warp Server. Therefore you do not need the AP/DBCS kit any more. These four DBCS versions are:

- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

This chapter provides some guidance in using these DBCS versions of TCP/IP V3.x for OS/2 Warp.

4.1 Using DBCS TCP/IP V3.x for OS/2 Warp

This section describes the information you need to use the DBCS TCP/IP V3.x for OS/2 Warp.

4.1.1 Environment Variables

After your DBCS TCP/IP for OS/2 Warp has been installed and configured, you should take note of the following environment variables:

LANG The LANG environment variable is set by the TCP/IP V3.x for the OS/2 Warp installation program. Values for the LANG environment variable for DBCS OS/2 depend on each country's specification. For example, to set the LANG environment variable with the traditional Chinese language used in Taiwan, this setting in CONFIG.SYS should be as follows:

```
SET LANG=Zh_TW
```

LOCPATH A locale is the definition of the subset of the user's environment that depends on language and cultural conventions.

In TCP/IP V3.0 for OS/2 Warp, the locale path in CONFIG.SYS is set by the TCP/IP installation program. The default setting is as follows:

```
SET LOCPATH=C:TCPIPLOCALE
```

Note: If your LOCPATH environment variable is set already, be sure that the value for TCP/IP V3.0 is set prior to the existing value. For example:

```
SET LOCPATH=C:TCPIPLOCALE;C:IBMLANXPG4LOCALE;
```

In TCP/IP V3.1 for OS/2 Warp, the locale path in CONFIG.SYS is set by the MPTS (the version shipped with OS/2 Warp Server) installation program. For example:

```
SET LOCPATH=C:IBMI18NLOCALE
SET I18NDIR=C:\IBMI18N
```

4.1.2 Code Page Translation

When data is transferred in ASCII format, there must be a translation between the ASCII representation used on the workstation and the version recognized by other hosts on the network. Translation is very important for special characters used in languages other than English. When you specify a code page translation table, the text that you send onto the network and that which you receive from the network is translated accordingly.

The following applications allow you to select a code page translation table:

- FTP
- FTPD
- FTP-PM
- Gopher
- LPD
- LPR
- LPRMON
- NewsReader/2
- Talk
- Telnet
- Telnetd
- TelnetPM
- TFTP
- TFTPD
- WebExplorer

You can use one of the code page translation tables provided or define your own single-byte character set (SBCS) code page translation table.

To specify the code page translation name to be used with the above applications, use the -cp parameter. The possible values for the -cp parameter are as follows:

NONE No code page translation occurs.

BIG5 Code page 950 to Republic of China Big-5 (8-bit) translation.

DECMULTI Code page 850 to DEC Multinational translation.

EUC Code page 932 or 942 to Japanese Extended UNIX Code translation.

Code page 949 or Korean KSC 5601 Extended UNIX Code translation. (No translation is done between these two code pages since they are equivalent.)

Code page 950 to Republic of China Big-5 Extended UNIX Code translation. (No translation is done between these two code pages since they are equivalent.)

GB8	Code page 1381 to Simplified Chinese GB Code (8-bit) translation. (No translation is done between these two code pages since they are equivalent.)
ISO1	Code page 850 to ISO code page 8859-1 translation.
ISO2	Code page 852 to ISO code page 8859-2 translation.
ISO5	Code page 855 to ISO code page 8859-5 translation.
	Code page 866 to ISO code page 8859-5 translation.
	Code page 915 to ISO code page 8859-5 translation. (No translation is done between these two code pages since they are equivalent.)
ISO6	Code page 864 to ISO code page 8859-6 translation.
ISO7	Code page 813 to ISO code page 8859-7 translation. (No translation is done between these two code pages since they are equivalent.)
ISO8	Code page 862 to ISO code page 8859-8 translation.
ISO2022	Code page 932 or 942 to Japanese ISO code page 2022 translation.
JIS	Code page 932 or 942 to Japanese ISO code page 2022 translation.
KSC	Code page 949 to Korean KS code (KSC 5601) (8-bit) translation.
WIN1250	Code page 852 to MicroSoft Windows code page 1250 translation.
WIN1251	Code page 855 to MicroSoft Windows code page 1251 translation.
	Code page 866 to MicroSoft Windows code page 1251 translation.
	Code page 915 to MicroSoft Windows code page 1251 translation.
WIN1253	Code page 813 to MicroSoft Windows code page 1253 translation.
WIN1256	Code page 864 to MicroSoft Windows code page 1256 translation.
USER	A user-created translation table is used.

Notes:

1. The -cp parameter is not case sensitive. For example, both ISO1 and iso1 are valid choices.
2. For a particular country, some of these listed values for -cp parameters might be unavailable. You should refer to each country's OS/2 Warp specification.

Normally, the default is for no translation to be done; so if you accept the default, no code page translation table needs to be specified. But for marketing reasons in some countries, the default is to use a specified code page translation table. For example, the default in Taiwan is to specify the code page 950 to BIG5 translation, so you just enter the application command without the -cp parameter. The following is an example of using the Telnet application in Taiwan:

```
telnet rs6ktw3
```

The Telnet application runs as follows:

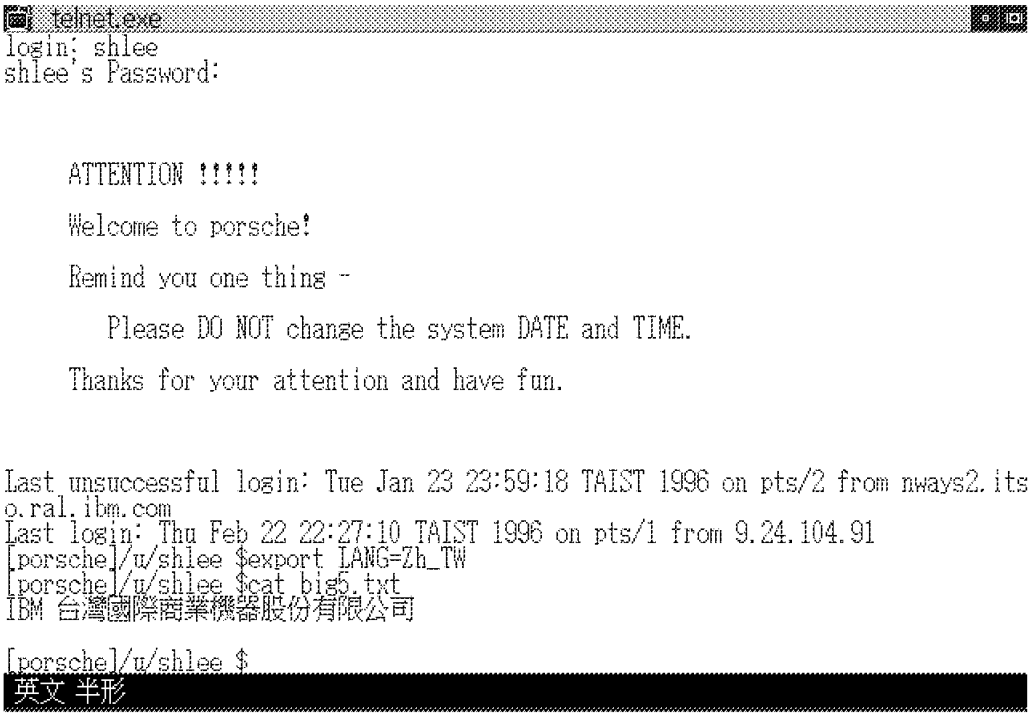


Figure 14. DBCS Telnet

For more examples of TelnetPM, FTP-PM, and NewsReader/2 refer to Figure 15, Figure 16 on page 43, and Figure 17 on page 43

Figure 15. DBCS TelnetPM

Figure 16. DBCS FTP-PM

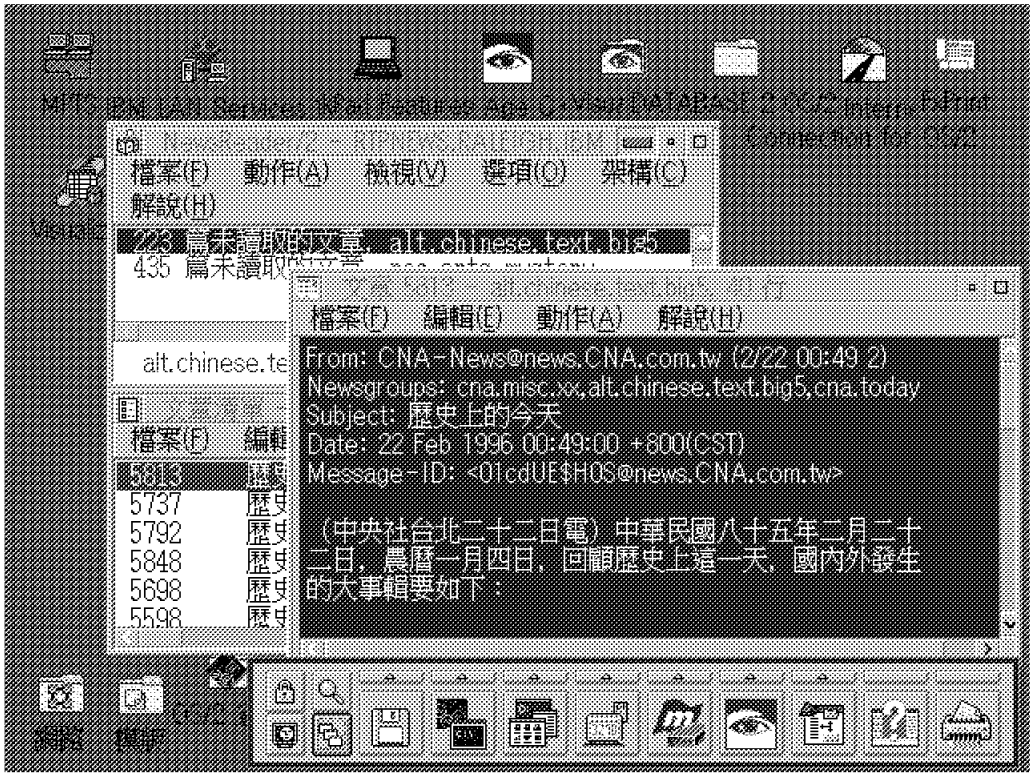


Figure 17. DBCS NewsReader/2

4.1.3 DBCS Outline Font Setting for WebExplorer

Basically, for improving performance and removing restrictions on DBCS systems, WebExplorer by default uses the MINCHO bit-mapped font for text display. Nevertheless, the MINCHO bit-mapped font set contains only several fixed-size fonts. Thus, sometimes, the text can not be displayed if the font sizes are not supported by the MINCHO font set at that screen resolution.

The following is a way of setting up WebExplorer to use one DBCS outline font so that all sizes of font can be displayed correctly.

1. Install a DBCS outline font for OS/2. If your OS/2 Warp has DBCS Windows, it can use the DBCS TrueType font that comes with Windows. Just register the font to OS/2 Warp by using the OS/2 Font Palette.

   Note: DBCS OS/2 Warp contains a TrueType font driver.

   The following shows the OS/2 Font Palette:

Figure 18. OS/2 Font Palette

2. Execute WebExplorer at least once and end it to create the initialization file EXPLORE.INI under your ETC subdirectory.
3. Edit the initialization file EXPLORE.INI. To specify the proper DBCS outline font to be used with WebExplorer, edit the following two lines:

   ```
   DBCS_PrimaryFont= <your DBCS outline font name>
   DBCS_SecondaryFont= <your DBCS outline font name>
   ```

4. Execute WebExplorer again. WebExplorer will still try to use MINCHO to display all text. But if font sizes cannot be displayed by the MINCHO font, WebExplorer will use the DBCS outline font that is specified in the EXPLORE.INI file.

The following is an example of the EXPLORE.INI file:

```
OS/2 視窗

; Web Explorer INI file
; (edit this file with care)

[screen]
xleft=4
ybottom=0
width=640
height=480
fontfamily=MINCHO
fontsize=Normal
DBCS_PrimaryFont=細明體
DBCS_SecondaryFont=細明體
textcolor=black
linkcolor=blue
visitcolor=darkpink
backcolor=palegray

[cache]
CacheOn=Yes
CacheMem=Yes
cachedocs=16
cacheimages=32

--- 尚有 ---
英文 半形
```

Figure 19. EXPLORE.INI for DBCS WebExplorer

The following shows the running WebExplorer:

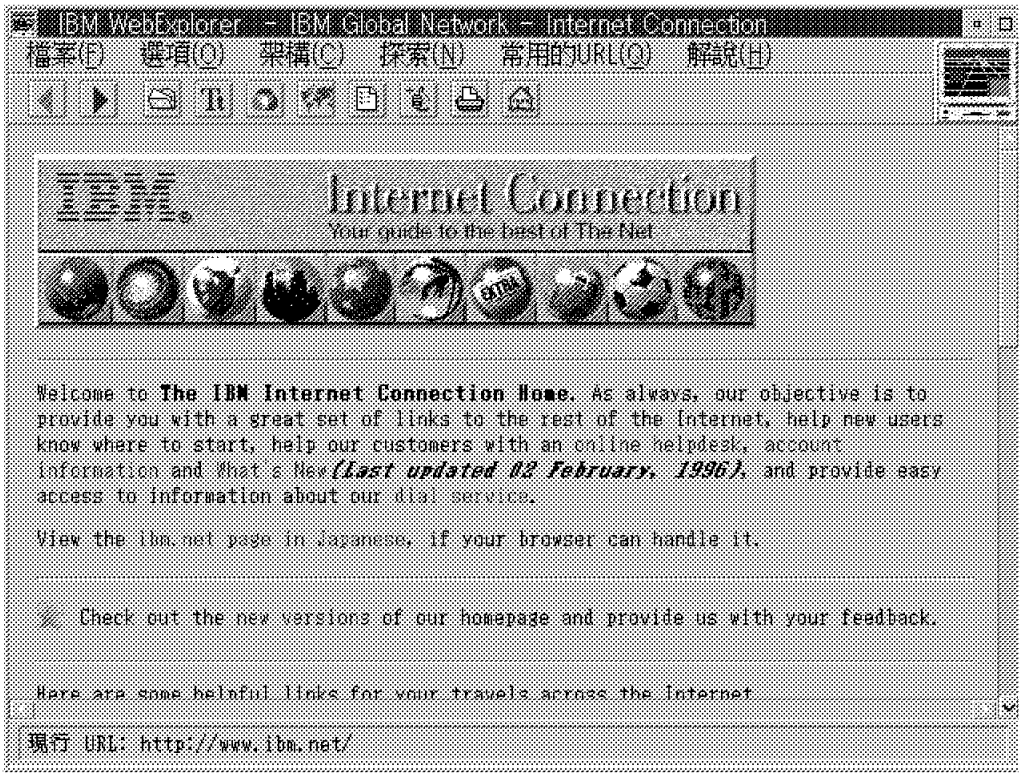


Figure 20. DBCS WebExplorer

Chapter 5. Dynamic IP

The main focus of this chapter is the installation and setup of Dynamic Host Configuration Protocol/Dynamic Domain Name Services (DHCP/DDNS) Clients and Servers. As Dynamic IP is the major new enhancement of TCP/IP V3.1 for OS/2 and may be new to the majority of TCP/IP users and system administrators, we first want to give a short, general introduction to Dynamic IP.

The second part of this chapter describes the concepts, protocols and configuration of DHCP and DDNS. The different scenarios will provide you with a deeper understanding of the interactions between of client and server and DHCP and DDNS.

5.1 Introduction

This section describes the purpose of dynamic IP and the benefits that can be derived from it. The idea behind Dynamic IP is to make the use and configuration of TCP/IP easier and more flexible for the end user and the administrator.

Before Dynamic IP, adding a new workstation to an IP network required the provision of several parameters and information to configure the TCP/IP software. Network components, such as a domain name server, were also required. The administrator of the IP network was responsible for managing IP addresses and hostnames and users had to apply for both.

A new TCP/IP host will normally require the following information:

- IP address
- IP subnet mask
- Default router address
- Local hostname
- Domain name
- Name server address

Additional parameters, such as other server addresses, time zones or protocol-specific configurations, may be necessary in some cases.

Keeping track of that information in a large TCP/IP network may not always be an easy task for network administrators, especially if users or machines, or both, change their location frequently. IP address lists and domain name-server databases have to be updated manually in order to keep track of any changes in the network.

From a user's point of view, a system administrator would have to be called to provide the necessary information in order to install a TCP/IP system. If the user moves to another location, this information must not be taken; the user will have to be assigned at least a new IP address, if not a new hostname as well. Users, particularly mobile ones, could potentially cause chaos in a TCP/IP network.

Even if workstations will be automatically installed using software distribution techniques, the TCP/IP configuration parameters have to be pre-assigned per distribution client.

The Bootstrap Protocol (BootP), as described in RFCs 951 and 1497, was introduced to the TCP/IP community in 1985 to provide automatic assignment of some TCP/IP configuration parameters to a new TCP/IP host. A table has to be maintained at BootP servers to enter information specific to any client that has been planned for installation. Typically, clients are identified by their LAN adapter′s hardware address, which has to be known to the system administrator in charge of a BootP server before he can prepare a new client entry in the database. Even though some manufacturers, nowadays, put the adapter hardware address on a label on the backplane of their LAN adapters, this ends up being a tedious process if many hosts have to be installed in a short period of time.

5.1.1 Benefits of Dynamic IP

To address the problems of manually updating centrally maintained information files and of a user manually configuring a TCP/IP workstation, the Dynamic Host Configuration Protocol (DHCP) has been developed. It is described in an IETF DHCP working group Internet draft and in RFCs 1533, 1534, 1541, and 1542. A DHCP server need not be preconfigured with a workstation′s LAN address in order to submit the necessary TCP/IP configuration to it.

With DHCP in place, the assignment of IP addresses and other configuration information has become a lot easier. But one problem still persists - how would a domain name server learn about those dynamically assigned IP addresses and hostnames so it can update its database accordingly? This can be solved by using the Dynamic Domain Name Services (DDNS) as proposed by an IETF DNSIND working group Internet draft.

Having DHCP and DDNS available gives system administrators the advantage of a high degree of flexibility and automation, and users do not have to worry about TCP/IP configuration parameters anymore. People in charge of information technology investment budgets may also prefer to spend their money on open standards which will give them the assurance that products from different vendors will coexist in their TCP/IP networks.

IBM is actively participating in the design and implementation of DHCP and DDNS, and it has coined the term *Dynamic IP*. To summarize, the objectives of Dynamic IP and its benefits to TCP/IP system administrators and users are as follows:

- Provides automatic IP network access and host configuration
- Simplifies IP network administration
- Leverages existing IP network products and infrastructure
- Employs only open standards
- Allows customers to administer site-specific host environments
- Enables customized, location-sensitive host serving

The following gives an example of using Dynamic IP with a mobile Lotus Notes client.

5.1.1.1 Example: Dynamic IP and Lotus Notes

Before Dynamic IP the usage of Lotus Notes as a mobile system was limited or too inflexible to use. Lotus Notes gives you the ability to store local copies of the databases that you need to access, or to hold your outgoing mail in a special local database. What makes it mobile is that you carry your data with you. The next time a connection to your Notes server is established, Lotus Notes will update all the information between your system and the Notes server. This is a good concept, but what happens if you don't have the disk space to hold local copies of databases or if your mail should be exchanged immediately? In this case you need a connection to your home server.

In order to establish a connection to your home server from a foreign location, you have to reconfigure your Lotus Notes client. You have to call the administrator of the IP network and apply for a temporary IP address. With that information you have to reconfigure TCP/IP and reboot your system. This procedure makes it really difficult and inflexible to use Lotus Notes as a mobile system.

Now, with Dynamic IP, that will change. You can set up your Lotus Notes client at your notebook using Dynamic IP. Every time you connect to a LAN with your notebook, a new IP configuration will be received from a DHCP server. That gives you the ability to access the IP network without contacting the network administrator. You can access other resources on the network immediately. In order to be known to other users, you can register your host with its usual hostname at a dynamic domain name server. All that will happen automatically when the DHCP and DDNS client is set up on your Notes system.

Dynamic IP is a value add for Lotus Notes and makes it a real mobile system. Lotus Notes is only one product of many that will profit from Dynamic IP. Dynamic IP will make it easy and transparent for the end user to change the location.

5.1.2 System Components

The following table gives a brief description of the different types of network components that comprise Dynamic IP:

Table 5 (Page 1 of 2). Dynamic IP Components in TCP/IP 3.1

System Component	Description
DHCP Server	DHCP servers provide the addresses and configuration information to DHCP and BootP clients on the network. DHCP servers contain information about the network configuration and about host operational parameters, as specified by the network administrator.
DHCP Client	DHCP clients receive their configuration from a DHCP server. TCP/IP will be configured automatically after receiving the information.
DDNS Server	Dynamic DNS servers are a superset of today's static DNS BIND servers. The dynamic enhancements enable client hosts to dynamically register their name and address mappings in the DNS tables directly, rather than having an administrator manually perform the updates. The DDNS server also supports static hostnames that can be entered manually by the administrator.

Table 5 (Page 2 of 2). Dynamic IP Components in TCP/IP 3.1	
System Component	**Description**
DDNS Client	IP Hosts (whether they are dynamic or static) can register their IP address and a hostname with the DDNS server.
BootP Relay Agents (or BootP Helpers)	BootP relay agents may be used in IP router products to pass information between DHCP clients and servers. BootP relays eliminate the need for having a DHCP server on each subnet to service broadcast requests from DHCP clients.

5.2 The Dynamic Host Configuration Protocol (DHCP)

The Dynamic Host Configuration Protocol (DHCP) provides a framework for passing configuration information to hosts on a TCP/IP network. DHCP is based on the Bootstrap Protocol (BootP), adding the capability of automatic allocation of reusable network addresses and additional configuration options. DHCP captures the behavior of BootP relay agents, and DHCP participants can interoperate with BootP participants.

In contrast to BootP, DHCP offers the ability to assign an IP address to a client for a limited amount of time, and it also offers a way to supply all required configuration parameters for a client. This is not possible with BootP.

Before a DHCP client is able to communicate over the IP network, the client has to receive an IP address from an IP address server (DHCP server). The DHCP server manages a pool of IP addresses. If an IP address from that pool is available the server will assign an IP address to the client for a limited time. It is said that the IP address is leased by the client for that period. Depending on the configuration of the DHCP client, the client can receive multiple configuration parameters from the DHCP server (for example router and name server addresses). Before the lease time of the IP address has expired, the client has to renew the lease. The client sends a request for renewing the lease to the DHCP server. If the server renews the client′s lease the client can still use the IP address. Otherwise, the client has to stop using the assigned IP address.

The following sections provide a brief outline of the DHCP client and server protocol. The communication flow between client and server is explained, while requesting an IP address and renewing a lease.

5.2.1 Requesting an IP Address

This section describes the initial interaction between DHCP clients and servers. If a client uses multiple IP interfaces, each of them must be configured by DHCP separately.

- When a client host is started and the DHCP client is initialized for the first time, the client will broadcast a DHCPDISCOVER message on the network, sending it to UDP port 67, the BootP server′s well-known port. The client itself uses UDP port 68, the BootP client′s well-known port. Using these ports, and also using the BootP message format as explained later, will ensure that a DHCP server can service both DHCP and BootP clients. The client at this stage is said to be in *INIT* state.

- If a DHCP server is not located on the same IP subnet as the client, an intermediate IP router may act as a BootP relay agent and forward any DHCP and BootP messages to a DHCP server (or to another intermediate IP router that has the same capability). In this case, the router will insert its own IP address from the subnet on which the client is located so that any DHCP servers can decide if they have an appropriate IP address on offer for that particular client request.
- In order to be able to send initial DHCP broadcast messages, a DHCP client configures its IP interface(s) with an address of 0.0.0.0 and sends the broadcast to IP address 255.255.255.255.

  In order to receive DHCP reply messages at a client whose IP stack has not been configured, the TCP/IP implementation at the client must be able to pass on IP packets that are sent to the client′s hardware address to the IP layer in that system. Otherwise, DHCP servers (and eventually involved BootP relay agents) must use broadcast frames to submit their information to the client. A client will indicate its ability to receive unicast datagrams rather than broadcast by not setting the broadcast bit in the flags field of a DHCP message.
- DHCP servers that receive DHCPDISCOVER messages will respond with a DHCPOFFER message if they have any IP addresses available. If no addresses are available at a server, it will not respond at all. A DHCP server will include an available IP address and other options in that message. Servers may also check if an offered IP address is not already in use. They can do so using an ICMP echo request (PING). Servers may also temporarily reserve any offered IP address so that they will not be offered to several DHCP clients at the same time.
- A client may receive several DHCPOFFER messages from a number of DHCP servers, and it is up to the implementation of the client software to decide which server′s offer the client should finally decide to accept. If a server has been selected, the client broadcasts a DHCPREQUEST message to that server whose IP address is contained in the server identifier option from the previous DHCPOFFER message.
- The server that receives a DHCPREQUEST message from a client will finally commit the requested IP address and optimal parameters to its configuration and acknowledge that to the client by sending a DHCPACK message. If that server at that time cannot, for whatever reason, supply any of the requested configuration parameters, it will send a DHCPNACK message instead. The client will then have to repeat the whole acquisition process, starting with a DHCPDISCOVER message.
- After receiving DHCPACK, the client should also check if the offered IP address is already in use. This can be done using ARP rather than PING since, at that time, the client has no IP host address it can use. If the offered address is already in use, the client responds with a DHCPDECLINE message to the server; otherwise it will configure its IP interface(s) according to the values obtained from the DHCP server. The client is now fully configured or in the *BOUND* state.
- After sending a DHCPDECLINE message, the client must restart the whole acquisition process, starting with a DHCPDISCOVER message. The server, in this case, must mark that address as not available, and it may notify the administrator with an error message.

- If a client does not receive any DHCPOFFER messages, it will continue to broadcast DHCPDISCOVER messages at random intervals for a certain period of time before it will notify the user with an error message that it could not obtain any TCP/IP configuration parameters.
- When a client no longer needs a given TCP/IP configuration, it may inform the server by using a DHCPRELEASE message. The server will then mark the IP address as available. This message will not be acknowledged by the server.

5.2.2 Renewing a Lease

This section describes the interaction between DHCP clients that have already been configured, and servers. If a client uses multiple IP interfaces, each of them must be configured separately by DHCP.

- After a DHCP client has applied the TCP/IP configuration parameters that it has obtained from a DHCP server, it has also received a lease time during which the client is rightfully entitled to use the given configuration. Two timers, T1 and T2, will start to tick down. While T1 will expire before T2, T2 will expire before the end of the assigned lease time. According to the latest IETF Internet draft, T1 defaults to (0.5 x lease time), and T2 defaults to (0.875 x lease time), but either timer can be set by the server through DHCP options.
- When timer T1 expires, the client will send a DHCPREQUEST message to the server asking to extend the lease for the given configuration. This state of a client is called the *RENEWING* state. The server would usually respond with a DHCPACK message indicating the new lease time to which T1 and T2 will then be reset accordingly.
- If no DHCPACK is received until timer T2 expires, the client enters the *REBINDING* state. It now has to broadcast a DHCPREQUEST message to extend its lease. This request can be confirmed by a DHCPACK message from any DHCP server on the network.
- If the client does not receive a DHCPACK message after its lease has expired, it has to stop using its current TCP/IP configuration and may start over from INIT state as described earlier.
- If a client has been configured before and is rebooted, it may want to use the previous configuration values, which may have been stored in a file on the client′s hard disk. In that case, the client would broadcast a DHCPREQUEST message containing the desired parameters in the appropriate option fields. DHCP servers will respond with DHCPACK messages if they can supply the requested configuration. If no DHCPACK messages are received by the client, it may wait and then start over from INIT state as described earlier.
- If a client is using external configuration values (external to DHCP), which it may have obtained through manual configuration, it would assemble a DHCPINFORM message containing its current configuration and any additionally desired parameters. If the client knows a DHCP server′s IP address, it will send this message to that address; otherwise it will broadcast the message. A server will respond to that request using a DHCPACK message which only contains the additionally required options for the client. If the client does not receive any replies, it should notify the user of that problem and continue operation using suitable defaults.

The following figure shows the state transition diagram for a DHCP client to illustrate the previous descriptions. Start reading the diagram from the INIT box. Broken lines indicate broadcast messages.

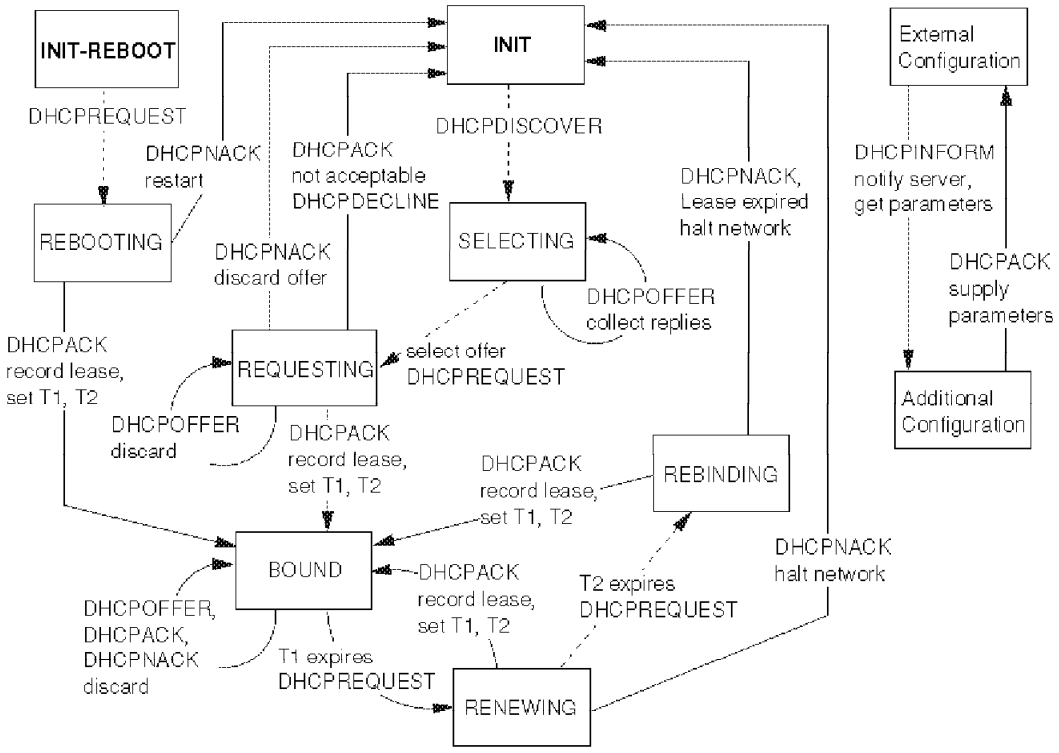


Figure 21. DHCP Client State Transition Diagram

5.2.3 DHCP Message Types and Message Format

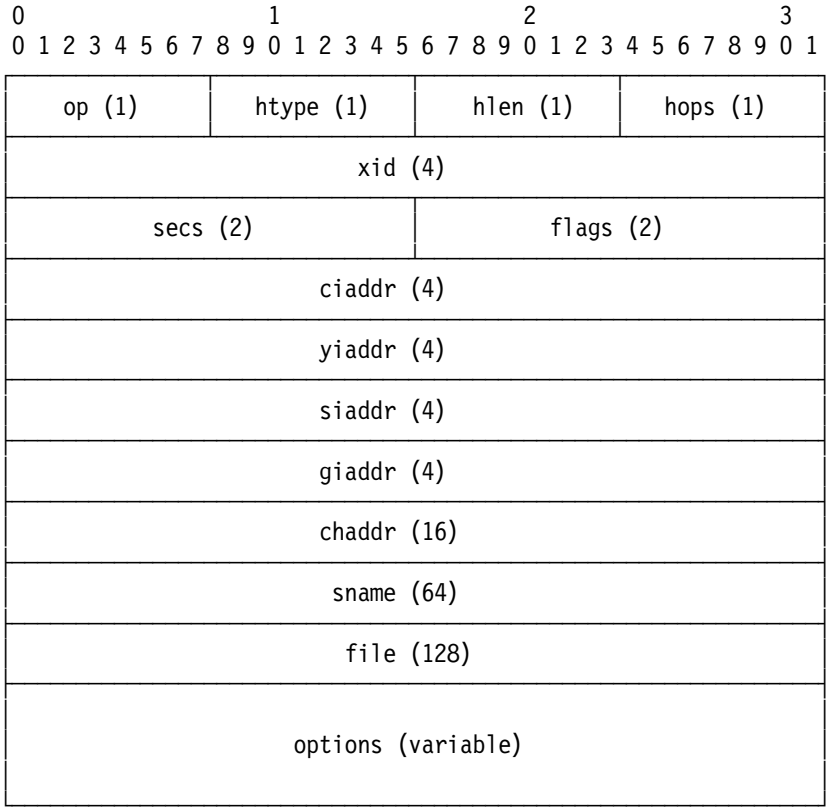
The following table lists the types of messages that can flow between DHCP client and DHCP server systems.

Table 6 (Page 1 of 2). DHCP Message Types

Message	Direction	Use
DHCPDISCOVER	Client to server	Locate available servers
DHCPOFFER	Server to client	Offer available configuration parameters
DHCPREQUEST	Client to server	1. Request offered parameters 2. Confirm correctness of previously allocated address 3. Extend lease on particular address
DHCPACK	Server to client	Commit requested address

Table 6 (Page 2 of 2). DHCP Message Types		
Message	**Direction**	**Use**
DHCPNACK	Server to client	1. Requested address cannot be supplied 2. Requested address is invalid because client has moved to a different IP subnet 3. Lease for an address has expired
DHCPDECLINE	Client to server	Address is already in use
DHCPRELEASE	Client to server	Address is no longer needed; current lease will be cancelled
DHCPINFORM	Client to server	Asking for additional parameters only; address already configured

DHCP uses the BootP message format, as defined in RFC 951, which is shown in the diagram below:



The next table explains the fields used within a DHCP message.

Table 7. DHCP Message Fields

Field	Number of Bytes	Description
op	1	Message operation code; will be 1 for BOOTREQUEST in messages sent from a client to a DHCP server and 2 for BOOTREPLY in messages sent from a DHCP server to clients.
htype	1	Hardware address type as used by ARP: for instance 6 for token-ring.
hlen	1	Hardware address length: for instance 6 for token-ring.
hops	1	Set to 0 by clients. This field may optionally be used by BootP relay agents if client and server are not on the same IP subnet.
xid	4	Transaction ID: a random number chosen by the client. This field is used by clients and servers to associate messages and responses between a client and a server.
secs	2	This field is filled in by the client. It represents the number of seconds that have elapsed since the client began the address acquisition or the lease renewal process.
flags	2	Only broadcast flag used to determine if client is able to accept IP unicast datagrams.
ciaddr	4	Client IP address. Filled in by the client. Set to 0 or client's IP address.
yiaddr	4	"Your" (client) IP address. Filled in by the DHCP server.
siaddr	4	Server IP address. Returned by the server in DHCPOFFER and DHCPACK messages.
giaddr	4	Gateway IP address. Inserted when a BootP relay agent is being used.
chaddr	16	Client hardware address. Filled in by the client.
sname	64	Server hostname. An optional field containing a null-terminated string.
file	128	BootP file name. Used when a DHCP server is employed to provide operating system startup files for BootP clients.
options	variable	Optional parameters. See explanations below.

Though the options field has a variable length, DHCP clients must be able to receive messages with an options field of a length of 312 bytes. This implies that a client must be configured to receive a message of 576 bytes, which is the minimum IP datagram size that a client must be prepared to accept anyway.

A DHCP server may also use the server host name and/or file fields to transmit additional DHCP options. It will then inform a client about this by coding a special option. The client will then evaluate those fields after it has gone through the regular options.

DHCP options are grouped by categories, as shown in the following table.

Table 8. DHCP Options

Group	Range	Description
Base options	1-18	BootP vendor extensions as defined in RFC 1497.
IP layer parameters per host	19-25	Options that affect the operation of the IP layer on a per-host basis.
IP layer parameters per interface	26-33	Options that affect the operation of the IP layer on a per-interface basis. Multiple requests should be possible to configure multiple interfaces separately.
Link layer parameters per interface	34-36	Options that affect the operation of the data link layer on a per-interface basis.
TCP parameters	37-39	Options that affect the operation of the TCP layer on a per-interface basis.
Application and service parameters	40-49	Options to configure miscellaneous applications and services.
DHCP extensions	50-61, 77	Options that are specific to DHCP.
Application and service extensions	64-76	Additional options to configure miscellaneous applications and services.
User-defined extensions	78-127	Reserved for future use.
Site-specific options	129-253	Options used for experimental usage or to provide site-specific configuration parameters.

Options 128 and 254 are reserved. Additional options may be registered with the Internet Assigned Numbers Authority (IANA), by sending E-mail to iana@isi.edu. The first four bytes in the options field should always be hexadecimal 63.82.53.63, the ″magic cookie″ as mentioned in RFC 951.

5.3 OS/2 DHCP Server Configuration and Administration

Product differentiation and the value of DHCP server products lie in their ease of use and in the flexibility in setting up policies that can be made available to administrators. The IBM OS/2 DHCP server includes a graphical configuration program that facilitates the creation and maintenance of the DHCP server database. The following programs come with the DHCP server for configuration and administration and are discussed in more detail later.

Table 9 (Page 1 of 2). DHCP Server Programs

Program	Description
\MPTN\BIN\DHCPSCFG.EXE	Graphical server configuration program
\TCPIP\BIN\DSTAT.EXE	Shows the status of the DHCP server

Table 9 (Page 2 of 2). DHCP Server Programs	
Program	**Description**
\TCPIP\BIN\DINIT.EXE	Reinitialization of the DHCP server
\TCPIP\BIN\DADMIN.EXE	Administration tool. Delete leases, reinitialize server and view server status
\TCPIP\BIN\DHCPSD.EXE	DHCP server

To set up the DHCP server, different files have to be created or will be created by the graphical configuration program. The following table shows all files, that are important to run and administer the server properly:

Table 10. DHCP Server Files		
Program	**Created by ...**	**Description**
\MPTN\ETC\DHCPSD.CFG	Administrator or server configuration program (DHCPSCFG.EXE)	Contains the configuration data of the server.
\MPTN\ETC\DHCPSD.DAT	Server configuration program (DHCPSCFG)	Contains the private key of the DHCP server for authentication with the dynamic domain name server.
\MPTN\ETC\DHCPS.AR	DHCP server	Stores status of server configuration.
\MPTN\ETC\DHCPS.CR	DHCP server	Stores status of server configuration.
\DHCPSD.LOG	DHCP server	Log file of the server. Different logs can be enabled in the server configuration.

The DHCP server administrator has different options to set up the server:

- Flexibility to configure hosts individually, based on a designated ″class″ or based on their location in the network.
- Configure site-specific applications on client hosts with information defined centrally at DHCP servers, effectively extending and customizing the Dynamic IP system to serve the needs of the enterprise.
- Use vendor-specific options to supply specific configuration parameters to clients from different vendors.

There are three ways of supporting clients with the OS/2 DHCP server:

1. Dynamic
2. Automatic
3. Manual

When used dynamically, the DHCP server assigns IP addresses from an address pool for a limited period of time (leased). The client must then periodically renew its lease of an IP address, but this is done automatically without user or administrator intervention.

When used automatically, the DHCP server assigns IP addresses from an address pool for an unlimited period of time (permanent).

When used manually, the DHCP server assigns a specific, predefined address to a specific client. This type of IP address assignment can be used to support BootP clients with the DHCP server.

The easiest way of creating the DHCP server configuration file dhcpsd.cfg is to use the graphical configuration program. Changes to the configuration file can easily be made with an OS/2 editor.

The IBM DHCP Server provides configuration information to clients based on statements contained in the server's configuration file and based on information provided by the client. The server's configuration file defines the policy for allocating IP addresses and other configuration parameters. It is a "map" that the server uses to determine what information should be provided to the requesting client. You must create the server configuration file before you start your DHCP server.

You can use the DHCP Server Configuration utility, DHCPSCFG, to create configuration files for your IBM DHCP Servers and, optionally, to create the Dynamic DNS data files. While you can use this utility to create configuration files for other servers, you must then move the configuration files to the appropriate server. This tool does not remotely configure or control DHCP servers.

The following figure shows the DHCP Server Services folder from which the DHCP server program and the DHCP server configuration program can be started.

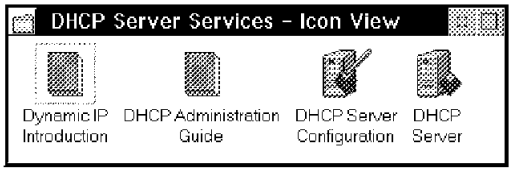


Figure 22. DHCP Server Services Folder

To start the DHCP server configuration program, double-click on the appropriate icon in the DHCP Server Services folder. The configuration program offers you a graphical interface to administer your DHCP server parameters. The following figure shows the DHCP server configuration program load with the configuration file DHCPSD.CFG.

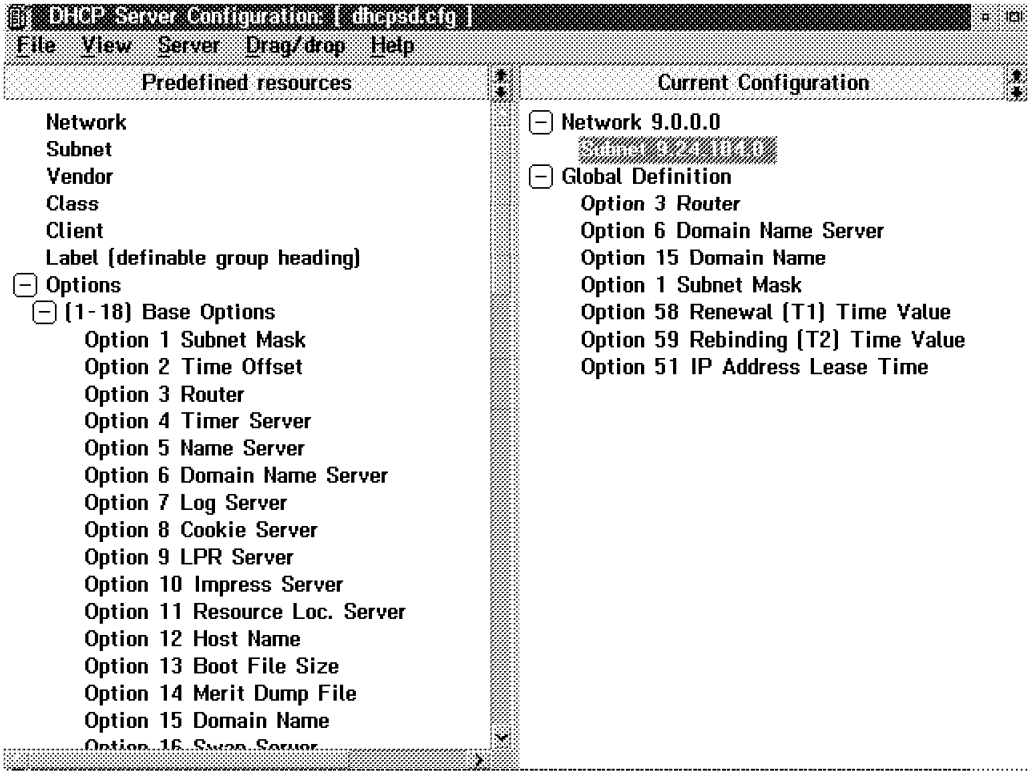


Figure 23. DHCP Server Configuration Panel

On the left side of the configuration program, the Predefined resources window is displayed. Items that can have a set of definitions are prefixed with a plus sign (+). Click there to expand any item to reveal parameters located one level below.

On the right side of the configuration program, you can see the Current Configuration window. To add items, select the appropriate parameter from the Predefined resources window, then click on it with the right mouse button. Hold down the right mouse button, and drag the item from the left side to the right side of the configuration program, then drop it onto the current configuration by releasing the right mouse button.

To remove an item from the Current Configuration window, simply drag the item to the OS/2 shredder and drop it there.

Because the DHCP server respects the position of information in the configuration file, you can create a hierarchy of configuration parameters by nesting items within the DHCP configuration. This allows you to specify that some configuration values are to be served to all clients, while other configuration values are to be served only to certain clients.

Parameters (options) specified on the first level are considered global and apply to all clients. Parameters (options) specified under a conditional statement, such as a Network statement, are considered local and apply only to clients that meet the criteria of the conditional statement. Conditional statements can also be nested. If a parameter is specified in more than one place, the lowest level statement (which is the most specific) is used.

Only the items options and vendor are used to specify the parameters that are passed from the server to the client. All other items are conditional statements and specify which and how DHCP clients are served by a DHCP server. Usually a client that applies for an IP address is a member of a group that is handled by the DHCP server in the same way. A group is specified by the item′s network, subnet, class or client. A group can also consist of only one member, which means that you can define options for single clients.

You can specify, for example, that all clients within the network 9.0.0.0 and the subnet 9.24.104.0 will be assigned an IP address from the IP address pool 9.24.104.20 to 9.24.104.165. Generally you can specify that the options 3, 6,15,1, 58,59 and 51 should be passed to a client. The options hold information about router, domain name server, domain name and so on for the client. In Figure 23 on page 59 you can see how the items are arranged to build a group for the specified network. Global Definitions is only a label to structure the configuration. Placing all options on the same level as Network would have the same effect.

The following table summarizes the configuration parameters of the Predefined resources window and describes their purposes.

Table 11. DHCP Server Configuration - Predefined Resources Window

Configuration Item	**Configuration Data**
Network	The network statement specifies one network that is administered by a server. A network starts at a base IP network address and may consist of one or more subnets or a range of IP addresses. There may be multiple network statements indicating that a server will control more than one network.
Subnet	The subnet statement specifies one subnet under a network statement. A subnet starts at a base IP subnet address and may include all IP addresses of that subnet or only a specified range of addresses. There may be multiple subnet statements under a network statement.
Class	A specification for a set of clients. May include a range of IP addresses and a set of options. DHCP clients which request this class will be given the specified options and valid addresses. This configuration can be used to group clients according to business organization.
Client	A specific definition for a client. May be used to serve clients individually, to exclude clients from participating in DHCP, or to serve BootP client requests.
Label	A comment that will be inserted in the configuration file to make it more readable.
Vendor	A specific set of configuration parameters to be used with a client from a certain vendor.
Options	Any of the DHCP options and the values that will be served to DHCP and BootP clients, as appropriate.

To configure the values of the items you have dragged and dropped to the current configuration you can double-click on the item. A configuration window for the specific item will appear.

Note: You can only double-click on an item in the Current Configuration window, not in the Predefined resources window.

Figure 24 shows the network configuration of the DHCP server from the above example.

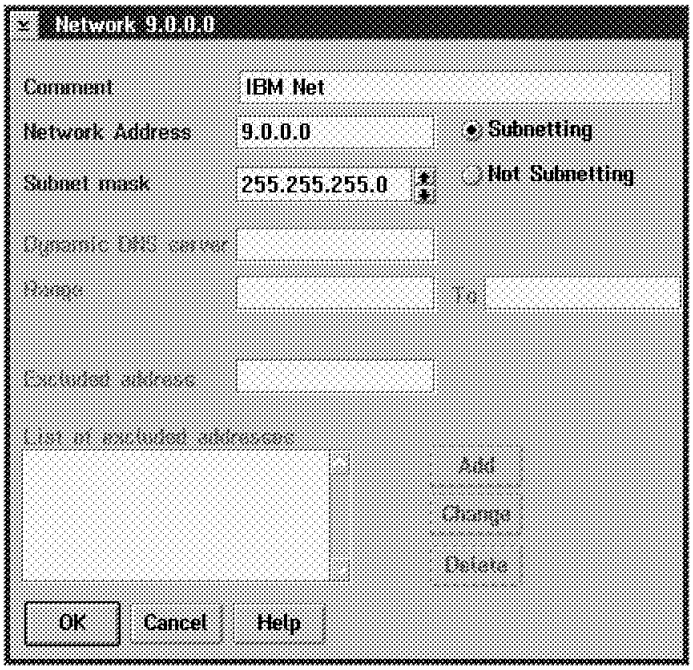


Figure 24. DHCP Network Configuration

The following table summarizes the network configuration parameters and describes their purposes.

Table 12 (Page 1 of 2). DHCP Server Configuration- Network Menu

Configuration Item	Configuration Data
Comment	Specify a descriptive comment for this network.
Network Address	Enter the base IP address for this network. You should always enter a base IP address here. Network addresses must be of the form 9.0.0.0 (class A), 129.32.0.0 (class B), or 199.17.21.0 (class C).
Subnet mask	If you clicked on the Subnetting button, enter the subnet mask for this network here. The DHCP server will then use all possible IP host addresses for the given network and subnet mask combination. You cannot specify a subnet mask if you clicked on the Not Subnetting button. In that case, you have to specify a range of IP addresses to be used by the DHCP server. **Note:** When you use subnetting, you cannot specify a DDNS server and IP addresses to be excluded on the network menu. Those parameters must be configured on the respective subnet menus.
Dynamic DNS server	Enter the IP address of a DDNS server that will be updated by this DHCP server with inverse name resolution information.

Table 12 (Page 2 of 2). DHCP Server Configuration- Network Menu	
Configuration Item	**Configuration Data**
Range	If you clicked the Not Subnetting button, specify the range of IP addresses, within this network, to be used by the DHCP server. The DHCP server will then use only IP host addresses that are within the specified range. You cannot specify a range if you clicked on the Subnetting button. In that case, you have to specify a subnet mask for this network.
Excluded address	Specify any IP addresses that you want to exclude from the specified subnet or range. Typically, this will be addresses of routers and servers, such as primary DDNS servers. The DHCP server will reserve those addresses and will not lease them to clients.

Specifying the network as subnetted, you have to configure a subnet item. Therefore simply drag the subnet item from the predefined resources window to the Current configuration window and drop it onto the Network item. To configure the item, double-click it.

In the subnet configuration the subnet address and the IP address pool must be defined. All other parameters are optional. Figure 25 shows the configuration of the subnet for the above example:

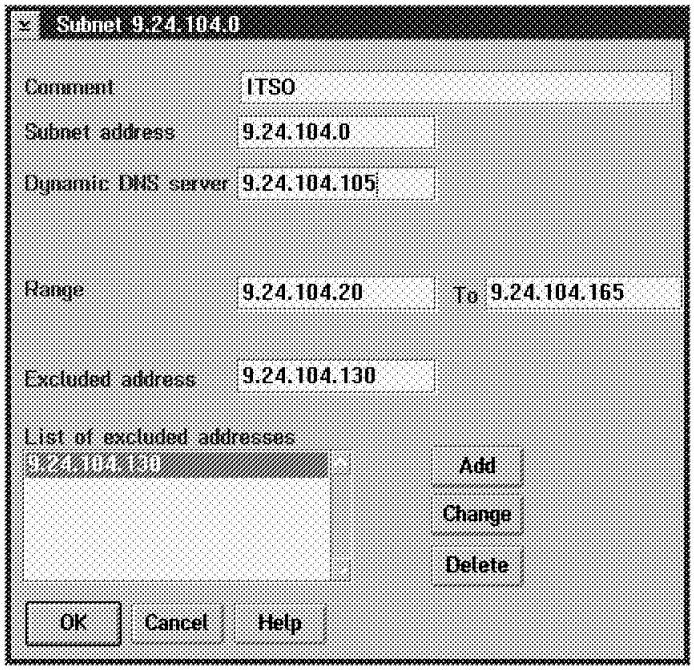


Figure 25. DHCP Server Subnet Configuration

The only difference compared to the network configuration panel is the field subnet address. Here you specify the IP address of the subnet that you want to define. Only to clients of that subnet will an IP address from the IP address pool be assigned.

You can configure the Class, Client, Label, Vendor and Options menus in a similar way. To use, for instance, a Label for global definitions simply drag the

label item from the Predefined resources window and drop it on the Current configuration window. This process will create a tree of configuration items.

In Options, the parameters for the clients are defined. Figure 26 shows the definition of the T1 parameter for a client. The T1 parameter specifies the renewal interval of the client's IP address lease.

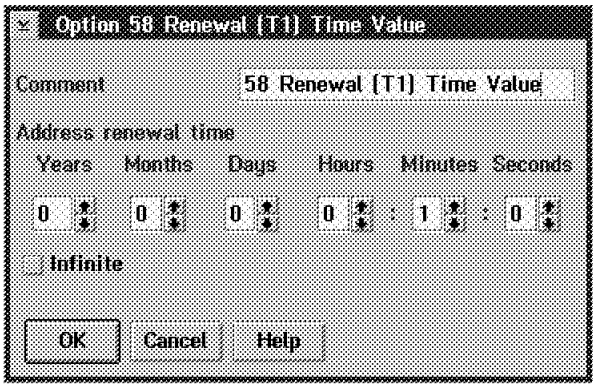


Figure 26. DHCP T1 Option Configuration Panel

The scope of an option covers the configuration item where it is specified, for instance a network, and all items below that. Options that are specified outside any item have a global scope.

As you can see in Figure 27, apart from the Predefined resources window in the DHCP Server Configuration panel, there is a User-defined resources window from which to drag items to a configuration. For either side of the configuration program, there is a Scratch pad window for testing. The User-defined resources window and the Scratch pads can be accessed by clicking on the up or down arrows in the windows on top of either side.

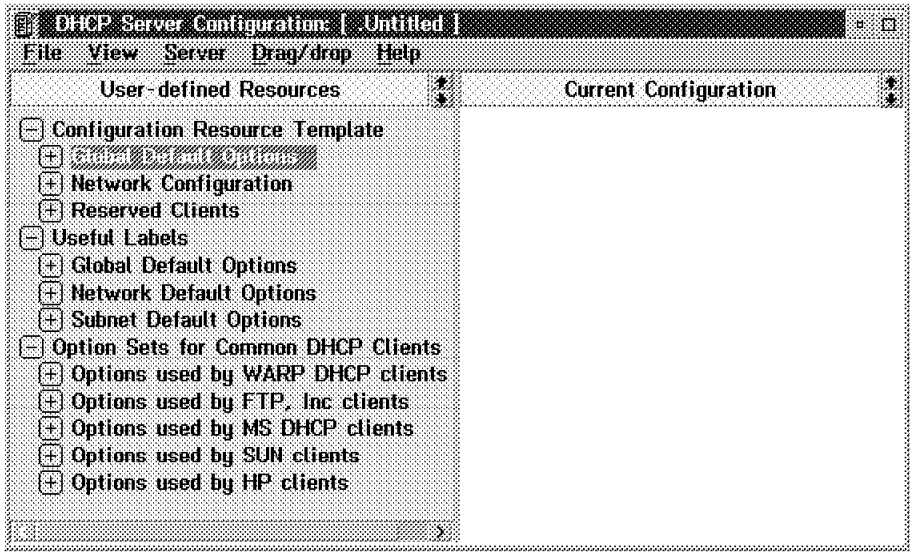


Figure 27. DHCP User Defined Resources

After creating this hierarchy of items in the DHCP Server Configuration panel (as shown in Figure 23 on page 59) and specifying the parameters that the server

should pass to its clients, you have to set up some additional parameters concerning the DHCP server.

Select **Server** and then **View/change server parameters** from the server configuration menu. In the Server Parameter panel you can specify the lease time for IP addresses. The default lease time is the time that a client is allowed to hold an IP address. After that time, the IP address is invalid if it is not renewed. The lease time expire-interval specifies the time interval for the DHCP server to check if leases have expired or are still valid. The time specified in these fields can reach from seconds to years.

The OS/2 DHCP server can be configured to log any activities and client requests, which is very helpful for problem determination and security. To activate logging, check the options you want to log from the Server pull-down menu in the configuration program shown in the following figure:

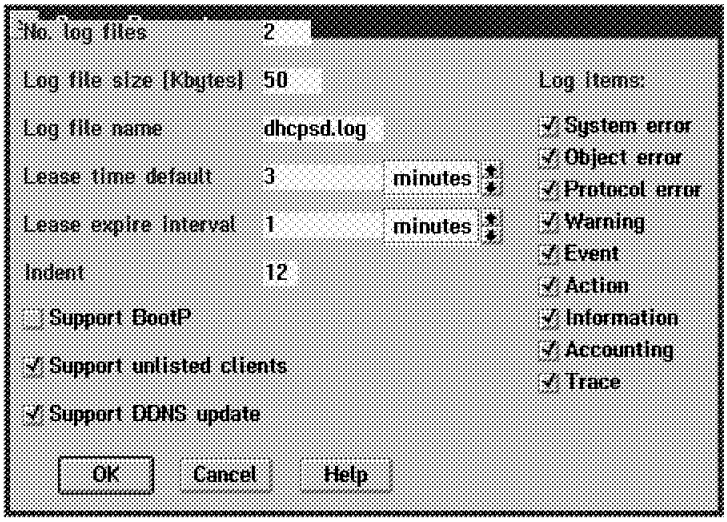


Figure 28. DHCP Server Parameters

The following table summarizes the server configuration parameters and describes their purposes.

Table 13 (Page 1 of 2). DHCP Server Configuration - Server Parameters

Configuration Item	Configuration Data
No. log files	Specify how many log files the DHCP server should maintain. The server will gradually fill up the log files and then continue by overwriting the oldest file.
Log file size (KBs)	Specify the maximum file size of any log file.
Log file name	Specify the name of the current log file. Completed log files will use the name with consecutive numbers as extensions.
Lease time default	Specify the default lease time for IP addresses.
Lease expire interval	Specify the time interval for the DHCP server to check if leases have expired or are still valid.
Indent	Specify the number in pixels that the DHCP server configuration program should use to indent items in the configuration tree.

Table 13 (Page 2 of 2). DHCP Server Configuration - Server Parameters	
Configuration Item	**Configuration Data**
Log items	Click on the type of information you want the server to write to the log file(s).
Support BOOTP	Click here, if you want to support BootP clients with this DHCP server.
Support unlisted clients	Click here, if you want to support DHCP clients in a dynamic way without having to configure specific information per client.
Support DDNS update	Click here, if you want the DHCP server to update a DDNS server with inverse hostname resolution information. The following statement in the DHCP server configuration file includes the command that is sent to the DDNS server to update PTR records for inverse mapping: `updateDNS "nsupdate -f -r%s -s"d;ptr;*;a;ptr;%s;s;%s;0;q""` The %s variables will be evaluated by the DHCP server as follows: 1. IP address 2. Fully qualified hostname 3. Lease time

In order to support DDNS updates, you have to select the respective option on the Server Parameters window. Then click on the **Update DDNS data file** option on the File menu of the DHCP server configuration program. This will create the DHCPSD.DAT file where information about the primary name server and the encryption key to be used in DDNS updates are stored. In order to actually enable the DDNS update function, you must merge the information from the DHCPSD.DAT file into the DDNS.DAT file that will be created by the DDNSZONE command when you configure the DDNS server. How to set up the DDNS server is explained later in this chapter.

When you have finished the DHCP server configuration, you can save the parameters to a file using the Save option from the File pull-down menu on the menu bar. By default, a DHCPSD.CFG file will be used by the DHCP server. This file will be searched in the directory where the ETC environment variable points to, normally the \MPTN\ETC subdirectory of the OS/2 boot drive.

The following example shows a DHCP server configuration file that has been created using the configuration program.

```
numLogFiles          2
logFileSize          50
logFileName          dhcpsd.log
leaseTimeDefault     3 minutes
leaseExpireInterval  1 minutes
supportBOOTP  no
supportUnlistedClients  yes
logItem     SYSERR
logItem     OBJERR
logItem     PROTERR
logItem     WARNING
logItem     EVENT
logItem     ACTION
logItem     INFO
logItem     ACNTING
logItem     TRACE
#.indent 12

updateDNS "nsupdate -f -r%s -s"d;ptr;*;a;ptr;%s;s;%s;0;q""
```

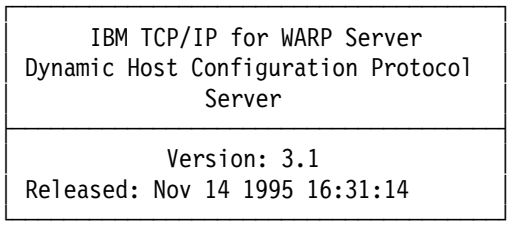
```
network     9.0.0.0 255.255.255.0      #.name IBM Net
{
    subnet     9.24.104.0 9.24.104.20-9.24.104.165    #.name ITSO
    {
        #.ddns 9.24.104.105
        client 0 0 9.24.104.105        #.exclu
    }
}
#.cat Global Definition              {
    option 3       9.24.104.1          #.name 3 Router
    option 6       9.24.104.105        #.name 6 Domain Name Server
    option 15      itso.ral.ibm.com    #.name 15 Domain Name
    option 1       255.255.255.0       #.name 1 Subnet Mask
    option 58      60                  #.name 58 Renewal (T1) Time Value
    option 59      120                 #.name 59 Rebinding (T2) Time Value
    option 51      180                 #.name 51 Lease Time
#.cat }
```

In the example above, the DHCP server controls IP addresses in the range from 9.24.104.20 to 9.24.104.165. It will send updates to the Dynamic DNS server 9.24.104.105, and it will therefore exclude this address from the list of addresses available to DHCP clients. Furthermore, DHCP options 1, 3, 6, 15, 51, 58 and 59 will be supplied to DHCP clients. In this example, BootP clients will not be supported by the DHCP server. All log files are enabled. The default lease time is three minutes and will be checked every minute. The client will be configured to renew its lease every minute and try to rebind after two minutes, when the renewing of the lease fails.

To start the OS/2 DHCP server, double-click on the appropriate icon in the DHCP Services folder. Likewise, you can start the server by entering the following command from an OS/2 command prompt:

```
DHCPSD
```

The following figure shows the OS/2 DHCP server program.




```
Server Initialized at Fri Jan 10 11:22:21 1996
```

The DHCP server will output its logging data to a DHCPSD.LOG file which may look as follows:

```
01/25/96 15:16:16 START:  ...log_initialize: ***************************
01/25/96 15:16:16 START:  ...log_initialize: *     NEW LOG FOLLOWS     *
01/25/96 15:16:16 START:  ...log_initialize: * | | | | | | | | | | | | *
01/25/96 15:16:16 START:  ...log_initialize: * V V V V V V V V V V V V *
01/25/96 15:16:16 START:  ...log_initialize: ***************************
01/25/96 15:16:16 SYSERR: ...log_initialize: Logging ENABLED
01/25/96 15:16:16 OBJERR: ...log_initialize: Logging ENABLED
01/25/96 15:16:16 PROTERR:...log_initialize: Logging ENABLED
01/25/96 15:16:16 WARNING:...log_initialize: Logging ENABLED
01/25/96 15:16:16 EVENT:  ...log_initialize: Logging ENABLED
01/25/96 15:16:16 ACTION: ...log_initialize: Logging ENABLED
01/25/96 15:16:16 INFO:   ...log_initialize: Logging ENABLED
01/25/96 15:16:16 ACNTING:...log_initialize: Logging ENABLED
01/25/96 15:16:16 TRACE:  ...log_initialize: Logging ENABLED
01/25/96 15:16:16 INFO:   ...profile_repository_initialize: end of string not found
01/25/96 15:16:16 INFO:   .....am_initMapper: previous map files did not exist or has been removed; new mapping
01/25/96 15:16:16 INFO:   .....am_initMapper: no previous mapping to adopt
01/25/96 15:16:16 INFO:   ...getPortNum: dhcps/udp unknown service, assuming port 67
```

```
01/25/96 15:17:16 TRACE:  ...SelectFunc: Alarm sounded
01/25/96 15:17:16 TRACE:  ...event_timeout: Garbage collection (every 60 seconds)
01/25/96 15:18:16 TRACE:  ...receiveMailbox: Alarm sounded
01/25/96 15:18:16 TRACE:  ...event_timeout: Garbage collection (every 60 seconds)
01/25/96 15:18:21 TRACE:  ...receiveMailbox: DHCP comm descriptor selected
01/25/96 15:18:21 TRACE:  ...receiveMailbox: size of incoming packet is 548
01/25/96 15:18:21 INFO:   .....primeOptions: Option: 53, length:1
01/25/96 15:18:21 INFO:   .....primeOptions: Option: 50, length:4 value: 2724730889 (0xa2681809)
01/25/96 15:18:21 INFO:   .....primeOptions: Option: 61, length:7
01/25/96 15:18:21 INFO:   .....primeOptions: Option: 12, length:5
01/25/96 15:18:21 INFO:   .....primeOptions: Option: 15, length:16
01/25/96 15:18:21 TRACE:  .....identifiableClient: DHCP option Client-identifier specified
01/25/96 15:18:21 TRACE:  .....legibleRequest: DHCP msg type DHCPDISCOVER
01/25/96 15:18:21 TRACE:  ...process_bootrequest: request is self-consistent
01/25/96 15:18:21 TRACE:  .........am_queryMapper: cannot find client 6-0x10005a21a05d in client records
01/25/96 15:18:21 TRACE:  .......am_queryClient: client 6-0x10005a21a05d is not known to address mapper, ask clientele
01/25/96 15:18:21 TRACE:  .........cl_queryClientele: client 6-0x10005a21a05d authenticated by clientele list
01/25/96 15:18:21 INFO:   .........am_addressClient: client 6-0x10005a21a05d suggested 9.24.104.162 in range
01/25/96 15:18:21 INFO:   .........am_addressClient: client 6-0x10005a21a05d had no previous mapping, getting one
01/25/96 15:18:21 ACTION: .....processDISCOVER: address 9.24.104.162 has been reserved
01/25/96 15:18:21 INFO:   .......getPortNum: dhcpc/udp unknown service, assuming port 68
01/25/96 15:18:21 INFO:   ...generate_bootreply: generating a DHCPOFFER reply
01/25/96 15:18:21 INFO:   ...FetchHwType: Found the HW type for interface 0 = 6
01/25/96 15:18:22 TRACE:  ...transmitMailbox: transmitting to (9.24.104.162 #68)
01/25/96 15:18:26 TRACE:  ...receiveMailbox: DHCP comm descriptor selected
01/25/96 15:18:26 TRACE:  .receiveMailbox: size of incoming packet is 548
01/25/96 15:18:26 INFO:   .....primeOptions: Option: 53, length:1
01/25/96 15:18:26 INFO:   .....primeOptions: Option: 50, length:4 value: 2724730889 (0xa2681809)
01/25/96 15:18:26 INFO:   .....primeOptions: Option: 54, length:4 value: 1768429577 (0x69681809)
01/25/96 15:18:26 INFO:   .....primeOptions: Option: 61, length:7
01/25/96 15:18:26 INFO:   .....primeOptions: Option: 12, length:5
01/25/96 15:18:26 INFO:   .....primeOptions: Option: 15, length:16
01/25/96 15:18:26 TRACE:  .....identifiableClient: DHCP option Client-identifier specified
01/25/96 15:18:26 TRACE:  .....legibleRequest: DHCP msg type DHCPREQUEST
01/25/96 15:18:26 TRACE:  ...process_bootrequest: request is self-consistent
01/25/96 15:18:26 TRACE:  .......locateExchange: client id matches an active exchange
01/25/96 15:18:26 TRACE:  .....processREQUEST: OFFER was selected by client 6-0x10005a21a05d
01/25/96 15:18:26 TRACE:  ..........locateClientRecord: located client 6-0x10005a21a05d in client records
01/25/96 15:18:26 TRACE:  .....processREQUEST: address 9.24.104.162 has been bound to 6-0x10005a21a05d
01/25/96 15:18:26 INFO:   .......getPortNum: dhcpc/udp unknown service, assuming port 68
01/25/96 15:18:26 INFO:   ...generate_bootreply: generating a DHCPACK reply
01/25/96 15:18:26 INFO:   ...FetchHwType: Found the HW type for interface 0 = 6
01/25/96 15:18:26 TRACE:  ...transmitMailbox: transmitting to (9.24.104.162 #68)
01/25/96 15:19:16 TRACE:  ...receiveMailbox: Alarm sounded
01/25/96 15:19:16 TRACE:  ...event_timeout: Garbage collection (every 60 seconds)
```

In the example above, you can see a DHCP server that has been started and initialized for the first time. When the server is reinitialized or restarted it tries to adopt the latest active configuration. The server then receives a DHCPDISCOVER message from a DHCP client that has been started and replies with a DHCPOFFER message. The client then sends a DHCPREQUEST message to the server to request the configuration parameters. The server checks the requested parameters and responds with a DHCPACK message. In fact, this example matches the DHCP client log file example that is shown in 5.4, “OS/2 DHCP Client Configuring” on page 69.

When the OS/2 DHCP server has been initialized, it will store the current status of its configuration in the \MPTN\ETC\DHCPS.AR and \MPTN\ETC\DHCPS.CR files. The server will attempt to restore that information again whenever it is restarted.

To check the status of your DHCP server you can use the program dstat or enter the command `dadmin -s`. Depending on your IP address pool and the number of clients, the status of your server will look somewhat like the following:

```
PLEASE WAIT....Gathering Information From the Server....PLEASE WAIT
Status of DHCP server dos2 (9.24.104.105) as of Thu Jan 25 15:20:10 1996
IP Address      Status    Lease Time  Start Time   Last Leased  ClientId
9.24.104.161    Free
9.24.104.162    Leased       0:03:00  01/25 15:18  01/25 15:18  0x10005a21a05d
```

To reinitialize your DHCP server when you have made changes to the configuration file, enter the command `dinit` or `dadmin -i`. The dadmin program is the administration tool for your DHCP server and has the following parameters:

Table 14. DADMIN Parameter

Parameter	Description
-?	Display help message.
-v	Execute in verbose mode.
-f	Don't prompt when deleting a lease. Force it to yes.
-h <host>	DHCP Server being used (local server if not specified).
-d <ipaddress>	DELETE the lease for the specified IP address.
-i	ReINITIALIZE the specified server.
-s	Display address pool STATUS of the specified server.

5.3.1 Configuring Site-Specific Options for OS/2 Warp TCP/IP

To code specific options for an OS/2 Warp TCP/IP client could be done using the vendor option (43), but the syntax of that option is rather complicated. An easier way to supply specific configuration information to OS/2 Warp TCP/IP clients is to use some of the site-specific options, along with application and services options.

On the DHCP client, a program must be run to evaluate those options and set configuration parameters accordingly. In case of an OS/2 Warp client, the DHCPIBM.CMD file is supplied with Adapter and Protocol Services. It is a REXX command file that evaluates site-specific options and applies the values to the TCP/IP for OS/2 configuration. To activate this mechanism, you must uncomment the line for one or more options in the DHCP client configuration file. Please see 5.4, "OS/2 DHCP Client Configuring" on page 69 for more information.

The following table summarizes the configuration parameters for OS/2 Warp TCP/IP clients that can be supplied by site-specific DHCP options:

Table 15 (Page 1 of 2). DHCP Server Configuration - Site-Specific Options

Option Number	Description	Modified file
9	IP address of the default LPR server	TCPOS2.INI
71	IP address of the default NewsReader/2 server	TCPOS2.INI
200	Device name of the default LPR printer	TCPOS2.INI
201	IP address of the default Gopher server	TCPOS2.INI
202	URL of the default WWW home page	EXPLORE.INI
203	URL of the default WWW proxy server	EXPLORE.INI
204	IP address of the default WWW news server	EXPLORE.INI
205	IP address of the default SOCKS server, and optionally IP address of the default SOCKS name server	TCPOS2.INI
206	NFS client mount string	FSTAB.INI
207	X Window System default font path	PMX.INI

Table 15 (Page 2 of 2). DHCP Server Configuration - Site-Specific Options		
Option Number	**Description**	**Modified file**
208	The xdmcp command-line for the X Window System display manager	PMX.INI

To configure the site-specific options for OS/2 Warp with the DHCP server configuration program, use option 78 (user-defined option) as many times as you need for the number of options you want to configure. Figure 29 shows the panel for option 78. All you have to do is enter the option number followed by a description in the Comment field; then enter the number again in the Option number field.

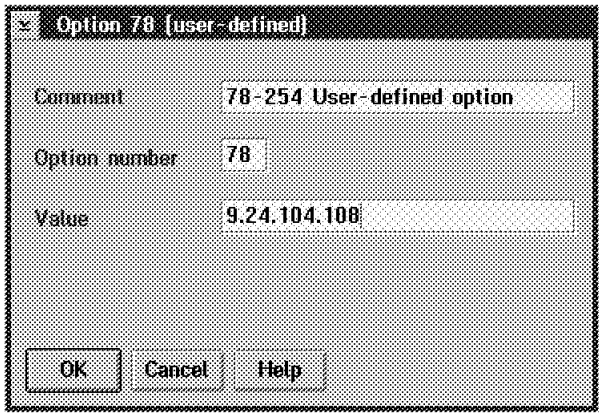


Figure 29. DHCP Server Configuration - Site-Specific Options

Notes:

1. When you expand the Options item on the Predefined resources window, you may only see options from 1 to 76. You have to expand the (62-76) App/service2 item, there you will find option 78 at the bottom of the list.
2. The text that is displayed for each site-specific option in the Current configuration window will remain Option 78 as long as you do not save and reload the configuration file.

You can, of course, configure site-specific options manually, if you prefer.

5.4 OS/2 DHCP Client Configuring

The Dynamic IP client programs will be installed with Adapter and Protocol Services. If you choose to use DHCP at the OS/2 Warp Server TCP/IP Services Installation menu, your TCP/IP interfaces will not be configured using the IFCONFIG command and any parameters that you have configured manually. Instead, the DHCP client will be started to get the necessary parameters from a DHCP server, and the DDNS client will be used to update the configuration of a Dynamic Domain Name Server, if one exists.

The following example shows a TCP/IP initialization that resulted from using manual configuration. It is contained in the \MPTN\BIN\SETUP.CMD file, which will be executed at system start:

```
route -fh
arp -f
ifconfig lan0 9.24.104.105 netmask 255.255.255.0
route add default 9.24.104.1 1
route add net 9 9.24.104.1 1
ipgate off
```

The following example shows a TCP/IP initialization that resulted from selecting dynamic configuration. It is also contained in the \MPTN\BIN\SETUP.CMD file, which will be executed at system start:

```
route -fh
arp -f
dhcpstrt -i lan0
rem route add default
```

Notes:

1. DHCP interfaces must be initialized before any manually configured interfaces.
2. If multiple interfaces need to be configured dynamically, there must be a separate dhcpstrt statement for each of them. That means that the DHCP client must contact a server for each interface, one after the other.

To configure and run the DHCP client, different programs are involved. These programs and files are summarized in the following table:

Table 16. DHCP Client Files

File	Description
\MPTN\BIN\DHCPSTRT.EXE	Starts the DHCP client and waits for it to set up the IP address. It has the following parameters: `DHCPSTRT [-d maxwait] [-i lan<#>]` maxwait is the wait time in seconds rounded to 5 seconds. `lan<3>` specifies the interface requiring DHCP configuration.
\MPTN\BIN\DHCPCD.EXE	The DHCP client, executed in the background.
\MPTN\ETC\DHCPCD.CFG	Client configuration file.
\MPTN\BIN\DHCPIBM.CMD	Site-specific configuration command file.
\MPTN\ETC\DHCPC.DB	Stores client configuration.
\MPTN\DHCPMON.EXE	Monitor for DHCP client.
\DHCPCD.LOG	Log file created by the DHCP client.

The actual DHCP client program, DHCPCD.EXE, runs as a detached program since it must remain active until you shut down the system. After the TCP/IP stack has been configured with parameters that have been obtained by a DHCP server, the client has to renew the lease for that configuration as long as TCP/IP is required to be operational.

To view the current TCP/IP configuration, you can use the DHCP client monitor program that is shown in Figure 30 on page 71. You can start this program from the System Setup folder.

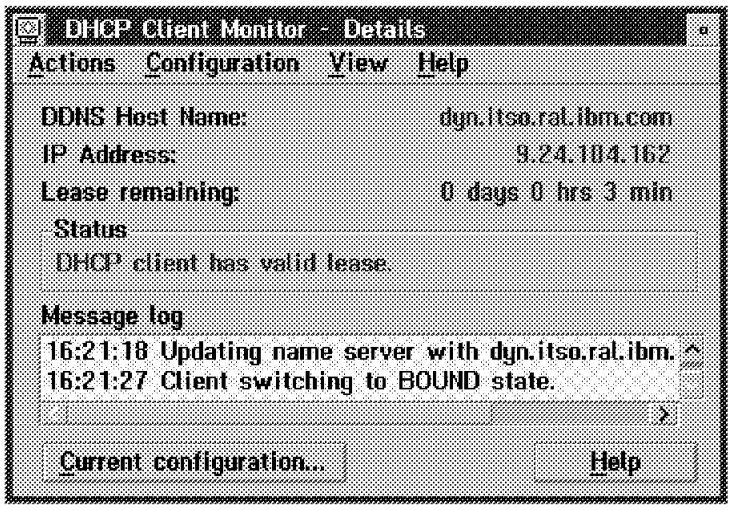


Figure 30. DHCP Client Monitor

The monitor will give you information about the client status. Depending on the client's status the icon of the monitor appears different on the desktop:

	This is how the DHCP client icon appears during normal startup, and when you are connected to the network with an unexpired lease.
	This is how the DHCP client appears when your DHCP server has not renewed your lease, and the DHCP client is attempting to renew your lease using any available DHCP server on the network. You may continue to use your current IP address until the lease expires.
	This is how the DHCP Client icon appears when either of the following conditions occurs: • You started your computer, but were unable to locate a DHCP server to provide you with an IP address or renewed your lease. • Your lease expired before a DHCP server renews your lease while you were on the network.

The status field of the monitor displays one of the following and is refreshed every 15 seconds:

- Discovering DHCP servers
- Renewing lease
- DHCP client is not running

The field DDNS Hostname displays the name of your client that is registered with the DDNS server. When you don't use a DDNS server, Not Registered is displayed. After receiving an IP address from your DHCP server the monitor shows the address in the field IP address; otherwise Not Configured is shown. The time that your lease of the IP address is vaild is displayed in the Lease remaining field. After this time has expired, your IP address is not valid any longer.

The Message log gives you more detailed information about the messages the client sends to the server. It also shows the state of the client. When the client is in BOUND state, your IP address is still valid. After the timer T1 expires, the

client switches to the renewing state. If renewing of the lease fails, it switches to the rebinding state (that is, when timer T2 expires).

If you want to review the configuration in more detail, click on **Current Configuration ...**. The following panel will be shown:

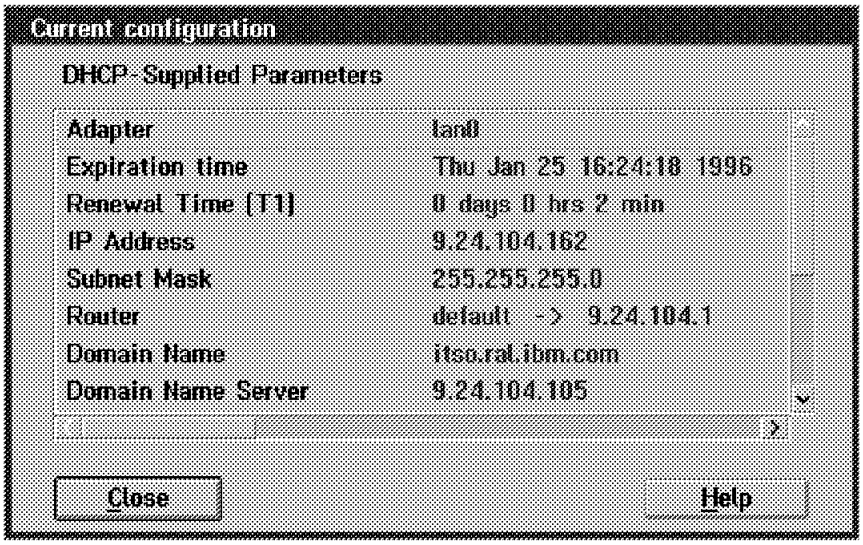


Figure 31. DHCP Client Configuration

Note: Starting and stopping the monitor will not affect the DHCP client.

The DHCP client can be configured by using the DHCPCD.CFG configuration file, which is normally contained in the \MPTN\ETC directory. In this file, you can specify what parameters the DHCP client should request from a server any time it is starting.

By default, the client identifies itself with its LAN adapter hardware address, and logging is enabled. Since the IBM OS/2 DHCP server allows grouping of clients that take the same set of parameters into classes, a client may want to obtain just those parameters if this workstation belongs to a certain class. This can be very helpful to separate workstations from different departments, while maintaining the capability of configuring any workstation dynamically. There is, however, administrative overhead involved, since the modified configuration files for the DHCP clients need to be supplied to the workstations during installation. Normally, this would be achieved by using electronic software distribution methods. The following table summarize the options that can be configured in a client's configuration file:

Table 17 (Page 1 of 2). Configuration Options

Option	Description
numLogFiles	Number of log files. If 0 is specified, no log file will be maintained and no log message is displayed anywhere. n is the maximum number of log files maintained. As the size of the most recent log file reaches its maximum size, a new log file is created.
logFileSize	Maximum size of a log file. When the size of the most recent log file reaches this value, it is renamed and a new log file is created.

Table 17 (Page 2 of 2). Configuration Options

Option	Description
logFileName	Name of the most recent log file. Less recent log files have the number 1 to (n - 1) appended to their names; the larger the number, the less recent the file.

logItem	Specifies, what should be logged.
	`SYSERR Log system errors.`
	`OBJERR Log object errors between objects in the process.`
	`PROTERR Log protocol errors between client and server.`
	`WARNING Log warnings deserving attention from the user.`
	`EVENT Log events that occur to the process.`
	`ACTION Log actions taken by the process.`
	`INFO Log information that might be useful.`
	`ACNTING Log who was served when.`
	`TRACE Log code flow for debugging.`

interface	DHCP interface. ifName is the name of the network interface.
clientid	The client ID to use in all communication with the server. The clientid value should always be MAC, denoting that the hardware address for the particular interface should be used as the client ID.
updateDNS	A command string to execute to cause the DNS server to be updated with the new IP address for the given name. This is explained in more detail in the DDNS server and client section of this chapter.

option	An option requested by this client. Option has the following format:
	`option <code> [<value>] [exec <string>]`
	For example:
	`option 200 exec "test.cmd %s"`

vendor	Special syntax for the specification of the vendor extensions field. It is followed by a set of curly braces. Inside the curly braces, the options and values for the vendor extensions field are specified. An exec function for the vendor option should be placed on the same line as the "vendor" keyword, using the same syntax as on the other option lines. An exec string on an option inside the vendor extensions options is not valid. It is ignored.
reject	Specifies that if this option code is returned by the server, this option should be ignored by the client. Its value should not be used.
otherOptions	Specifies how all other options should be handled by the client. This refers to any options not specifically requested with an "option" statement or rejected with a "reject" statement. The default is that all options are accepted.
ifconfig_options	Specifies the options and flags to be used with ifconfig when initializing the interface. This refers to the options that can be specified by the ifconfig command. For example: `ifconfig_options "802.3 -icmpred"`

The following example shows a configuration file for the DHCP client. The default file name is DHCPCD.CFG.

```
# Basic options required

clientid   MAC
interface lan0

# Uncomment as desired for logging

numLogFiles   4
logFileSize   100
logFileName   dhcpcd.log
logItem       SYSERR
logItem       OBJERR
logItem       PROTERR
logItem       WARNING
logItem       EVENT
logItem       ACTION
logItem       INFO
logItem       ACNTING
logItem       TRACE

option 1
option 3
option 6
option 15 itso.ral.ibm.com

# The following are requested for interoperability with some servers which
# need explicit requests.


#updateDNS "nsupdate -h%s -d%s -s"d;a;*;a;a;%s;s;%s;311040Q;q" -q"

# The following are options for which IBM supplies an installation
# script, dhcpibm.cmd, to automatically configure the IBM application
# with the served value.  Uncomment them if desired.

#option 9   exec "dhcpibm.cmd 9 %s"       # LPR Server
#option 71  exec "dhcpibm.cmd 71 %s"      # Default NewsReader/2
#option 200 exec "dhcpibm.cmd 200 %s"     # Default LPR Printer
#option 201 exec "dhcpibm.cmd 201 %s"     # Gopher Server
#option 202 exec "dhcpibm.cmd 202 %s"     # Default WWW Home Page
#option 203 exec "dhcpibm.cmd 203 %s"     # Default WWW Proxy Server
#option 204 exec "dhcpibm.cmd 204 %s"     # Default WWW News Server
#option 205 exec "dhcpibm.cmd 205 %s"     # Default Socks Server
#option 206 exec "dhcpibm.cmd 206 %s"     # NFS Servers and Mount Points
#option 207 exec "dhcpibm.cmd 207 %s"     # Default X Font Server
#option 208 exec "dhcpibm.cmd 208 %s"     # Default X System Display Manager
option 12 dyn
```

In this example, the client will identify itself using its LAN adapter hardware address (MAC) and it will use DHCP to configure one IP interface on the LAN (lan0). The client will also request specific options from a DHCP server, and it will identify itself as belonging to a certain vendor and user class. This may help a DHCP server to supply options to this client that are specific to a set of clients that form this user class.

An update string is also provided to add the client's hostname resource records to a dynamic domain name server.

Towards the end of the configuration file, a user program can be invoked to evaluate if site-specific options have been supplied by a DHCP server. Such a program will then apply those parameters to the client's TCP/IP configuration. In case of an OS/2 WARP client, the DHCPIBM.CMD file is supplied with Adapter and Protocol Services. It is a REXX command file that evaluates site-specific options and applies the values to the TCP/IP for OS/2 configuration. To activate this mechanism, you have to uncomment the line for one or more options in the DHCP client configuration file.

A log file is provided by the DHCP client for problem determination purposes. Logging information will normally be written to the DHCPCD.LOG file, but logging is turned off by default. An example of a DHCP client log file is shown below:

```
01/25/96 16:30:57 START:  ....log_initialize: ***************************
01/25/96 16:30:57 START:  ....log_initialize: *     NEW LOG FOLLOWS     *
01/25/96 16:30:57 START:  ....log_initialize: * | | | | | | | | | | | | *
01/25/96 16:30:57 START:  ....log_initialize: * V V V V V V V V V V V V *
01/25/96 16:30:57 START:  ....log_initialize: ***************************
01/25/96 16:30:57 SYSERR: ....log_initialize: Logging ENABLED
01/25/96 16:30:57 OBJERR: ....log_initialize: Logging ENABLED
01/25/96 16:30:57 PROTERR:....log_initialize: Logging ENABLED
01/25/96 16:30:57 WARNING:....log_initialize: Logging ENABLED
01/25/96 16:30:57 EVENT:  ....log_initialize: Logging ENABLED
01/25/96 16:30:57 ACTION: ....log_initialize: Logging ENABLED
01/25/96 16:30:57 INFO:   ....log_initialize: Logging ENABLED
01/25/96 16:30:57 ACNTING:....log_initialize: Logging ENABLED
01/25/96 16:30:57 TRACE:  ....log_initialize: Logging ENABLED
01/25/96 16:30:57 INFO:   ....probeIfs: client has 1 previously recorded lease
01/25/96 16:31:09 INFO:   ....probeIfs: Initialized interface lan0

01/25/96 16:31:09 TRACE:  ..........FetchHWAddress: interface 0 [802.5] physical address 10005a21a05d
01/25/96 16:31:09 INFO:   ..........FetchHwType: Found the HW type for interface 0 = 6
01/25/96 16:31:09 TRACE:  ....probeIfs: [ifconfig lan0 0  broadcast 255.255.255.255]

01/25/96 16:31:09 TRACE:  ....probeIfs: ifconfig successful for 0
01/25/96 16:31:09 INFO:   ....probeIfs: Getting media ADDRESS
01/25/96 16:31:09 TRACE:  ........FetchHWAddress: interface 0 [802.5] physical address 10005a21a05d
01/25/96 16:31:09 INFO:   ........FetchHwType: Found the HW type for interface 0 = 6
01/25/96 16:31:10 INFO:   ..main: number of interfaces needing DHCP configuration is 1
01/25/96 16:31:10 TRACE:  ....probeIfStatus: a previous lease (9.24.104.162 : 180) has expired
01/25/96 16:31:10 TRACE:  ......process_event: expectation fulfilled
01/25/96 16:31:13 TRACE:  ........process_fsm: generating a DISCOVER
01/25/96 16:31:13 TRACE:  ........process_fsm: generating message with xid = 2435
01/25/96 16:31:13 INFO:   ........process_fsm: Check of against MTU size 1500
01/25/96 16:31:13 TRACE:  ..........transmitMailbox: transmitting to (255.255.255.255 #67)
01/25/96 16:31:13 INFO:   ..........process_timer: Ta Seconds = 30
01/25/96 16:31:13 TRACE:  ........process_fsm: state transition to SELECTING
01/25/96 16:31:13 TRACE:  ........SelectFunc: DHCP comm descriptor selected
01/25/96 16:31:13 TRACE:  ......client_event: received packet xid = 2435
01/25/96 16:31:13 INFO:   ........primeOptions: Option: 53, length:1
01/25/96 16:31:13 INFO:   ........primeOptions: Option: 58, length:4 value: 1509949440 (0x5a000000)
01/25/96 16:31:13 INFO:   ........primeOptions: Option: 59, length:4 value: 2634022912 (0x9d000000)
01/25/96 16:31:13 INFO:   ........primeOptions: Option: 54, length:4 value: 1768429577 (0x69681809)
01/25/96 16:31:13 INFO:   ........primeOptions: Option: 1, length:4 value: 16777215 (0x00ffffff)
01/25/96 16:31:13 INFO:   ........primeOptions: Option: 3, length:4 value: 23599113 (0x01681809)
01/25/96 16:31:14 INFO:   ........primeOptions: Option: 6, length:4 value: 1768429577 (0x69681809)
01/25/96 16:31:14 INFO:   ........primeOptions: Option: 15, length:16
01/25/96 16:31:14 INFO:   ........primeOptions: Option: 51, length:4 value: 3019898880 (0xb4000000)
01/25/96 16:31:14 TRACE:  ........legibleReply: DHCP message type DHCPOFFER
01/25/96 16:31:14 TRACE:  ....receiveEvent: Expecting xid 2435  , Got xid 2435
01/25/96 16:31:14 TRACE:  ......process_event: expectation fulfilled
01/25/96 16:31:14 ACTION: ..........record_offer: recorded offer (9.24.104.162 : 180) from server 9.24.104.105
01/25/96 16:31:14 INFO:   ..........record_offer: Pseudo T2 limit set to 1
01/25/96 16:31:14 INFO:   ..........record_offer: Pseudo T3 limit set to 1 interval 300
01/25/96 16:31:14 INFO:   ............count_option_match: Counted 1 matches in offer from 9.24.104.105.
01/25/96 16:31:18 TRACE:  ........receiveMailbox: Alarm sounded
01/25/96 16:31:18 INFO:   ..........updateLeaseRecord: t1 is 90
01/25/96 16:31:18 INFO:   ..........updateLeaseRecord: t2 is 60
01/25/96 16:31:18 ACTION: ........process_fsm: Using lease (9.24.104.162 : 180) from server 9.24.104.105
01/25/96 16:31:18 TRACE:  ........process_fsm: generating a REQUEST
01/25/96 16:31:18 TRACE:  ........process_fsm: generating message with xid = f84
01/25/96 16:31:18 INFO:   ........process_fsm: Check of against MTU size 1500
01/25/96 16:31:18 TRACE:  ..........transmitMailbox: transmitting to (255.255.255.255 #67)
01/25/96 16:31:18 INFO:   ..........process_timer: Ta Seconds = 30
01/25/96 16:31:19 TRACE:  ........process_fsm: state transition to REQUESTING
01/25/96 16:31:19 TRACE:  ........receiveMailbox: DHCP comm descriptor selected
01/25/96 16:31:19 TRACE:  ......client_event: received packet xid = f84
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 53, length:1
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 58, length:4 value: 1509949440 (0x5a000000)
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 59, length:4 value: 2634022912 (0x9d000000)
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 54, length:4 value: 1768429577 (0x69681809)
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 1, length:4 value: 16777215 (0x00ffffff)
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 3, length:4 value: 23599113 (0x01681809)
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 6, length:4 value: 1768429577 (0x69681809)
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 15, length:16
01/25/96 16:31:19 INFO:   ........primeOptions: Option: 51, length:4 value: 3019898880 (0xb4000000)
01/25/96 16:31:19 TRACE:  ........legibleReply: DHCP message type DHCPACK
01/25/96 16:31:19 TRACE:  ....receiveEvent: Expecting xid f84  , Got xid f84
01/25/96 16:31:19 TRACE:  ......process_event: expectation fulfilled
01/25/96 16:31:19 TRACE:  ........process_fsm: Checking address for clash.
01/25/96 16:31:19 TRACE:  ..........arpcheck: deleting old arp entry.
01/25/96 16:31:19 TRACE:  ..........arpcheck: sending pings.
01/25/96 16:31:21 TRACE:  ..........arpcheck: checking arp table.
01/25/96 16:31:21 ACTION: ..process_fsm: announcing the new IP address 9.24.104.162 obtained from server 9.24.104.105
01/25/96 16:31:21 ACTION: ..........record_offer: recorded offer (9.24.104.162 : 180) from server 9.24.104.105
01/25/96 16:31:21 INFO:   ..........record_offer: Pseudo T2 limit set to 1
```

```
01/25/96 16:31:21 INFO:   ..........record_offer: Pseudo T3 limit set to 1 interval 300
01/25/96 16:31:21 INFO:   ............count_option_match: Counted 1 matches in offer from 9.24.104.105.
01/25/96 16:31:21 INFO:   ..........updateLeaseRecord: t1 is 90
01/25/96 16:31:21 INFO:   ..........updateLeaseRecord: t2 is 60
01/25/96 16:31:21 ACTION: ........process_fsm: Using lease (9.24.104.162 : 180) from server 9.24.104.105
01/25/96 16:31:21 TRACE:  ........process_fsm: Plugboard starts now...

01/25/96 16:31:21 TRACE:  ..........SetOptions: Inside SetOptions
01/25/96 16:31:21 INFO:   ..........SetOptions: ddnscfg verification: hostname dyn domain itso.ral.ibm.com
01/25/96 16:31:21 TRACE:  ............exec_set_ipaddress: ip address = 9.24.104.162
01/25/96 16:31:21 TRACE:  ............exec_set_ipaddress: cmd = ifconfig lan0 netmask 9.24.104.162
01/25/96 16:31:21 INFO:   ..........SetOptions: Option 1 received
01/25/96 16:31:22 TRACE:  ..........SetOptions: running builtin_exec for 1.
01/25/96 16:31:22 TRACE:  ............exec_set_mask: cmd = ifconfig lan0 netmask 255.255.255.0
01/25/96 16:31:22 INFO:   ..........SetOptions: Option 3 received
01/25/96 16:31:22 TRACE:  ..........SetOptions: running builtin_exec for 3.
01/25/96 16:31:22 TRACE:  ............exec_set_routes: SetOptions: Route option
01/25/96 16:31:22 TRACE:  ............exec_set_routes: Router Option command is = route add default 9.24.104.1 1
01/25/96 16:31:23 INFO:   ..............route_add: dest 0 router 9186801 type GATEWAY
01/25/96 16:31:23 INFO:   ..........SetOptions: Option 6 received
01/25/96 16:31:23 TRACE:  ..........SetOptions: running builtin_exec for 6.
01/25/96 16:31:24 TRACE:  ............exec_set_dns_server: SetOptions: Domain Name Option
01/25/96 16:31:24 TRACE:  ............exec_set_dns_server: Domain = [itso.ral.ibm.com]

01/25/96 16:31:26 INFO:   ..........SetOptions: Option 15 received
01/25/96 16:31:26 INFO:   ..........SetOptions: Option 51 received
01/25/96 16:31:26 INFO:   ..........SetOptions: Option 53 received
01/25/96 16:31:26 INFO:   ..........SetOptions: Option 54 received
01/25/96 16:31:26 INFO:   ..........SetOptions: Option 58 received
01/25/96 16:31:26 INFO:   ..........SetOptions: Option 59 received
01/25/96 16:31:26 INFO:   ..........SetOptions: Option 255 received
01/25/96 16:31:26 ACTION: ........process_fsm: assigned net address 9.24.104.162 to interface 0
01/25/96 16:31:26 INFO:   ..updateDNS:"nsupdate -hdyn-ditso.ral.ibm.com-s"d;a;*;a;a;9.24.104.1 6 2 ;s;180;...
01/25/96 16:31:26 TRACE:  ..updateDNS: Start "nsupdate -hdyn -ditso.ral.ibm.com -s"d;a;*;a;a;9.24.104.162;...
01/25/96 16:31:49 TRACE:  ..updateDNS: Alarm sounded
01/25/96 16:31:56 TRACE:  ........process_fsm: state transition to BOUND
```

In the example above, you can see a DHCP client whose lease is expired. The client has to request a new lease. It will therefore broadcast a DHCPDISCOVER message to find an DHCP server. The DHCP server replies with a DHCPOFFER message which holds an IP address that can be used by the client. After the client receives its IP address, it requests the configuration parameters from the server by sending a DHCPREQUEST message. The DHCP server will answer with a DHCPACK message and supplies the parameters to the client. In fact, this example matches the DHCP server log file example that is shown in the previous section.

In this example the client is also configured as a DDNS client. That is why it updates the DDNS Server with its hostname. This can be seen in the last statements of the above log file, before the client switches to the BOUND state.

When the OS/2 Dynamic IP client has been initialized, it will store the options received from the DHCP and DDNS servers in the \MPTN\ETC\DHCPC.DB file and it will also modify the original DHCPCD.CFG file. The client will attempt to request the stored information again whenever it is restarted.

After a specified time the client has to renew its lease for the IP address. The renewing time is stored in the parameter T1, which is supplied by the server in the above example. When the time (T1) is expired, the client sends a REQUEST message to the DHCP server to renew its lease. Therefore the client state changes from BOUND to RENEWING. When the DHCP server renews the client's lease, it will answer with a DHCPACK message.The rest of the procedure is the same as when applying for a new lease. The following part of a log file shows the messages exchanged between client and server:

```
01/25/96 16:31:56 INFO:   ....receiveEvent: Timer T1 expired
01/25/96 16:31:56 TRACE:  ......process_event: expectation fulfilled
01/25/96 16:31:56 TRACE:  ........process_fsm: generating a REQUEST
01/25/96 16:31:57 TRACE:  ........process_fsm: generating message with xid = 2Oa4
01/25/96 16:31:58 TRACE:  ..........transmitMailbox: transmitting to (9.24.1O4.1O5 #67)
01/25/96 16:31:58 TRACE:  ........process_fsm: state transition to RENEWING
01/25/96 16:31:58 TRACE:  ........receiveMailbox: DHCP comm descriptor selected
01/25/96 16:31:58 TRACE:  ......client_event: received packet xid = 2Oa4
01/25/96 16:31:58 INFO:   ........primeOptions: Option: 53, length:1
01/25/96 16:31:58 INFO:   ........primeOptions: Option: 58, length:4 value: 1509949440 (0x5a000000)
01/25/96 16:31:59 INFO:   ........primeOptions: Option: 59, length:4 value: 2634022912 (0x9d000000)
01/25/96 16:31:59 INFO:   ........primeOptions: Option: 54, length:4 value: 1768429577 (0x69681809)
01/25/96 16:31:59 INFO:   ........primeOptions: Option: 1, length:4 value: 16777215 (0xOOffffff)
01/25/96 16:31:59 INFO:   ........primeOptions: Option: 3, length:4 value: 23599113 (0x01681809)
01/25/96 16:31:59 INFO:   ........primeOptions: Option: 6, length:4 value: 1768429577 (0x69681809)
01/25/96 16:31:59 INFO:   ........primeOptions: Option: 15, length:16
01/25/96 16:31:59 INFO:   ........primeOptions: Option: 51, length:4 value: 3019898880 (0xb4000000)
01/25/96 16:31:59 TRACE:  ........legibleReply: DHCP message type DHCPACK
01/25/96 16:31:59 TRACE:  ....receiveEvent: Expecting xid 2Oa4  , Got xid 2Oa4
01/25/96 16:31:59 TRACE:  ......process_event: expectation fulfilled
01/25/96 16:31:59 ACTION: ..........record_offer: recorded offer (9.24.1O4.162 : 18O) from server 9.24.1O4.1O5
01/25/96 16:31:59 INFO:   ..........record_offer: Pseudo T2 limit set to 1
01/25/96 16:31:59 INFO:   ..........record_offer: Pseudo T3 limit set to 1 interval 3OO
01/25/96 16:31:59 INFO:   ............count_option_match: Counted 1 matches in offer from 9.24.1O4.1O5.
01/25/96 16:32:OO INFO:   ..........updateLeaseRecord: t1 is 9O
01/25/96 16:32:OO INFO:   ..........updateLeaseRecord: t2 is 6O
01/25/96 16:32:OO ACTION: ........process_fsm: Using lease (9.24.1O4.162 : 18O) from server 9.24.1O4.1O5
01/25/96 16:32:OO TRACE:  ........process_fsm: Plugboard starts now...

01/25/96 16:32:OO TRACE:  ..........SetOptions: Inside SetOptions
01/25/96 16:32:OO INFO:   ..........SetOptions: ddnscfg verification: hostname dyn domain itso.ral.ibm.com
01/25/96 16:32:OO TRACE:  ............exec_set_ipaddress: ip address = 9.24.1O4.162
01/25/96 16:32:OO TRACE:  ............exec_set_ipaddress: cmd = ifconfig lanO netmask 9.24.1O4.162
01/25/96 16:32:O1 INFO:   ..........SetOptions: Option 1 received
01/25/96 16:32:O1 TRACE:  ..........SetOptions: running builtin_exec for 1.
01/25/96 16:32:O1 TRACE:  ............exec_set_mask: cmd = ifconfig lanO netmask 255.255.255.O
01/25/96 16:32:O1 INFO:   ..........SetOptions: Option 3 received
01/25/96 16:32:O1 TRACE:  ..........SetOptions: running builtin_exec for 3.
01/25/96 16:32:O1 TRACE:  ............exec_set_routes: SetOptions: Route option
01/25/96 16:32:O1 TRACE:  ............exec_set_routes: Router Option command is = route add default 9.24.1O4.1 1
01/25/96 16:32:O1 INFO:   ..............route_add: dest O router 9186801 type GATEWAY
01/25/96 16:32:O1 INFO:   ..............route_add: route exists!
01/25/96 16:32:O1 INFO:   ..........SetOptions: Option 6 received
01/25/96 16:32:O2 TRACE:  ..........SetOptions: running builtin_exec for 6.
01/25/96 16:32:O2 TRACE:  ............exec_set_dns_server: SetOptions: Domain Name Option
01/25/96 16:32:O2 TRACE:  ............exec_set_dns_server: Domain = [itso.ral.ibm.com]

01/25/96 16:32:O2 INFO:   ..........SetOptions: Option 15 received
01/25/96 16:32:O2 INFO:   ..........SetOptions: Option 51 received
01/25/96 16:32:O2 INFO:   ..........SetOptions: Option 53 received
01/25/96 16:32:O2 INFO:   ..........SetOptions: Option 54 received
01/25/96 16:32:O2 INFO:   ..........SetOptions: Option 58 received
01/25/96 16:32:O2 INFO:   ..........SetOptions: Option 59 received
01/25/96 16:32:O3 INFO:   ..........SetOptions: Option 255 received
01/25/96 16:32:O3 ACTION: ........process_fsm: assigned net address 9.24.1O4.162 to interface O
01/25/96 16:32:O3 INFO:   ..updateDNS: "nsupdate -hdyn -ditso.ral.ibm.com -s"d;a;*;a;a;9.24.1O4.162;s;18O;...
01/25/96 16:32:O3 TRACE:  ..updateDNS: Start "nsupdate -hdyn -ditso.ral.ibm.com -s"d;a;*;a;a;9.24.1O4.162;...
```

5.5 BOOTstrap Protocol

The BOOTP protocol enables a client to get its IP address, a gateway address and the address of a name server from a BOOTP or DHCP server machine. You can use it to update your network-relevant information from a central point, the BOOTP or DHCP server. You can use it on a token-ring network with token-ring bridges, which use the Source-Routing protocol. It does not work with the token-ring to Ethernet bridge (IBM 8209).

Note: The BOOTP server and the BOOTP client have to be on the same physical token-ring LAN or Ethernet segment. It is not possible to run BOOTP over a SLIP or an X.25 interface.

The BOOTP 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 TCPIPETC directory. You must change this file to contain hardware

addresses, IP addresses, gateway addresses, and name server addresses for your local network.

To find the hardware addresses of the installed terminals, use the NETSTAT command with the -n parameter, on the installed terminal.

You must verify that the SERVICES file in the TCPIPETC directory contains the following two lines:

```
sbootp  67/udp  #bootp server
cbootp  68/udp  #bootp client
```

These statements have to be in both machines, the server and the client.

The following BOOTPTAB file is an example of a configuration with host CID server as the BOOTP server:

```
# tcpipetcbootptab: database for bootp server BOOTPD

# Blank lines and lines beginning with '#' are ignored.
#
# Legend:
#
# first field -- hostname
#   (full domain name)
#
# bf -- bootfile (not supported)
# ds -- domain name server address list
# gw -- gateway address list
# ha -- host hardware address (follows ht)(hexadecimal)
# hd -- home directory (not supported)
# hn -- send hostname (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.

global.dummy:\
        :sm=255.255.255.0:\
        :hd=/bootpd/trypd:bf=null:\
        :ds=9.24.104.108:

# Next, the subnets default route information.

subnet.dummy:\
        :tc=global.dummy:gw=9.24.104.1

# Last, the individual information.

dos.itso.ral.ibm.com: tc=subnet.dummy: ht=ethernet:\
     ha=10005a26399c: ip=9.24.104.68: hn:
os2.itso.ral.ibm.com: tc=subnet.dummy: ht=ethernet:\
     ha=400030010002: ip=9.24.104.69: hn:
cid.itso.ral.ibm.com: tc=subnet.dummy: ht=ethernet:\
     ha=400040010003: ip=9.24.104.70: hn:
```

Note: The ht= (host hardware type) parameter must be either ETHERNET or ETHER, even if the physical connection is token-ring. The hn parameter for the hostname to send to must be blank. Every input to the BOOTPTAB file is sensitive to the syntax. To create your own BOOTPTAB file, rename the original sample BOOTPTAB to, for example, BOOTPTAB.ORI, and copy this file to BOOTPTAB. Do not erase the original BOOTPTAB file.

A successfully started BOOTP process will show you the following screen on the OS/2 BOOTP client os2.

```
[C:]bootp

ifconfig lan0 9.24.104.69 netmask 255.255.255.0
route add default 9.24.104.1 1
add net default: router 9.24.104.1: File exists
Name server: 9.24.104.108
Hostname os2.itso.ral.ibm.com
```

The next screen shows the BOOTP request from OS/2 BOOTP client os2 and DOS BOOTP client dos in the BOOTPD window of host cid.

```
[C:]bootpd -d -d -d -d

bootpd: reading "D:\TCPIP\ETC\BOOTPTAB"
bootpd: read 7 entries from "D:\TCPIP\ETC\BOOTPTAB"
bootpd: request from hardware address 400030010002
bootpd: vendor magic field is 99.130.83.99
bootpd: sending RFC1048-style reply
bootpd: request from hardware address 10005A26399C
bootpd: vendor magic field is 99.130.83.99
bootpd: sending RFC1048-style reply
```

You can start an OS/2 BOOTP client without TCP/IP for OS/2 started, but after an unsuccessful BOOTP you should run at least the SETUP.CMD to set the network interfaces to a legal status. Each interrupted BOOTP process can result in unpredictable errors if the machine is not rebooted or the SETUP.CMD not run.

5.5.1 BOOTP from a DOS Workstation

The BOOTP client allows a client machine to discover its own Internet address, subnet mask, default router, hostname, domain name, domain name servers, and print and time server from a remote BOOTP protocol server. The following is the format of the BOOTP command:

```
BOOTP [-? -?? | <hware_name>]
```

where:

-? displays a list of parameters and their meaning.

-?? displays a more detailed list of parameters

<hware_name> is the name interface defined by CUSTOM that is to be set by the execution of BOOTP.

To use BOOTP on your PC you should do the following:

1. Verify that a BOOTP server is running on the network.
2. Use CUSTOM to create an interface, for example, nd0. Enter any valid Internet address in the IP Address field, for example 1.1.1.1 - (note that this will be replaced once you execute BOOTP) and leave the Network Mask field blank.

To execute BOOTP, you must enable TCP/IP for DOS with the TCPSTART command. Once TCP/IP for DOS is enabled, you can then execute BOOTP.

The following is an example of the response that is displayed as a result of issuing the command BOOTP nd0 to an OS/2 BOOTP server:

```
C:>bootp nd0

ifconfig nd0 9.24.104.68 netmask 255.255.255.0
route add default 9.24.104.1
writing to the route daemon: No space left on device
add net default: gateway 9.24.104.1: No space left on device
Name server: 9.24.104.108
Hostname dos.itso.ral.ibm.com
```

The OS/2 BOOTP server only sends the data that was defined in the BOOTPTAB file.

You can run the TCP/IP for DOS BOOTP client to an OS/2 BOOTP server as long as you are on the same physical LAN segment as the BOOTP server. It does not run on a SLIP connection. There is only the request for the data at the BOOTP server, but no response to the BOOTP client. To start the BOOTP client on DOS, TCP/IP has to be already initialized.

5.6 The Domain Name System

Apart from using an IP address to identify a TCP/IP host, it is also possible to use a hostname. A set of hosts and names can be grouped to form a domain. Domains are arranged hierarchically to form the domain name system (DNS), descending like a tree from a root domain.

You can refer to hosts in your domain by hostname only; however a name server requires a fully qualified domain name. For instance, a hostname of:

```
host
```

and a domain name of:

```
domain.parent.root
```

will make a fully qualified domain name of:

```
host.domain.parent.root
```

with hostname and domain names, according to the domain hierarchy, separated by periods. The local resolver combines the hostname with the domain name before sending the address resolution request to the domain name server.

TCP/IP applications contact a name server whenever it is necessary to translate a domain name into an Internet address, or when information is required about a domain. The name server performs the translation if it has the necessary information. If it does not have the necessary information, the name server can contact other name servers, which in turn can contact other name servers. This process is called a *recursive query*. Alternatively, a name server can simply return the address of another name server that might hold the requested information. This is called a *referral response* to a query. Name server implementations must support referrals, but are not required to perform recursive queries.

5.6.1.1 The IN-ADDR.ARPA Domain

The DNS defines a special domain called IN-ADDR.ARPA to translate Internet addresses to domain names. The format of an in-addr.arpa name is the reverse octet order of an IP address concatenated with the `in-addr.arpa` string. For example, the address 9.67.43.100, would be 100.43.67.9.in-addr.arpa. This scheme is sometimes referred to as reverse name resolution.

5.6.2 Domain Name Resolver and Domain Name Server

The resolver is comprised of a few routines that build query packets and exchange them with the name server. TCP/IP for OS/2 applications have `gethostbyname()` and `gethostbyaddr()` routines compiled into their code that resolves hostnames into Internet addresses. This code accesses a name server, host table, or both, depending on your configuration.

5.6.2.1 The HOSTS File

If you are not using a name server but you still need to map hostnames to IP addresses, you need to create a HOSTS file in the TCPIPETC subdirectory. In this file, each line contains an IP address and a hostname that you want associated with it.

```
9.24.104.10  kwichman
9.24.104.36  lnotessv
9.24.104.162 wgrode
9.24.104.163 shlee
```

Note: The contents of a HOSTS file are entirely local to the system that this file resides on. A domain name server can be thought of as a centralized HOSTS file.

5.6.2.2 The RESOLV File

Each system that is going to use the name server needs to know its Internet address. Multiple name servers can be defined in the TCPIPETCRESOLV file at the DNS clients to have backup variants. If none of the name servers in the list responds, then the HOSTS file is searched to find the address for the given name. This situation is responsible for very slow response times during the connection setup. You can change the configuration in the RESOLV file as shown in the following example:

```
;domain         domainname
;nameserver     IP address
domain          itso.ral.ibm.com
nameserver      9.67.38.108
```

1. The domain statement defines the local domain for the system.
2. This is the first name server that is asked to resolve the name. If you want to use the HOSTS file only, do not code the name server statement.
3. If the first name server does not respond, the next available name server in the list is asked.

The OS/2 resolver accesses the local host table only in the following cases:

- You have not defined a name server in the RESOLV file.
- The name server either does not respond, or the name server responds with an `unknown host` error.

The name server communicates with other name servers to query records outside its zone, to answer queries about zones for which it has authority, and to transfer zones both to and from other name servers.

The name server has a function similar to the HOSTS file; its prime function is to translate names into Internet addresses. The name server database can be administered centrally. It is no longer necessary to maintain many HOSTS files on various systems.

The following illustrates the search flow for domain name resolution:

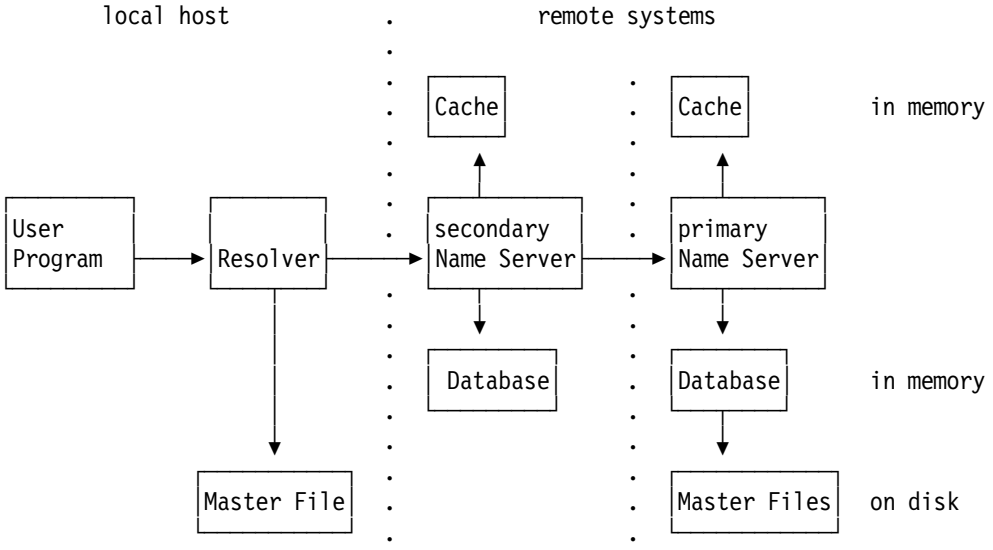


Figure 32. Domain Name Resolution

Notes:

1. A user program that needs to resolve hostnames to Internet addresses (or the other way round) would:
   a. Contact the resolver on the local host if no RESOLV file exists; the master file would then be the local HOSTS file.
   b. Contact the name server specified in the RESOLV file; the master file would then be the name server's NAMED.DOM file.
2. If the first name server could not resolve the name, it might forward the request to another name server that ranks higher in the DNS hierarchy. The NAMED.CA file specifies which server that would be.
3. The name server database is built in memory from the master files.
4. The name server cache holds recent resolve requests.
5. A secondary name server would have no master files.

5.7 The Dynamic Domain Name Services (DDNS)

Today's Domain Name System (DNS) servers support only queries on a statically configured database. The Dynamic DNS (DDNS) protocol defines extensions to the Domain Name System to enable DNS servers to accept requests to update the DNS database dynamically. These extensions provide support for adding and deleting a set of names and associated resource records within a single zone automatically.

The extensions assume that DNS security extensions, as defined by the IETF DNSSEC working group, have been implemented, but are not necessarily in use. DNS security extensions are used in DDNS to authenticate hosts that request to enter or change entries in the DDNS server database.

Without client authentication, another host, with perhaps malicious intent, may impersonate an unsuspecting host by remapping the address entry for the unsuspecting host to that of its own. After the remapping occurs, data (for example, logon passwords!) intended for the unsuspecting host is effectively intercepted by the malicious, ″spoofing″ host. IBM implements fail-safe RSA public-key digital signature technology to secure the DNS database updates and eliminate the possibility of ″spoofing″. IBM is the first company to introduce products that support Dynamic DNS and associated DNS security extensions.

The following paragraphs provide a brief outline of the RSA encryption standard and the DDNS client and server protocol.

5.7.1 RSA - Cryptography

Since the IBM OS/2 DDNS server and client products implement not only dynamic DNS but also DNS security functions, we would like to explain, in very brief terms, the usage of cryptographic processes, courtesy of RSA Data Security, Inc., Redwood City, California.

Secret Key Cryptography: This method uses a secret key to encrypt a message. The same secret key must be used again to decrypt the message. This means that the key must be sent along with the message that exposes it to whomever may be eavesdropping on the conversation. Secret keys are very fast in terms of processing, and it is not easy to break them, even though they are exposed through the communication process.

Public Key Cryptography: This method uses a combination of a modulus and a pair of exponents, called the public key and the private key. Exponents and modulus must be used together to encrypt or decrypt a message, but only the modulus and the public exponent are communicated since they are important to everyone who wants to send or receive encrypted messages using this method. The private exponent will never be publicly exposed. This ensures that no one else can decrypt messages that have been intended for a specified recipient, nor can anyone else, disguised as that recipient, intercept a message.

Encryption and Authentication: Encryption means that a message will be scrambled before it can be sent over a communications link. The plain message itself will never be sent in order to ensure privacy. Authentication is used to ensure that a message has indeed originated from the source that is specified in the message, and that the message has not been altered in transit. It additionally serves the purpose of non-repudiation, which means that whoever has digitally signed a message cannot claim later that he or she has not done so. In this case, the plain message itself will be sent since there is no need for privacy. The message will also be used to generate a digital signature by using one of the aforementioned cryptographic methods, preferably public keys.

Hash Functions A hash function is a computation that takes a variable-size input and returns a fixed-size string, which is called the hash value. If the hash function is one-way, that means hard to invert, it is also called a message-digest function, and the result is called a message digest. The idea is that a digest represents concisely the longer message or document from which it was

computed; one can think of a message digest as a digital fingerprint of the larger document.

The RSA Encryption Standard: This standard public key encryption method, along with the MD5 hash function, is used with the IBM DDNS products in OS/2 Warp Server. The principle of the RSA algorithm is as follows:

1. Take two large primes, p and q.
2. Find their product n = p x q ; n is called the modulus.
3. Choose a number, e, less than n and relatively prime to (p-1) x (q-1).
4. Find its inverse, d, mod (p-1) x (q-1), which means that e x d = 1 mod (p-1) x (q-1).

e and d are called the public and private exponents, respectively. The public key is the pair (n,e); the private key is d. The factors p and q must be kept secret or destroyed.

An example of RSA privacy (encryption) would be the following:

Suppose Alice wants to send a private message, m, to Bob. Alice creates the ciphertext c by exponentiating:

```
c = m^e mod n
```

where e and n are Bob′s public key. To decrypt, Bob also exponentiates:

```
m = c^d mod n
```

and recovers the original message, m; the relationship between e and d ensures that Bob correctly recovers m. Since only Bob knows d, only Bob can decrypt.

An example of RSA authentication would be the following:

Suppose Alice wants to send a signed document, m, to Bob. Alice creates a digital signature s by exponentiating:

```
s = m^d mod n
```

where d and n belong to Alice′s key pair. She sends s and m to Bob. To verify the signature, Bob exponentiates and checks that the message, m, is recovered:

```
m = s^e mod n
```

where e and n belong to Alice′s public key.

Thus encryption and authentication take place without any sharing of private keys: each person uses only other people′s public keys and his or her own private key. Anyone can send an encrypted message or verify a signed message, using only public keys, but only someone in possession of the correct private key can decrypt or sign a message.

To make encryption methods secure, a fairly large modulus should be chosen since it becomes increasingly difficult to break a large number into factors in order to determine the original primes. RSA uses a minimum length of 512 bits for the modulus, which would convert to a number with approximately 155 digits.

Due to security concerns, public key systems that use a key length of more than 512 bits must not be exported from the United States.

For encryption, in reality, RSA is combined with a secret-key cryptosystem, such as DES, to encrypt a message by means of an RSA digital envelope. Data Encryption Standard (DES) is one of the most widely used secret key algorithms and was originally developed by IBM.

Suppose Alice wishes to send an encrypted message to Bob. She first encrypts the message with DES, using a randomly chosen DES key. Then she looks up Bob′s public key and uses it to encrypt the DES key. The DES-encrypted message and the RSA-encrypted DES key together form the RSA digital envelope and are sent to Bob. Upon receiving the digital envelope, Bob decrypts the DES key with his private key, then uses the DES key to decrypt the message itself.

For authentication, in reality, RSA is combined with a hash function, such as MD5.

Suppose Alice wishes to send a signed message to Bob. She uses a hash function on the message to create a message digest, which serves as a digital fingerprint of the message. She then encrypts the message digest with her RSA private key (this is the digital signature) which she sends to Bob along with the message itself. Bob, upon receiving the message and signature, decrypts the signature with Alice′s public key to recover the message digest. He then hashes the message with the same hash function Alice used and compares the result to the message digest decrypted from the signature. If they are exactly equal, the signature has been successfully verified, and he can be confident that the message did indeed come from Alice. If, however, they are not equal, then the message either originated elsewhere or was altered after it was signed, and he rejects the message.

Note that for authentication, the roles of the public and private keys are converse to their roles in encryption, where the public key is used to encrypt and the private key to decrypt. In practice, the public exponent is usually much smaller than the private exponent; this means that the verification of a signature is faster than the signing. This is desirable because a message or document will only be signed by an individual once, but the signature may be verified many times.

5.7.2 DDNS Client to Server Interaction

When a DDNS client is initialized for the first time, it must be given the following information:

1. A hostname to be registered with a DDNS server
2. An IP address that goes along with that hostname
3. A default DDNS server to be updated with the given information

The hostname could be supplied by a DHCP server; it could be chosen by a user who observes the initialization process, or it could be obtained from a configuration file which has been supplied by a system administrator. It could also be contained in an existing name server, of course, but that does not have to be the case. The following discussion may be helpful in finding out which technique is most suitable for your installation.

Notes:

1. Using a DHCP server to supply hostnames in addition to IP addresses will relieve a user from any involvement in the TCP/IP configuration process of

his or her workstation. It will, however, place a significant burden on the administrator of the DHCP server. If a DHCP server would assign IP addresses dynamically and have hostnames go along with them, a user's hostname may change every time he or she starts TCP/IP. This will render electronic mail and other applications unusable. Moreover, if a DHCP server would store a fixed assignment of IP address and hostname per client, this could be considered a step backwards since there would be no difference in using BootP and a static Domain Name Server.

2. A better implementation of a DHCP server may, however, issue an inverse domain name query to a DDNS server to check if there is an existing mapping of a name to the IP address that the DHCP server is about to offer to a DHCP client. If this is the case, the DHCP server will include this host name in its offer, and the DDNS client can use it to update the DDNS server accordingly.
3. If a user can choose the hostname, it may already be in use and thus be rejected by the DDNS server. In this case, the user should be given one or more attempts to enter a hostname that is not already in use. In this case, the IP address and the DDNS server name can be obtained from a DHCP server easily. This method will leave a system administrator with little or no work, since client registration will be handled by the Dynamic IP software, and the configuration of the DHCP server can be rather generic.
4. Providing a configuration file to supply a hostname for DDNS initialization will give the system administrator the option to assign hostnames to workstations but still have IP addresses assigned automatically by DHCP. This will not impede electronic mail, for instance, since that hostname is not subject to change. This method could be used for electronic software distribution environments, and it will involve little overhead to system administrators since the response files for the client installation would have to be prepared anyway. The DHCP and DDNS server configurations could be rather generic, again.

The interaction between DDNS client and server, and the role of a DHCP server in that scenario, can be summarized as follows:

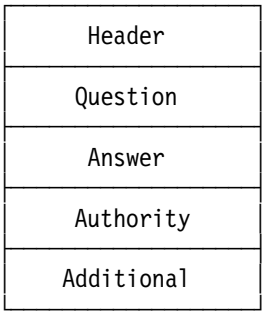
- Once the DDNS client has been provided with the required information, it will contact the name server by using the address that it has received from the DHCP server. A user may also provide this information, along with a hostname. The client will ask that name server for the name of the primary DDNS server for this zone or domain.
- The name server will send back the name of the primary DDNS server, which may be itself. It is also possible to run DHCP and DDNS servers on the same system.
- The DDNS client will then send an update request for the resource records which are associated with the client's hostname. If all goes well, the server will commit the changes to its database, and the client will be known to other hosts by the associated hostname.
- If the specified hostname is already registered in the DDNS database with a different client, the user will be notified to enter another name.
- Since DNS security is in place, the client will also send its public encryption key, and it will sign all resource records with a digital signature. The key and signature, together, will allow anyone to verify that it was indeed this client that created the records and that the information contained in the client's

records is valid. Only that client can, later on, make changes to those records. This means, that a client can register a hostname with the DDNS Server and that hostname is only available for that client. The hostname is reserved for the client even if the hostname is not used by the client. For the purpose of maintenance, a system administrator should also have the permission to change and/or delete any resource records in the DDNS database.

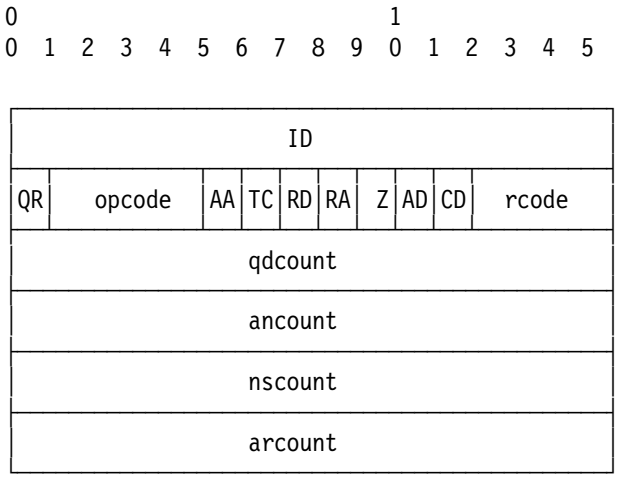
- Once the registration of a DDNS client is completed, other hosts may perform a hostname to IP address query for this client in order to send information to it.
- A name server should normally also support inverse queries, or IP address to hostname mappings, so the DDNS server must be updated with that information as well. In a Dynamic IP environment, a DHCP server can carry out the job of updating a DDNS server with the inverse address resolution information for a client. This is done in the following way: after the user has specified a hostname in the DDNS client configuration menu, and after the DDNS client has successfully registered that name with the DDNS server, the DHCP client will send a lease renewal request message to the DHCP server. The client will include the newly learned hostname in that message, thus indicating to the DHCP server that a DDNS update should occur for that information. The possibility of inverse name queries may also enable a DHCP server to find a hostname for a client that has not supplied one during initialization.

5.7.3 DDNS Message Format

DDNS uses the domain name message format, as defined in RFC 1035, which is shown in the diagram below:



The header section of a DDNS message is always present and has the following format:



The DDNS header format has added the AD and CD bits to the original DNS header format. The AD (authentic data) bit is used by a DDNS server to indicate that it has verified the data in a message. The CD (checking disabled) bit is used by a DDNS client to indicate that it will accept data from old DNS servers (non-verified data) as well as from secure DDNS servers.

DDNS introduces a new type of message, the *update* message. DDNS update messages have no section count fields, but a new opcode (5) and new return codes (6-10) that are not known to existing static DNS servers. The following types of update requests can be distinguished:

Table 18. DDNS Update Operations

Type	**Description**
ADDNAMENEW	Supplies resource records (RRs) with new names to be added.
ADDNAMEEXIST	Supplies RRs with existing names to be added.
ADD	Supplies RRs with new or existing names to be added.
DELETE	Specifies RRs to be deleted.
ZONEAUTHORITY	Supplies the SOA (start of authority) RRs of the zone to be updated.

A typical DDNS transaction involves one or more update requests to the DDNS database and the processing and adding of signatures for the resource records (RRs) that have been updated. Traditional DNS queries will, of course, not be subject to authentication.

5.8 OS/2 DDNS Server Configuration

The domain name server maps a hostname to an Internet address or an Internet address to a hostname. The domain name server is like a telephone book that contains a person's name, address, and telephone number. For each host, the name server can contain Internet addresses, nicknames, mailing information, and available well-known services (for example, DNS, SMTP, FTP, or TELNET).

When a client needs to communicate with another host, the client uses either the Internet address of the remote host or sends a query to the domain name system (DNS) for hostname resolution.

Before the name server can resolve a query, it must be supplied with resource records that either define a zone or a remote name server.

5.8.1 Types of Domain Name Servers

If a name server defines local data and has authority for that zone, it is called a *primary name server*. If a name server zone transfers a zone from a remote primary name server, it is called a *secondary name server*. Once a secondary name server receives zone data, it has authority for that zone. The secondary name server refreshes its zone data using the value defined in the zone's authority record.

A name server that does not have authority for any zone is called a *caching-only name server*. A caching-only name server must communicate with a remote name server in the Internet that has access to zone data. The OS/2 name server runs as a primary, secondary, or caching-only server.

5.8.2 DDNS and DNS Configuration

There is no explicit configuration utility for the DDNS server as there is for the DHCP server. You can either create new DDNS server configuration files, or you can migrate an existing DNS configuration to dynamic DNS server configuration files. In this section, we show you how this can be done in both cases.

The following are the three ways to use the OS/2 DDNS server:

1. Static DDNS server
2. Dynamic secure DDNS server
3. Dynamic pre-secured DDNS server

When used as a static DDNS server, there is nothing you have to do but use your existing DNS configuration files with the DDNS server. It will then work exactly the same way as the previous DNS server.

When used in dynamic secure mode, the DDNS server will allow clients to update their resource records dynamically using encryption keys that have been created by the clients themselves.

When used in dynamic pre-secured mode, the DDNS server will only allow those clients to update their records to which an encryption key has been provided that has been generated at the server.

5.8.3 Creating a New DDNS Server Configuration

Several files are needed to configure and administrate a DDNS server. The following table summarize these files:

Table 19 (Page 1 of 2). DDNS Server Files

File	Description
\TCPIP\BIN\NAMED.EXE	DDNS Server

Table 19 (Page 2 of 2). DDNS Server Files	
File	**Description**
\MPTN\ETC\NAMED.BT	The nameserver boot file that contains the path and file names for any other configuration files. It will be examined by the DDNS server at startup.
\MPTN\ETC\NAMED.DOM	The nameserver domain file that contains information about the zones for which this server will be authoritative, and all mappings from names to IP addresses (ordinary or forward name resolution).
\MPTN\ETC\NAMED.REV	The nameserver reverse file that contains information about the mappings from IP addresses to names (inverse or reverse name resolution).
\MPTN\ETC\NAMED.CA	The nameserver cache file contains information about other name server.
\TCPIP\BIN\DDNSZONE.CMD	The DDNSZONE command will also create the DDNS.DAT file that contains the private encryption keys to sign any updates to the zone resource records.
\MPTN\ETC\DDNS.DAT	Private keys of the server. This file should include the key stored in the DHCPSD.DAT file when the DHCP server dynamically updates the DDNS server.
\TCPIP\BIN\NSUPDATE.EXE	Updates the DDNS server
\TCPIP\BIN\NSSIG.EXE	Gives information about the status of the domain
\TCPIP\BIN\NSLOOKUP.EXE	Queries the name server
\MPTN\ETC\SYSLOG.CNF	Configuration of the DDNS servers log files.
\MPTN\ETC\SYSLOG	DDNS server log file.

Follow the steps below to create DDNS server configuration files from scratch. The following are the files required for a minimum configuration:

1. Create the \MPTN\ETC\NAMEDB directory, or create a NAMEDB directory under the directory where the ETC environment variable points to. Normally, this directory should have been created during OS/2 Warp Server installation.
2. Create the DDNS configuration files. Those files are plain ASCII files, so you can create them, for instance, with the OS/2 system editor. You can also modify the samples that are shipped with OS/2 Warp Server and contained in the \TCPIP\SAMPLES\ETC\NAMEDB directory. Normally, those sample files should also be found in the \MPTN\ETC\NAMEDB directory.
3. Create the DDNS server boot file. The boot file points the DDNS server to its database files. The database files, as explained later, hold the name and address information.

   On a primary DDNS server, the boot file contains one line for each file to be read. This line is comprised of four fields: the word primary, starting in the first column, the domain the server is authoritative for, the file name and the dynamic keyword, which specifies that the zone can be dynamically updated by DDNS client hosts. Also, the dynamic keyword can be qualified with either the keyword secured (the default), or presecured to specify the security policy for the zone.

   Additionally, the keyword nokeytosec can be specified as the last keyword to indicate that KEY and SIG records should not be transferred to secondary

name servers for the zone. The nokeytosec should be used when you are using a traditional, static DNS server as a secondary to a dynamic zone, and the static DNS server fails during a zone transfer because it fails to recognize the new KEY and SIG resource records used in dynamic DNS zones.

The DDNS servers cache file is also specified in the boot file. The cache file is not for general cache data. It only contains the root name server hints. The domain name "." refers to the root domain. The last field of the cache statement is the file name of the file, where the cache data is defined.

A name server boot file might look as follows:

```
; NAMED.BT file for name server configuration.
;
; type       domain                      source file or host
;
primary  itso.ral.ibm.com       d:\\mptn\\etc\\namedb\\named.dom   dynamic
;
primary  104.24.9.in-addr.arpa  d:\\mptn\\etc\\namedb\\named.rev   dynamic
;
cache    .                      d:\\mptn\\etc\\namedb\\named.ca
;
```

4. Create the domain file (\MPTN\ETC\NAMED.DOM). The NAMED.DOM file contains information about the name of the domain name server and its operating parameters. It then contains entries that map names to IP addresses for each workstation in the domain.

   Like a static domain name server the DDNS server knows two control entries. A control entry performs special functions within the cache file (NAMED.CA) and domain data files (NAMED.DOM and NAMED.REV). The following two control entries are defined:

   $ORIGIN This entry indicates that the specified domain name (a character string) be appended to each hostname in this file that does not end with a period (.). It resets the current origin for relative domain names to the stated name.

   ```
   $ORIGIN itso.ral.ibm.com
   ```

   $INCLUDE This inserts the specified file into the current file. The client can specify a domain name that sets the relative domain name origin for the included file. The use of $INCLUDE is optional.

   ```
   $INCLUDE c:\\mptn\\etc\\namedb\\named.inc
   ```

   There are special characters that can be used within the definition of a resource record:

   @ A free-standing @ denotes the current domain (origin).

   . One free-standing dot represents the null domain name of the root.

   \X A backslash followed by any character other than a digit denotes that the character (X) be used, not its special meaning. For example, in a mailbox specification, you can use a backslash followed by a dot ("\.") to place a dot in the local part of the name. The name server treats the dot as a normal character and not the end of the name.

( ) Parentheses group the data that crosses a line. Line terminations are not recognized within parentheses.

; A semicolon begins a comment. The remainder of the line is ignored.

All entries in the domain file are entered in a special resource record format. The resource record format is the basis for all entries in the cache file (NAMED.CA) and domain data files (NAMED.DOM and NAMED.REV). The resource record format is:

```
domain_name    ttl    address_class    record_type    record_specific_data
```

domain_name This can be either a hostname (for example, client1) or a reverse IP address (for example, 104.24.9.in-addr.arpa.). Instead @ can be used, which stands for domain specified in the boot file and is used in the SOA resource record.

ttl The ttl (time-to-live) indicates the number of seconds that a record is valid in a cache. The default is the ttl value contained in the SOA record.

address_class Specifies the address class for this entry. The allowable values are:

ANY The wildcard value ANY is defined to match any of these classes.

CHAOS CHAOS system (obsolete)

HESIOD HESIOD class

IN The Internet. Most Domain Name Systems (including TCP/IP Version 3.1 for OS/2 Warp Server) support only the Internet.

record_type Indicates the type of record for this entry. The following is a list of valid record types.

Table 20 (Page 1 of 2). Record Types

Type	Description
A	The address record contains the dotted-decimal IP address for the domain name identifying the record.
ANY	Any record type for the domain name.
AXFR	The query type used by secondary name servers to transfer all records in the zone (the query class is set to IN when using the AXFR query type).
CNAME	The canonical name record is used to provide alias or alternative information for a domain name.
HINFO	The host information record type contains a string specifying the CPU type and operating system of a node.
KEY	The KEY record holds the RSA public key component of the creator of the record. There is one KEY RR per host plus one zone KEY RR representing administrative authority for the zone.
MAINB	Any mailbox records for the domain name.

Table 20 (Page 2 of 2). Record Types	
Type	**Description**
MB	The mailbox record contains the domain name of a host workstation to receive mail for the user specified in the domain name field.
MG	The mail group member record specifies the mail address of a client belonging to the mail group specified in the domain name field.
MINFO	The mailbox information record specifies the mail address of the client responsible for the mail group specified in the domain name field.
MR	The mail rename name record specifies a mailbox that is a rename of the mailbox specified in the domain name field.
MX	The mail exchanger record identifies a host capable of acting as a mail exchange for the domain specified in the domain name field. A mail exchange runs a mail agent that delivers or forwards mail for the domain name specified in the first field.
NS	The name server record contains the domain name of a name server for the current zone.
NULL	The null resource record contains any information.
PTR	The domain name pointer record is mainly used to store data for the in-addr.arpa domain, and contains the domain name referenced by an IP address.
SIG	The SIG record holds an RSA digital signature, which is generated at the client and included in a dynamic update to the DDNS server. There is one SIG record for each type of RR. The DDNS server uses the SIG RR along with a host's KEY RR to verify that the originator of an update message was in fact the owner of a resource targeted by the updated message.
SOA	The Start of Authority record is unique to a zone. This record contains the administrative details of the zone.
TXT	The text string record contains descriptive text.
WKS	The well-known services record stores the protocol number of multiple services in a single record. Each of the defined TCP/IP services has a unique protocol number.

record_specific_data Contains information appropriate for the data type indicated in the data type field, in the format defined for that specific data type. See the data type description to determine its specific requirements.

Table 21 (Page 1 of 4). Record Specific Data	
Record Type	**Data Entered After Record Type**
A	Dotted-decimal IP address.
ANY	Any record type.
AXFR	Query type (the query class is set to IN when using the AXFR query type).
CNAME	The CNAME characters are followed by the canonical name.
HINFO	The CPU characters are followed by the CPU information.

Table 21 (Page 2 of 4). Record Specific Data

Record Type	Data Entered After Record Type
KEY	Record specific data: **flags** This field indicates the type of resource record for which this KEY RR is provided. **protocol** This field indicates the protocols (in addition to DDNS) that are to be secured for authentication by this KEY RR. **algorithm** This field indicates what encryption algorithm should be used with this key; in case of IBM Dynamic IP this field has a value of 1 which means that the RSA/MD5 algorithm is being used. **public key** The actual public key to be used for authentication. This field is structured in a public exponent length field, the public key exponent portion, and the public key modulus portion.
MAINB	Any mailbox records.
MB	Hostname.
MG	Mail address.
MINFO	Mail address.
MR	Mail rename name.
MX	Hostname.
NS	Domain name.
NULL	Text with less than 65535 octets in length.
PTR	Domain name.

Table 21 (Page 3 of 4). Record Specific Data

Record Type	Data Entered After Record Type
SIG	**type covered** This field indicates the type of RR covered by this signature. **algorithm** This field indicates what encryption algorithm should be used with this key; in case of IBM Dynamic IP this field has a value of 1 which means that the RSA/MD5 algorithm is being used. **labels** This field indicates the number of labels (host and domain name strings separated by dots) in the SIG owner name. **original** TTL The original time to live for the signed resource record is included in order to avoid caching name servers to decrement this value. This value is protected by the signature, and it is different from the TTL of the SIG record itself. **signature** The time until this signature is valid. This value is represented in a expiration number of seconds starting from 1 January 1970, GMT (ignoring leap seconds). **time signed** The time when this signature has actually been signed, represented in the same format as mentioned above. **key footprint** This field determines, depending on the applicable encryption algorithm, how to decode the signature. **signer′s name** The fully qualified domain name of the signer generating this SIG RR. **signature** The actual digital signature that authenticates an RR of the type indicated in the type covered field.

Table 21 (Page 4 of 4). Record Specific Data

Record Type	Data Entered After Record Type
SOA	• The domain name of the name server responsible for the zone. • The mail address of the user responsible for the zone. • The serial number of the zone database, which identifies the current revision of the data. • The refresh interval, which indicates the length of time (in seconds) a secondary server for this zone should allow between refreshes from the master server. • The retry interval, which indicates the length of time (in seconds) a secondary server for this zone should allow before retrying a failed refresh. • The expiration time, which indicates the length of time a secondary server should consider its data valid in the event it is not able to contact the master server. If there is no contact with the master server prior to the expiration time, the secondary server will consider its data stale, and it will no longer respond to queries for this zone. • The minimum time to live (ttl), which is attached to all data given in response to a query or a secondary transfer, and determines the amount of time the data may be cached. • The IncrTime parameter specifies the time, in seconds, by which the zone serial number must be updated after an update has been committed. • The DeferUpdCnt parameter specifies the maximum number of update requests that could be processed before updating the zone serial number.
TXT	Descriptive text.
WKS	Unique protocol number.

A name server domain file might look as follows:

```
;******************************
;*  Start of Authority Records  *
;******************************
;
@   IN  SOA  dhcpsd.itso.ral.ibm.com.   kwichman.raleigh.ibm.com. (
        42       ; Serial number for this data (yymmdd##)
        86400    ; Refresh value for secondary name servers
        300      ; Retry value for secondary name servers
        864000   ; Expire value for secondary name servers
        3600     ; Minimum TTL value
        300 )    ; dynamic update increment time
    IN  NS   dhcpsd.itso.ral.ibm.com.
;
kwichman       IN  A   9.24.104.10
;
lnotessv       IN  A   9.24.104.36
wtrn1its       IN  CNAME lnotessv
```

5. Create the reverse file \MPTN\ETC\NAMED.REV. The NAMED.REV file defines a special domain called in-addr.arpa to translate IP addresses to hostnames. The format of an in-addr.arpa name is the reverse octet order of an IP address concatenated with the in-addr.arpa string. The in-addr.arpa name has four labels, which correspond to the four octets of an IP address. All four octets must be specified. For instance, the IP address 9.67.30.143 would be represented as 143.30.67.9.in-addr.arpa in the NAMED.REV file.

   A name server reverse file might look as follows:

```
;*******************************
;*  Start of Authority Records  *
;*******************************
;
@   IN  SOA  dhcpsd.itso.ral.ibm.com.   kwichman.raleigh.ibm.com. (
        42       ; Serial number for this data (yymmdd##)
        86400    ; Refresh value for secondary name servers
        300      ; Retry value for secondary name servers
        864000   ; Expire value for secondary name servers
        3600     ; Minimum TTL value
        300 )    ; dynamic update increment time
    IN  NS   dhcpsd.itso.ral.ibm.com.
;
; Addresses for the canonical names;

10     IN   PTR  kwichman.itso.ral.ibm.com
36     IN   PTR  lnotessv.itso.ral.ibm.com
```

6. Start the DDNS server by typing NAMED or double-clicking on the **DDNS server** icon.

7. Enter DDNSZONE to create the public encryption key pairs for the zone resource records in the domain and reverse files. Ignore any messages from DDNSZONE command, that the DDNS server should not be started.

   After the DDNSZONE command has processed the files, they look as follows:

   - NAMED.DOM file:

     ```
     ;*******************************
     ;* Start of Authority Records   *
     ;*******************************
     ;
     @ IN KEY 80 0 1 AQPOzUYWvAUyZhYxogDcrtxOZOH33V31Tmrs1Db1WYiyI4Y7Mmoz6Vm3XY/QTMHOyeHcVAMKmuba+rW4/+IkMeP3
     @   IN  SOA  dhcpsd.itso.ral.ibm.com.   kwichman.raleigh.ibm.com. (
             42       ; Serial number for this data (yymmdd##)
             86400    ; Refresh value for secondary name servers
             300      ; Retry value for secondary name servers
             864000   ; Expire value for secondary name servers
             3600     ; Minimum TTL value
             300 )    ; dynamic update increment time
         IN  NS   dhcpsd.itso.ral.ibm.com.
     ;
     kwichman       IN  A   9.24.104.10
     ;
     lnotessv       IN  A   9.24.104.36
     wtrn1its       IN  CNAME lnotessv
     ```

   - NAMED.REV file:

     ```
     ;*******************************
     ;* Start of Authority Records   *
     ;*******************************
     ;
     @   IN KEY 80 0 1 AQPR+3ObXCgcjmlBfKSnN4fD6vVH/AUIwincGNeD1MAuz2BTQSQ/bJkXLA3nxfV+HxKfxWp
                       tkRckwzxEk1DD3DSB
     @   IN  SOA  dhcpsd.itso.ral.ibm.com.   kwichman.raleigh.ibm.com. (
             42       ; Serial number for this data (yymmdd##)
             86400    ; Refresh value for secondary name servers
             300      ; Retry value for secondary name servers
             864000   ; Expire value for secondary name servers
             3600     ; Minimum TTL value
             300 )    ; dynamic update increment time
         IN  NS   dhcpsd.itso.ral.ibm.com.
     ;
     ; Addresses for the canonical names;
     10     IN   PTR  kwichman.itso.ral.ibm.com
     36     IN   PTR  lnotessv.itso.ral.ibm.com
     ```

   - Create the cache file (NAMED.CA). The name server needs to know the servers that are the authoritative name servers for the root domain of the network. To do this we have to prime the name server's cache with the addresses of these higher authorities. The location of this file is specified in the NAMED.BT file. This file uses the Standard Resource Record Format. An example for a cache file is the following:

```
; define parent(root) domain nameserver (Note trailing dot)
;
 .                  99999999 IN     NS      leda2.cwp.ibm.com.
leda2.cwp.ibm.com. 99999999 IN      A      9.14.1.3
```

- The DDNSZONE command will also create the DDNS.DAT file that contains the private encryption keys to sign any updates to the zone resource records in the domain and reverse files. This is shown in the following example:

```
itso.ral.ibm.com ns-updates.itso.ral.ibm.com
Pb7bySIfzXcWIxQ131Oc9xOW6aotedZP35y/q4QzPQEpZQb2l5NoMbjOF1r/uA7AuQUmF2y5bcDa+gEoh7wPsXlEhZOv1Hn4Dw3Od6G/9ej
G7JfugtWhjGi2ucsd1CxFLPkUaJPNQ6gkSDPsPCVNqWrp8O8A44QxROokSJbuq67wozjl74OUOHl9irqkQR/xUKAbEB53p6TpSYrdK4C4JQ
JvrSiEKhk5ntukqiOPUH2458TsSCUOhYSX+MOsez8HSvht0+n+7HjtpTYSI2S6AG/jXuMkFjxMxUMZAOHKpky5+LPCj8f+veKa9zJ274GPC
lRxtBSmgwOjVTyOOylGtAkyK7Fd1V2TPAd+4bT4vJasi2km2kGF9L+cYEVV/hoTSWwwjs2nOVwfCNy1D/99vKM2CeBR4bjRZkmhsYHM8a3q
b7er4ndMcNgmuZCaXJtq4THZp UdNczoPAQPS5nBuY34O4dOkWsDcjsvQSwpAKMIGNaGphB+xNKNTPsf9DMy8Lx65OxQ16IcGwH/hO33VgM
5CWyl3EOWDVmqz

24.1O4.9.in-addr.arpa ns-updates.itso.ral.ibm.com
KlexSRMP/q/kkbpWEDZK9xXdBQAJNGpVktyU3QV8hAETUWDZ+uCj2siYOzYzfJKb9DhNgVVltSGyC9IE/UV+3RrPK2XvFQbEwdKdM6klUBK
QXY8ymczE3JAYeJyUedCg7OO4GLAoZDMGsxj5cb/sPGadizODQ6V4HteO+Cr211N22azL5/ee8QktdOpLtmLzmZmDu94qF9i4JcyNHij1Pg
IFbJhl9UGMxmhTyzYZJUIcMwc1c1M41oUwnCNDRhON+X7TuObYfYX6wE3FsRkVADoOmn4hosLshF4VPeYOg59Z5zKS+H2a8JAZrMZfQcs7g
fi1LsV3x+PHTiie/7Mzah51ChY6a2DRI7OdSBOamfDRmbBCCxfHAlAUE2W8yxXsvKJD+auirBWygBT6B/9ZNUp6NnMrWPIcON6MTXsxpU27
1B41USVn6vp9MTXziyai+EAa2 5QMKRA=AQPE7HDQaltur1bT7Zv1nEP7318TJXv82rZX67rdVzew3Ts++KQ/ggimUPk/EodzISfYfhEyNb
DcgIno9aAbqqs7
```

8. If you have a DHCP server configured for DDNS updates, you need to add the information from the DHCPSD.DAT file to the DDNS.DAT file. The DHCP.DAT file will be created by using the graphical DHCP configuration program. Select **File** and then **Update DDNS data file** from the menu. The DHCP.DAT file looks something like the following:

```
*.1O4.24.9.in-addr.arpa 9.24.104.1O5
UkrFyWVpIAK47E7x7tU899UvLWnWqiipyhqFX2kWCQEyUNy2Z2LxYYBMtdSxJJxLDUG6u5MF2hWryL+RYK3SJgUA4JaoaTLf+WDuVYadYWk
gOjrOfuXGqyPHWirowX2t5v92j37kr8N7Xgt/glVaU1gGkqraPH2NIULsYviOGDEEp7MQJbFOSyo83CfsvcSkbsUf2/iMjIXokx726x2Flm
Ex9lXt1kRBgqai15PlJ6vEQcUDVkm4vOHwffGAmA7I2kD3pLY3vkip246/1OViALMuzipuIOwzpLIb9rJyOCaUcKy+5ZQP2e/cRXAlliX3V
T4fLoYN64TdSjH1p95qr2jggU+dyEFSK/OEOx6m6YZ/OG26sDTcH9SQZ2RjoPNEjcgOesemlQEhUHshcmHfRj8AnnC1gVyKd/bjt524j+oo
swbxVCeQ7niTV7UjGlYpajnAK 5ia65U=AQPEcHDCeX13WWKXfEvONnBeiB5pJ/7KkeA1dU8m1bDp626v39CtM//C3EPiabXA2hCof1MKfO
isH8yhE6367ckj
```

9. Finally, copy the SYSLOG.CNF file from the \TCPIP\SAMPLES\ETC\NAMEDB directory to the directory that contains the name server files. This file configures the logging options for the DDNS server and will also be examined at server startup. Normally, it should be there already. The SYSLOG.CNF file may look like the following example:

```
##############################
# system log configuration file #
##############################
#
#   Here is a list of all the keywords whose value can be specified
#   in this file:
#   Keyword          Effect
#   -------------    ------------------------------------------------------
#   numLogFiles      The number of log files desired.
#   logFileSize      The Size of log files in K bytes.
#   logFileName      The name of the most recent log file.
#   logItem          One item to be logged.
#
#   logItem                    One item that will be logged.
#               LOG_EMERG      system is unusable
#               LOG_ALERT      action must be taken immediately
#               LOG_CRIT       critical conditions
#               LOG_ERR        error conditions
#               LOG_WARNING    warning conditions
#               LOG_NOTICE     normal but signification condition
#               LOG_INFO       informational
#               LOG_DEBUG      debug-level messages
#
#
numLogFiles     4
logFileSize     1OO
logFileName     syslog.
logItem         LOG_EMERG
logItem         LOG_ALERT
logItem         LOG_CRIT
logItem         LOG_ERR
```

```
logItem         LOG_WARNING
logItem         LOG_NOTICE
logItem         LOG_INFO
```

Note: KEY and SIG resource records as well as encryption keys always use a single line. We have indented the examples for illustration purposes only.

5.8.4 Migrating an Existing DNS Configuration to Dynamic IP

Before you migrate a name server from static DNS to dynamic DDNS you should decide if you want to:

- Leave existing resource records as they are and allow new ones to be created and updated dynamically. This will allow existing systems to keep their hostnames, but they will not be able to update their resource records dynamically unless a system administrator deletes them.
- Delete all existing resource records and start with a dynamic domain from the beginning.

Follow the steps below to migrate existing DNS server configuration files to Dynamic IP.

1. Modify your existing DNS configuration files (NAMED.BT, NAMED.DOM, NAMED.REV) to resemble the files as shown in the example above (before the DDNSZONE command has been run). In the case of a NAMED.BT file, you have to remove the domain statement, and you have to add the dynamic or dynamic secure keywords to the primary statements for the authoritative DNS server that you are upgrading.
2. Start the DDNS server and ignore any messages in the following DDNSZONE command that might instruct you to stop the server.
3. Use the DDNSZONE command to create the encryption keys.
4. If you have a DHCP server configured for DDNS updates, you need to add the information from the DHCPSD.DAT file to the DDNS.DAT file. The DHCPSD.DAT file will be created by choosing **Update DDNS data file** from the File menu of the DHCP server configuration program.
5. Copy the SYSLOG.CNF file to set DDNS server logging options.

5.8.5 Dynamic DNS Server Administration

To start the OS/2 DDNS server, double-click on the appropriate icon in the DDNS Services folder that is shown in Figure 33.

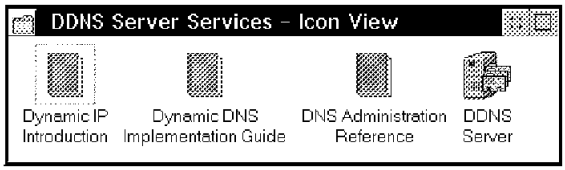


Figure 33. DDNS Services Folder

Likewise, you can start the server by entering the following command on an OS/2 command prompt:

```
NAMED
```

The following figure shows the OS/2 DDNS server program.

```
┌──────────────────────────────────────┐
│ IBM OS/2 Warp Domain Name Server (NAMED) │
│          TCP/IP Version 3.1          │
└──────────────────────────────────────┘

  bootfile = F:\MPTN\ETC\NAMEDB\NAMED.BT
```

The DDNS server performs Dynamic DNS database updates in the appropriate domain file as the updates occur. Therefore, do not edit domain files for dynamic zones while the DDNS server is running. Further, dynamic domains cannot be dynamically reinitialized with new configuration information using the traditional nssig -HUP command which is explained in more detail later.

Also, note that if you enter comments into a domain file for a dynamic zone, the comments will be deleted when the first update to the domain is made. Domain file comments are not maintained because they would degrade the performance of the file update process.

You may change the configuration information for a dynamic domain in the following different ways:

1. Manually, by editing the domain files only after shutting down the DDNS server.
2. Dynamically, by using nsupdate while the DDNS server is running.

When you first set up your dynamic domain, you used ddnszone to create a ″zone key″. ddnszone created the zone key RSA key pair, put the public key in the appropriate domain file, and put the private key in the ETC\DDNS.DAT file. The zone private key stored in ETC\DDNS.DAT is used by nsupdate when signing update requests for the administrator of the zone. The server then examines the signature to identify update requests from the zone administrator versus those from ordinary hosts. The zone key gives the possessor the power to use nsupdate to create, modify, and delete any host′s record in a dynamic domain.

Once the zone key information is generated and the DDNS server started, the administrator can take the ETC\DDNS.DAT file with him or her and administer the zone remotely using nsupdate.

You can use nsupdate in an interactive fashion where you are prompted through a series of subcommands and associated input values to create and execute DNS update operations on a host record. Alternatively, if you know the sequence of operations and input values beforehand, you can use nsupdate in batch mode and specify a subcommand sequence in the -s <subcommand string> command-line parameter. You have the following commands:

```
optional parameters are:
  -kkeyfile -hhostname -ddomainname -pprimaryname
  -rIPaddress(for in-addr.arpa hostname) -s″command string″
switches:
  -a administrator mode
  -g key generation mode
  -q quiet (no prompts)
  -v verbose output
  -? display help
```

The following are example console sessions using nsupdate in interactive mode. All examples assume the administrator has already set up zone key and that the private key component for the zone is included in the local ETC\DDNS.DAT.

How an Administrator Removes and Locks Out a Hostname: The following example demonstrates an administrator's input and the system responses when removing and locking out a hostname.

1. Generate a new key for the host

```
[C:\]nsupdate -g -h warpspeed.dynozone.sandbox -p netadmin.dynozone.sandbox
--- NSUPDATE Utility ---
---
Key Gen ..... succeeded ...
```

2. Delete a user's A and KEY RRs and add a new KEY RR for new, administrator generated key

```
[C:\]nsupdate -a -h warpspeed.dynozone.sandbox -p netadmin.dynozone.sandbox
--- NSUPDATE Utility ---

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> DELETE
---
InitDDNSUpdate ..... succeeded ...
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): a
....ip addr: *
DDNSUpdate_A (Delete *) ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> DELETE
---
InitDDNSUpdate ..... succeeded ...
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): key
DDNSUpdate_KEY DELETE *
succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> ADD
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): key
DDNSUpdate_KEY (Add Flags 0000 Protocol 0 Algid 1
      Keylen 64 Key[0-15]: AQPS8Oe7uGuuNIdA ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> SEND
..sig Expiration (secs from now, ENTER for 3600):
..sig KEY pad (ENTER for default of 3110400):
DDNSSignUpdate ...succeeded
DDNSFinalizeUpdate ...succeeded
DDNSSendUpdate ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> QUIT
```

How an Administrator Creates an Alias for the Dynamic Zone: The following example demonstrates an administrator's input and the system responses when creating an alias for the dynamic zone.

```
[C:\]nsupdate -a -h ns-updates.dynozone.sandbox -p netadmin.dynozone.sandbox
--- NSUPDATE Utility ---

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> ADD
---
InitDDNSUpdate ...... succeeded ...
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): cname
....hostname: netadmin
DDNSUpdate_CNAME (Add netadmin.dynozone.sandbox) ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> SEND
..sig Expiration (secs from now, ENTER for 3600):
..sig KEY pad (ENTER for default of 3110400):
DDNSSignUpdate ...succeeded
DDNSFinalizeUpdate ...succeeded
DDNSSendUpdate ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> QUIT
```

The nsupdate command is used to create cryptographic keys and to apply digital signatures. It is used by both the DHCP server and DDNS client.

To query the DDNS server database, use the following OS/2 command:

```
NSLOOKUP hostname|ip address
```

This command works like a shell and allows you to perform subsequent queries on a name server.

To view the status of the DDNS server, or to take a dump of the DDNS server's database, use the following OS/2 command:

```
NSSIG
```

NSSIG can be called with the following parameters:

-HUP Reads the NAMED.BT file from the disk and reloads the domain name server's database. All previously cached data is lost. This is useful when you have made a change to a data file and you want NAMED's internal database to reflect the change.

-INT Dumps the domain name server's database and cache file to the MPTN\ETC\NAMEDB\NAMED.DMP file. You can then look at the file to see whether the database was loaded correctly. Refer to the sample NAMED.DMP file for a representative dump file.

-IOT Dumps the status of the domain name server to the MPTN\ETC\NAMEDB\NAMED.STS file.

-USR1 Increases by 1 the level of debugging messages displayed by the domain name server.

-USR2 Stops the domain name server from displaying debugging messages.

Note: With the former, static, version of the OS/2 DNS server, NSSIG could be used to reload the name server database without taking the server down. This cannot be done with the DDNS server anymore.

5.9 The DDNS Client

The actual OS/2 client programs for DHCP and DDNS are supplied with Adapter and Protocol Services, at least as far as the OS/2 Warp Server product is concerned. Those clients will also be available to other OS/2 systems as a software upgrade to TCP/IP 3.0 for OS/2 product or component.

To set up your host as a DDNS client you have to enable DDNS in your TCP/IP Configuration folder.

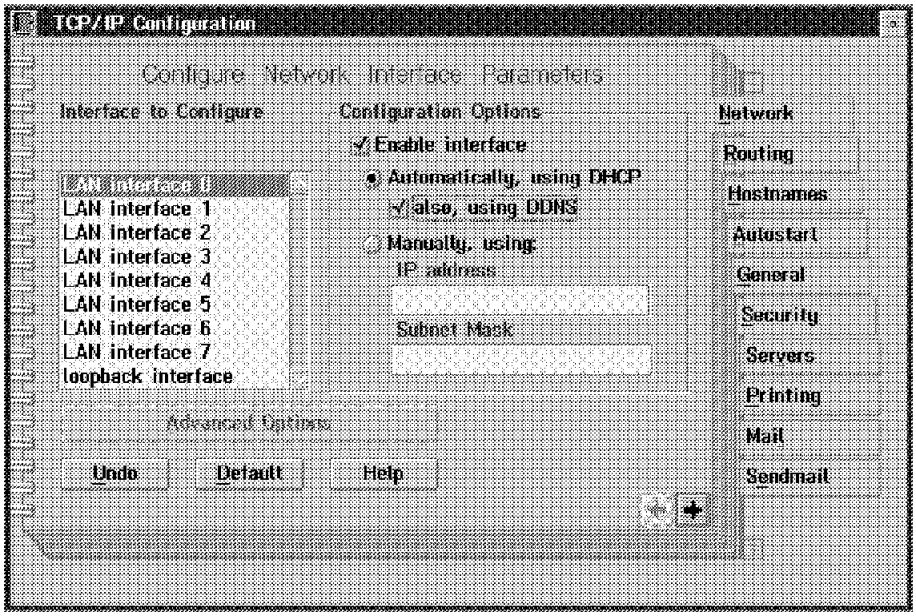


Figure 34. DDNS Client Configuration Program (1 of 2)

When you initialize Dynamic IP for the very first time on your workstation, and a DDNS server will be used for name resolution, a hostname for your workstation must be supplied. This can be done in the following ways:

1. A hostname is statically defined in the name server. In this case, your hostname changes whenever you receive a different IP address from the DHCP server. With Dynamic IP, this should not be an option.
2. The DHCP server supplies a hostname along with an IP address. This places a burden of work on the system administrator, and it also means that your hostname changes when the IP address changes. That should not be the case, especially when electronic mail or NFS are being used.
3. You can choose a hostname by yourself.

In the latter case, the DDNS client configuration program will be used, as shown in the following. If the name you specify already exists, the name server will notify you, and you must select a different name.

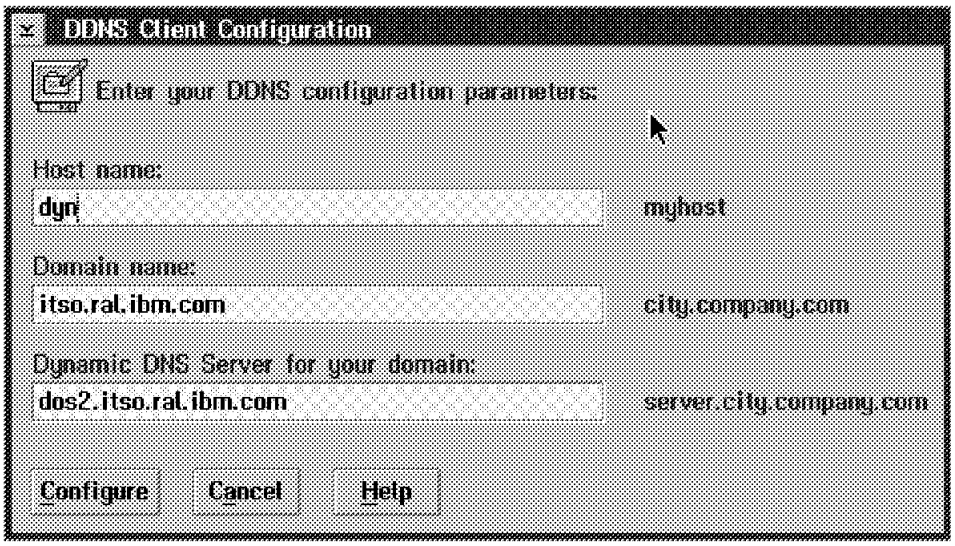


Figure 35. DDNS Client Configuration Program (2 of 2)

The first time you start your DDNS Client Configuration program (DDNSCFG.EXE) to register with a hostname, a public and a private encryption key will be created. Your private key will be stored in your DDNS.DAT file. For every hostname that is registered or reserved for you, you have one entry in the DDNS.DAT file. In the following example a DDNS.DAT file is shown with two private keys, one for each reserved hostname:

```
dyn1.itso.ral.ibm.com dos2.itso.ral.ibm.com
2K9SxqCef7pKToU2/7oK98T5wX1Q9qYH1B1GAQD/iwH34bI7XAaX4CgewOagDf/J2PMzODYdT2mIJQuYHNDQeGjUm4Y5WcJU6qb/3zsdgoB
68XSVDAL8jb79P3eggy+ja2TUjTqZM3HWehxWSdOUNA56T1irouxkHjhuOgwNDR/1SRu8+mZg/HFO7U7jdKbW2wxNh/5Ae5IaGeptFraf76
3aPM/ZGhuZOY3ko/+N4Xgx/rnjmdJdkKmvYi5DKHHZ5TPc96E6V7svp7lOUu3nAAr9IZljcJnzYBARylLvJZLMszNzEjr5JpT+iwW5zBJDB
RQjhUi/OQkpbBtrKKeatDvOLxCNilmGpuerokVUnTvtahJAd5dd9CLqxPctbTWfplR2qC7ogZFJIyWZO17X+TcbNAedHyBM/i/ww7h/hyzZ
7oeORJ4QsgOFzCGYbh9fbE6YR 5maDZO= AQPeStP81U9YZXuOlcm2QCvOv6HgjGYvZqJGYcZ4qtLnvF5Z/RB1TNOf4afygMVOD27kpgaWw
tCEq/JXWKOylPD9

dyn2.itso.ral.ibm.com dos2.itso.ral.ibm.com
5GzoWYVSLPlQIqfrBloa96BJGIX9IcQFlQbxpBvgxgFpFxcx/ToCDrvECuAZt7ANe9wd5hKQiy51ymEAfov5cfvAfFDw4FOxSGPJ+fIrWzr
6feAYARTs94l/1pjd3PK8gDMpGobsxNUzYjgWWFNGelG4QQBfJvOL7FJxcMAHOWkaSsjLPYXhHE5PcnR1SW9GF8KhJewslZwYwUcC9+VVbl
QTflEw5B2jPOE2Az8snkoRRoLSLIbgdBnDvglmG81smjcgA1g+qyqdCfH+2oUrALEVXLjpNQ3ofSYXyHJipMfsaVLHhT6Tcpwxq9QrkGyx8
yZYBkwgJmOyCvJYFV16OfgUC+eT8S9+7O5rbBGm3vrKhsDZOnpwlwhwEfIfmsybFoS5Rcw+9Lf2Q7GndSDsXqylSvFCtaZvI2lidLHIlH6x
pz7c7+R2WKvXT+QJQkjQsA3dM7qhSsri AQPHJiMS1rdX2MwmIQ3JNNGmQeAO9Swd6kVCpsdx+5JyNsgwlhTym2fxcBCWOtMiQqtjPrjlYk
JZxIWSWbw6Ueuj
```

The name server stores your name and the IP address that has been supplied by a DHCP server. If that address changes later, the DDNS client and DHCP server will simply update the records in the name server which should not involve any user interaction.

The following statement in the DHCP client configuration file includes the command that is sent to the DDNS server to update a client's A record for name resolution:

```
updateDNS "nsupdate -h%s -d%s -s"d;a;*;a;a;%s;s;%s;3110400;q" -q"
```

The %s variables will be evaluated by the DDNS client as follows:

1. Hostname
2. Domain name
3. IP address
4. Lease time

After updating the DDNS server, the server stores the new information in the domain and reverse file. The updated domain record could look like the following:

```
$ORIGIN ral.ibm.com.
itso            IN      KEY     0x0080   0  1 1AQOwCZ8xspbyX+ZTYlO7JgjDMlNebK6+OIENAZXGud7J
                                             syaA6lOoOeeW5ozihKelLrcrFIiqTqaAOSo3MbWxPoAP      ;Cl=4
                IN      NS      dhcpsd.itso.ral.ibm.com.  ;Cl=4
                IN      SOA     dhcpsd.itso.ral.ibm.com. kwichman.raleigh.ibm.com. (
                                             1555 86400 300 864000 3660 300 );Cl=4
$ORIGIN itso.ral.ibm.com.
dyn1    4660    IN      SIG     A 1 4 4660 822589163 822588983 0x8eb6 dyn1.itso.ral.ibm.com
                                mj1yGt45snrOBXK9ON1h5Q4KZ91n/WA6ypXQqdi8DWeZdpNboVdHWFtqadRmdTDdhFlOHC99sp7en
                                DXhXpwfvA==;Cr=auth
        4660    IN      SIG     KEY 1 4 4660 825699563 822588983 0x8eb6 dyn1.itso.ral.ibm.com
                                Wt2ML44646Fk3tR3S5JOGVtiJVFsgq9oKhRZaQaml+51P1kk9ZTuPKZkRNgnqj1PaUKRIwzeJiFdp
                                iieFKEnIQ==  ;Cr=auth
        3600    IN      KEY     0x0000   0   1 AQOwXqFgb1yOCDrPw4fLECcEJKlnnf1vFFT2QDuj//T/2PPaQhYYFtKCfLeJIN
                                C72X4bgKgMjix9TNKJQVc1to4v      ;Cr=auth
dyn2    3600    IN      KEY     0x0000   0   1 AQPeStP81U9YZXuOlcm2QCvOv6HgjGYvZqJGYcZ4qtLnvF5Z/RB1TNOf4afygM
                                VOD27kpgaWwtCEq/JXWKOylPD9      ;Cr=auth
        4660    IN      A       9.24.104.162   ;Cr=auth
        4660    IN      SIG     A 1 4 4660 82260435 822603855 0xf094 dyn2.itso.ral.ibm.com
                                f61vP+ysk/FFeglULtUCbJTv9KjcHFlWVHxfv2wC7WzJ/bQNrkz5vIZxKUaO7kcD/x7/TnZl1gYcS
                                LrGAh1cFw==  ;Cr=auth
        4660    IN      SIG     KEY 1 4 4660 825714435 822603857 0xf094 dyn2.itso.ral.ibm.com
                                OWbxVHNd3EXy/qaYVUzrsGk9OrYtuIpM8zBQ7ChlYHmFBtWDoWOSfAoKc3tfpuqf/9W9FY7fkyrvI
                                jwXV4UFCQ==  ;Cr=auth
kwichman IN     A       9.24.104.10     ;Cl=4
lnotessv IN     A       9.24.104.36     ;Cl=4
wtrn1its IN     CNAME   lnotessv        ;Cl=4
```

When you are using DHCP, the registered hostname appears in the DHCP monitor.

5.10 Dynamic IP Scenarios

The following sections provide you with some useful examples on how to set up Dynamic IP for use in different environments and different platforms.

5.10.1 Simple Operational Scenario

This section provides an example of a very simple Dynamic IP scenario involving only a client and a server on a single IP subnet.

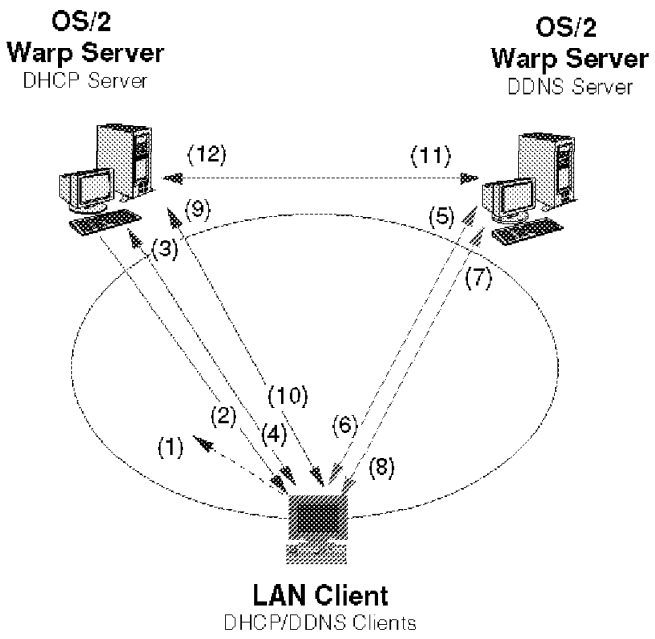


Figure 36. Simple Dynamic IP Scenario

For a very simple scenario, the DHCP and DDNS servers in the figure above would be on the same OS/2 Warp Server system. We have separated those functions only for a better illustration of the Dynamic IP operation.

The following steps describe the Dynamic IP operation in the scenario shown above, when the client is started for the first time:

1. DHCPDISCOVER, broadcast by the DHCP client.
2. DHCPOFFER, sent by the DHCP server containing configuration options.
3. DHCPREQUEST, broadcast by the DHCP client.
4. DHCPACK, sent by the DHCP server.
5. DNS query for primary name server, sent by the DDNS client.
6. DNS authoritative reply, sent by the DDNS server.
7. DDNS update query, sent by the DDNS client containing the hostname specified by the user.
8. DDNS acknowledgement, sent by the DDNS server.
9. DHCPREQUEST lease renewal, sent by the DHCP client supplying the hostname specified by the user.
10. DHCPACK, sent by the DHCP server.
11. DDNS update query, sent by the DHCP server to update the DDNS database with the inverse mapping information for the hostname and IP address.
12. DDNS acknowledgement, sent by the DDNS server.

The following example shows the DHCP server and DDNS server configuration files used in this scenario:

- DHCPSD.CFG file (DHCP server):

```
numLogFiles4 logFileSize 10
logFileName dhcpsd.log
leaseTimeDefault1 hours
leaseExpireInterval 10 minutes supportBOOTP no
supportUnlistedClients yes
logItem SYSERR logItem OBJERR
logItem PROTERR
logItem WARNING logItem EVENT
logItem ACTION
logItem INFO
logItem ACNTING logItem TRACE
#.indent 12
updateDNS "nsupdate -f -r%s -s"d;ptr;*;a;ptr;%s;s;%s;0;q""

network 9.24.104.0 9.24.104.20-9.24.104.165 #.name Test Network
{ #.ddns 9.24.104.105
client 0 0 9.24.104.100     #.exclu
client 0 0 200.200.200.105  #.exclu
option 3 9.24.104.1         #.name 3 Router
option 6 9.24.104.105       #.name 6 Domain Name Server
option 15 itso.ral.ibm.com  #.name 15 Domain Name
option 201 9.24.104.100     #.name 201 - Gopher Server}
```

- DHCPCD.CFG file (DHCP client):

```
# Basic options required
clientid MAC
interface lan0

# Uncomment as desired for logging
numLogFiles4
logFileSize 100 logFileName dhcpcd.log
logItem SYSERR
logItem OBJERR
logItem PROTERR logItem WARNING
logItem EVENT
logItem ACTION logItem INFO
logItem ACNTING
logItem TRACE
# The following are requested for interoperability with some servers which
# need explicit requests.
option 1  # Subnet Mask
option 3  # Router
option 6  # Domain Name Server option 15 # Domain Name
option 28 # Broadcast Address
option 33 # Static Routes
option 60 "IBMWARP_V3.1" # Vendor Class
option 77 "IBMWARP_V3.1" # User Class

#updateDNS "nsupdate -h%s -d%s -s"d;a;*;a;a;%s;s;%s;3110400;q" -q"
# The following are options for which IBM supplies an installation
# script, dhcpibm.cmd, to automatically configure the IBM application
# with the served value. Uncomment them if desired.
#option 9 exec "dhcpibm.cmd 9 %s" # LPR Server
#option 71 exec "dhcpibm.cmd 71 %s" # Default NewsReader/2
#option 200 exec "dhcpibm.cmd 200 %s" # Default LPR Printer
#option 201 exec "dhcpibm.cmd 201 %s" # Gopher Server
#option 202 exec "dhcpibm.cmd 202 %s" # Default WWW Home Page
#option 203 exec "dhcpibm.cmd 203 %s" # Default WWW Proxy Server
#option 204 exec "dhcpibm.cmd 204 %s" # Default WWW News Server
#option 205 exec "dhcpibm.cmd 205 %s" # Default Socks Server
#option 206 exec "dhcpibm.cmd 206 %s" # NFS Servers and Mount Points
#option 207 exec "dhcpibm.cmd 207 %s" # Default X Font Server
#option 208 exec "dhcpibm.cmd 208 %s" # Default X System Display Manager
```

- NAMED.BT file:

```
;
; NAMED.BT file for name server configuration.
;
; type domain source file or host
; primary itso.ral.ibm.com      c:\\mptn\\etc\\namedb\\named.dom dynamic
;
primary 104.24.9.in-addr.arpa c:\\mptn\\etc\\namedb\\named.rev dynamic;
```

- NAMED.DOM file:

```
$ORIGIN itso.ral.ibm.com.
test IN KEY 0x0080 0 1 AQPS5nBuY34O4d0kWsDcjsvQSwpAKMIGNaGphB+xNKNTPsf9DMy8Lx65OxQ16IcGwH/hO33
                       VgM5CWyl3EOWDVmqz ;Cl=5
     IN SOA ns-updates.test.itsc.austin.ibm.com. ns-updates.test.itsc.austin.ibm.com. (
                       95112502 86400 300 864000 3600 300 ) ;Cl=5
     IN NS ns-updates.test.itsc.austin.ibm.com. ;Cl=5 $ORIGIN test.itsc.austin.ibm.com.
```

```
martin    IN CNAME ns-updates.test.itsc.austin.ibm.com. ;Cl=5
localhost IN A 127.0.0.1 ;Cl=5
client1   IN KEY 0x0000 0 1 AQO3P+UqipNXsuijeL3yyfJLw9PagI+NZg9oXrgYI1cSKOAo+WwPOxpEqUsjOhFsKNo4VOq
                            6LH1LKl7XcytwAIO1 ;Cr=auth
4660      IN   A 200.200.200.2 ;Cr=auth
4660      IN   SIG  A14 4660 817359867 817356267 0x8d00 client1.test.itsc.austin.ibm.com
                            tDCJdEVGFPTPAt8nN+Oz3IuOFgWhomCORcKaY3xhBbJalnLvFOKmG+D//JJ+7RmM+rqrW9A
                            K7qQslvIyum6NPw== ;Cr=auth
4660      IN SIG KEY 14 4660 820470267 817356268 0x8d00 client1.test.itsc.austin.ibm.com
                            ecK2L1zhtyVnNrI24/Viitl41reduDy7TU8dxSCoGoc9zc4IIGEy4E4uVPud4fjessH8XS+
                            H2UVjLXhr66y6Gg== ;Cr=auth
ns-updates IN   A 200.200.200.10 ;Cl=5
```

- NAMED.REV file:

```
$ORIGIN 200.200.in-addr.arpa.
200 IN KEY 0x0080 0 1 AQPE7HDQaltur1bT7Zv1nEP7318TJXv82rZX67rdVzew3Ts++KQ/ggimUPk/EodzISfYfhE yNbDcgIno9aAbqqs7
;Cl=5
IN SOA ns-updates.test.itsc.austin.ibm.com. ns-updates.test.itsc.austin.ibm.com. (
95112502 86400 300 864000 3600 300 );Cl=5
IN NS ns-updates.test.itsc.austin.ibm.com.;Cl=5 $ORIGIN 200.200.200.in-addr.arpa.
2 IN KEY 0x0000 0 1 AQPvxnJUi6hiHzRJC/beIJDsfFtumzD2He33CvM5mYOPMGTYVkOYR+DUNTtdlGOwm2ONFvo
5uVAOdRdIuIMfb4UN ;Cr=auth 4660 IN PTR client1.test.itsc.austin.ibm.com. ;Cr=auth
4660 IN SIG PTR 1 4 4660 817430323 817426724 0x856f 2.200.200.200.in-addr.arpa sPfLnGeDm9i+N/jyLDn
VRPl8tKTYMQT2zsf135nqFRR+AyrZrCPSEICA4UmK8787IQXMmcawczAjOUgrNgtlIA== ;Cr=auth
4660 IN SIG KEY 1 4 4660 817430323 817426724 0x856f 2.200.200.200.in-addr.arpa vnsYFxdSJNQl+YmheIk
fxvZ1Ia3jeyMuS7YOPyTcVH7bqXJgoys1eIvvmMgGYEBHb+YU3lyt2tZARqpA+FfQeQ== ;C r=auth
10 IN PTR ns-updates.test.itsc.austin.ibm.com. ;Cl=5
```

- SYSLOG.CNF file:

```
numLogFiles4 logFileSize 100
logFileName syslog.
logItem LOG_EMERG
logItem LOG_ALERT logItem LOG_CRIT
logItem LOG_ERR
logItem LOG_WARNING logItem LOG_NOTICE
logItem LOG_INFO
```

Notes:

1. KEY and SIG resource records as well as encryption keys always use a single line. We have indented the examples for illustration purposes only.
2. You will realize that the format of the NAMED.DOM and NAMED.REV files looks quite different from the examples shown in 5.5.6.1, ″Creating a New DDNS Server Configuration″ on page 230. This format will be used after the first update has occurred to the DDNS server, no matter what the format has been before, so you don′t have to worry about it. Both formats work, but only the second one will be used in actuality.

5.10.2 Complex Operational Scenario

This section provides an example of a more complex Dynamic IP scenario involving a client and a server on different IP subnets and also involving a Bootp client and a BootP relay agent.

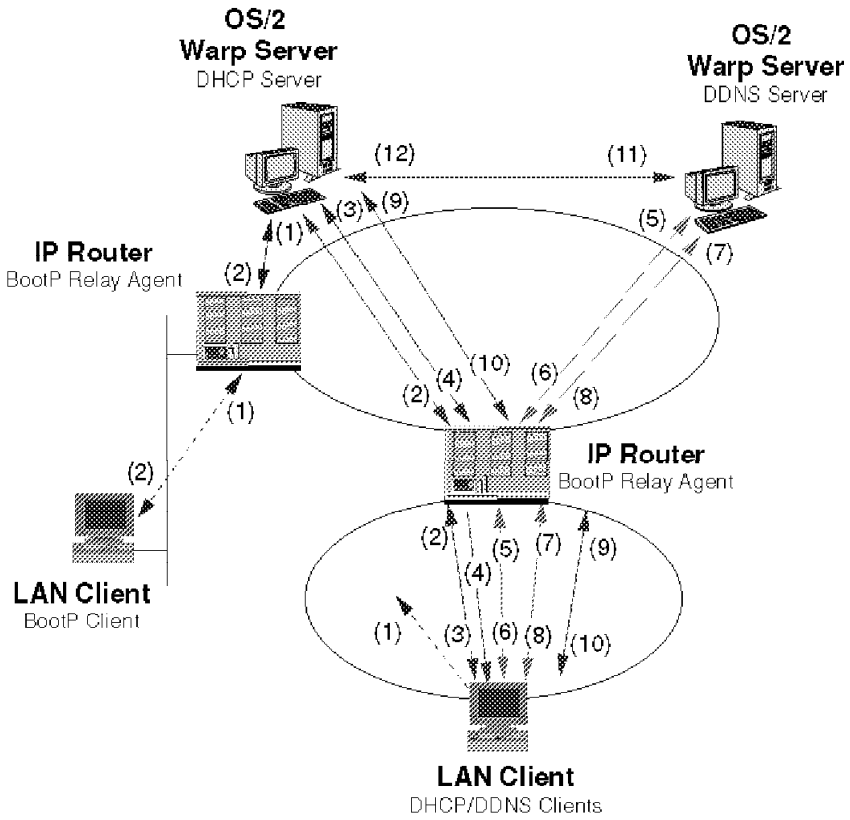


Figure 37. Complex Dynamic IP Scenario

The following steps describe the Dynamic IP operation in the scenario shown above, when the Dynamic IP client is started for the first time:

1. DHCPDISCOVER, broadcast by the DHCP client on the local subnet and forwarded by the BootP relay agent in the IP router.
2. DHCPOFFER, sent by the DHCP server containing configuration options and forwarded by the BootP relay agent in the IP router.
3. DHCPREQUEST, broadcast by the DHCP client on the local subnet and forwarded by the BootP relay agent in the IP router.
4. DHCPACK, sent by the DHCP server and forwarded by the BootP relay agent in the IP router.
5. DNS query for primary name server, sent by the DDNS client.
6. DNS authoritative reply, sent by the DDNS server.
7. DDNS update query, sent by the DDNS client containing the hostname specified by the user.
8. DDNS acknowledgement, sent by the DDNS server.
9. DHCPREQUEST lease renewal, sent by the DHCP client supplying the hostname specified by the user.
10. DHCPACK, sent by the DHCP server.
11. DDNS update query, sent by the DHCP server to update the DDNS database with the inverse mapping information for the hostname and IP address.
12. DDNS acknowledgement, sent by the DDNS server.

The following steps describe the BootP operation in the scenario shown above, whenever the BootP client is started:

1. BootP request, broadcast by the BootP client on the local subnet and forwarded by the BootP relay agent in the IP router
2. BootP reply, sent by the DHCP server containing configuration options and forwarded by the BootP relay agent in the IP router.

5.10.3 Using Multiple Dynamic IP Servers

The following considerations should be considered when you want to install multiple Dynamic IP servers for backup purposes:

1. The address ranges of DHCP servers must not overlap.
2. DHCP servers do not communicate or consolidate their configurations between each other.
3. DHCP servers can support multiple subnets.
4. Only one DDNS server can be authoritative for a domain and can accept update requests.

You can still provide at least some functional backups in the following ways:

1. Distribute IP addresses of one or more subnets across multiple DHCP servers. If one server fails, only the range of IP addresses that this server was managing will be unavailable. Just make sure that you do not overlap IP address ranges when you are setting up multiple DHCP servers.
2. Use secondary name servers. If the primary DDNS server fails, no more update requests can be processed in this zone, but the latest available database will be held in secondary name servers to answer queries. However, if the primary server is down for a longer time than the resource records in the secondaries′ databases are valid, the whole zone will gradually become unavailable.

5.10.4 Connecting an AIX DCHP Client to an OS/2 DHCP Server

Since Version 4.1.4 AIX also supports DHCP. Not only the DHCP client but also the DHCP server is delivered with this AIX release.

The following scenario shows how to set up your OS/2 DHCP server and the AIX DHCP client. The same machine that runs the DHCP server also runs the DDNS server for dynamic domain name resolution in that scenario. Furthermore the DHCP server is connected via token-ring to the 9.24.104 subnet and via Ethernet to the 9.24.105 subnet. The AIX client is part of the 9.24.105 subnet. The DHCP server is responsible for supplying IP addresses for both subnets. The following figure shows the configuration:

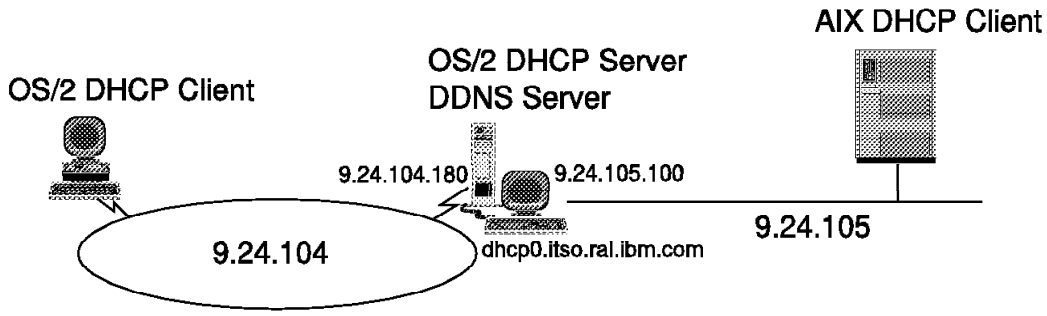


Figure 38. AIX DHCP Client Scenario

5.10.4.1 Setting Up the DHCP and DDNS Server

As the DHCP server is responsible for both subnets, you have to create two different entries in the DHCP server configuration file (dhcpsd.cfg). When you use the DHCP Server Configuration program your configuration screen should look like the following. Your configuration could look different depending on the option you supply.

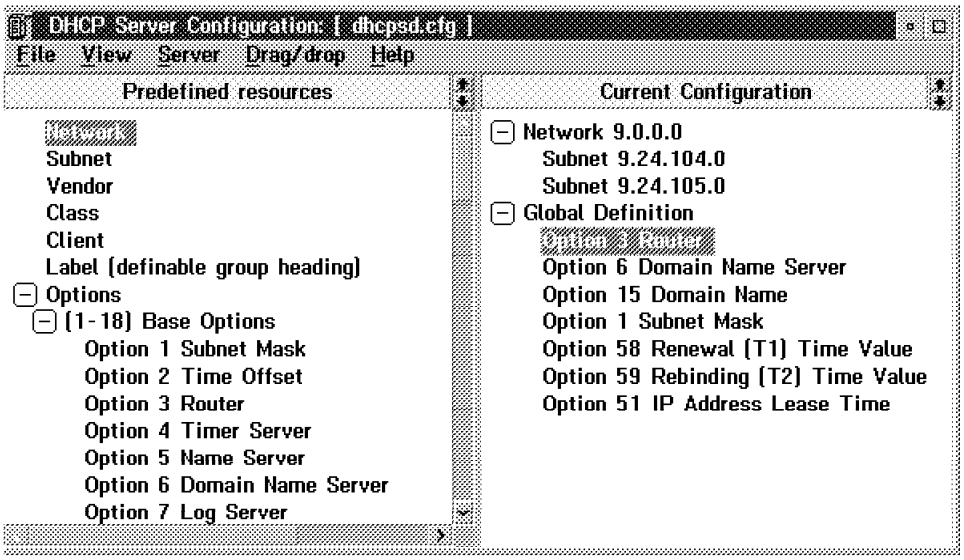


Figure 39. DHCP Server Configuration

Your configuration file will look like the following. The only change we made to that file after its creation was to include the -pdhcp0.itso.ral.ibm.com option in the updateDNS statement. This will specify the DDNS server that should be updated by the DHCP server.

```
numLogFiles          2
logFileSize          50
logFileName          dhcpsd.log
leaseTimeDefault     3 minutes
leaseExpireInterval  1 minutes
supportBOOTP  no
supportUnlistedClients  yes
logItem     SYSERR
```

```
logItem     OBJERR
logItem     PROTERR
logItem     WARNING
logItem     EVENT
logItem     ACTION
logItem     INFO
logItem     ACNTING
logItem     TRACE
#.indent 12

updateDNS "nsupdate -r%s -pdhcp0.itso.ral.ibm.com -s"d;ptr;*;a;ptr;%s;s;%s;0;q""

network     9.0.0.0 255.255.255.0      #.name IBM Net
{
    subnet     9.24.104.0 9.24.104.181-9.24.104.182    #.name ITSO
    {
        #.ddns 9.24.104.180
    }
    subnet     9.24.105.0 9.24.105.101-9.24.105.110    #.name (blank)
    {
        #.ddns 9.24.104.180
    }
}
#.cat Global Definition                    {
    option 3       9.24.104.1    #.name 3 Router
    option 6       9.24.104.180    #.name 6 Domain Name Server
    option 15       itso.ral.ibm.com    #.name 15 Domain Name
    option 1       255.255.255.0    #.name 1 Subnet Mask
    option 58       120     #.name 58 Renewal (T1) Time Value
    option 59       60     #.name 59 Rebinding (T2) Time Value
    option 51       180    #.name 51 Lease Time
#.cat }
```

To set up your DDNS server you should create a bootfile defining two reverse files, one for each subnet. The bootfile (named.bt) looks like the following in our example:

```
;
; NAMED.BT file for name server configuration.
;
; type       domain                       source file or host
;
primary  itso.ral.ibm.com       d:\\mptn\\etc\\namedb\\named.dom dynamic
;
primary  104.24.9.in-addr.arpa  d:\\mptn\\etc\\namedb\\named104.rev dynamic
primary  105.24.9.in-addr.arpa  d:\\mptn\\etc\\namedb\\named105.rev dynamic
;
cache . d:\\mptn\\etc\\namedb\\named.ca
;
```

There are two reverse files. Named104.rev for the 9.24.104 subnet and named105.rev for the 9.24.105 subnet. Only one domain file (named.dom) is used to hold the hostnames of the itso.ral.ibm.com domain. The domain and reverse files are created like it is explain in the previous sections. The ddnszone command is used to create the public and private keys for these zones. Also the key stored in the dhcpsd.dat file must be copied to the ddns.dat file like explained in the previous section.

The new named105.rev reverse file could look like the following, once the AIX client has registered to it with the name aixdhcp:

```
$ORIGIN 24.9.in-addr.arpa.
105     IN      KEY     0x0080   0   1 AQPWhe77Yq6hCc6M7GJewCQwrW51AgVzOG/xKWbyumi4Bc3w9twn
                                       EOEVzvk/j4woFJ6Urz+dGywyd4nwVtpfstOj  ;Cl=5
        IN      NS      dhcp0.itso.ral.ibm.com. ;Cl=5
        IN      SOA     dhcp0.itso.ral.ibm.com. kwichman.raleigh.ibm.com. (
                252 86400 300 864000 3600 300 ) ;Cl=5
$ORIGIN 105.24.9.in-addr.arpa.
102     IN      KEY     0x0000   0   1 AQOiZqGKbRNWK9nO+/rT1TIyj8H9oyusLqHbXMFrdwDlthLjdGoRm
                                       Q9JLoyRbSqrRbtTZPgDT1OSPdEdm19lFeOt ;Cr=auth
        4660    IN      PTR     aixdhcp.itso.ral.ibm.com. ;Cr=auth
        4660    IN      SIG     PTR 1 4 4660 826822537 826822357 0xe315 102.105.24.9.in-addr.arpa
                                        SRtaUMdv3Fh+nWAcOwl/4XajjhARKG+cnHiG5Owa1JpKxqdj
                                        EBONqOZ6qEBGkJnOLm82uxqHGyNDqK9gmOP/BA== ;Cr=auth
        4660    IN      SIG     KEY 1 4 4660 826822537 826822358 0xe315 102.105.24.9.in-addr.arpa
                                        OjS5iwxcLSWonBIkesvCnNxPMCvbYWg52GJahLXncOtwAOLa
                                        1W5GsX7mHIc9rjJxvOxS4nLUTIlPlkonuPEMNg== ;Cr=auth
```

5.10.4.2 Setting Up the AIX DHCP Client

To set up your AIX machine as a DHCP client you use the smit or smitty command. Enter `smit tcpip` and you get the configuration screen for TCP/IP. Select **Use DHCP for TCPIP Configuration & Setup** to set up your DHCP client. A list displaying your interfaces pops up.

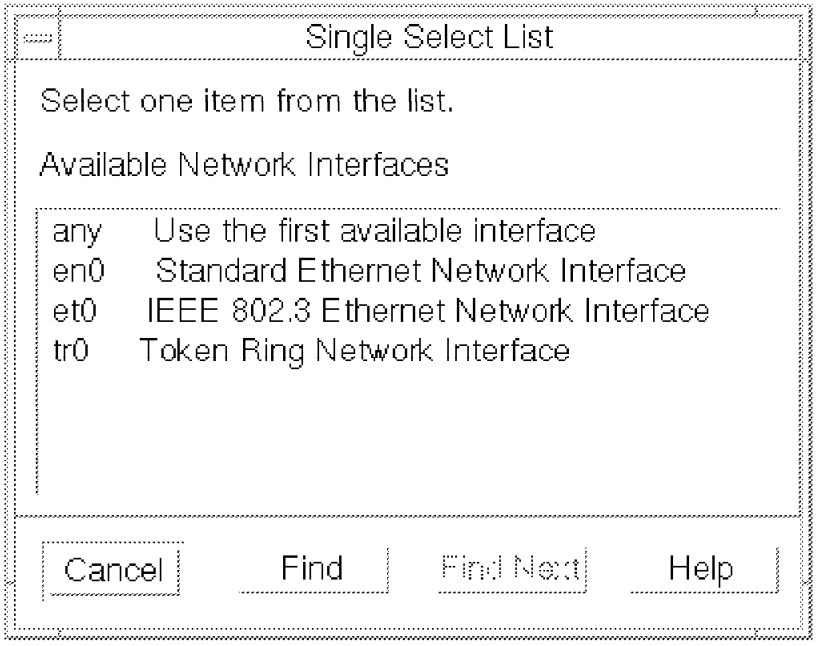


Figure 40. AIX DHCP Client Interface

Double-click the interface that you want to configure for using DHCP. In our scenario we select the interface en0 (Standard Ethernet Network Interface). You get the following configuration screen to configure that interface for DHCP.

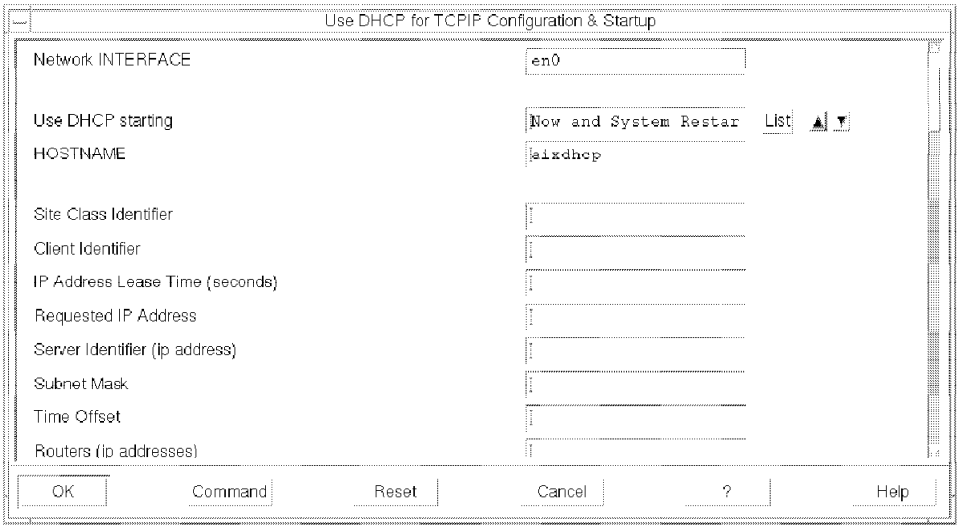


Figure 41. AIX DHCP Client Configuration

The field Network INTERFACE should display en0 for the Ethernet interface. Select **Now and System Restart** in the Use DHCP starting field. In the HOSTNAME field you should specify the hostname that your machine should register with the DDNS server. Once you have completed these fields click on **OK** to create the client configuration file (dhcpcd.ini) and to activate the DHCP client.

Your AIX DHCP client configuration file should look like the following:

```
# @(#)02 1.5 src/tcpip/etc/dhcpcd.ini, dhcp, tcpip41C, 9535B 8/30/95 18:15:42
#
# dhcpcd.ini -- DHCP Client configuration file
#
#
# This file contains directives that can be specified by the
# to configure the client.
#
#

numLogFiles 4
logFileSize 100
logFileName /usr/tmp/dhcpcd.log
logItem SYSERR
logItem OBJERR
logItem PROTERR
logItem WARNING
logItem EVENT
logItem ACTION
logItem INFO
logItem ACNTING
logItem TRACE

#sniffer "ifsniff -c -itr0"
#option 60 foo
#updateDNS "nsupdate -h%s -d%s -s"d;a;*;a;a%s;s;%s;3110400;q" -q"
clientid MAC
```

```
interface en0
{
 option 12 "aixdhcp"
 option 19 0
 option 20 0
 option 27 0
 option 29 0
 option 30 0
 option 31 0
 option 34 0
 option 36 0
 option 39 0
}
```

Once the AIX DHCP client is started, it connects to the DHCP server and receives its IP address and the specified options. You can use the command `ifconfig en0` to see the configuration of the Ethernet interface. The DDNS server will be updated by the DHCP server with the specified hostname.

To manually update the DDNS server you can use the nsupdate command. Enter `nsupdate -pddns_server_name` to start nsupdate in the interactive mode.

5.10.5 Interoperation with OEM and Legacy Hosts

As mentioned earlier, a benefit of Dynamic IP, using only open networking standards, is that IBM products interoperate with OEM IP networking products. More specifically, Dynamic IP clients may be served by OEM DHCP and DNS servers. Dynamic IP DHCP servers may serve OEM BootP or DHCP clients. Dynamic IP DNS servers are a functional superset of existing DNS servers and may be seamlessly inserted into existing customer DNS server hierarchies.

5.10.5.1 Connecting Windows NT Clients to an OS/2 DHCP Server

When you install Windows NT 3.5 with Microsoft TCP/IP support, or when you configure Microsoft TCP/IP on Windows NT at a later time, you can choose to manually configure TCP/IP parameters or use DHCP.

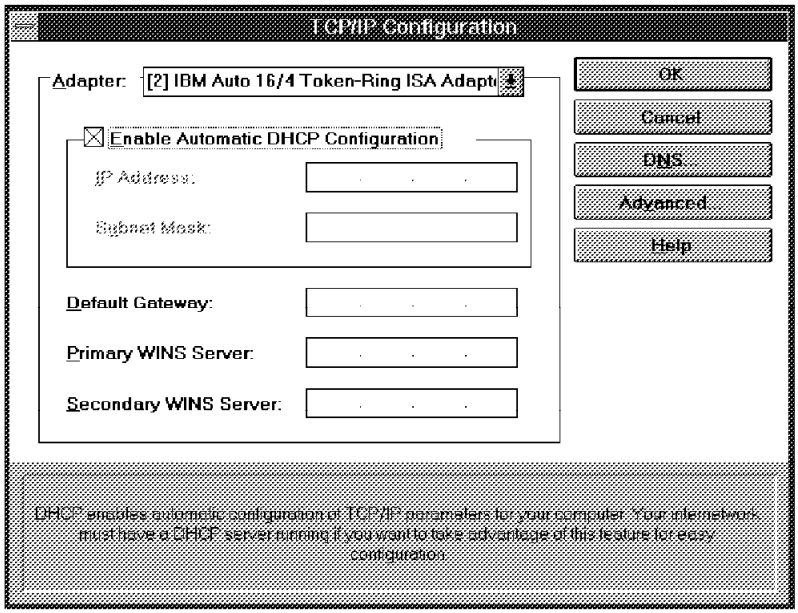


Figure 42. Windows NT TCP/IP Configuration

We have successfully connected a Windows NT DHCP client to the IBM OS/2 DHCP server. Windows NT cannot participate in DDNS.

5.10.5.2 Connecting Windows 95 Clients to an OS/2 DHCP Server

When you install Windows 95 with Microsoft TCP/IP support, or when you configure Microsoft TCP/IP on Windows 95 at a later time, you can choose to manually configure TCP/IP parameters or use automatic configuration (DHCP). Figure 43 shows the TCP/IP configuration menu of a Windows 95 system.

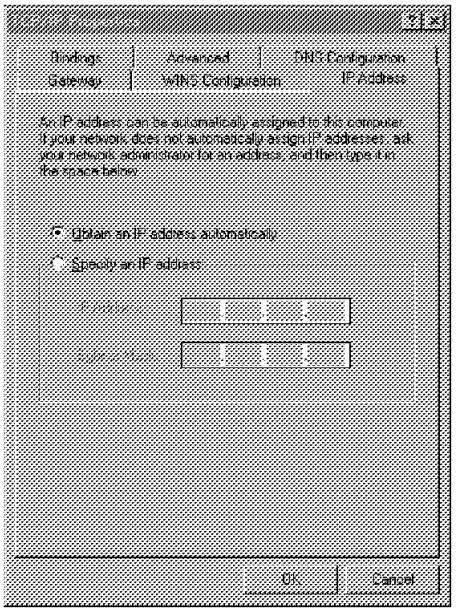


Figure 43. Windows 95 TCP/IP Configuration

We have successfully connected a Windows 95 DHCP client to the IBM OS/2 DHCP server. Windows 95 cannot participate in DDNS.

5.10.5.3 Connecting IBM Dynamic IP Clients to a Windows NT DHCP Server

A Windows NT 3.5 Advanced server system offers a DHCP server to be installed as an option of Microsoft TCP/IP support. Figure 178 shows the DHCP server configuration menu of a Windows NT system.

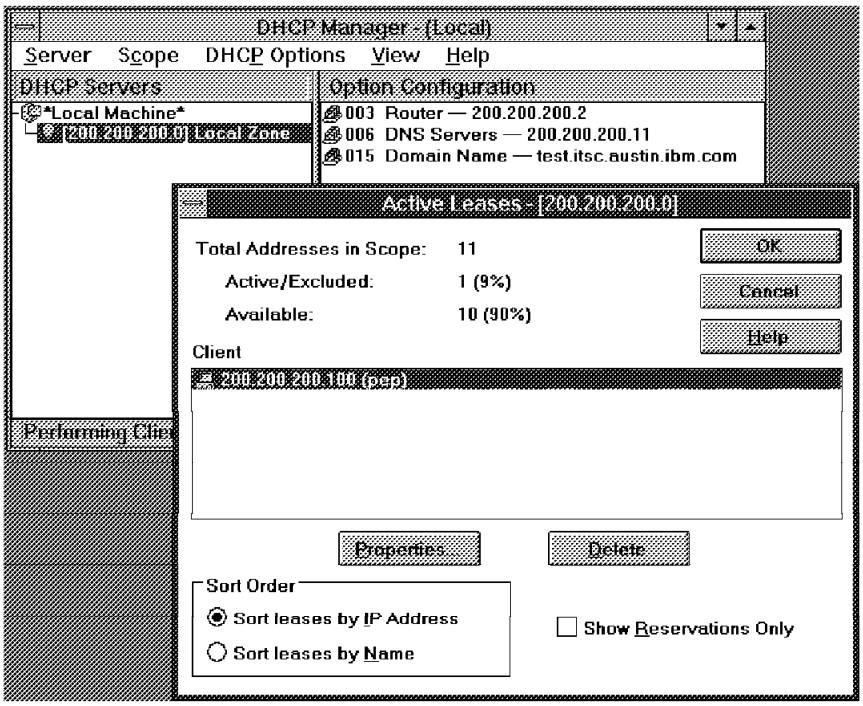


Figure 44. Windows NT DHCP Server Configuration

We have successfully connected the OS/2 DHCP client to a Windows NT DHCP server.

Windows NT also uses the Windows Internet Name Service (WINS), for which it supplies client and server programs. This service works as a name server for NetBIOS over TCP/IP P-node, M-node, and H-node systems, providing a mapping service between NetBIOS names and IP addresses. WINS works in a dynamic way that is similar to DDNS, but it does not provide any client authentication. WINS also cannot be used as a name server in the DNS hierarchy.

Chapter 6. Electronic Mail

This chapter describes the mail services in TCP/IP for OS/2, which are based on the Simple Mail Transfer Protocol (SMTP). The Multipurpose Internet Mail Extensions (MIME) to SMTP are also covered in this chapter.

The basis of mail services in TCP/IP is the SendMail program (sendmail.exe), which provides client and server functions that are compliant with SMTP. UltiMail Lite is also provided with TCP/IP 3.x. This program gives you enhanced mail handling facilities as a Presentation Manager application. You should note that UltiMail Lite requires basic send and receive functions provided by SendMail.

Another mailing system besides UltiMail Lite is Lotus Notes. Lotus Notes is called a Groupware product that enables different users to work together in an easy and effective way. Nevertheless the connectivity to users on the Internet is an important factor for most users. This chapter discusses the Lotus Notes gateway for SMTP and shows how to exchange mail between both systems.

SendMail uses SMTP to route mail from one host to another, allowing you to exchange mail with other hosts that support SMTP. SendMail functions as a mail router, that listens for and receives mail from the network and sends mail to the network. It stores incoming mail in a mail directory. You can then use UltiMail Lite to look at the mail; therefore, UltiMail Lite does not have to be running to receive mail.

6.1 SendMail

SendMail is the basic application to receive and send mail. This section shows you how to set up SendMail and how to use it to send and receive mail.

6.1.1 Configuration of SendMail

Before you try to use SendMail, you must configure the software. Configuration information for SendMail is stored by default in the file sendmail.cf. Usually the sendmail.cf file is stored in your \MPTN\ETC directory. You can create your own configuration file for SendMail and save it under a different name. For example, UltiMail Lite uses the file sendmail.uml to set up SendMail for its purpose.

Sendmail.cf is a very complex configuration file. For your application, to send and receive mail you should only change the following items:

OQ Specifies the path to the mail queue. This is where SendMail temporarily holds mail before delivering it to the recipient.

Cw Specifies your host (workstation) name and any aliases you may want to designate. You can use the TCP/IP hostname command to display your host name.

Dw Specifies your host (workstation) name only. This should be the same as the Cw entry. (You cannot specify alias names.)

DD Specifies the local domain name.

For example, if a workstation is kwichman.itso.ral.ibm.com and the path to the mail queue is D:\MPTN\ETC\MQUEUE, you would change the fields in the sendmail.cf file as follows:

```
OQD:\MPTN\ETC\MQUEUE
Cwkwichman
Dwkwichman
DDitso.ral.ibm.com
```

In sendmail.cf are more parameters that can be customized. Do not change any entries other than those recommended unless you are certain about the effect of your changes. If the sendmail.cf file is incorrect, you might not be able to send or receive mail. The following gives you more information about useful parameters:

DR Specifies a gateway to which mail is delivered instead of direct delivery. (DRYour.Internal.Gateway)

DV Specifies the hostname of your mail gateway. Mail that cannot be delivered directly is sent to this address to be relayed. (DVYour.External.Gateway)

Mlocal Specifies aspects of the local mail handler. A sample Mlocal entry is shown below. It assumes that:

- Your mail agent is installed into D:\TCPIP\MAIL.
- D:\TCPIP\MAIL\FOLDERS\INBOX contains incoming mail.

You would type it as one line in the sendmail.cf file:

```
Mlocal, P=D:\TCPIP\MAIL\MAILAGENT.EXE, F=lsm,
A=-dest D:\TCPIP\MAIL\FOLDERS\INBOX -to $u
```

Parameters:

P Specifies a mail handler to be activated. Set this field to the path and name of your mail handler (MAILAGENT.EXE).

F l, specifies a local handler for final delivery, s strips quotes from user IDs and m specifies that the mailer can handle multiple users on the same host in one transaction. If m is not specified, your MAILAGENT.EXE is called once for each user.

A Specifies the argument to be passed to the mail handler. For MAILAGENT.EXE, it could be specified as follows:

```
A=-dest dir -to users
```

Where dir specifies the directory where the mail should be saved (the InBox of your Mail server) and users specifies a SendMail macro containing the users to whom this mail is to be delivered. This value should be set to $u. If m is specified in the F=field, the $u macro contains a list of all users to which this mail is addressed.

OTxt Designates how long mail is held in the mail queue until it can be delivered to a server that is down. Undelivered mail is held while SendMail attempts to retransmit it. The xt entry is the time interval that undelivered mail is retained for retransmission before it's deleted from the mail queue:

- x is an integer

- t is the unit of time (d= days, m= minutes,h= hours, w= weeks)

Example:

0T8h

OXn Sets the Load Limiting variable. n is an integer specifying the maximum number of concurrent SendMail instances permitted. The default is six.

You should also ensure that the following directories exist in your ETC subdirectory:

MAIL Incoming mail is stored in this directory. This is configurable.

MQUEUE Outgoing mail and temporary files are stored in this directory.

6.1.2 Starting SendMail

Normally, SendMail should be run continuously as a background process to allow you to send and receive mail. You can either configure SendMail for autostart, and it will start when TCPSTART.CMD is executed, or you can execute the sendmail.exe from an OS/2 command prompt.

6.1.2.1 Configuring for Autostart

1. Start the TCP/IP configuration notebook.
2. Click on the **Autostart** tab of the notebook.
3. Select **sendmail** from the Services to autostart list.
4. Select **Autostart service** if SendMail should be started a utomatically when your machine is started. The Autostart option is saved in the TCPSTART.CMD file.
5. Select **Detached** or **Foreground session**. When Detached is selected SendMail is started when TCP/IP starts. SendMail does not appear in the OS/2 task list. When SendMail is started as a foreground session it is started in an OS/2 window.
6. Customize the default parameters. The parameters are the same as those used for starting SendMail from the command line. These parameters are explained in the next section.

   Your panel should look like this:

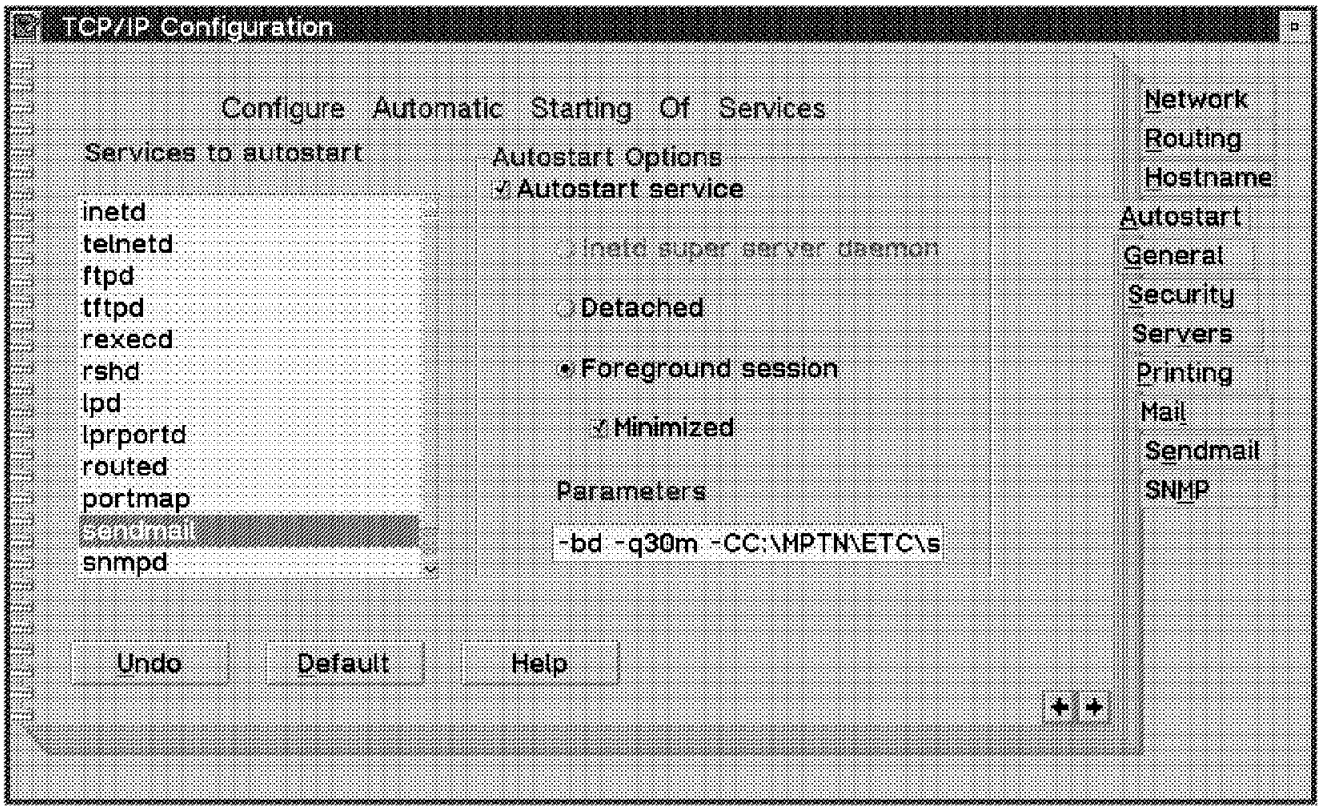


Figure 45. SendMail Configuration (Autostart)

7. Close the configuration menu by double-clicking on the top left-hand corner of the window.

   Click on **Save**. The following message will appear. If you don't use Ultimail Lite select **No**. Otherwise it is recommended to let the system do the configuration of SendMail for use with UltiMail Lite.

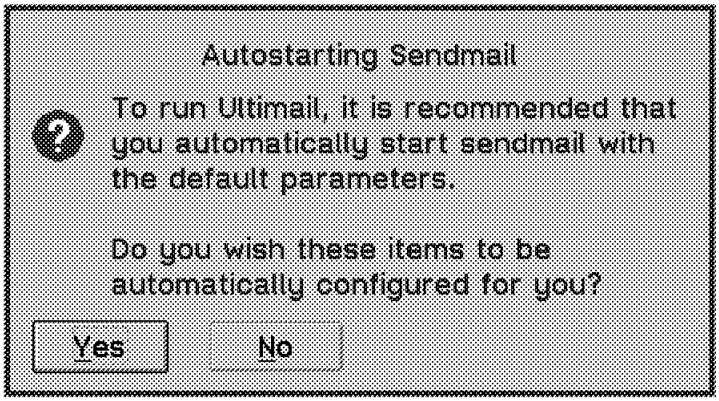


Figure 46. Configuration of UltiMail Lite

From now on, anytime you execute TCPSTART.CMD the SendMail service will automatically be started.

6.1.2.2 Starting SendMail From an OS/2 Command Prompt

As shown in the following example, SendMail can be started from the command line with the same parameters that are specified in the autostart menu.

```
[C:tcpipbin]SendMail -bd -q30m
IBM OS/2 SendMail VERSION 1.3.14
reading C:\TCPIP\ETC\sendmail.cf
starting SendMail process
SendMail process started
```

The sendmail command has the following format:

```
sendmail -bd -qtime -Cconfiguration_file
```

The following is a summary of the SendMail parameters:

-bt Starts SendMail as a server.

-qtime Specifies how often the mail queue can be processed. Enter the time as a number and a letter, where the letter is one of the following:

- s for seconds
- m for minutes
- h for hours
- d for days
- w for weeks

An example would be:

```
-q30m
```

-Cconfiguration_file Specifies the name and the location of the configuration file to be used by the sendmail server. Use this parameter to overwrite the default SendMail configuration file sendmail.cf. For example:

```
-CC:\MPTN\ETC\sendmail.uml
```

6.1.3 Sending Mail

SendMail is usually used by an E-mail application like UltiMail Lite to send mail. Nevertheless, it is possible to use only the sendmail command to send your mail to other systems using SendMail.

The first thing you have to do, in order to send mail, is to create a file that contains the desired information. This file can then be sent to other systems.

Use the sendmail command to send your file. The format of sendmail is the following:

```
sendmail -Cconfigurationfile -af filename -f user@localhost user@remotehost
```

or

```
sendmail -Cconfigurationfile -af filename -t
```

The following is a summary of the parameters used to send mail:

-Cconfigurationfile Specifies the path and file name of the SendMail configuration file. Sendmail.cf is the default.

-af filename Specifies the name of the file that contains the mail message.

-f user@localhost Identifies the user and hostname of the sender.

user@remotehost Identifies the user and hostname of the receiver.

-t Specifies that sendmail retrieves the addressing and subject information from the header of the message rather than from the command line. SendMail scans the note for the following tags:

- To:
- From:
- BCC: (blind carbon copy)
- CC: (carbon copy)
- Subject:

Note: OS/2 is not a multiuser operating system. The user field of an address in an item of mail sent to an OS/2 system is not significant; however, it is required. The significant part of the address is the hostname, with the domain name expansion if domain names are used.

The following shows an example on how to use SendMail to send text created with an editor. The text is stored in the file testmail.txt and contains the address of the receiver and the sender.

```
To:  dlboone@raleigh.ibm.com
From:  kwichman@kwichman.itso.ral.ibm.com
Subject:  SendMail

David,

This is a short message sent with SendMail that comes with TCP/IP for OS/2.

Have a nice day,

Klaus
```

To send the file enter the following command:

```
[D:\]sendmail -af testmail.txt -t
IBM OS/2 SENDMAIL VERSION 1.3.14
reading C:\MPTN\ETC\sendmail.cf 10
02/07/96 12:16:51 mail delivered from:  user@kwichman.itso.ral.ibm.com
```

If you specify the wrong address where you want to send the mail to, your mail will be queued. Depending on how SendMail is set up on your system it will try to deliver the mail again after a certain amount of time.

```
[D:\]sendmail -af testmail.txt -f kwichman@kwichman.itso.ral.ibm.com
dlboone@raleigh.ibm.edu
IBM OS/2 SENDMAIL VERSION 1.3.14
02/07/96 12:16:26 mail delivered from:  user@kwichman.itso.ral.ibm.com
dlboone@raleigh.ibm.edu Host unknown
Cannot send mail to dlboone@raleigh.ibm.edu - mail will be queued
```

6.1.4 Debugging SendMail

As it can be very difficult to configure the SendMail configuration file with all its parameters, it might be useful to trace the SendMail activities. SendMail is started in the debugging mode by adding the parameter -d at the end of your SendMail command. In the example above, the debug option gives you the following information:

```
[D:\]sendmail -af testmail.txt -t -d
IBM OS/2 SENDMAIL VERSION 1.3.14
setoption d=background
setoption o=
setoption r=15m
setoption Q=c:\mptn\etc\mqueue
setoption s=
setoption T=5d
setoption H=c:\mptn\etc\sendmail.hf
setoption A=c:\mptn\etc\aliases
setoption S=c:\mptn\etc\sendmail.st
queuename: assigned id AA0858, env=1786
setsender()

--parseaddr(user@kwichman.itso.ral.ibm.com)
parseaddr-->179a=user@kwichman.itso.ral.ibm.com: mailer 1 (local), host ', user user'
        next=0, flags=0, alias 0
        home="(null)", fullname="(null)"
EOH
----- collected header -----
Received: ?sfrom s .by j (v/Z) id i; b
Date: a
From: (null)
Message-Id: <t.i@j>
Resent-Date: a
Resent-From: q
Resent-Message-Id: <t.i@j>
Full-Name: x
To: dlboone@raleigh.ibm.com
sendto: dlboone@raleigh.ibm.com
   ctladdr=[NULL]

--parseaddr(dlboone@raleigh.ibm.com)
parseaddr-->436O=dlboone@raleigh.ibm.com: mailer 0 (smtp), host raleigh.ibm.com',
user dlboone@raleigh.ibm.com'
        next=0, flags=0, alias 0
        home="(null)", fullname="(null)"

recipient: 436O=dlboone@raleigh.ibm.com: mailer 0 (smtp), host raleigh.ibm.com',
user dlboone@raleigh.ibm.com'
        next=0, flags=0, alias 8
        home="(null)", fullname="(null)"
From: kwichman@kwichman.itso.ral.ibm.com
Subject: SendMail
----------------------------

SENDALL: mode b, sendqueue:
436O=dlboone@raleigh.ibm.com: mailer 0 (smtp), host raleigh.ibm.com',
user dlboone@raleigh.ibm.com'
        next=0, flags=0, alias 8
        home="(null)", fullname="(null)"
queueing AA0858
queueing 436O=dlboone@raleigh.ibm.com: mailer 0 (smtp), host raleigh.ibm.com',
user dlboone@raleigh.ibm.com'
        next=0, flags=0, alias 8
        home="(null)", fullname="(null)"
remotename(dlboone@raleigh.ibm.com)
remotename(kwichman@kwichman.itso.ral.ibm.com)

--deliver, mailer=0, host=raleigh.ibm.com', first user=dlboone@raleigh.ibm.com'
remotename(user@kwichman.itso.ral.ibm.com)
remotename => user@kwichman.itso.ral.ibm.com'

send to 436O=dlboone@raleigh.ibm.com: mailer 0 (smtp), host raleigh.ibm.com',
user dlboone@raleigh.ibm.com'
        next=0, flags=0, alias 8
        home="(null)", fullname="(null)"
openmailer: "IPC" "raleigh.ibm.com"
makeconnection (netmail.raleigh.ibm.com [9.67.1.118])
makeconnection: 519
remotename(dlboone@raleigh.ibm.com)
remotename => dlboone@raleigh.ibm.com'
remotename(dlboone@raleigh.ibm.com)
```

```
crackaddr(dlboone@raleigh.ibm.com)
crackaddr=>g'
remotename => dlboone@raleigh.ibm.com'
remotename(kwichman@kwichman.itso.ral.ibm.com)
crackaddr(kwichman@kwichman.itso.ral.ibm.com)
crackaddr=>g'
remotename => kwichman@kwichman.itso.ral.ibm.com'
dropenvelope 1786 id="AA0858" flags=3
```

6.1.5 Scenario: Sending Mail Over a Firewall or Mail Gateway

The following scenario explains how to set up SendMail to send mail over a firewall or a mail gateway. This is also a good example for UltiMail Lite since it uses SendMail to exchange mail.

In this example a user at host kwichman.itso.ral.ibm.com (which is an OS/2 system) will send mail to a user on the host mailhost.lotus.com. The lotus.com domain is in the Internet and the domain ibm.com is behind a firewall. Therefore mail has to be sent over the firewall using a mail gateway. The following figure shows the hosts and the domains. The mail gateway is the host netmail in the raleigh.ibm.com domain.

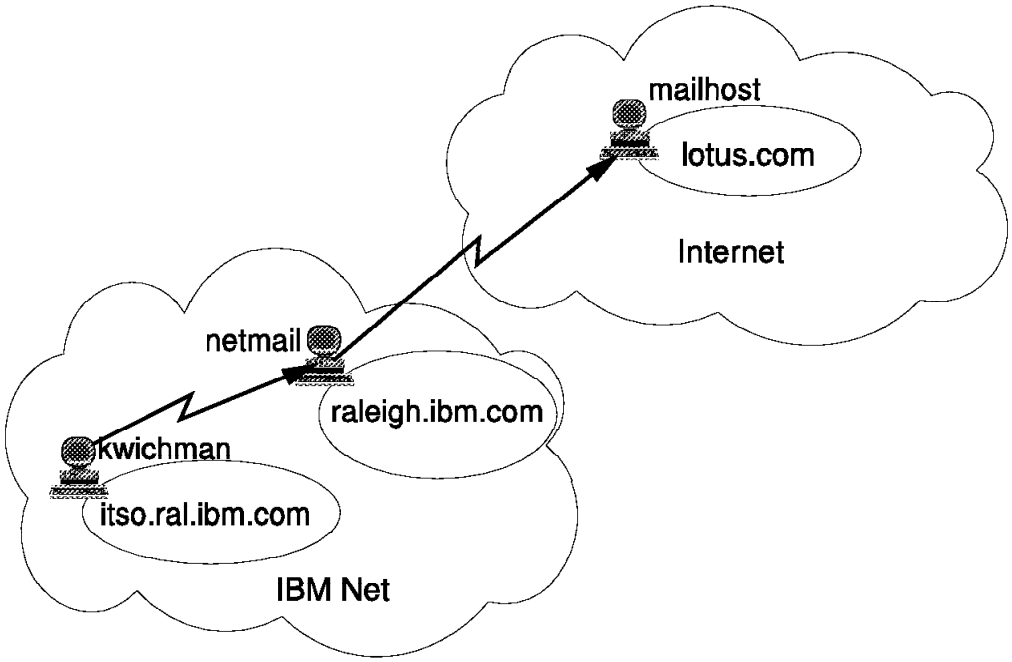


Figure 47. Sending Mail Over a Firewall

In order to send mail over the mail gateway you have to register at the gateway.

The mail gateway will then translate your original address to an address known by the systems behind the firewall. For example if your original address was kwichman@kwichman.itso.ral.ibm.com, the mail gateway would translate your address to kwichman@raleigh.ibm.com. The mail gateway changes the From field of your mail to the new address. This address can then be used by the receiver of your note to reply. It is possible to reply to this address because the mail gateway is seen by the Internet. Once the mail gateway receives mail addressed with your mail gateway address, it will translate this address to the original address and send the mail to your machine.

The configuration file sendmail.cf on the OS/2 system kwichman.itso.ral.ibm.com should look like the following:

```
Dwkwichman.itso.ral.ibm.com
Cwkwichman.itso.ral.ibm.com

######################################################################
#                                                                    #
#  Sendmail                                                          #
#  Copyright (c) 1983  Eric P. Allman                                #
#  Berkeley, California                                              #
#                                                                    #
#  Copyright (c) 1983 Regents of the University of California.       #
#  All rights reserved.  The Berkeley software License Agreement     #
#  specifies the terms and conditions for redistribution.            #
#                                                                    #
#  This configuration file was created specifically for sendmail on the #
#  IBM OS/2 Operating System.  Please avoid making changes to this file #
#  because any changes will change the operation of sendmail.        #
#                                                                    #
#  Created by: William Chung for IBM Ultimedia Mail/2 "Lite"         #
#              and IBM NR/2                                          #
#              IBM T.J. Watson Research Center, Hawthorne, NY        #
#                                                                    #
#              March 21, 1995                                        #
#                                                                    #
######################################################################

# The fully qualified (with domain) name of the internal gateway
# DRYour.Internal.Gateway
DR

# The fully qualified (with domain) name of the external gateway
# DVYour.External.Gateway
DVnetmail.raleigh.ibm.com

# The fully qualified (with domain) name of the external mail hub
# DHYour.External.Mail.Hub

# The fully qualified (with domain) name of the internal mail hub
# DIYour.Internal.Mail.Hub
DI

# External user id
# DPYourExternalUserID

# The local domain
# DDetc\mail
DDitso.ral.ibm.com

# Version # of this file
DZ2.12um

# Official canonical hostname.
Dj$w

# Standard macros

# SMTP initial login message
De$j Sendmail $v/$Z ready at $b
# Name used for error messages
DnMailer-Daemon
# UNIX header format
DlFrom $g  $d
# Delimiter (operator) characters
Do.:%@![=/[]
# Format of a total name
Dq$?x$x <$g>$|$g$.

# Options

# Process messages in the background.
Odbackground
# Accept oldstyle addresses
Oo
# SMTP read timeout
Or15m
# Queue directory - this must be changed if TCP/IP is moved!
OQc:\mptn\etc\mqueue
# Always queue for safety
Os
# Time to live in the queue
OT5d
```

```
# Message precedences
# Note: use equal weight so we can let relay decide what to do

than default sendmail.cf...

# Sendmail configuration file *must* end with a newline - do not remove below newline
OHc:\mptn\etc\sendmail.hf
OAc:\mptn\etc\aliases
OSc:\mptn\etc\sendmail.st
```

6.2 Talk

This section describes the services provided by the Talk protocol in TCP/IP for OS/2.

Talk is used to send and receive interactive electronic messages to and from another host. Before Talk can be used the Talk daemon must be running on both systems that want to communicate by using Talk.

The Talk services consist of:

TALKD.EXE The TALK server. This program needs to be running before any other host can initiate an interactive session with you. However, it will start automatically when you try to initiate an interactive session with another host.

TALK.EXE The TALK client. This program can be used to initiate an interactive session with another host. It contacts the TALK server on the other host which notifies the user that they should start a TALK client. Once this has been done, an interactive talk session will be established and you will be able to type messages to each other.

6.2.1 Talk Configuration

Talk is an easy to use and easy to install application. As it comes with TCP/IP it is installed automatically. You can configure Talk for autostart or start the Talk daemon manually everytime you want to use that service.

6.2.1.1 Configuring for Autostart

1. Open the TCP/IP configuration notebook.
2. Click on the **Autostart** tab of the notebook.
3. Select **talkd** from the Service to autostart list.
4. Select **Autostart service**.
5. Select **Detached** or **Foreground session**. When Detached is selected talkd is started when TCP/IP starts; Talkd does not appear in the OS/2 task list. When talkd is started as a foreground session it is started in an OS/2 window.

   Your panel should look like this:

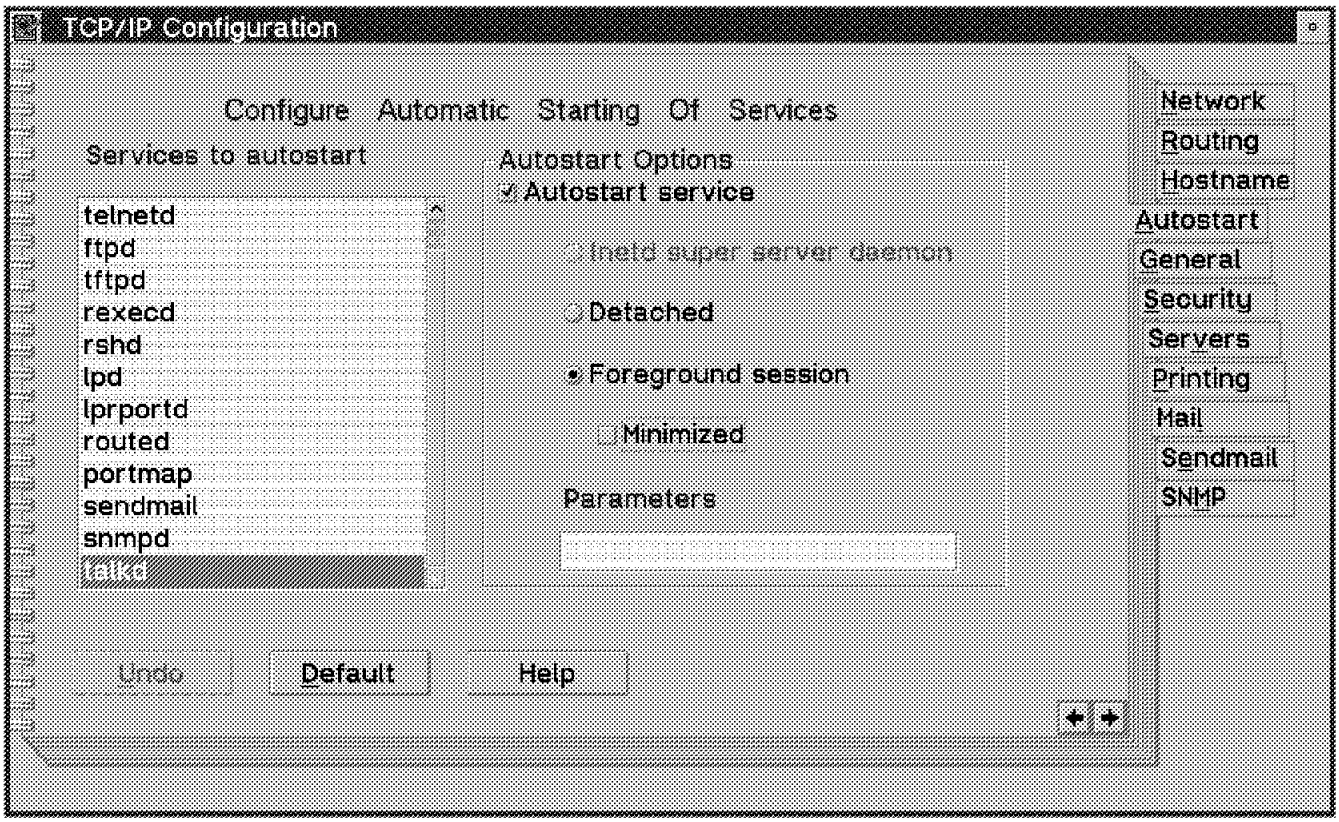


Figure 48. Talk Configuration (Autostart)

6. Close the configuration menu, by double-clicking on the top left-hand corner of the window.

   Click on **Save**.

   From now on, any time you execute TCPSTART.CMD, the TALK daemon will automatically be started.

6.2.1.2 Starting Talk from an OS/2 Command Prompt

To start Talk daemon from an OS/2 command prompt, enter `talkd`.

6.2.2 Using Talk

The following is an example of the steps required to establish a conversation between two OS/2 hosts, kwichman and walter in the domain itso.ral.ibm.com:

1. Ensure that you have talkd running on both systems that will partake in a conversation.
2. At the host kwichman, enter:

   ```
   talk os2user@walter
   ```

   If you want to talk to a user on an OS/2 system you have, to specify os2user as the user on that host. If you talk to a user on a multiuser system you specify the real user name.
3. At the host walter, the talkd session will beep and the following message will appear:

```
Message from Talk_Daemon@ at 15:01 ...
talk:  connection requested by os2user@kwichman.itso.ral.ibm.com.
talk:  respond with:  talk os2user@kwichman.itso.ral.ibm.com
```

Then, you should enter this command at a new OS/2 command prompt:

```
talk os2user@kwichman
```

4. Both machines now have an OS/2 session running Talk. You can enter text of your message and when you press Enter, the text will be sent to the other host which is partaking in the conversation. Your panel will look like this:

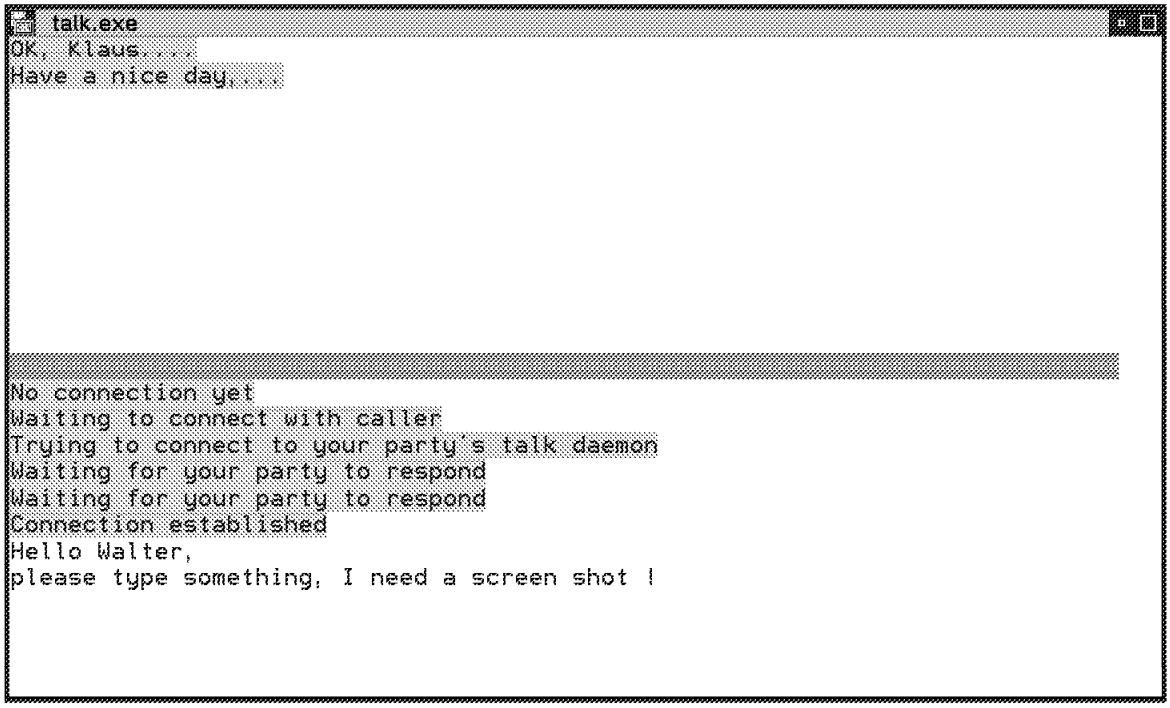


Figure 49. Talk Connection

6.3 UltiMail Lite

UltiMail Lite is a multimedia electronic mail system. With UltiMail Lite you can receive, read, create, and send electronic mail that contains a variety of media types including text, enriched text, images, audio, spreadsheets, and word processor documents. UltiMail Lite replaces the UltiMail/2 kit and LaMail from previous TCP/IP versions.

The UltiMail/2 kit came with an UltiMail client and a server. The new UltiMail Lite comes only as a client version. The client provides an interface to the user to write E-mail messages and to include multimedia information.

UltiMail Lite uses either SendMail to exchange mail with other systems or can connect to a POP (Post Office Protocol) server to receive mail. To receive mail from a POP server UltiMail Lite uses the Post Office Protocol. To send mail UltiMail Lite does not use the POP server and sends its mail directly via SendMail. As a stand-alone system that is not connected to a POP server, UltiMail Lite uses SendMail to send and receive mail.

6.3.1 The Advantages of UltiMail Lite

Some of the advantages of using UltiMail Lite to handle your electronic mail are the following:

- Interoperability
- Support for multimedia mail
- Compatibility with OS/2 Workplace Shell
- Use of OS/2 multimedia extensions

6.3.1.1 Interoperability

Because UltiMail Lite implements standard protocols, you can integrate it into a network with other mail handlers. UltiMail Lite uses TCP/IP′s Simple Mail Transfer Protocol, along with the mail formats specified by RFC 822 and MIME (RFC 1521), to send mail through your network. You can communicate with many types of systems, based not only on OS/2, but also other workstation and mainframe platforms available from IBM and other companies. UltiMail Lite uses standard protocols, rather than proprietary ones. This means that you can use it in your existing network. It also means that if you expand your network in the future, you can continue to use UltiMail Lite on your OS/2 systems.

6.3.1.2 Support for Multimedia Electronic Mail

Electronic mail has always lagged behind ″paper″ mail because, until recently, electronic mail has been limited to plain text, while paper mail can include ornate text, pictures and full color. You can even send audio and video tapes through the mail, although the recipient would require a tape player on which to play them.

UltiMail Lite takes electronic mail beyond the paper mail boundaries by providing you with the ability to send and receive mail using all these media types.

6.3.1.3 Compatibility with the OS/2 Workplace Shell

UltiMail Lite is designed to fit smoothly into OS/2′s Workplace Shell, providing a familiar environment in which to manipulate your mail. UltiMail Lite can be viewed as a master folder containing all of your electronic mail folders. Each of these mail folders can contain other folders and letters. Within the mail folders, letters are represented as envelopes and can contain enclosures (called letter parts).

You can handle all UltiMail Lite objects (folders, letters, and letter parts), just as you handle other objects on your OS/2 desktop, with the mouse and the keyboard.

UltiMail Lite conforms to the Common User Access (CUA) definition, making it consistent with other programs that you use.

6.3.1.4 Use of OS/2 Multimedia Extensions

UltiMail Lite uses OS/2′s multimedia extensions, available with OS/2, to communicate with special video capture and audio hardware. This means that as OS/2 supports new hardware adapter cards, UltiMail Lite will also support them. You will not have to wait for updates to UltiMail/Lite.

6.3.2 Setting up TCP/IP For UltiMail Lite

To set up TCP/IP for your UltiMail Lite system, you have to open the TCP/IP configuration notebook. Most items can be configured using the TCP/IP configuration notebook. After closing the notebook the changes will be saved in the configuration files of SendMail and UltiMail Lite. If your system operates in an unusual environment, manual changes to the configuration files may be required.

To configure your UltiMail Lite environment follow these steps:

- Open the TCP/IP configuration notebook and click on the label **Mail**. You will see the first page of two.

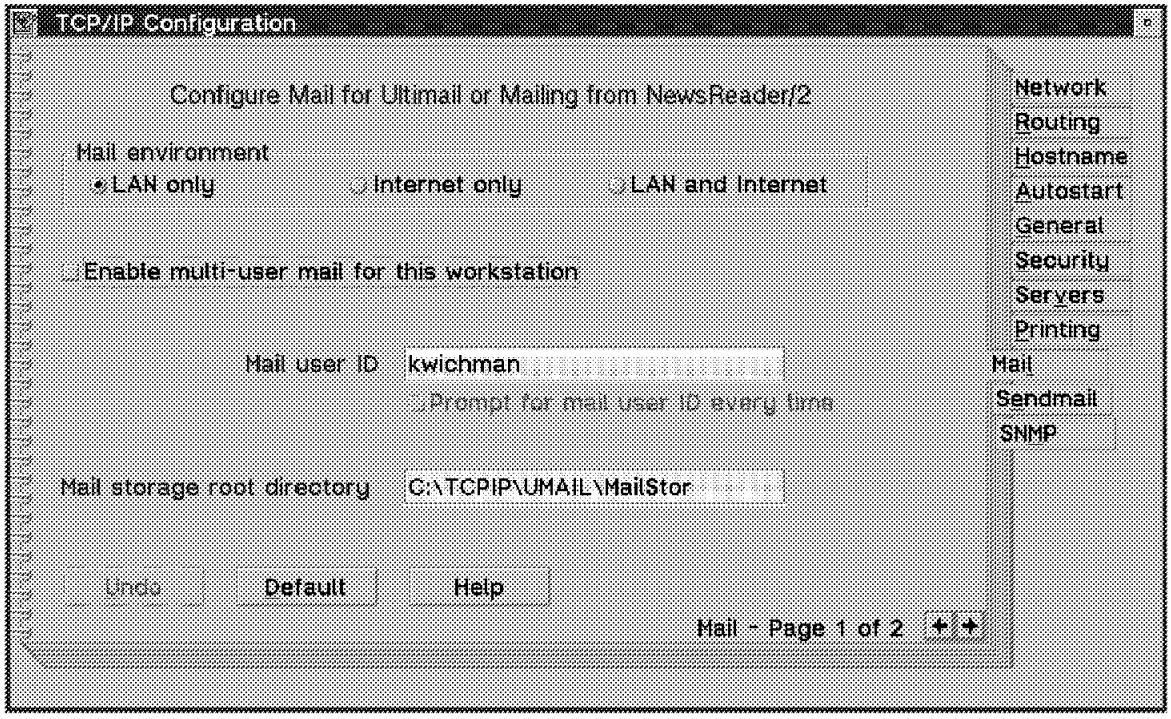


Figure 50. Configure Mail for UltiMail Lite

- Select one of the following:
  - **LAN only**: If your host is connected to a LAN. This option should also be selected if your LAN has a connection to the Internet and you want to access the Internet.
  - **Internet only**: If you want to connect to an Internet provider via a modem.
  - **LAN and Internet**: If you want to connect in both ways described.
- Select **Enable multi-user mail for this workstation** if you want to allow multiple users (such as individual members of your office) to have individual UltiMail Lite accounts on this workstation.

  If you allow multiple users on this workstation, you can also specify whether you want UltiMail Lite to prompt you for the mail user ID each time it is started.

  If an ID is specified, it will be used as the default. The default mail user ID is your host name.

- Specify the mail storage root directory. For UltiMail Lite the entry should be:
  `C:\TCPIP\UMAIL\MailStor`
- Click on the right arrow in the right corner at the bottom of the notebook to see the next configuration page.

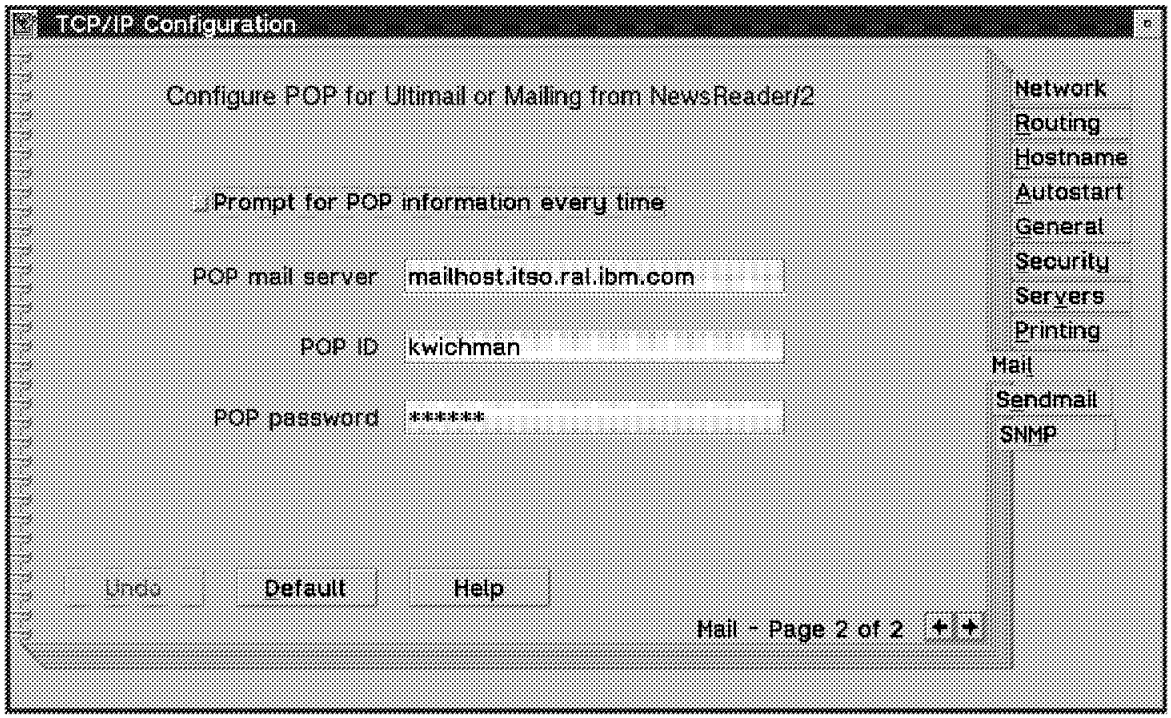


Figure 51. POP Configuration for UltiMail Lite

If you want to use a POP server to receive your mail, configure your host for using that server on this page.

- If you want UltiMail Lite to prompt you for your POP information each time you start up, select this check box.
- Specify the hostname of the mail server assigned to you on your LAN. It is recommended to use a hostname instead of the IP address.
- Specify your user ID assigned to you for access to the mail server on your LAN. If the POP ID field is blank, the mail user ID is used as the default.
- Specify your password for access to the POP server.
- Select the label **Sendmail** to see the next configuration page.

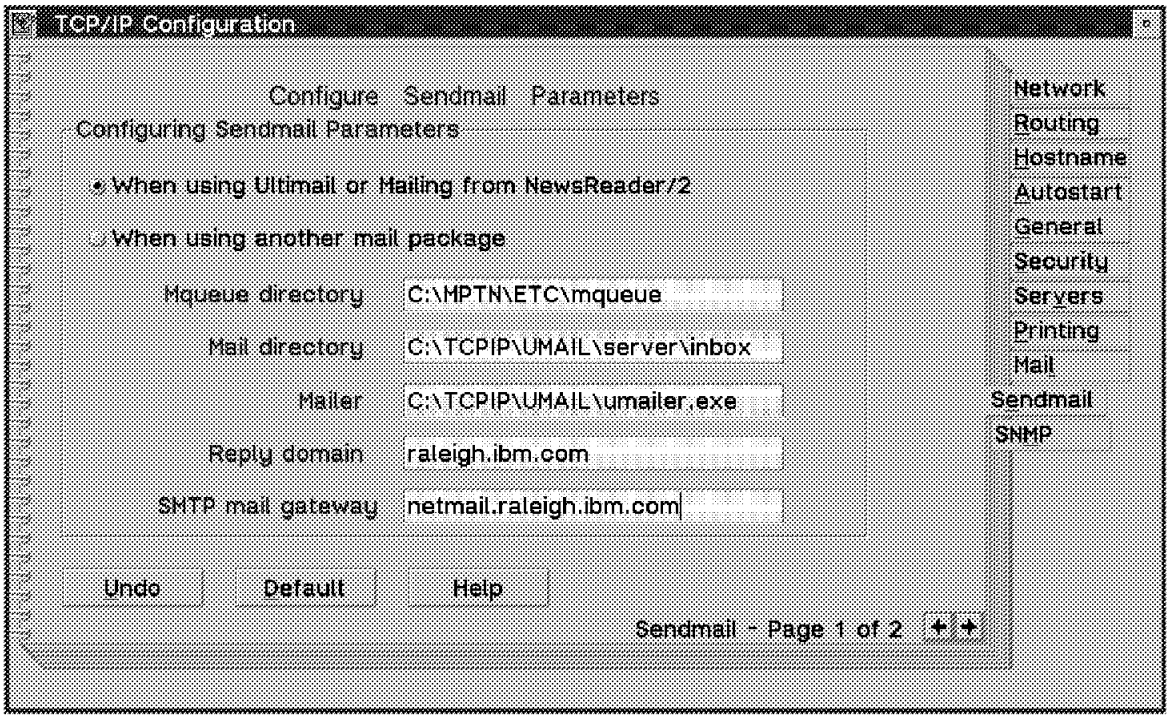


Figure 52. Configuring Sendmail for UltiMail Lite

This page and the next make changes to the SendMail configuration file.

- Select **When using Ultimail Lite or Mailing from NewsReader/2**.

  The UltiMail Lite configuration information is saved in the SENDMAIL.UML file. Information for other mail packages would be stored in the SENDMAIL.CF file.

- The next three fields should be set with the correct values by default. Depending on your directory structure you should see the following values:

  ```
  Mqueue directory   C:\MPTN\ETC\mqueue
  Mail directory     C:\TCPIP\UMAIL\server\inbox
  Mailer             C:\TCPIP\UMAIL\umailer.exe
  ```

  The Mqueue directory will define the OQ parameter of your SendMail configuration file. Mail directory and Mailer are defined in the Mlocal parameter.

- Specify the name of the domain in which your mail server resides. The domain name includes all subdomains and the root domain separated by periods. This is not a required field, and if the Reply Domain field is left blank, no default will be provided.

  The reply domain is set in the DI parameter of your SendMail configuration file.

- Specify the hostname of the SMTP gateway that you use. The SMTP mail gateway routes the mail to the recipients. The SMTP mail gateway is analogous to a POP server.

  If your connection is through a LAN, the SMTP mail gateway is optional. If your network uses an SMTP mail gateway, enter its hostname.

  If your connection is through a service provider, the SMTP mail gateway host name is assigned by your provider.

This will set the DR parameter in your SendMail configuration file.

- Switch to the next page of the SendMail configuration.

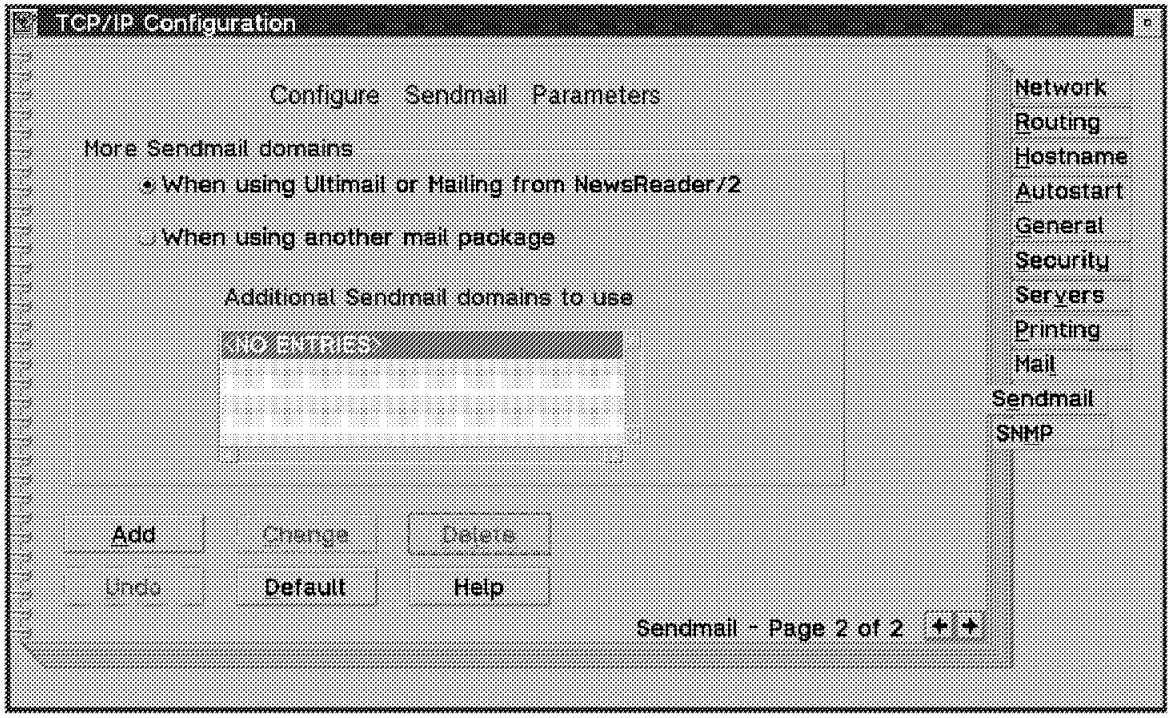


Figure 53. Adding Additional Domains to Sendmail

- If you are in a LAN mail environment you can specify up to three additional domains. Click on **Add** to add an additional domain.

  An additional domain is specified to ensures that outgoing, LAN-based mail destinations beyond your local domain are reached.

  For example, your local domain is raleigh.ibm.com, and you want to send mail to the following destinations:

  - hostname.raleigh.ibm.com (local domain address)
  - hostname.atl.ibm.com (internal network address)
  - hostname.eos.ncsu.edu (Internet address)
  - hostname.eng.mit.edu (Internet address)

  If no additional SendMail domains were specified in the list box, all of the mail would be delivered successfully except that addressed to hostname.atl.ibm.com because it is not part of the local domain or the Internet.

  To ensure that the internal network address mail (for example, ibm.com) would be delivered to any addressee outside of your local domain, you would add ibm.com to the Additional Sendmail Domains to Use list box.

- There are more SendMail parameters that will be affected by the settings in the TCP/IP notebook.

  CW Hostname and domain name

  DW Hostname and domain name

  DD Your domain name

- Close the configuration notebook by double-clicking in the upper left corner. Click on **Save** to save your changes. If SendMail is not configured for autostart yet, click on **Yes** when the following message appears. This ensures that the sendmail.uml configuration file is used by SendMail and modified correctly.

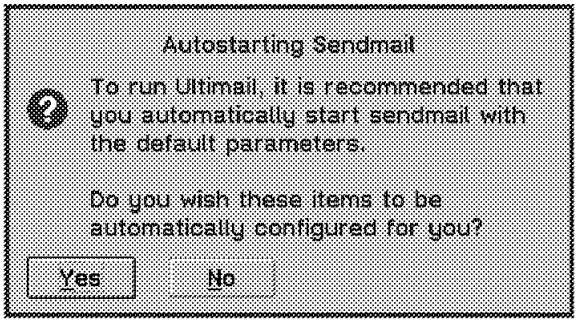


Figure 54. Autostart Sendmail

- If you use an external mail gateway, you must specify the DV parameter in your sendmail.uml configuration file. Use an editor that preserves tabs to make the changes to the file. See 6.1, “SendMail” on page 119 for more information.
- Your UltiMail Lite configuration is now complete.

6.3.3 Setting Up Your UltiMail Lite Environment

To set up your UltiMail Lite environment you have to start the program from the UltiMail Lite folder.

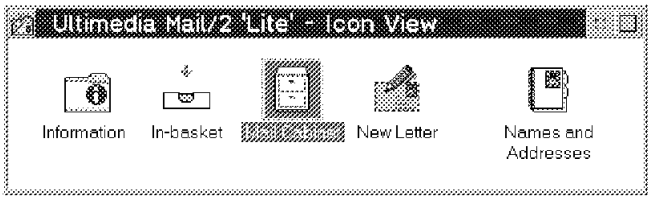


Figure 55. UltiMail Lite Icon View

The first time you start UltiMail Lite by double-clicking one of the **In-basket**, **Mail Cabinet** or **New Letter** icons you have to specify your user name, your password and your reply domain.

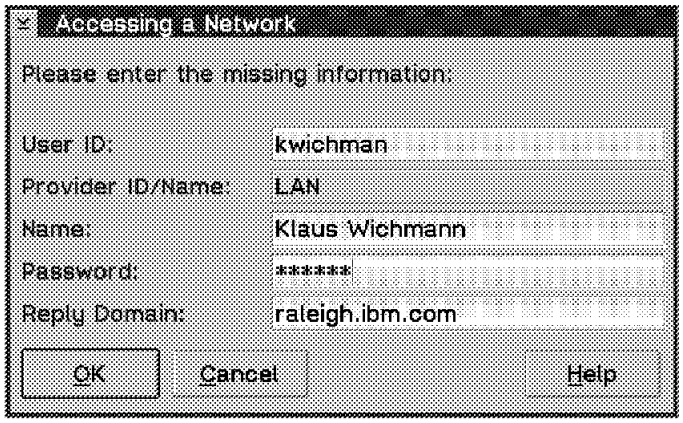


Figure 56. UltMail Logon

Fill out the following fields:

User ID This user ID is your primary ID and can be used to access the Internet or a LAN, depending on what you have set up in your TCP/IP configuration. If you picked only one network to identify the kind of user you are, you will only receive mail for that primary ID. If you picked both in your TCP/IP configuration, you will be asked if you want both networks to be linked. Choose to link them; otherwise, you will only receive mail for your primary ID.

A subdirectory with that name will be created in your tcpip\umail\mailstor directory. That directory stores your mail and other information needed by UltiMail Lite, such as the index or your address book.

The default is your mail user ID that you have specified in your TCP/IP configuration notebook.

Provider ID/Name This is only available if you picked both Internet and LAN. You can enter your user ID for the provider (or provider nickname in case of the Dial Other Internet Providers dialer). If you picked only one, this field will be gray-shaded and contains the string Internet or LAN respectively.

Name This is the name that you want to appear as.

Password The access password for UltiMail Lite. This password is needed to access your mail. If the field does not contain a line of asterisks, type your password. If it is the first time you are using UltiMail Lite, the system will ask you to enter the password twice.

Figure 57. Reenter Password

Reply Domain The domain where you want reply mail to be sent. By default your reply domain is your hostname plus your local domain name, that you have specified in the TCP/IP configuration notebook.

Once you have filled out all necessary fields, click on **OK**. When you have selected **Prompt for POP information every time** in the TCP/IP configuration notebook, the following panel will appear:

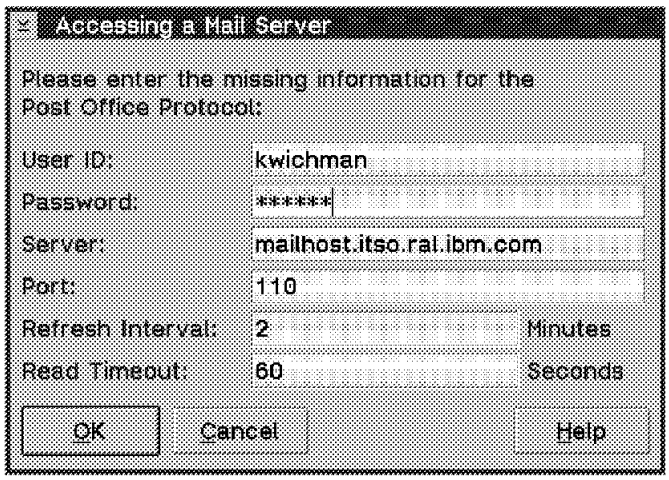


Figure 58. Prompt for POP Server

The Accessing a Mail Server window allows you to define the information that UltiMail Lite uses to retrieve your mail via the Post Office Protocol (POP). Information necessary for retrieving mail via POP is normally supplied to UltiMail Lite by one of the dialer applications if you are connected to an Internet service provider. This window is also displayed if POP information is missing. If you use a POP server on the LAN, you can also configure POP information through the TCP/IP Configuration program.

Fill in the following information:

Server Full network name of the POP server where UltiMail Lite retrieves your mail.

User ID Your account on the POP server.

Password Password for your account on the POP server.

Port Network port number on which UltiMail Lite contacts the machine to retrieve mail. The default port number is 110. You can type `none` to indicate you do not want to use a POP server.

Refresh Interval Amount of time after which UltiMail Lite contacts the POP server to check for new mail.

Read Timeout This time indicates how long UltiMail Lite waits for a response from the POP server before it gives up.

When you have filled in the information, click on **OK**. UltiMail Lite is now ready to send and receive mail. The next section helps you customize UltiMail Lite to fit your individual needs.

6.3.4 UltiMail Lite Customization

This section describes how to use the UltiMail Lite Settings notebook to set up the UltiMail Lite for your individual use. Your configuration will be stored in the profile \TCPIP\UMAIL\UMAIL.PRO.

To open the UltiMail Lite Settings notebook, do the following:

1. Start UltiMail Lite by double-clicking on the **Mail Cabinet** icon at the UltiMail Lite folder.
2. Choose **Settings** from the Cabinet menu.

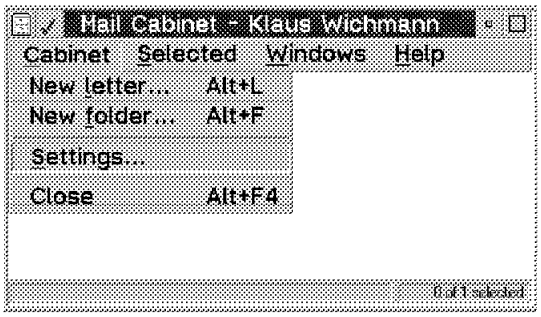


Figure 59. Opening the UltiMail Lite Settings Notebook

You will see the UltiMail Lite Settings notebook with the first page of the Letter section on top. This is the same page as if you had selected **Settings** from the Letter menu of the New Letter application. It is recommended to use the Settings notebook of the Mail Cabinet because it contains all other configuration notebooks.

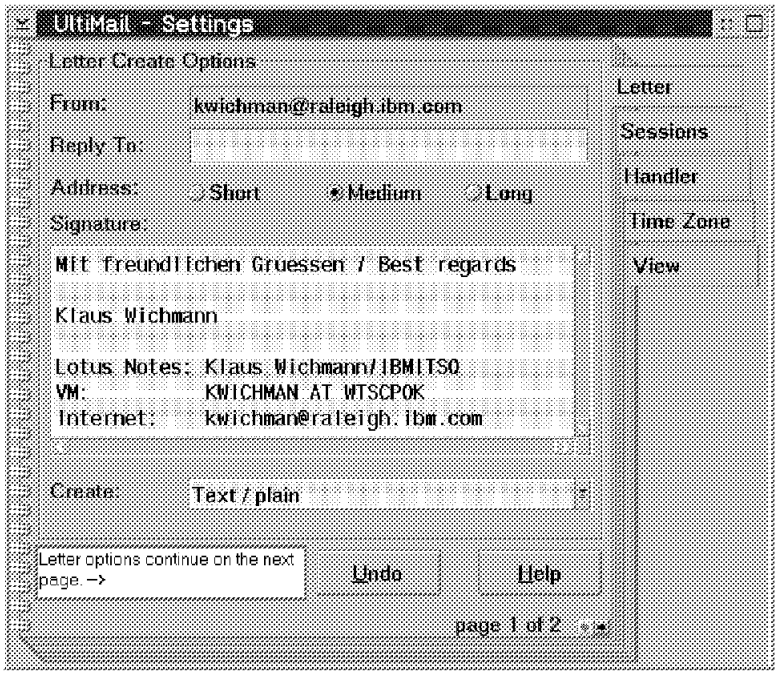


Figure 60. UltiMail Lite Configuration Notebook - First Letter Page

To set or change the options for your letter, do the following:

1. If you want replies sent to an address other than your user ID, type the address in the Reply To field.
2. Select **Short**, **Medium**, or **Long** to indicate the address format. Depending on your choice, your real name is displayed in front of your E-mail address.
3. If you want the first text-part in each letter to have a default signature, such as a greeting or your name, type the lines in Signature.
4. If you want a default letter-part to be created automatically for every letter, select the part from the list for Create. This letter part will be the default first letter-part for all new notes.

On the next page of Letter you have more options to customize your letters.

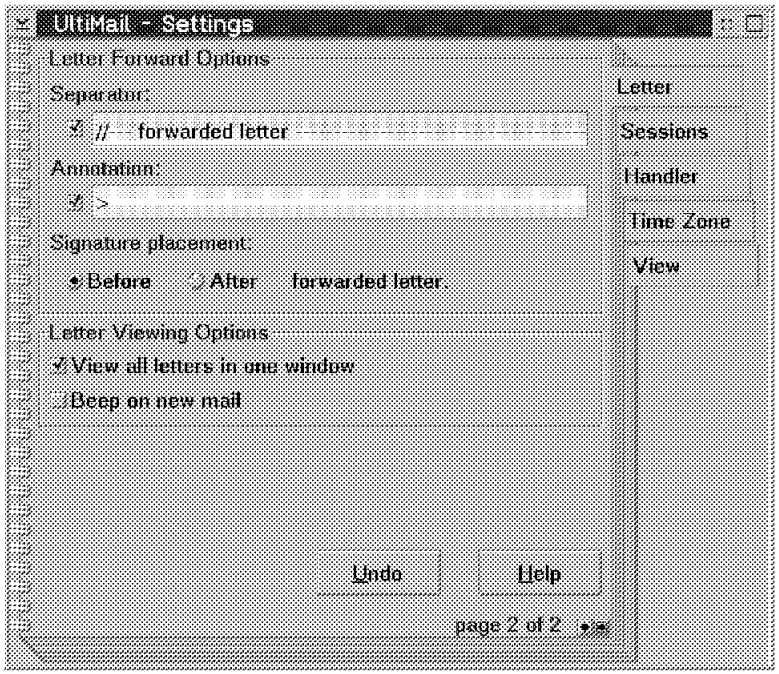


Figure 61. UltiMail Lite Configuration Notebook - Second Letter Page

- For Letter Forward Options, select **Separator** if you want a line separating your text from the forwarded text, **Annotation** if you want a mark put before each line of forwarded text, and **Before** or **After** if you want your signature before or after the forwarded letter.
- For Letter Viewing Options, select **View all letters in one window** if you want all letters displayed in one window, instead of opening a new window for each letter and select **Beep on new mail** if you want a beep to occur when new mail arrives in your in-basket.

It is recommended to keep the default values in the configuration pages for Sessions, Handler, Time Zone and View. If you need to make changes to one of these pages, please refer to the online help that comes with UltiMail Lite. Only two additional parameters, Retry and Interval, are mentioned here.

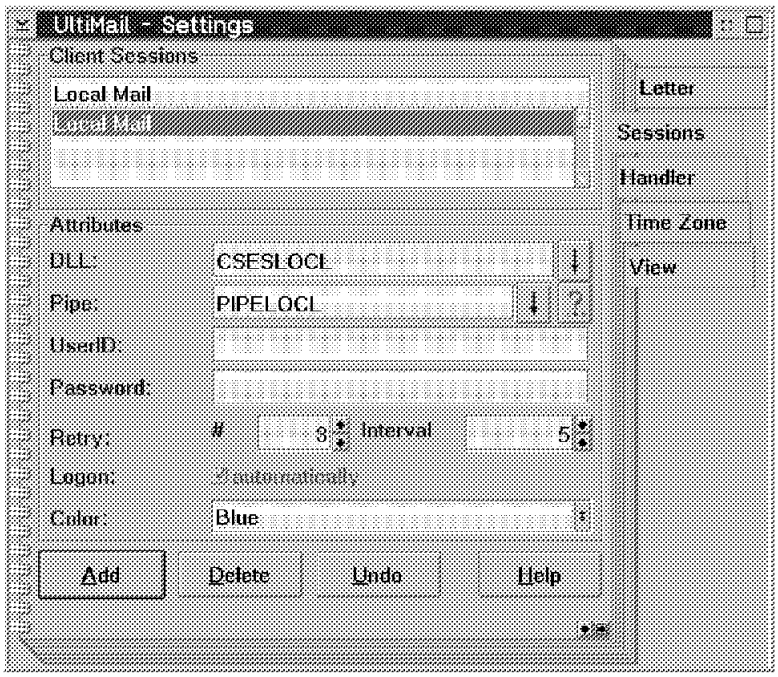


Figure 62. UltiMail Lite Configuration Notebook - Session Page

With Retry you specify the number of retries that UltiMail Lite attempts to send mail. After its expiration, it informs you that a connection could not be established. Interval specifies the time in seconds that UltiMail Lite waits after a connection fails until it retries to establish the connection again.

On the Sessions page of the settings notebook, you can define the sessions you can log on to. You are automatically logged on to your mail server and thus do not have to change this page unless you are logging on to additional sessions.

On the Handler page of the Profile settings notebook, you can select **Object handlers**. An object handler allows you to display a visual representation of data when creating or viewing letters. For example, an editor can be used to display text parts while a painting program can be used to display images. UltiMail Lite has a default object handler for each of the supported types of data.

On the Time Zone Options page of the Profile settings notebook, you can set the time zone. The time zone is used to convert mail received from other time zones to your time zone so that letters are correctly sorted according to the time they were created.

After you have customized UltiMail Lite for your needs, close the configuration notebook to save the settings in your profile.

6.3.5 Using UltiMail Lite

This section describes how to use UltiMail Lite. It shows how to create new entries in your private Names and Address book and how to send and receive mail. You also learn how to handle multimedia documents with UltiMail Lite.

When you open the UltiMail Lite icon view, you will see the following icons:

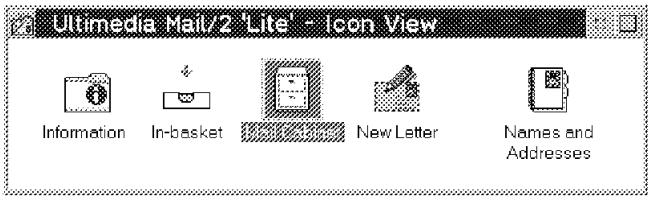


Figure 63. UltiMail Lite Icon View

Except for the Information folder which provides you with additional information on how to use UltiMail Lite, you see icons that help you to handle your E-mail in an easy way.

The major application is the Mail Cabinet. From the Mail Cabinet you can start any of the other applications. Therefore it is recommended to start the Mail Cabinet first.

Clicking on the **In-basket** shows your received mail. With the New Letter application you can create and send multimedia documents. The Names and Address book helps you to manage your addresses. These applications are explained in the following sections.

6.3.5.1 The Names and Address Book

You open the Names and Address book by double-clicking on the **Names and Addresses** icon. Your address book will look somewhat like the following:

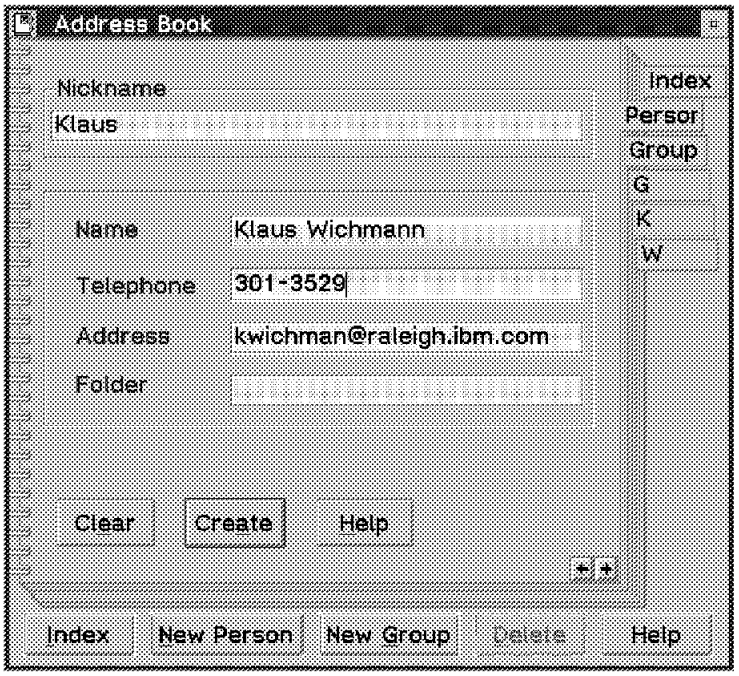


Figure 64. UltiMail Lite Address Book - Person Tab

Every time you create a new nickname in the address book starting with a new letter, the address book is enhanced with that letter. The above shows an address book with three tabs for G, K, W. There are also tabs for Index, Person and Group. The Person and Group tabs are used to create new people and groups. The index will show all entries of the address book regardless of person and groups.

How to Create a New Person in The Address Book: You can create a new person by clicking on **New Person** or selecting the tab **Person**. As shown in the last figure you fill in all information about that person. When you've done so, click on **Create** to create a new entry for that person in your Names and Address book.

The Person page contains the following fields:

Nickname Nickname for the person. This field is required and cannot contain spaces.

Name Name of the person.

Telephone Telephone number of the person.

Address The Internet or LAN address of the person.

Folder Default folder in which you want to store mail for this person.

The new nickname will appear in the index and in the alphabetically ordered tab list.

How to Create a New Group: You can create a new group of people by clicking on **New Group** or selecting the tab **Group**. The following panel will appear:

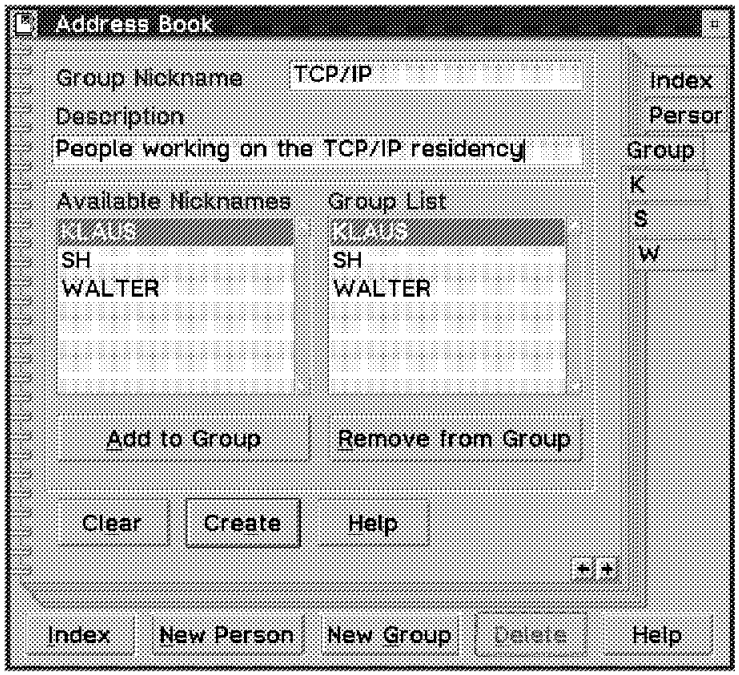


Figure 65. Create a Group

In the Available Nicknames list all entries in the address book are displayed by their nickname. To add a nickname to the group, you simply select the nickname and click on **Add to Group**. The nickname will be shown in the group list. When you have selected all the nicknames you want to be in the group and filled out all the information required in the other fields, click on **Create** to create a new entry in the Names and Address book.

The Group page contains the following fields:

Group Nickname contains the nickname for the new group. This field is required and cannot contain spaces.

Description contains a description of the group.

Available Nicknames contains the list of members available to be added to this group.

Group List contains a list of members currently belonging to this group.

After you click on Create, the group shows up in the index and in the alphabetically ordered tab listing.

6.3.5.2 How to Send Mail

To create a new letter either double-click the **New Letter** icon or select **New Letter** from the Cabinet menu of the Mail Cabinet. The following window appears:

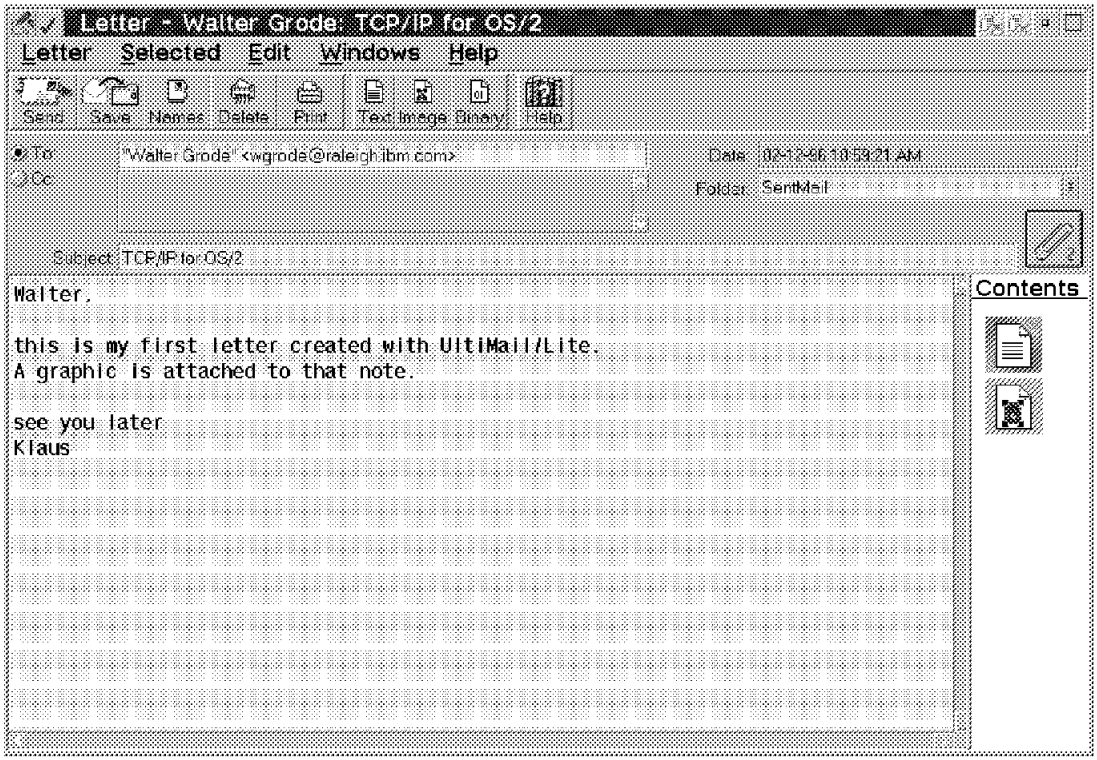


Figure 66. Create New Letter (1 of 2)

The following is a step-by-step description of how to create and send your letter:

- Fill in the To field. You can either type in the E-mail address or select a nickname from the address book. You request a list of all nicknames defined in your address book by selecting **Address letter** from the Letter menu or pressing mouse button 2 when the cursor is in the To field.
- Select the folder where you want to save your mail. You can only select a folder from the folder list. This list contains all folders that you have created in your Mail Cabinet. The default folder is SendMail.
- Fill in the subject of your mail.

- Clicking on the icon showing a paper clip will display the contents of the document. There are different icons for attached images, binary files or text. If you click on one of these icons, you switch to that document.
- Type in the text you want to send to the addressed person. This is plain text and can also be sent to destinations not supporting MIME.
- To create a graphic, click on the **Image** smarticon. The following window appears:

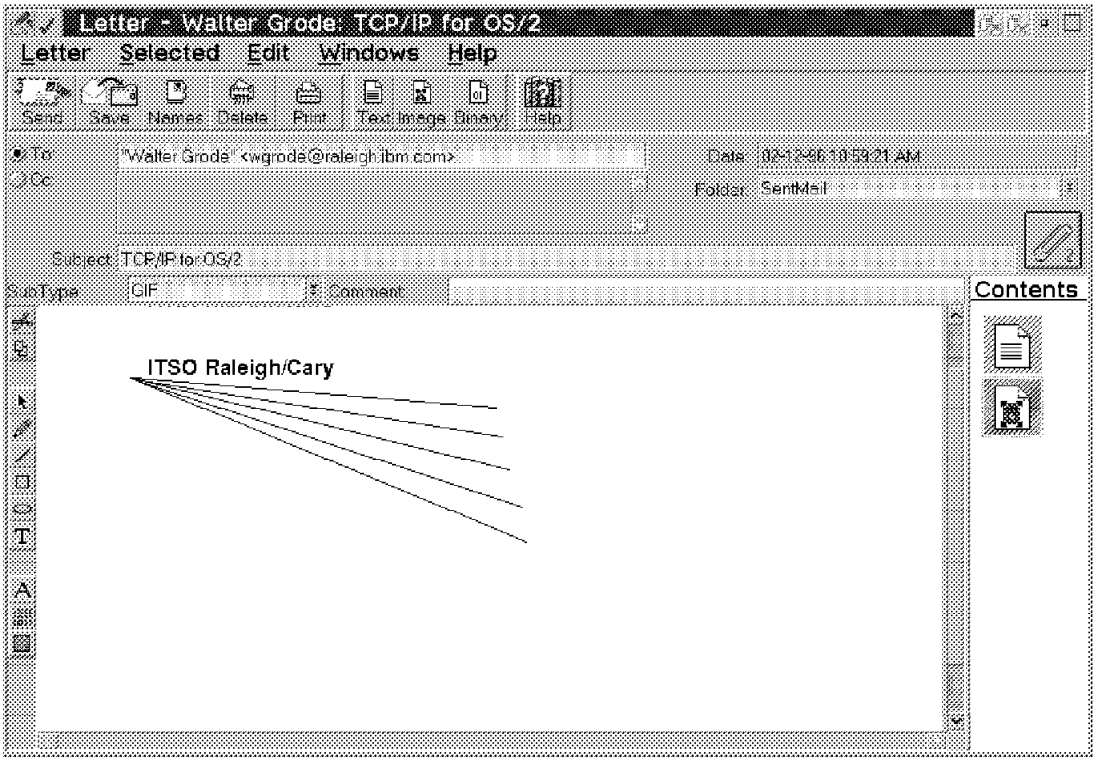


Figure 67. Create New Letter (2 of 2)

- You can create a graphic as in other drawing applications. A new icon for that attachment is shown in the contents list on the right side.
- If you want to switch back to your text, simply click on the **Text** icon from the contents list.
- When your letter is complete, you can save it by clicking on the **Save** smarticon. Your letter will be saved in the folder specified in the Folder field of your letter.
- To send your letter you click on the **Send** smarticon. Your letter will then be sent and confirmed by the following message:

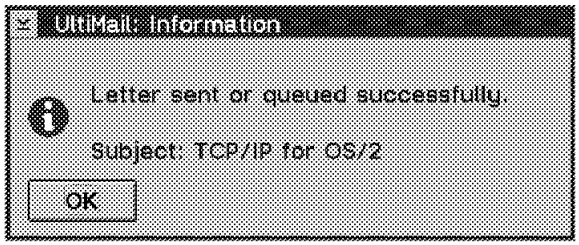


Figure 68. Sending a Letter

6.3.5.3 Reading Received Mail

To read your incoming mail, you have to open your in-basket. You can double-click on the **In-basket** icon or select **In-basket** from the window menu of the Mail Cabinet. You see a list of already read and new unread mail. The read mail is marked by an opened envelope and the unread mail is marked by a closed envelope. Your In-basket will look similar to the following:

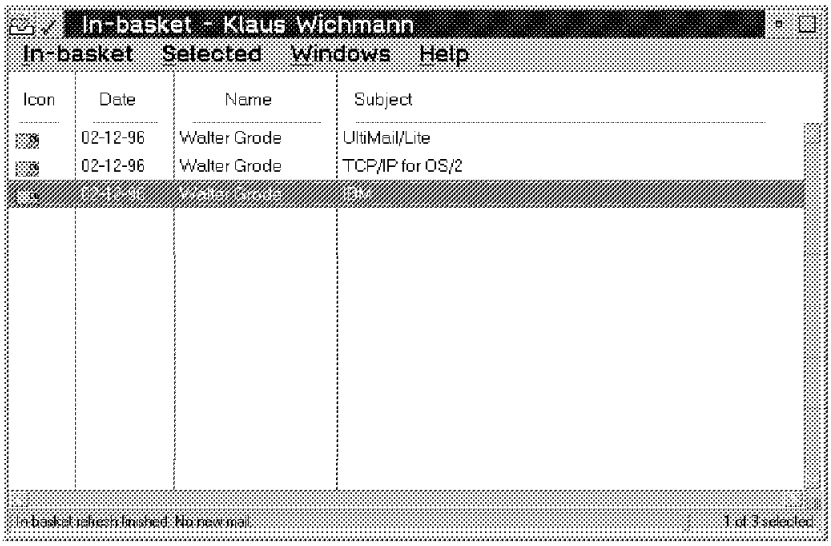


Figure 69. The In-Basket (1 of 2)

To read your mail, you simply double-click on the letter that you want to read. The letter is then opened and you can see its contents. The window for reading a letter is the same as for writing a letter. If you want to reply to mail, simply click on the **Reply** smarticon after you have opened the letter.

6.3.5.4 The Mail Cabinet

The Mail Cabinet helps to manage your mail. You can create different folders to store and categorize your mail. When you start UltiMail Lite for the first time, there is only one folder to save your mail. That folder is called SendMail. You can create new folders to organize your mail as you wish.

The following figure shows the Mail Cabinet with five folders to handle the mail. Within the icons, the number of documents contained in a folder is shown.

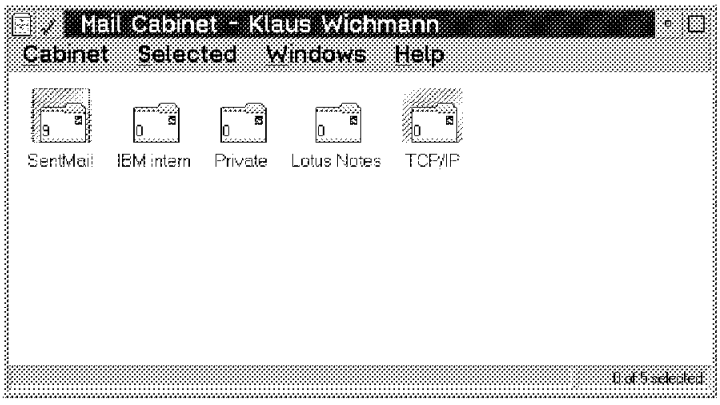


Figure 70. The In-Basket (2 of 2)

To create a new folder, select **New folder** from the Cabinet menu. Type in the name of the folder and click on **OK** to create the folder. The server field contains your hostname by default. If you want to create that folder on another system, select a server from the server list. When you have created the new folder, a new icon will be displayed in your Mail Cabinet window.

Once you double-click on an icon in the Mail Cabinet window a list of documents is shown. The list looks like your in-basket but stores all documents that you have saved in that particular folder. By double-clicking a document of that list, you can see its contents.

From the Mail Cabinet you can open the address book, create new letters or open your in-basket. You can start these applications from either the Cabinet or the Window menu.

6.3.5.5 The UltiMail Lite Tutorial

UltiMail Lite comes with a very good tutorial program. Double-click the **Tutorial** icon in the Information folder of the UltiMail Lite folder and see what you can do with UltiMail/Lite.

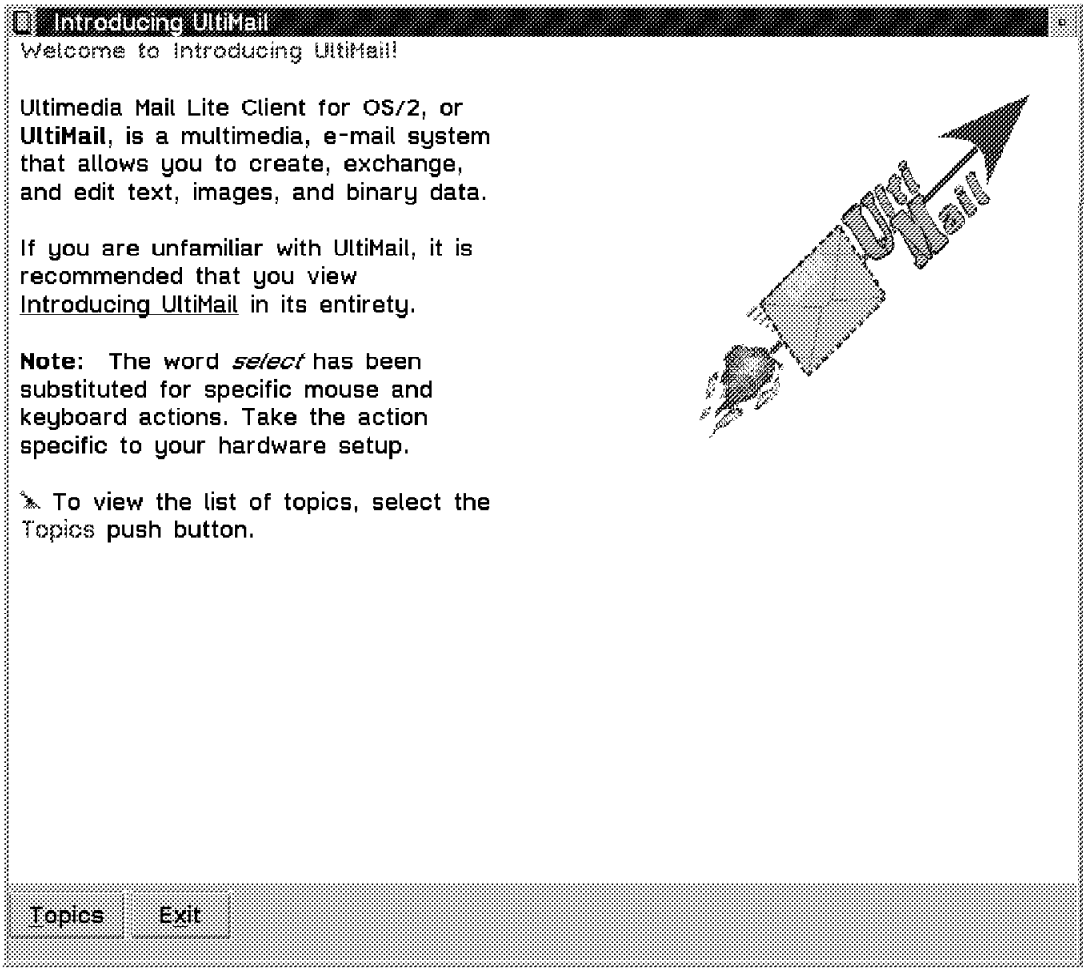


Figure 71. UltiMail Lite Tutorial Program

There are quizzes to check what you have learned.

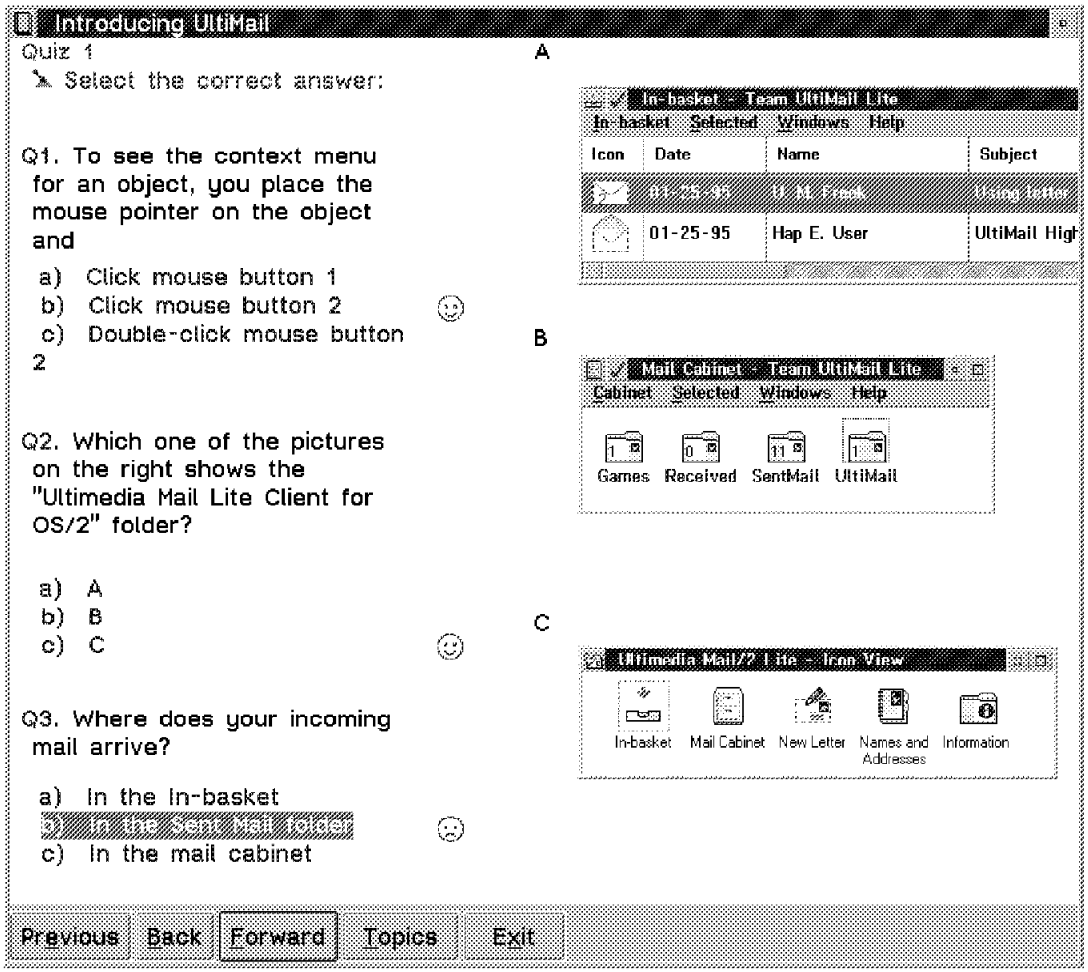


Figure 72. UltiMail Lite Tutorial Program Quiz

The tutorial is a great tool for you to start using the UltiMail Lite system.

6.4 SMTP (RFC 822) and MIME (RFC 1521)

This chapter describes the RFCs that are used by UltiMail Lite to send mail between users on the Internet.

6.4.1 SMTP (RFC 822)

RFC 822 is a standard for a mail format for use with the Simple Mail Transfer Protocol (SMTP). All systems that use SMTP can exchange mail, but each must understand certain information, such as sending and receiving addresses, in order to handle the mail successfully. RFC 822, which is used by virtually all SMTP handlers, specifies the format of the mail headers so that they may all handle each other's mail. UltiMail Lite uses the RFC 822 standard, and is compatible with other such systems.

6.4.2 MIME (RFC 1521)

The RFC 822 standard was written when all electronic mail was in plain text form, and it handles only plain text. The requirements of multimedia mail and of SMTP necessitate extensions to RFC 822. These extensions are in RFC 1521, known as Multi-purpose Internet Mail Extensions, or MIME. MIME specifies how to encode and encapsulate non-text attachments to electronic mail. It defines standard ways of packaging one or more separate objects into a message so that any MIME-compliant mail system will understand it. The UltiMail Lite implementation uses the term *letter* to mean a message, and the term *letter part* to mean one of a set of objects in a letter. UltiMail Lite uses RFC 1521 standard, making it fully compatible with these systems as well.

Since MIME is an extension to RFC 822, systems that use MIME can understand RFC 822 mail, and systems that use RFC 822, but not MIME, can understand the plain text portions of a MIME letter. That makes UltiMail Lite compatible with all RFC 822 systems, even those that don′t use MIME.

6.5 SMTP and Lotus Notes

Lotus Notes is a proprietary mail system. It′s easy to use and easy to exchange mail with other Lotus Notes users. It is even possible to include multimedia information like rich text, graphics, video and sound. Also, file attachments or references can be included in a document. Notes also helps you organize your information and mail in a user-friendly way. This makes Lotus Notes a great system for electronic mail.

As the Internet is growing every day and millions of people use E-mail on the internet, there is a need for Lotus Notes users to have the ability to exchange E-mail with Internet users. There are different products that enable Lotus Notes to exchange mail with other systems. For example, mail can be exchanged with IBM Profs, DEC′s VAX/VMS Mail, X.400 and others. As this book focuses on TCP/IP, the following shows how to exchange mail with SMTP systems. The gateway used to deliver mail is called the Lotus Notes Mail Gateway for SMTP. This gateway uses the SendMail application explained in the beginning of this chapter.

6.5.1 The Lotus Notes Mail Gateway for SMTP

Notes and SMTP messages have different formats. When a message is sent from one system to another it must be converted to a format that can be read in the target mail system.

A Notes mail message consists of a header, an optional body, and optional attachments. The Notes header consists of field name and value pairs, such as the To, From, Date, and Subject fields, which are analogous to the SMTP header. The Notes body can contain embedded textual information such as color, italics, icon information, and document links. The Notes body can support rich text format (RTF). Notes attachments may be any native filesystem file.

There are several types of Notes messages. The Memo form is a user-generated message; examples of system-generated messages are DeliveryReport, NonDeliveryReport, and ReturnReceipt.

An SMTP message consists of a header and an optional body. The SMTP header contains fields of information about the message and its destination. The

SMTP body is optional and contains short lines of seven-bit US-ASCII characters. The SMTP message body can include rich text through an ASCII representation.

The SMTP Gateway has to take care of the conversion between the different message formats. It also has to translate the different address formats to send the messages to the correct recipients.

6.5.2 Message Routing between Lotus Notes and SMTP

The following explains how messages are routed between SMTP systems in the Internet and Lotus Notes domains.

6.5.2.1 Notes to SMTP Message Routing

The SMTP Gateway is defined to Notes as a foreign domain. Mail sent through the SMTP Gateway is deposited in the gateway mail file of the foreign domain on the Notes server. The SMTP Gateway retrieves mail from that file, converts it to SMTP format, and submits it using SendMail.

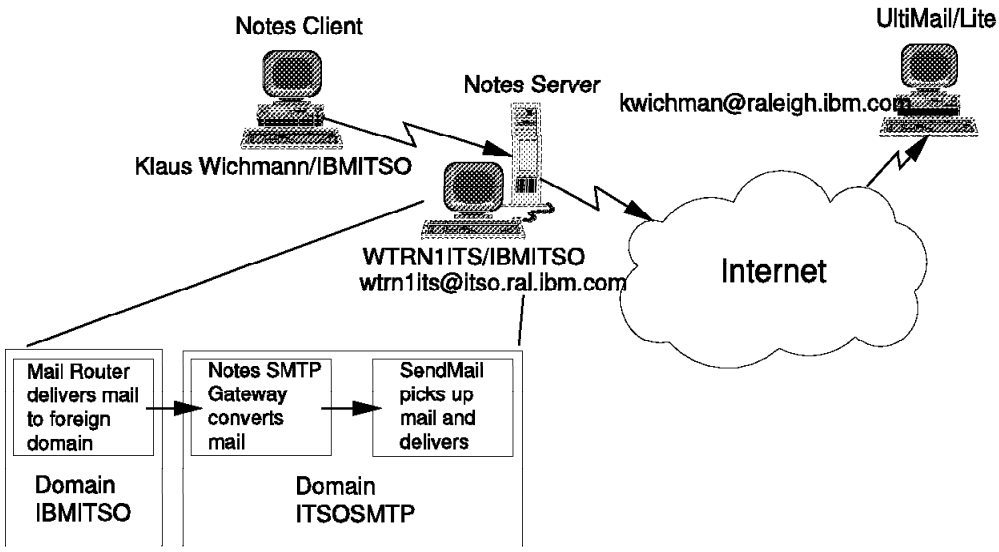


Figure 73. Notes to SMTP Mail Exchange (1 of 2)

6.5.2.2 SMTP to Notes Message Routing

All SMTP mail addressed to the Notes OS/2 server is redirected to the SMTP Gateway. This is done by reconfiguring SendMail during the SMTP Gateway installation (done automatically by the installation program) to use the delivery agent program (GWMAILER.EXE) to transfer mail to the SMTP directory. The SMTP Gateway retrieves the mail from this directory, converts it to Notes format, and submits it to the Notes mail router.

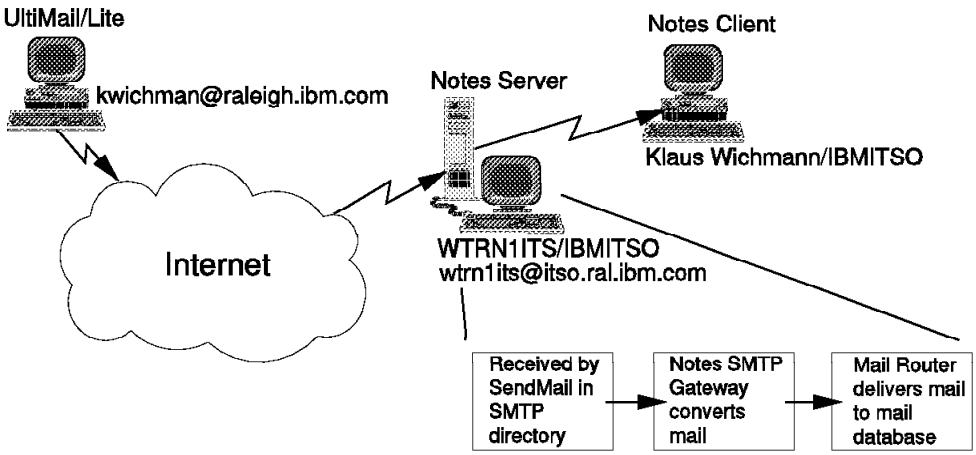


Figure 74. Notes to SMTP Mail Exchange (2 of 2)

6.5.3 Setting Up The SMTP Gateway

Installing the SMTP Gateway is an easy process. There are seven basic steps that need to be performed. The detail of the SMTP installation is not the subject of this book. Please refer to the SMTP gateway installation guide for more information. The basic installation steps are as follows:

1. Stop the Sendmail process.
2. Run the Installation program. To do so, insert the Notes Mail Gateway for SMTP disk in your disk drive, make the drive that contains the disk the current drive, enter `install`, and follow the directions on the screen. This copies all necessary files to your system and makes the changes to your sendmail.cf configuration file. The following is an example of a sendmail.cf configuration file using a mail gateway to send mail to the internet:

```
# DwYOUR-HOST-NAME
Dwwtrn1its.itso.ral.ibm.com
# CwYOUR-HOST-NAME
Cwwtrn1its.itso.ral.ibm.com
# DRYour.Internal.Gateway
DRraleigh.ibm.com
# DVYour.External.Gateway
DVraleigh.ibm.com
# DHYour.External.Mail.Hub
# DIYour.Internal.Mail.Hub
DIraleigh.ibm.com
# DPYourExternalUserID
# DDetc\mail
DDitso.ral.ibm.com
# Version # of this file
DZ2.12um
Dj$w

#
# Standard macros
#

# SMTP initial login message
De$j Sendmail $v/$Z ready at $b
# Name used for error messages
DnMailer-Daemon
# UNIX header format
DlFrom $g  $d
# Delimiter (operator) characters
Do.:%@![=/[]
# Format of a total name
Dq$?x$x <$g>$|$g$.
```

```
#
# Options
#

# Process messages in the background.
Odbackground
# Accept old style addresses
Oo
# SMTP read timeout
Or15m
# Queue directory - this must be changed if TCP/IP is moved!
OQc:\mptn\etc\mqueue
# Always queue for safety
Os
# Time to live in the queue
OT5d

   ...

#
# SMTP, Local and Program Mailer specifications
#

Msmtp,   P=[IPC], F=mDFMuX, S=10, R=0, A=IPC $h
# Mlocal,P=C:\TCPIP\UMAIL\umailer.exe , F=lsm, S=10, R=0, A=c:\mptn\etc\mail C:\TCPIP\UMAIL\SERVER\INBOX -to $u
Mlocal,  P=D:\TCPIP\BIN\gwmailer.exe, F=lsm, S=10, R=20, A=D:\TCPIP\notesgw -f$g $u
Mprog,   P=xxx, A=Required by sendmail but unused

   ...

# Sendmail configuration file *must* end with a new line - do not remove below new line
OHc:\mptn\etc\sendmail.hf
OAc:\mptn\etc\aliases
OSc:\mptn\etc\sendmail.st
```

3. Restart the SendMail process.
4. Create a foreign domain for the SMTP Gateway by opening the Notes Names and Address book, and choosing **Compose** - **Domain** - **Foreign**. Create a foreign domain name that is easy to remember (such as ITSOSMTP) because you use this name each time you address mail to SMTP users.

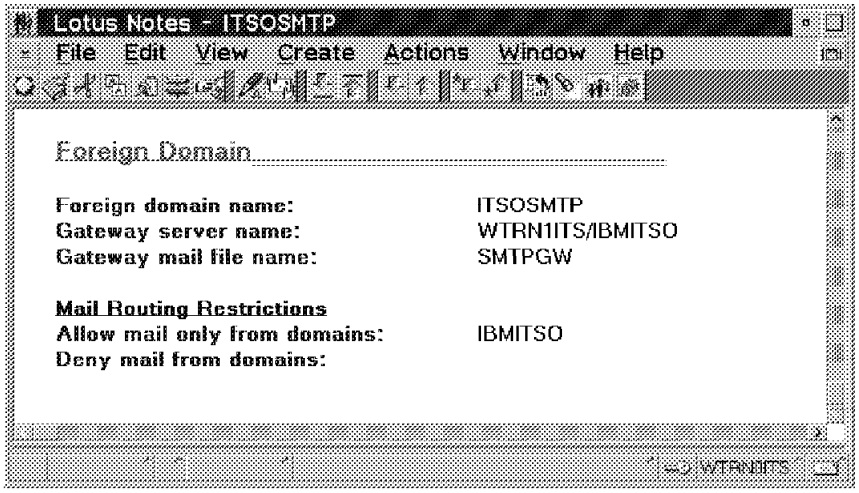


Figure 75. Creating a Foreign Domain

5. Create a foreign domain mail file database by choosing **File** - **New Database**. The file name must be the same name you gave to the gateway mail file when you created the foreign domain. In the Template box, select **Mail Router Mailbox**.

   This mail file stores messages that Notes users send to SMTP users until the SMTP Gateway picks them up and forwards them to SMTP.

6. Use the configuration forms to enter configuration options that fit your system requirements. First you have to complete the setup form. The following shows a sample setup form:

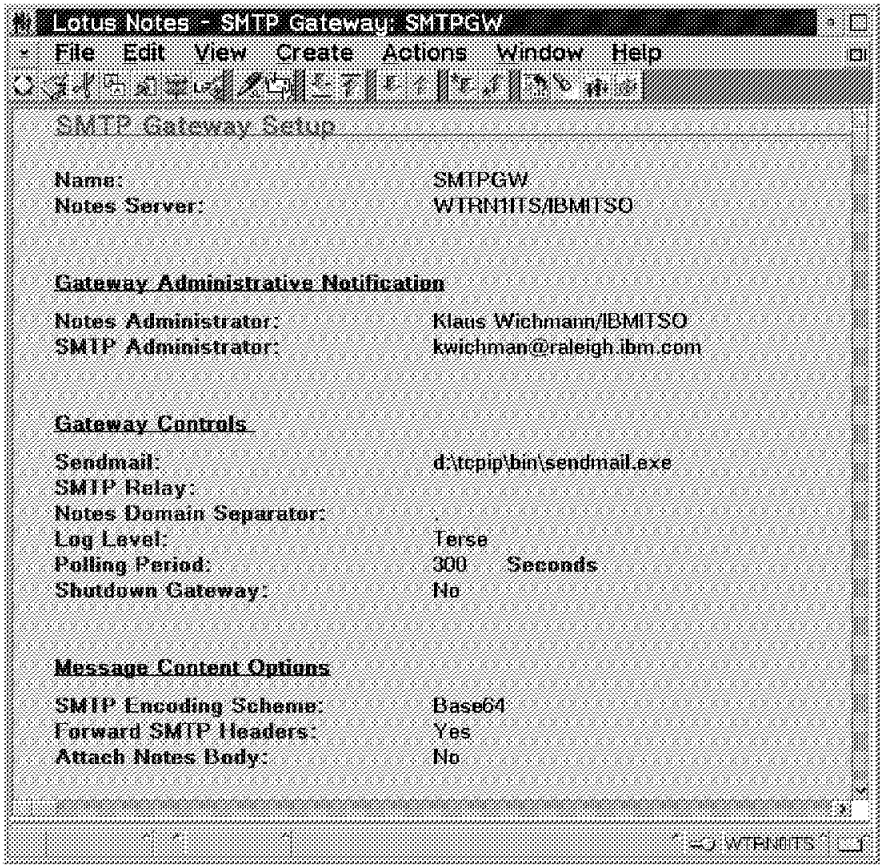


Figure 76. Completing a Setup Form

After completing that form, you create a SMTP connection document. The following figure shows an example:

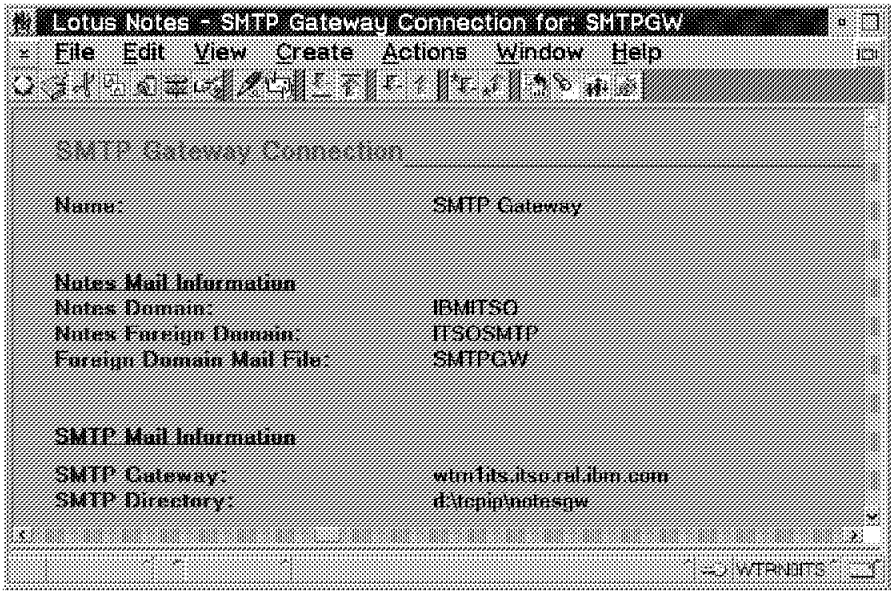


Figure 77. Creating a Connection Document

7. Start the SMTP Gateway by typing `load SMTPGW` at the Notes server prompt. From now on, restarting Notes restarts the SMTP Gateway.

6.5.4 Sending Mail From Lotus Notes to UltiMail Lite

As UltiMail Lite is a typical mail application on the Internet, it is interesting to see how Notes mail is sent to an UltiMail Lite system on the Internet. If you want to address a user on the Internet, use the following address format:

```
TO:  full_internet_address @ smtp_gateway_domain
```

Where full_internet_address is the user's ID with the host and domain name. For example kwichman @ mailhost.itso.ral.ibm.com would be a full Internet address. Here kwichman is the user ID, mailhost is the hostname of the host, where the user ID is known and itso.ral.ibm.com is the domain. The smtp_gateway_domain is the domain name of the Lotus Notes Domain in which the SMTP gateway resides. An example would be simply ITSOSMTP. The full address is then: kwichman @ mailhost.itso.ral.ibm.com @ ITSOSMTP.

The following figure shows an example of Notes mail sent to an Internet user. This example shows it is even possible to include file attachments in the note. As both systems support the MIME standard, binary information can be exchanged.

In this example, the config.sys file (which is an ASCII file) is attached to the Lotus Notes mail.

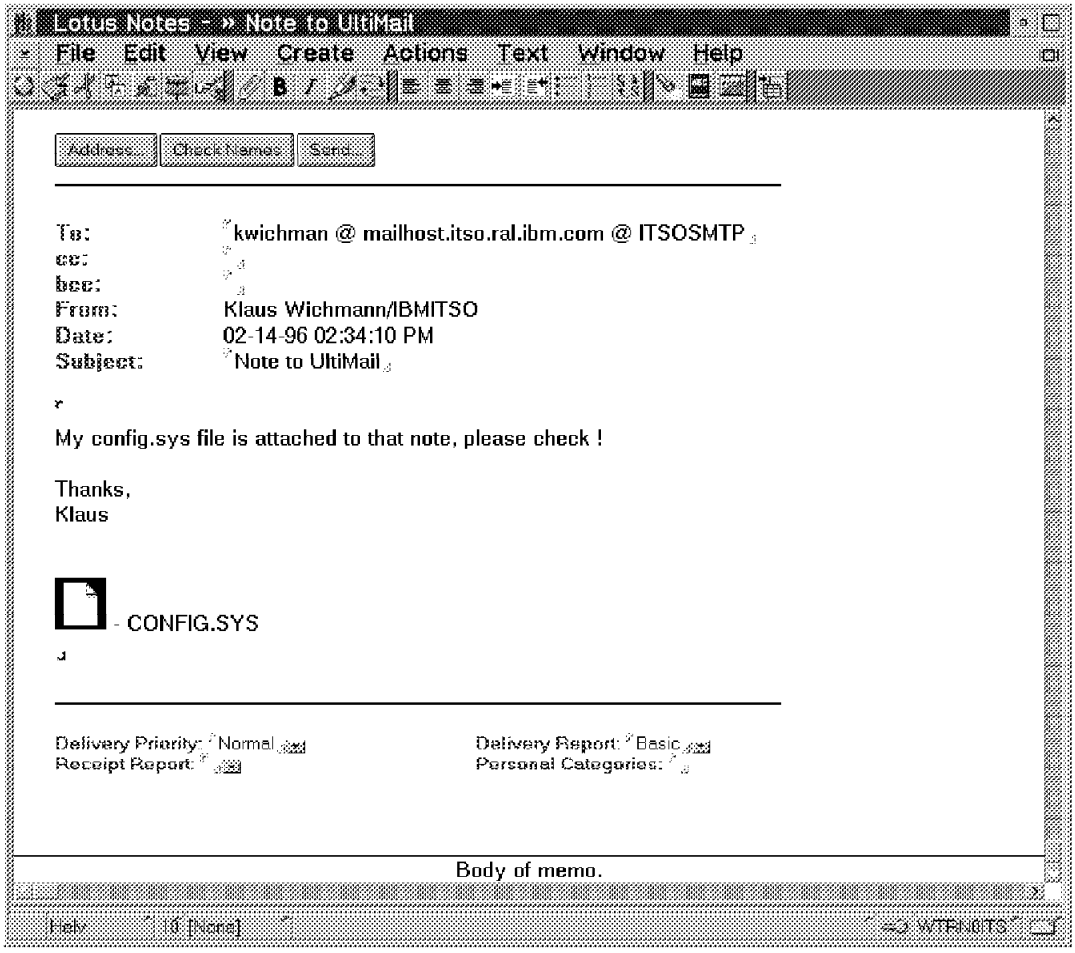


Figure 78. Sending Mail from Notes to UltiMail Lite

Once you send this note, the Lotus Notes mail router will route the mail to the SMTP domain (ITSOSMTP), which is the SMTP gateway. The SMTP gateway will convert the Lotus Notes mail and the attachment to SMTP mail. This SMTP mail is then delivered by SendMail over the Internet. On the other side, the mail is received by SendMail running on mailhost in the itso.ral.ibm.com domain. Once the UltiMail Lite Client connects to mailhost, it receives the mail using the Post Office Protocol (POP). The following figure shows what UltiMail Lite received:

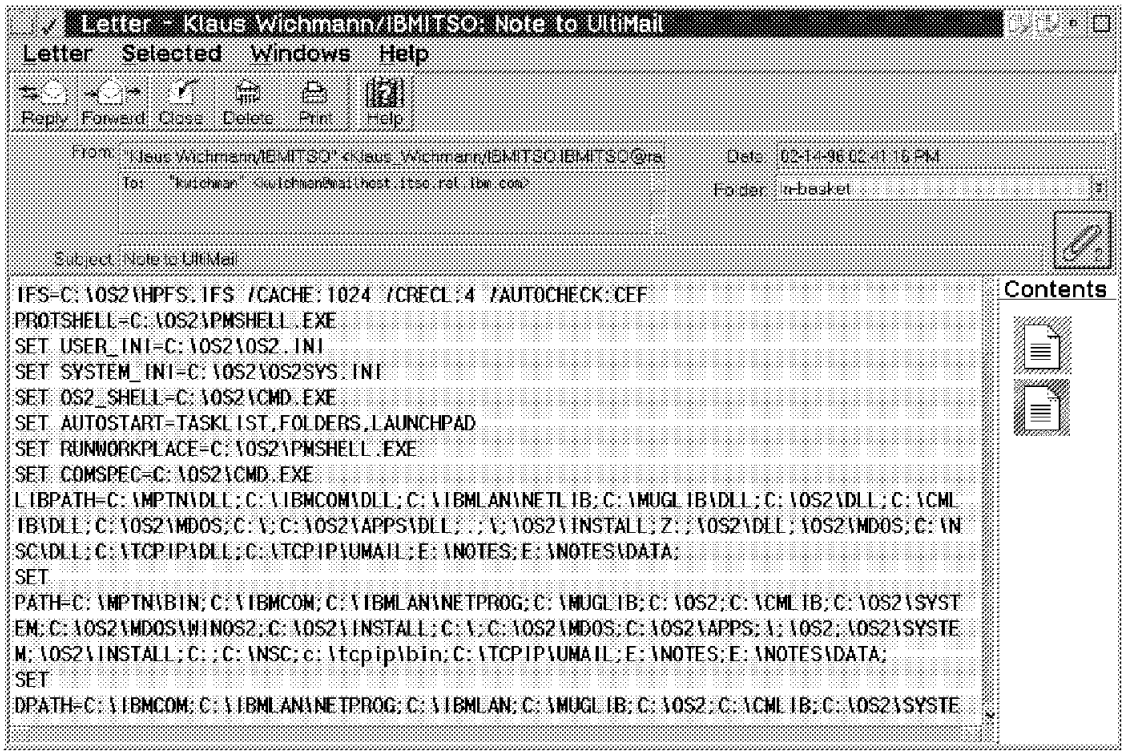


Figure 79. Mail Received By UltiMail Lite

As the above figure shows, UltiMail Lite received the Lotus Notes mail. The contents of the mail shows two documents. The first document is the text we wrote in the Notes mail and the second is the attached config.sys. If you click on the second document, the config.sys file is displayed in UltiMail Lite, since the config.sys is an ASCII file.

In a second example we tried to send a binary file attached to a Lotus Notes mail. In this example we attached an OS/2 bit map. The following figure shows how binary files are received by UltiMail Lite.

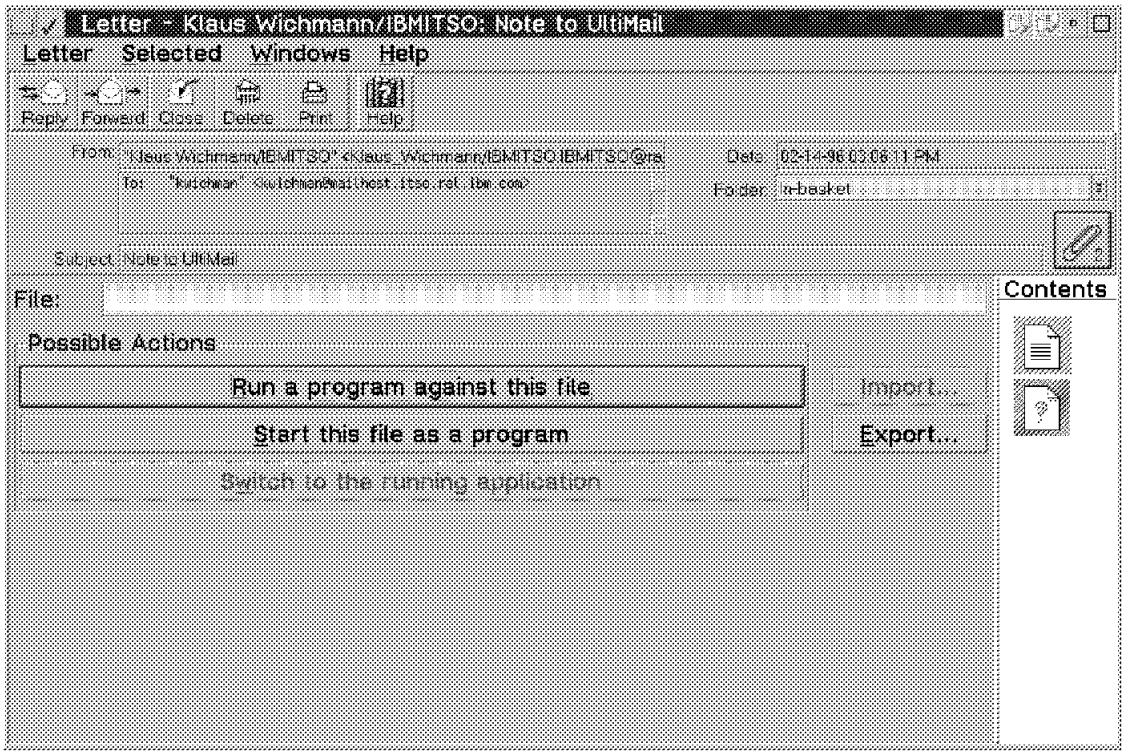


Figure 80. Receiving Binary Files with UltiMail Lite

If you click on the binary file, which is the document in the contents area marked by a question mark, you will get the dialog shown in the above figure. You can either execute the received file or save it to disk.

6.5.5 Sending Mail from UltiMail Lite to Lotus Notes

To send mail from the UltiMail Lite system on the Internet to a Lotus Notes system with a SMTP gateway, the address format is the following:

```
TO:  notes_full_name . notes_domain @ hostname . domainname
```

Where notes_full_name is the Lotus Notes full name assigned to the Lotus Notes user. An example would be Klaus_Wichmann/IBMITSO. All spaces must be replaced by an underscore. The notes domain is the domain in which the Lotus Notes user resides. For example, in the IBMITSO Notes domain, the hostname is the TCP/IP hostname of the SMTP mail gateway (for example, wtrn1its) and the domainname is the Internet domain of the SMTP mail gateway (for example itso.ral.ibm.com). A complete address of a Lotus Notes user would then be:

```
Klaus_Wichmann/IBMITSO.IBMITSO@wtrn1its.itso.ral.ibm.com
```

As in the following section, where mail was sent from Lotus Notes to UltiMail Lite, binary data can be attached to an UltiMail Lite note sent to a Lotus Notes user. The following figure shows a note addressed to a Lotus Notes user. To that note a gif image is attached.

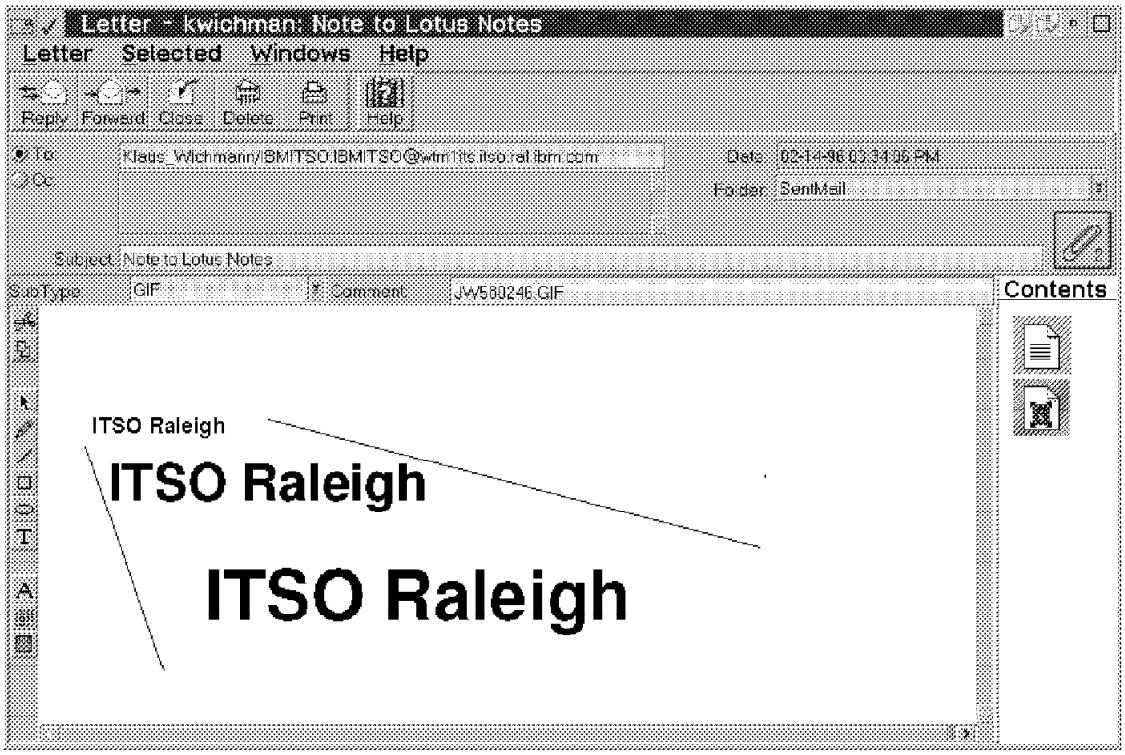


Figure 81. Sending Mail from UltiMail Lite to Lotus Notes

When you send the UltiMail Lite note, the mail is sent by SendMail to the Lotus Notes SMTP mail gateway (here: wtrn1its.itso.ral.ibm.com). The SMTP mail gateway will convert the mail to the Lotus Notes format and deliver it to the Lotus Notes mail router, which stores the mail in the addressed mail database. When the Lotus Notes user opens the mail database he or she will see the new mail. In this example, the Lotus Notes mail looks like the following:

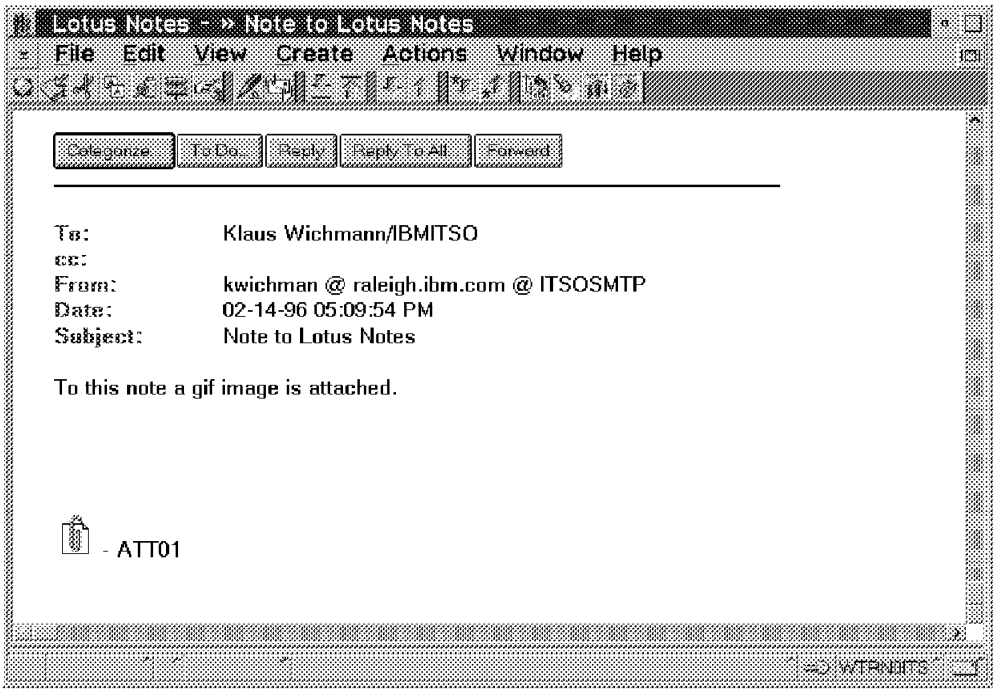


Figure 82. Receiving Binary Files from UltiMail Lite

Under the text that was entered by the UltiMail Lite user, you can see the attached gif image. By double-clicking on the paper clip symbol you can either detach or launch the file.

If you check the From field, you notice that the reply address is correct and all you have to do to reply to the UltiMail Lite user is to click on the Reply button on the top of your document.

Chapter 7. Internet Applications

This chapter covers typical applications used with the Internet. It explains how to set up these applications and how to use them. Some scenarios provide a deeper understanding of these Internet applications. The following applications are discussed:

- NewsReader/2
- Gopher
- WebExplorer
- Lotus Notes R4 (InterNotes)
- Network Dialer
- Netcomber

Basically there are two ways to connect to the Internet. You can be connected by having a constant connection to a network that is connected to the Internet, or you can be connected via a modem to a service provider. Both types of connections are covered in this chapter.

IBM′s new Internet product Netcomber, which is a kit containing all useful Internet applications, is discussed together with the access through an Internet provider. Netcomber is designed for dial-in Internet access.

Lotus Notes R4 is now supporting World Wide Web access. It implements a new strategy on how to access information on the Internet. Basically it converts HTML documents to Lotus Notes documents. These documents are stored in Lotus Notes databases and can be accessed by Lotus Notes users.

The IBM WebExplorer is the classic way to access Web pages on the Internet. The newest WebExplorer supports features such as Internet access over Socks servers and HTML 2.0.

Gopher is another application to surf the Internet in an easy way.

NewsReader/2 lets you subscribe to newsgroups on the Internet. On the Internet there are thousands of newsgroups and new information is added every day. NewsReader/2 helps you to handle this information.

7.1 NewReader/2

NewsReader/2 is an OS/2 application that is included in the TCP/IP package since TCP/IP 2.0. It can be used to access news servers on the Internet. Most news servers contain a broad range of newsgroups. Their content varies from technical computing subjects to recreational and cultural articles. With NewsReader/2 you can do the following:

- Subscribe to news groups
- Read new articles
- Post your own articles

7.1.1 Configuring NewsReader/2

There are many news servers available on the Internet. We recommend that you consult with your network administrator for the name or IP address of your news server. These news servers are commonly referred to by a name. This normally requires that you have access to a named server. Please see the previous section for information to connect to a named server.

We recommend that you try to PING the address of your news server before proceeding. In our examples, we are already connected to a named server, so we issued the following PING to ensure network access:

```
[C:tcpipetc]ping rtpnews.raleigh.ibm.com 10 1
PING rtpnews.raleigh.ibm.com: 56 data bytes
64 bytes from 9.67.10.155: icmp_seq=0. time=31. ms
64 bytes from 9.67.10.155: icmp_seq=1. time=0. ms
64 bytes from 9.67.10.155: icmp_seq=2. time=0. ms

----rtpnews.raleigh.ibm.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/10/31
```

We received a response from the address, and we can now confidently configure our NewsReader/2 application to use this address as our news server.

You can configure the address of the news server in the configuration notebook provided with TCP/IP for OS/2. Listed below are our recommended steps:

1. Open the **TCP/IP** folder view on your desktop:

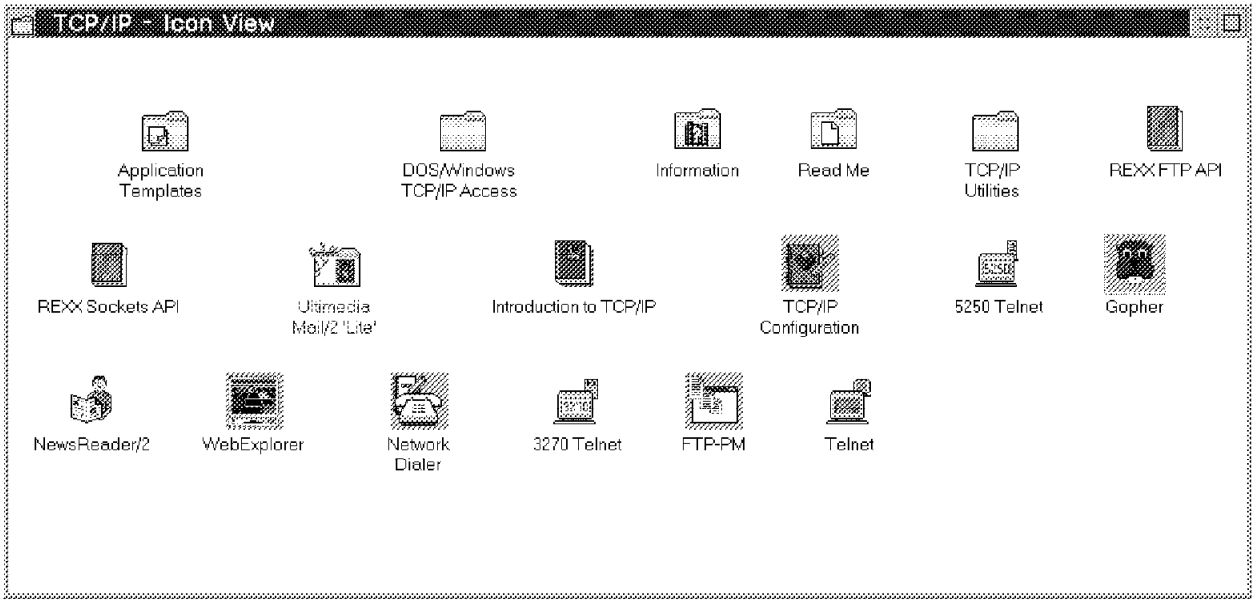


Figure 83. TCP/IP Folder

2. Open the configuration notebook, and then do the following:
   a. Click on **Servers**.
   b. Type in the name of your news server. Your panel should look like the following:

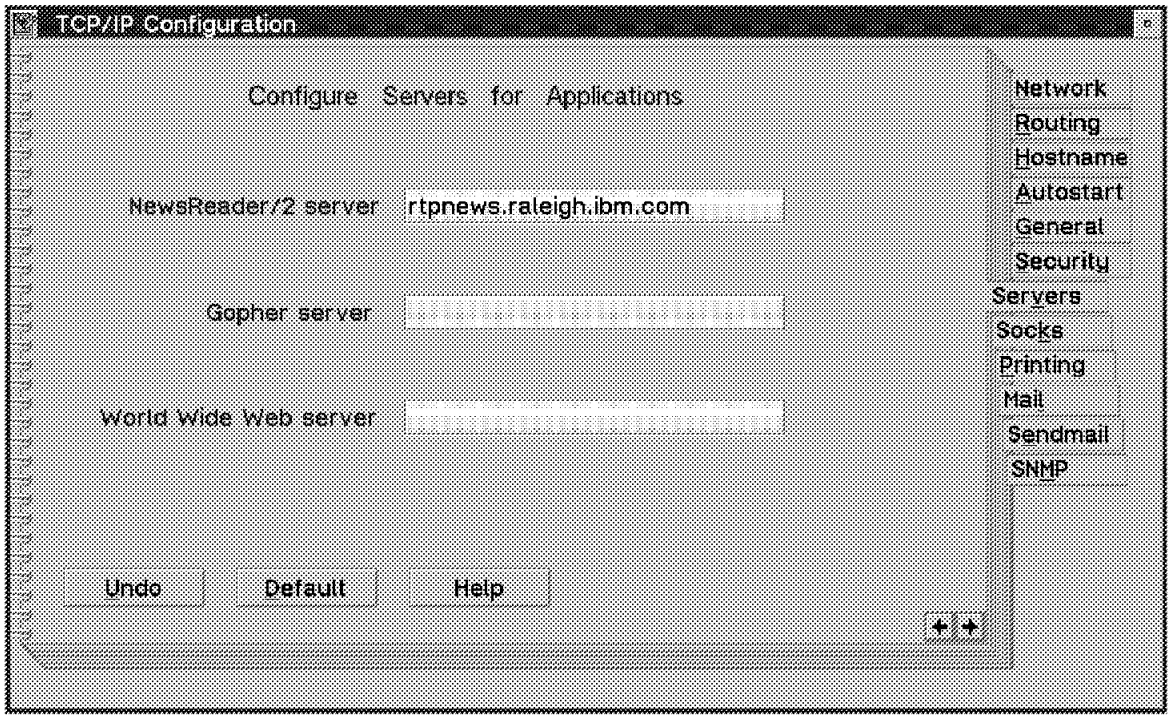


Figure 84. Configuration Notebook

c. Close the notebook, save any changes, and agree to changes to your CONFIG.SYS file. These steps will modify your CONFIG.SYS file to include an environment variable for the news server. This change takes effect when you reboot the machine.

You should now be able to go ahead and start NewsReader/2 without rebooting.

To provide information about the news server to workstations on a LAN, you can create an NR2.CTL LAN control file that is used to specify the name or IP address of the news server that NewsReader/2 should use. The file must be placed somewhere on the LAN so that it can be found in the DPATH of each workstation. The file should consist of a single line as follows:

```
NEWSSERVER=<server>
```

7.1.2 Starting NewsReader/2

The following are the two ways of starting NewsReader/2:

1. Enter the nr2 command from an OS/2 command prompt.
2. Click on the NewsReader/2 object in your TCP/IP folder on your OS/2 Workplace Shell desktop.

7.1.2.1 Starting from a Command Prompt

There are several ways to start NewsReader/2 from the command line. Usually you start NewsReader/2 with the server name to which you want to connect as follows:

```
nr2 rtpnews.raleigh.ibm.com
```

You can also specify the following parameters:

/nc Starts NewsReader/2 without connecting to the news server. You can use this option to import a NEWSRC file from a UNIX system. For example, you could import a news file (NEWS.GRP) from a UNIX-format NEWSRC file in your ETC subdirectory.

You can also use this option if you want to learn how to use NewsReader/2 but do not have a news server.

/u Starts NewsReader/2 in update mode. In update mode, NewsReader/2 accesses the news server and obtains an updated set of all newsgroups for later display in your All Groups window.

The NR2BATCH.ERR file logs any errors encountered while running the NewsReader/2 in the batch process NEWS.ALL building mode.

/s Causes NewsReader/2 to prompt you to enter the host name or IP address of a news server to access.

The first time you start NewsReader/2 on your system, it will establish its directory structure in your local ETC directory (this is where most of the related files are kept). The following is a list of the directories that are created and their contents:

Directory	Contents
ETCRNSPOOL	All temporary (.TMP) and deferred posting (.SAV) files are kept in this directory. Files for posting, mailing, and replying are POSTx.TMP, MAILy.TMP, and REPLYz.TMP, respectively. Numbers x, y, and z start at 1 for each session and increment as you perform each task. File ARTICLE.TXT is also created whenever you manipulate an article. All *.TMP files in this directory, as well as ARTICLE.TXT, are deleted at the start of each session.
ETCKILL	All killfiles are kept in this directory. The killfile for all groups is easily recognizable as ALLGROUP.KIL; however, the killfiles for each individual newsgroup are named using a numeric encryption.
ETCSIG	The directory holds the signature files available for use by NewsReader/2. Only files ending with .SIG are recognized for use by the Signature option from the NewReaders Option menu.
ETCLOG	This is where the NR2POST.LOG and NR2MAIL.LOG files are kept (if these log file options are being used). This directory can be used to hold any other log files that you wish to create by selecting **File** and then **Save As...** in the Article window.

7.1.2.2 Starting from the Workplace Shell

Double-click on the **NewsReader/2** icon in the TCP/IP folder.

7.1.3 Connecting to a News Server

1. Once you have started the application, the NewsReader/2 - Product Information is displayed on a panel.
2. NewsReader/2 now attempts to connect to your news server, and you will notice the following message in the bottom of the NewsReader/2 window:

   ```
   Connecting to rtpnews.raleigh.ibm.com
   ```

   This connection may take several minutes.
3. On the first occasion that you log on, you will not have a list of newsgroups available on your news server. NewsReader/2 prompts you if you want to download a list from the news server.

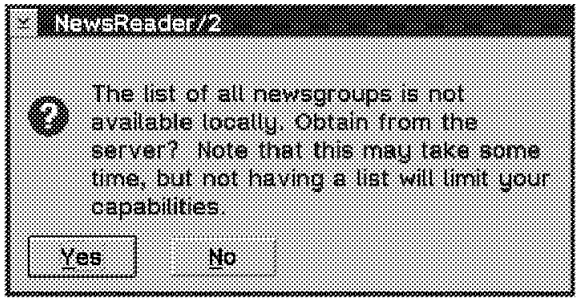


Figure 85. Downloading the Newsgroup List

If you click on **Yes**, NewsReader/2 downloads a list of newsgroups and creates the following files in the mptn\etc directory. If your connection to the Internet is via a service provider, these files are stored in the mptn\etc\rnspool\name directory, where name is the name of the service provider.

NEWS.GRP This file keeps track of the subscriptions and articles you have viewed. Each line in the file contains a newsgroup name followed by a list of read article number ranges.

NEWS.SAV This file is a backup of your NEWS.GRP file from the last session. If for some reason your NEWS.GRP file becomes lost or garbled, you can retrieve the lost information from this file. NewsReader/2 automatically uses the NEWS.SAV file if it is unable to find and/or use the NEWS.GRP file.

NEWS.ALL This binary file contains all newsgroups that are known to your news server, and whether or not you can post to each. This file is always required for adding newsgroups, and is required by default for making posts.

NEWS.TIM This file contains the time stamp of the last time you started NewsReader/2. The time stamp is used when the news server is queried for new groups. The format of the time stamp, in GMT, is as follows:

```
YYMMDD HHMMSS
```

NR2.INI This file is used to store all of the application's information from session to session.

Your NewsReader/2 panel then displays a status indicating that it is building NEWS.ALL, and eventually your NewsReader/2 service will start.

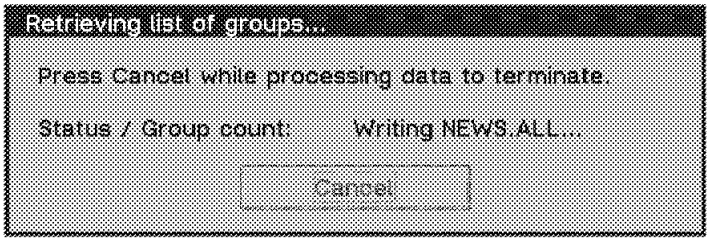


Figure 86. Building NEWS.ALL Pop-Up Window

7.1.4 Using NewsReader/2

Your NewsReader/2 application consists of several windows. These windows contain the following:

NewsReader/2	A list of the newsgroups to which you subscribed on this server
All Groups	A list of all the newsgroups available on this server
Article list	A list of the headers of articles on the currently selected newsgroup

On the first occasion that you use a news server you will need to browse the list of newsgroups in the All Groups window. You should select the groups that interest you by double-clicking on them. You can also select more than one newsgroup by clicking once on each newsgroup, then clicking on **Actions** and **Add Group(s)** from the action bar of the All Groups window. Your panel should look as follows:

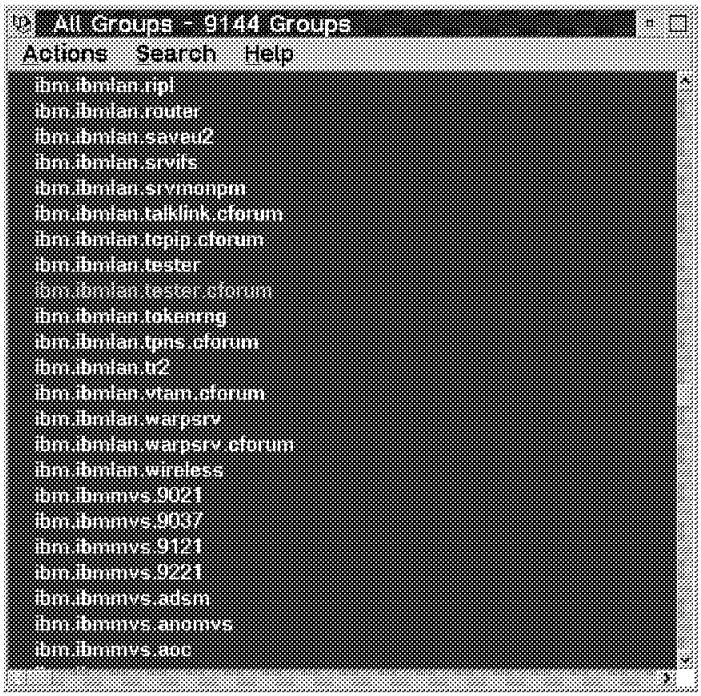


Figure 87. All Groups Window

Each of the news services is added to your NewsReader/2 panel, which will look something like the following figure. For each article in this list, the number of articles the newsgroup contains is shown.

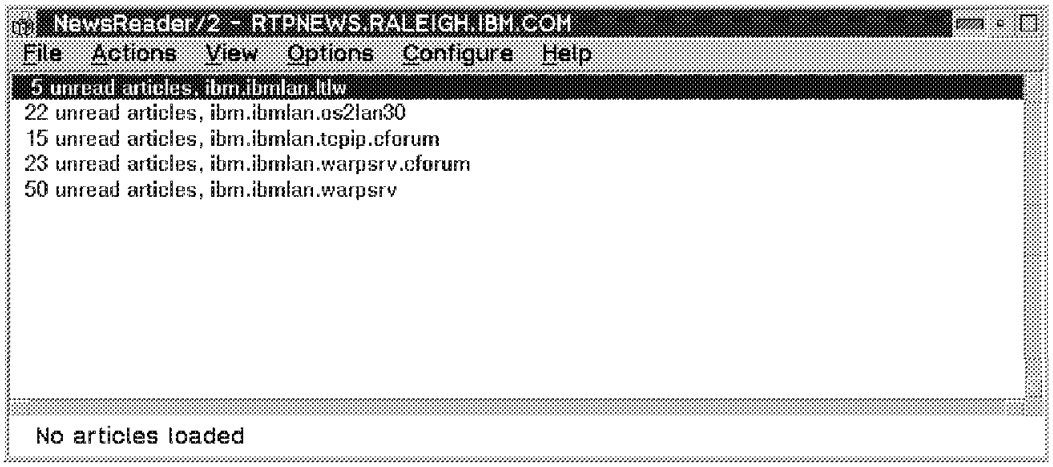


Figure 88. Subscribed Articles

When you double-click on one of the newsgroups, a list of headers showing the subject of each article is displayed in the following Article List panel:

Figure 89. Article List Window

To read an article you have to double-click on the specific article. An editor is opened to show the contents of the article.

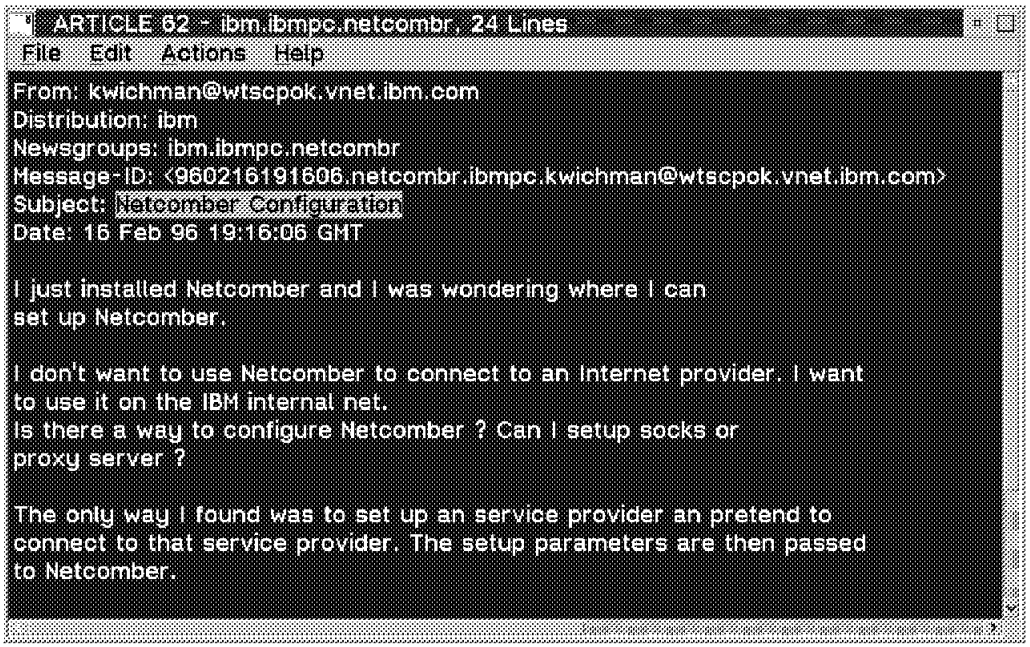


Figure 90. Reading an Article

You can read the article and then, using the various action bar alternatives, perform any of the following:

- Copy to the clipboard
- Print
- Save to a file
- Reply via mail
- Post reply
- Forward article

These are some basic steps to subscribe to newsgroups and to read articles in the newsgroups. Every time you exit your NewsReader/2 application and re-start it, the newsgroups to which you have subscribed are updated. You also get a message if new newsgroups exist at your news server. You can then download a new list of the current newsgroups available.

The next section will give you more information on how to reply to articles.

7.1.4.1 Posting

You can also contribute to some newsgroups by posting to the newsgroup. We recommend that you customize your headers for postings by clicking on **Configure** and **Posting**, which brings up the panel shown in Figure 91 on page 169.

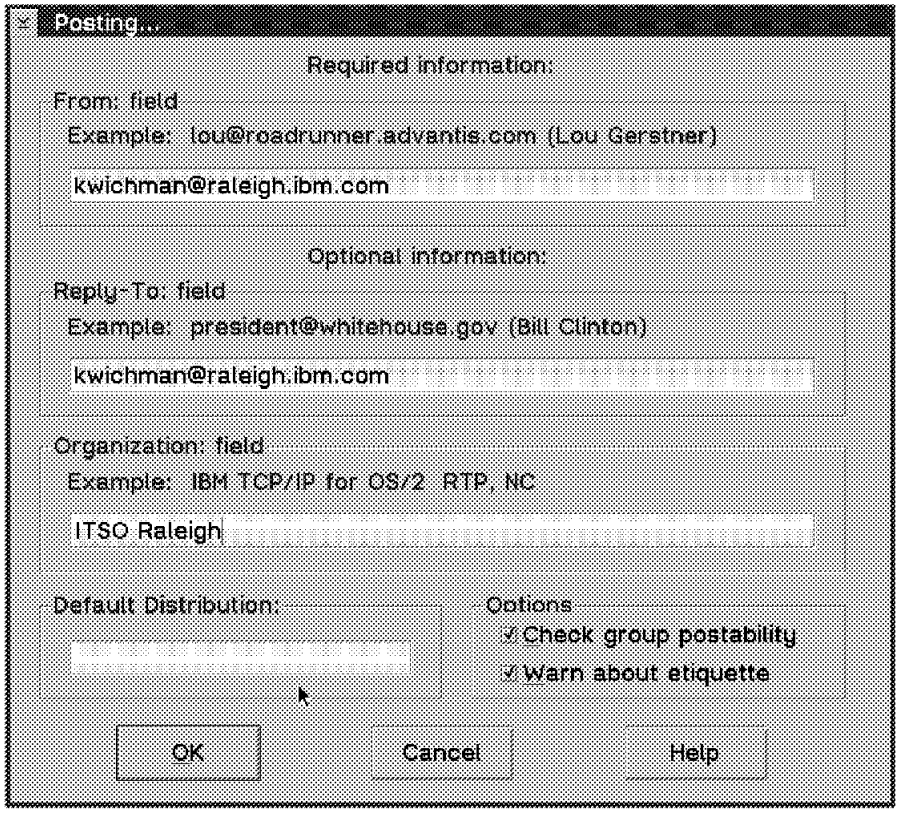


Figure 91. Configure Posting

We recommend that you highlight the Check group postability checkbox. NewsReader/2 then checks to see if you can post to a newsgroup before you type information for posting. This option is checked by default. The following shows an example message you may receive when you try to post to a newsgroup:

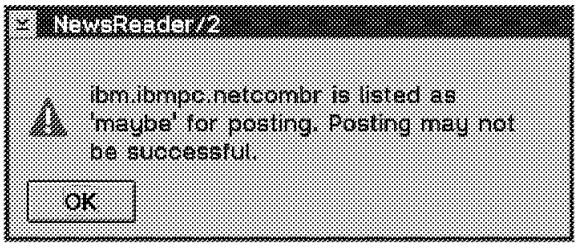


Figure 92. Post Message

Highlight the Warn about etiquette checkbox if you want the etiquette reminder window displayed each time you begin your posting activities. The reminder window reminds you that your postings should conform with the etiquette guidelines posted in news.announce.newusers. We recommend you read these guidelines once and do not highlight this option.

In order to post to a newsgroup, you should click on **Actions** and **Post** from the NewsReader/2 window. You will then add a new article to the selected newsgroup. It is also possible to reply to an article. Select or open the specific article and select **Post...** from the Articles List Action menu or **Post reply...** from the NewsReader/2 Editor Action menu.

If you reply to an article, you will get a panel like the following. If you post a new article, the panel looks almost the same. The original article is marked with a greater then sign on the left side.

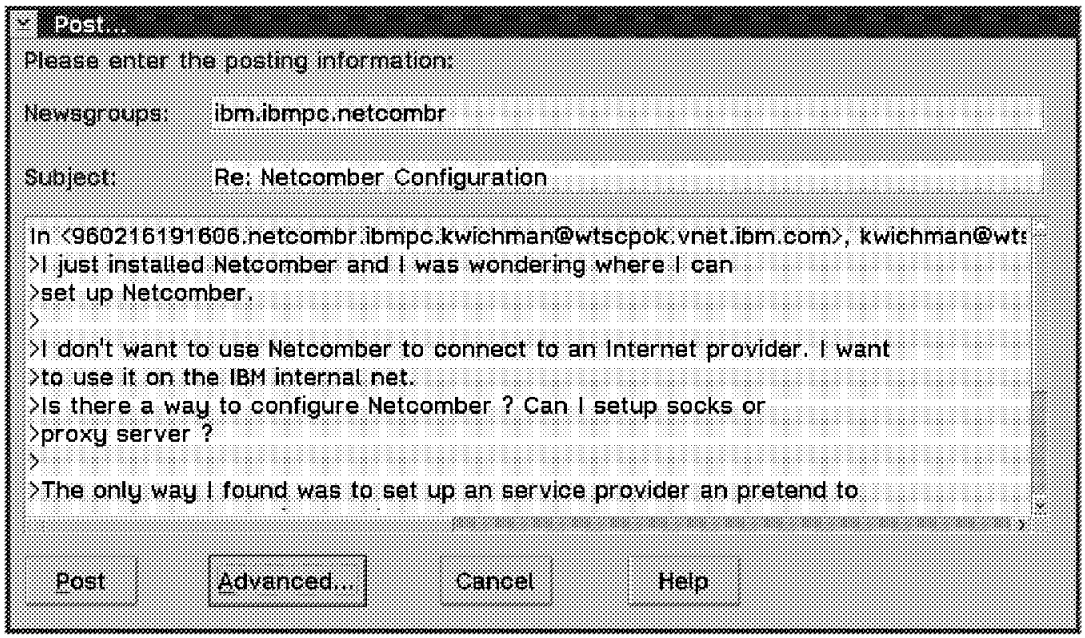


Figure 93. Post Reply

Type in your information and click on **Post** to post your article or reply to the newsgroup.

Selecting **Advanced** gives you the possibility to enter more header information than you have already configured for posting. The following figure shows the advanced posting options:

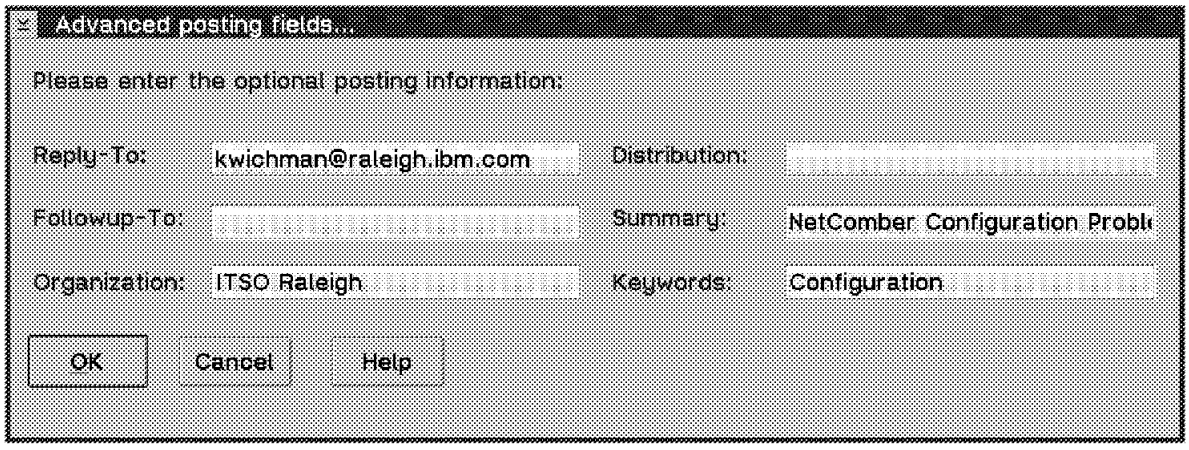


Figure 94. Advanced Posting Options

The fields contain the following information:

Distribution Enter an alternate distribution for your post in this field. The default distribution is displayed if previously configured.

Summary Enter a short summary for your post into this field.

Keywords Enter the keywords for your post into this field.

Reply-To Enter a Reply-To E-mail address if you want to override your default posting information.

Followup-To Enter the name of the newsgroup(s) that you want respondents to post to when they reply to your post.

Organization Enter the name of the newsgroup(s) that you want respondents to post to when they reply to your post.

Enter an organization name in the Organization field if you want to specify a different organization than the default posting information.

7.1.4.2 Other Features

There are many other fine features of the NewsReader/2 service such as the following:

Unsubscribe	Remove all subsequent posts with this header from your header list.
Block Author	Remove all subsequent posts submitted by an author from your header list.
Sort	Sort the header list by author, subject or number.
Search	Search a news group for a particular string.
Refresh	Refresh the subscriptions now.
Refreshing	Set the frequency to refresh the subscriptions.
Signature	Signature to be attached to the end of a post.
Export	Export newsgroup files compatible with UNIX.
Import	Import newsgroup files compatible with UNIX.

7.1.5 Changing News Servers

Each news server has its own idea of what the article numbers are. So, if you switch from one server to another, NewsReader/2 will get confused as to what articles have been seen and what have not.

You have to either erase the NEWS.GRP file in the local ETC directory and start again, or edit it and remove the article number information, leaving only the newsgroup names.

7.2 Gopher

Gopher is an easy-to-use and easy-to-configure application to surf the Internet with point and click. Unlike the World Wide Web, it does not support a graphical representation but represents information as icons that can be clicked on to view the information or receive it to a file.

The information that you can access is maintained in subdirectories on Gopher servers just as you might maintain files in a directory on your computer. When you open a Gopher menu item, the resulting window shows you the information in the associated directory. And, just as you might have directories on your computer that contain subdirectories which contain files, a Gopher menu might

contain items that represent other Gopher menus that contain items that represent information files.

The information contained in these files can be in various formats. For example, the information might be a text (ASCII) file, a graphic, or a binary file such as a computer program. Or, it could be a tool such as a search facility to locate the definition of a word or a communications tool used to access another computer on the Internet.

7.2.1 Configuration

To configure Gopher, you have to open the **TCP/IP Configuration** folder from the TCP/IP icon view. On the server page of the configuration folder you enter the name of the Gopher server that you want to access. Your configuration should look somewhat like the following:

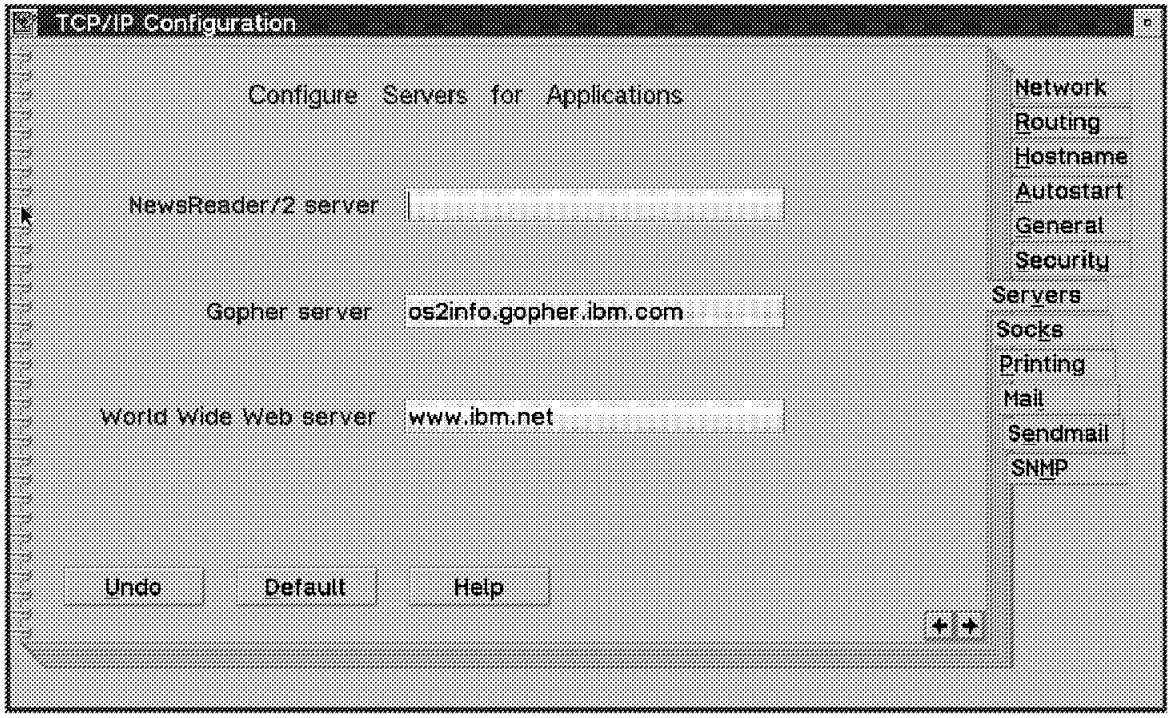


Figure 95. Gopher Configuration

Once you have completed your configuration, close the configuration folder and save the changes.

7.2.2 Starting and Customizing Gopher

To start gopher you either double-click on the **Gopher** icon in the TCP/IP icon view or enter the command `gopher` at the command prompt.

Gopher will then connect to the specified Gopher server. Your Gopher window should look somewhat like the following:

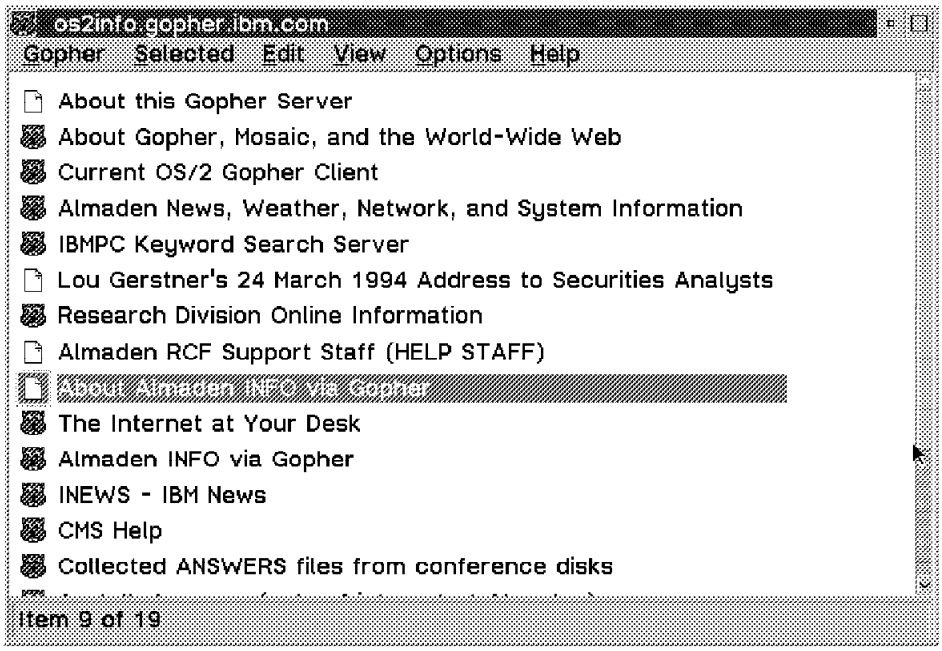


Figure 96. Gopher

When Gopher is started, the first thing you should do is to customize Gopher. Open the configuration notebook from the Options menu. The configuration notebook lets you define applications that should be started to display certain types of information. The following figure shows an example of the definition for ASCII documents. In this example, documents will be displayed by the internal Gopher viewer.

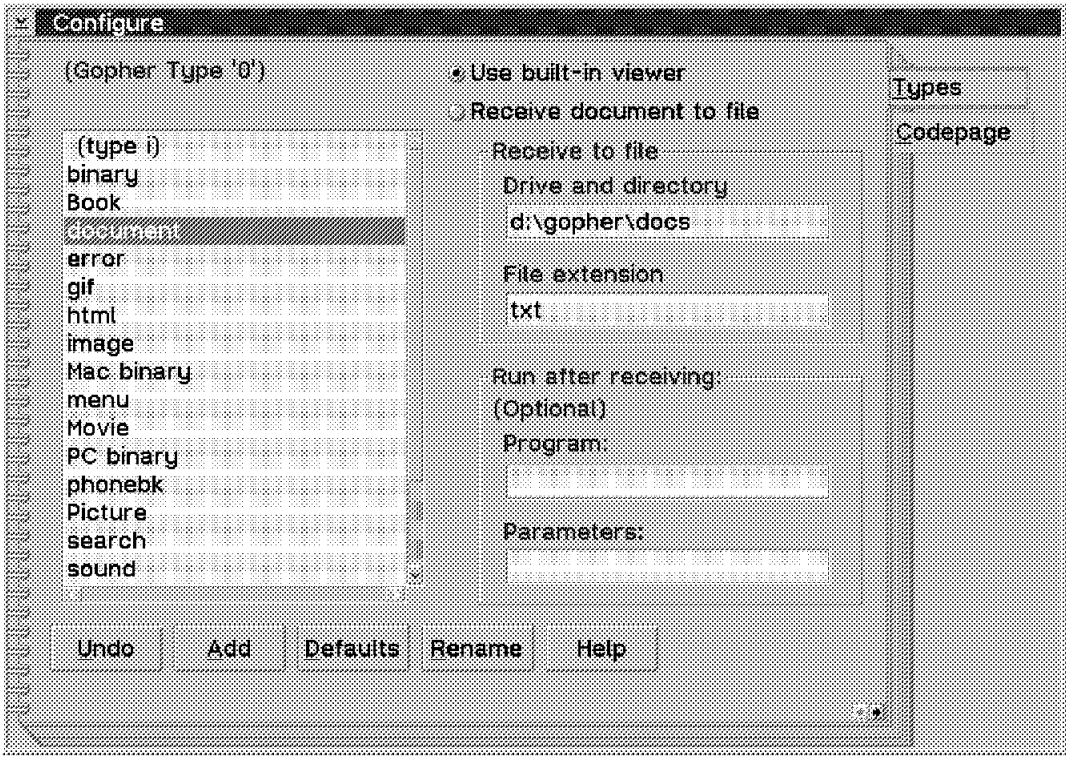


Figure 97. Gopher Customization

It is also possible to display documents using viewers other than the internal viewer. You then have to check the Receive document to file and enter the name of the program you want to run against this document in the Program field (for example EPM).

7.2.3 Using Gopher

Once you have started Gopher you will see a list of icons with an explanation to the right. Gopher knows a couple of different icons for different types of information. The following list shows you the most common icons:

Icon	Type of Information
	Menu
	ASCII document
	Binary document
	Image
	Telnet
	Hypertext
	Sound

When you double-click on an icon, either its contents are shown, or a panel to save the information to a disk appears.

If the item is a document, graphic (GIF*), hypertext document (HTML), Book file, movie, sound file, picture, or image file, Gopher transfers the file to the default directory and opens the file using the default viewer (as specified in the Configure window).

To transfer a file without running the default viewer or program against the file, do the following:

1. Highlight the item.
2. Select **Get** from the Selected pull-down menu.

For many Gopher items, when you transfer the item, the Save File As window is displayed.

Specify the drive and directory to which you want to receive the file and select the **Save** push button.

If you double-click on the **Menu** icon you will switch to the next menu. That menu will again display several icons that you can choose.

7.2.3.1 Bookmark Items

To maintain a list of useful Gopher menus or other Gopher items, you can bookmark items and menus. To bookmark a menu, select **Bookmark this menu** from the Gopher menu. To bookmark a Gopher item, select the item and select **Bookmark this item** from the Selected menu. All bookmarked items and menus will be displayed in the Bookmark List. To see this list, choose **Open bookmark window** from the Gopher menu. The following window is displayed:

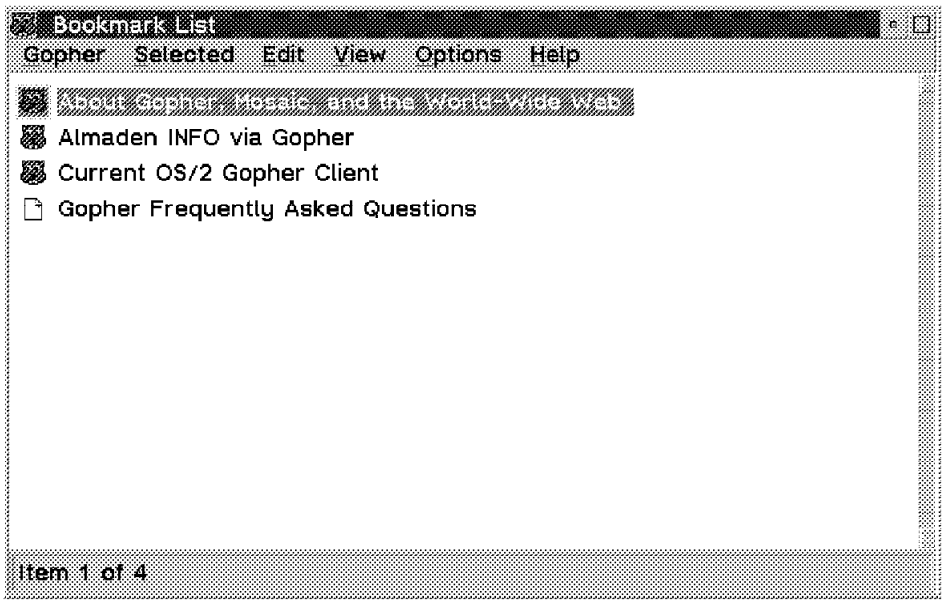


Figure 98. Gopher Bookmarks

The Bookmark List looks like a common Gopher menu. The only difference is that it is customized by you. It contains all the information that is important to you and gives you easy access to that information.

One special bookmark is the home bookmark. The home bookmark marks the menu that is loaded when you start your Gopher application. You can mark the current menu as your home bookmark by selecting **Make this menu the home bookmark** from your Gopher menu.

7.3 The WebExplorer

The World Wide Web, or WWW, is the newest information service to arrive on the Internet. The WWW is based on a technology called hypertext. Hypertext is a method of presenting information where selected words in text have links to other information. WWW is an attempt to organize all the information on the Internet into hypertext documents to form a super database.

WebExplorer is a World Wide Web browser. Compared to Gopher the WebExplorer is different in a number of important ways. First, the WWW is based on hypertext links and is structured around whatever the author feels

might be relevant to his or her document. A Gopher menu is not as flexible. WWW knows the contents of files to which the hypertext refers, whereas Gopher does not know details on the contents of a file until a user selects the file. Hypertext documents can integrate multimedia in the hypertext document. Gopher servers can't integrate multimedia files with other types of data. The advantage of hypertext is that in a hypertext document, if you want more information about a particular subject, you click on the highlighted item.

The World Wide Web is based on a client/server model with the following specifications:

- URL (uniform resource locator):

  URL is a standard for specifying an object on the Internet. Web browsers can access the following URL objects types:
  - HTTP (browse through HTML documents)
  - Gopher (browse through Gopher menus)
  - FTP (download files)

    This does not work if you are accessing the Internet through a firewall that has restricted FTP file transfers.
  - NNTP (Read news files)
  - Local files
  - WAIS (connect to a WAIS server)
- HTTP (HyperText Transfer Protocol)

  HTTP is an application-level transfer protocol for transferring the URL object. It can handle security and display different image formats and multimedia formats.
- HTML (HyperText Markup Language)

  The WWW hypertext documents are stored on HTTP servers. These documents are written in a language called HTML. The HTML language is similar to the IBM script markup language. Web browsers can understand the HTML language. An example of an HTML document is included at the end of this section.

WebExplorer supports all markup in the HTML V2.0 standard, including forms, ISO-Latin-1 fonts, inline graphics, glossaries, and more. Since some documents on the Internet are not compliant with the proposed standard, WebExplorer attempts to clean up the document by introducing markup where appropriate. It also supports HTTP V1.0 which includes GET and POST access methods, image maps, and MIME headers. IBM is committed to supporting these public standards as they evolve, including HTTP V3.0 when it is standardized.

WebExplorer does not support the as-yet-to-be-defined HTML V3.0 standard as well as the inline mail function. If you encounter an HTML document which contains imbedded inline mail commands, the system reports an error.

7.3.1 Configuring the WebExplorer

Configuring the WebExplorer is as easy as setting up Gopher. To configure the WebExplorer open the **TCP/IP Configuration** notebook and select the server page. You get the following screen:

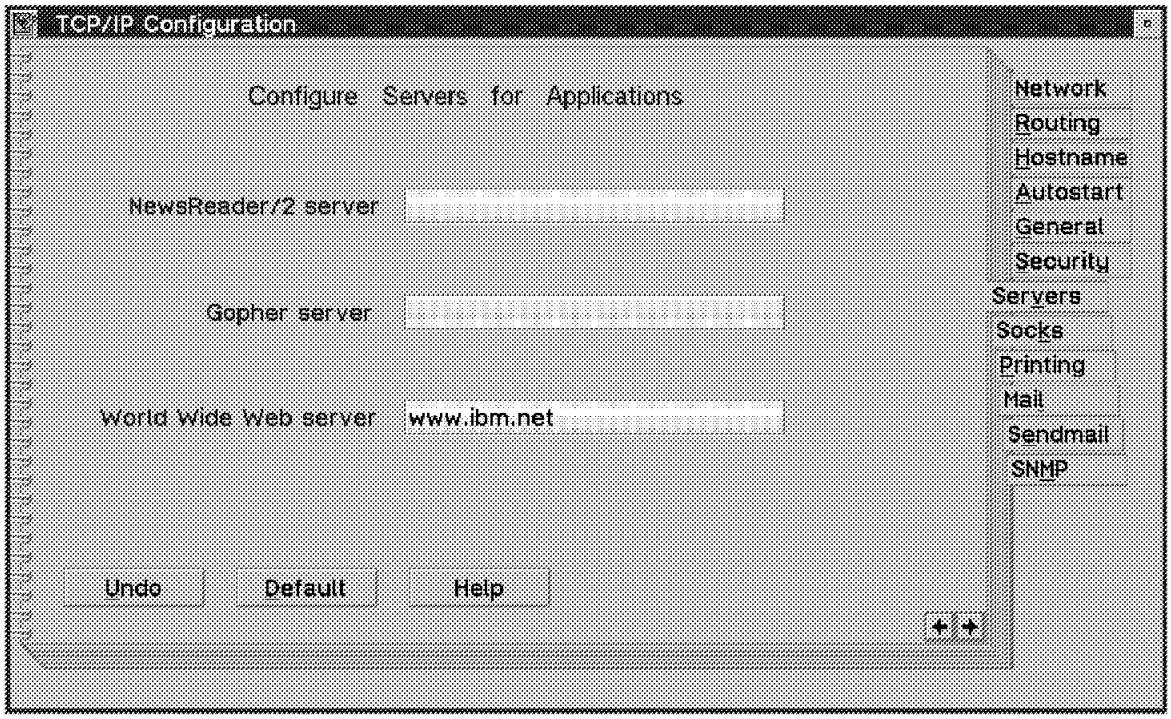


Figure 99. TCP/IP Configuration Notebook

You have to fill in your World Wide Web server which is the server your WebExplorer first tries to connect to after you have started the application.

When you have entered the name, close the configuration notebook and save the changes.

7.3.2 Starting and Customizing the WebExplorer

You can start the WebExplorer by double-clicking on the **WebExplorer** icon in the TCP/IP icon view or by entering explore at the command line.

When the server is started, it tries to connect to the World Wide Web server that you have specified in the configuration notebook of TCP/IP. Your WebExplorer should then look something like the following:

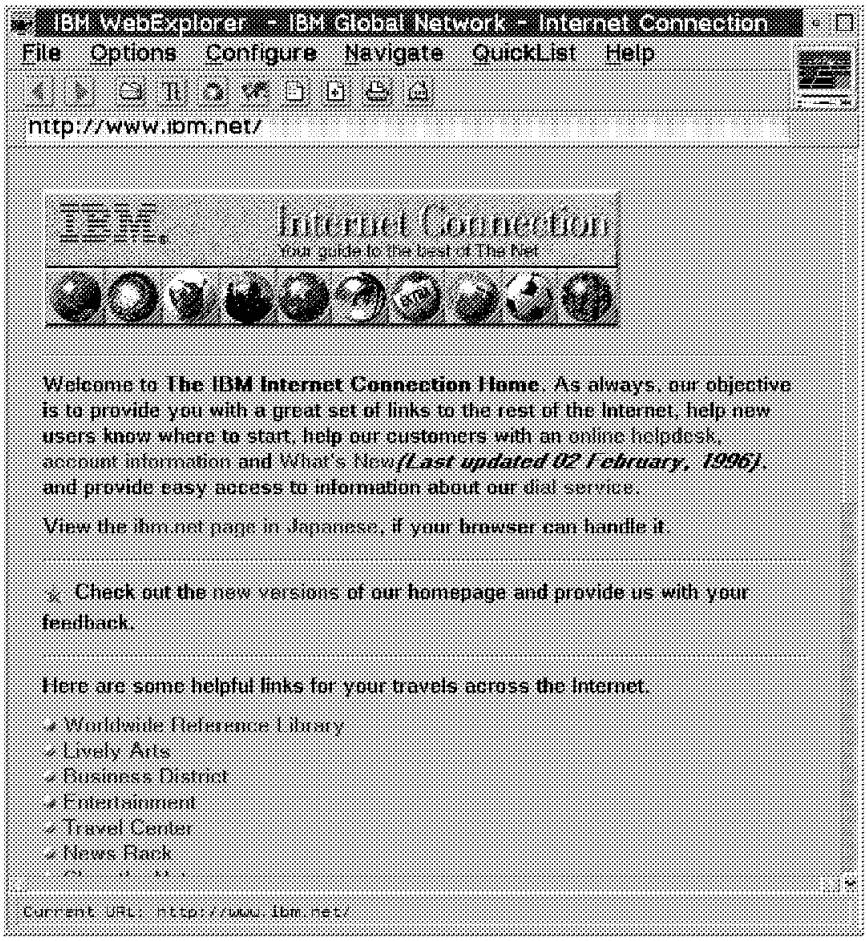


Figure 100. WebExplorer

To configure the servers that your WebExplorer will use, select **Servers** from the WebExplorer Configure pull-down menu. The following Configure Servers window is displayed:

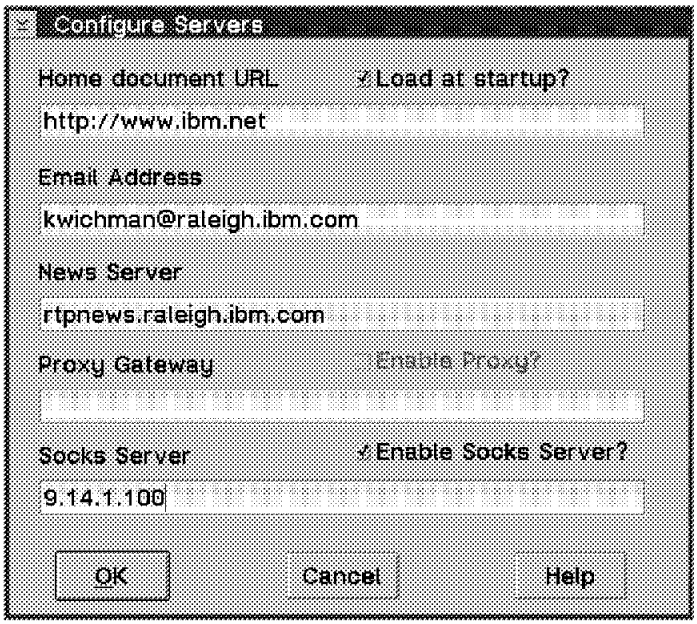


Figure 101. WebExplorer - Server Configuration

The different fields on this configuration page have the following meaning:

Home document URL Specify the Uniform Resource Locator (URL) that you want to use as your home document. If you want to use this URL as the default URL at startup, select the Load home document at startup checkbox.

Your E-mail address Specify the E-mail address that others are to use when sending electronic mail to you. An example of an E-mail address is kwichman@raleigh.ibm.com, where kwichman is the user ID and raleigh.ibm.com is the hostname of the mail server.

News server Specify the hostname or 32-bit dotted decimal notation Internet Protocol (IP) address of the default news server you want to use. An example of a news server name is rtpnews.raleigh.ibm.com.

Proxy gateway Specify the URL (http:// followed by the hostname or 32-bit dotted decimal notation IP address) of the default proxy gateway you want to use. Contact your system administrator to ensure that you are authorized to use the proxy services.

Socks server Specify the hostname or 32-bit dotted decimal notation Internet Protocol (IP) address of the default Socks server you want to use. Contact your system administrator to ensure that you are authorized to use the Socks services.

If you are accessing the WWW through a corporate network, you need to set up the WebExplorer to access your company firewall as a proxy server. WebExplorer is also socksified. You can set it up to contact the company Socks server.

When you complete this panel, click on **OK** to save the changes.

Note: Some changes take effect only after restarting the WebExplorer.

By default the current URL is not shown by the WebExplorer. It is recommended that you turn this on using the Options pull-down menu and selecting **Show current URL**. These changes update the EXPLORE.INI file.

Total Configurability: The size, position, colors, fonts, QuickList selections, network servers, and home page are all remembered between uses of the WebExplorer in the EXPLORE.INI file. By using the -i flag when starting the WebExplorer, users can specify a particular .INI file to use. This allows a network administrator to make one copy of the executable file accessible to users, while each user can maintain individual .INI files on their local disk.

Internal Viewer: By default, WebExplorer shows images by using its own internal mechanisms for handling GIF, JPEG, XBM, TIFF, and OS/2 BMP graphics. If you want to use your own program to view images, select **Internal Viewer** from the Options pull-down menu to toggle this feature on or off. WebExplorer supports true-color displays, that is, those with 65,000 or 1.67 million colors.

Text and Graphics Streaming: Text and graphics can be displayed as soon as they are received from the network. Using the Configure Loading option from the Configure pull-down menu, you can do the following:

- Specify whether you want to preview the document before images are loaded (fast load) or wait until the entire document is loaded to see it.
- Specify whether you want to display images while loading (streaming graphics) or wait until the images are complete before they display.
- Use the fast load method to display the text and put place holders on the images that take longer to load. If the graphics were created with height and width tags, WebExplorer automatically displays the correct amount of space for them. Otherwise, it reformats the display to fit the graphics as they come in.

Streaming graphics is independent of the fast load method. It controls how images are drawn once they begin to display. If you select streaming graphics, you can also say whether you want GIF-interlaced images to be drawn as precise or smeared.

7.3.3 Using the WebExplorer

Once you are connected to the World Wide Web, you should notice that there are highlighted items appearing in the document. These point to hypertext-linked documents at other locations. You can use these to begin surfing around the Internet by clicking on these highlighted text items.

To access an object, simply type in the object type and location in the command line area, as follows:

```
type://location name/directory/file name

For example:

http://ibm.austin.com/aix/test.html
```

The object can be any one of the supported URL types.

If you want to interact with a document before it is completely loaded, you can stop the loading process by pressing the escape (Esc) key or by selecting the animated icon, as shown in the top right-hand corner of Figure 100 on page 178.

7.3.3.1 Drag and Drop

You can use the drag and drop function to capture images and HTML from an open document and place them on your desktop where you can save, maintain, and reuse them as follows:

- To capture an image, place the cursor over an image, press and hold the right mouse button, drag the image to the desired location, and release.
- To capture HTML, place the cursor anywhere in the document except over an image, press and hold the Ctrl key and the right mouse button simultaneously, drag the image to the desired location, and release.

You can also maintain frequently used documents on your desktop by using the URL drag and drop feature. This allows you to drop the URL for a document on a file folder or any other place on your OS/2 Warp desktop to create a URL Workplace Shell Object.

- To drag a URL from WebExplorer move the cursor onto the WebExplorer window over any location except an image, press and hold the right mouse button, move the cursor (and object) to the desired location and release.

Once you have created a URL Workplace Shell object you can do the following:

- Drop the URL object onto an unopened WebExplorer icon and WebExplorer will open with the document at that URL.
- Drag and drop a URL onto an open WebExplorer window and it will open the document at that URL.
- Double-click on the **WebExplorer URL Workplace Shell** object to change the Settings notebook for the icon.

Note: Many applications, such as editors, do not allow copy-based drag operations so you will not be able to directly drop HTML, images, or URLs onto them. Instead, drag and drop them on to the desktop first and then drag them to the application.

7.3.3.2 Color WYSIWYG Printing

Advanced image processing techniques are used to accurately display images and text on both color and monochrome printers. The entire document is also reformatted on the fly to exactly fit the margins of the printer, producing high-quality output for archival or hardcopy distribution of Web documents.

To print a Web page, select **Print...** from the File menu of the WebExplorer.

7.3.3.3 Mailto Support

The following are the two ways you can send mail to other users on the Internet:

- By clicking on a mail address (userid@domain) when it appears as a highlighted link in a document.
- By specifying the mailto protocol and a mail address (mailto:user@domain) as a document URL. Select **Open document (URL)** from the File pull-down menu or use the current URL area under the Tool Bar.

Either of these will display a mailto window in which you can type your correspondence.

Figure 102. Mail To ...

Note: You must have your E-mail address specified. To do this, select **Servers** from the Configure pull-down menu to get the Configure Servers window.

7.3.3.4 NewsReader/2 Articles

Articles on a news server are displayed as a hierarchical tree of conversations. For example, one person posts an article, then someone replies, then someone replies to them, and so on. This function is available to WebExplorer only when the news server supports it.

To connect to your specified news server you enter news:* in the URL command line of your WebExplorer.

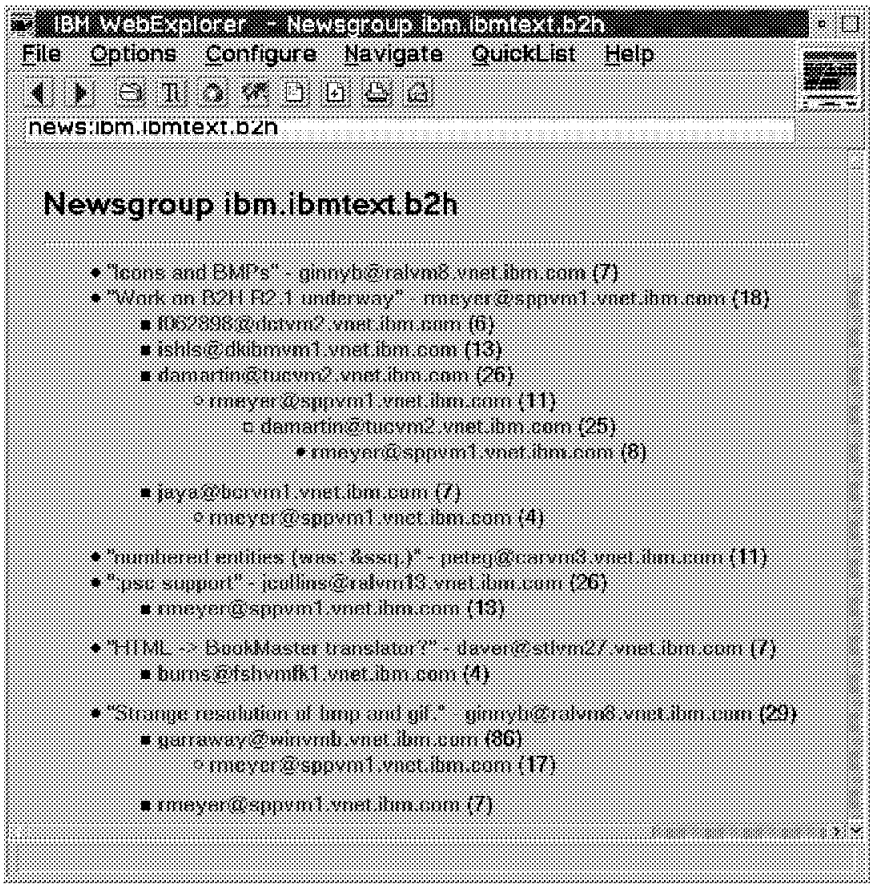


Figure 103. WebExplorer and Newsgroups

To read one of the displayed articles you simply click on the article that you are interested in. The article will be displayed and you can respond to that article or to the newsgroup. The article page will give you the option Post to newsgroup and Follow up.

Selecting **Follow up** displays a newspost window in which you can type your correspondence. The header fields are already defined.

Figure 104. Responding to a Newsgroup

When you have typed in your text you click on **Post Article** to add your article to the newsgroup.

Note: You must have your E-mail address and news server specified. Select **Servers** from the Configure pull-down menu to get the Configure Servers window.

7.3.3.5 QuickList Archival

The QuickLists are written to both the initialization file EXPLORE.INI as well as a separate WEBMAP.HTM file in HTML format. These Web maps may then be exchanged among user groups, renamed and organized into directories, or stored in databases to maintain an entire library of topical QuickLists.

To add the current Web page to your Quicklist, select **Add current document...** from the Quicklist menu of the WebExplorer. To load a document from the Quicklist, simply select the document from the Quicklist menu. You can also select **WebMap** from the Navigate menu of WebExplorer. Items from your Quicklist are marked with a red dot. Click on an item and the document will be loaded.

Future releases of WebExplorer will learn and categorize topical QuickLists on the fly using proven AI technologies. It is also planned to allow loading and saving of QuickLists into separate HTML files.

7.3.4 HTML Markup Language

The Hypertext Markup Language is used to describe the general structure of a document. It is not a page description language like Postscript. Postscript contains page size descriptions and font descriptions. HTML relies on the browser to provide the fonts. As numerous different types of browsers are used to access HTML documents, each based on different hardware types, a page size setting would be pointless. What might work well on one person′s system will not necessarily work well on another′s.

HTML is available in three levels. All browsers support HTML Level 1. A second level, termed HTML Level 2, is now available. HTML Level 2 is similar to Level 1 with a few changes, one being the ability to support what is termed interactive forms. Most browsers available today support HTML Level 1 and Level 2. A third level of HTML has been proposed, but has not been standardized yet. The third level provides support for features such as tables and mathematical symbols. Level 3 is generally not supported by most Web browsers.

7.3.4.1 HTML Structure

All HTML documents have a basic structure, as follows:

```
<HTML>
<HEAD>
<TITLE> This is the page title </TITLE>
</HEAD>
<BODY>
This is the information you want to give on your page
</BODY>
</HTML>
```

7.3.4.2 HTML Header Structure

The following will show you different kinds of headers:

HTML Tag: Every HTML element has a begin tag and an end tag (for example, HTML and /HTML). The first document structure tag is <HTML>, which indicates that the content of this file is in the HTML language. All of the text and HTML commands in your HTML document should go within the beginning and ending HTML tags. The format of the <HTML> tag is as follows:

```
<HTML>
.......
</HTML>
```

HEAD Tag: The <HEAD> tag specifies that the lines within the the HEAD tags form the document header. The format of the <HEAD> tag is as follows:

```
<HTML>
<HEAD>
.......
</HEAD>
</HTML>
```

TITLE Tag: Each HTML document needs a title. To give a document a title, use the <TITLE> HTML tag. The title tag always goes inside the document header (<HEAD>) tag. You can only have one title in a document. The title can only contain plain text. Pick a title that is short and descriptive. The title should also be relevant out of context. That is, someone else could link to your document from elsewhere on the Web, and the title should take this into account, as a title of *Part 3* would not work well if referenced from another site. The format of the <TITLE> tag is as follows:

```
<HTML>
<HEAD>
<TITLE> This a test title </TITLE>
</HEAD>
.......
</HTML>
```

7.3.4.3 HTML Body Structure

The following gives information on how to create the main body of your HTML document:

BODY Tag: The body of document must lie within the <BODY> tag. Elements which make up the body of an HTML document can include the following:

- Headings
- Paragraphs
- Comments
- Links
- Lists
- Images

The format of the <BODY> tag is as follows:

```
<BODY>
.......
</BODY>
```

Headings: HTML defines six levels of headings. Heading tags <H> include numbers to indicate the heading numbers. These numbers are not displayed in the heading itself. The headings appear as bigger, bolder, centered, underlined, or all in capital letters. Their appearance varies from browser to browser. The format of the <H> tag is as follows:

```
<BODY>
<H1>Test heading level 1</H1>
<H2>Test heading level 2</H2>
<H3>Test heading level 3</H3>
<H4>Test heading level 4</H4>
<H5>Test heading level 5</H5>
<H6>Test heading level 6</H6>
</BODY>
```

Paragraphs: Paragraph tags are used to indicate paragraphs within your document. A plain text paragraph is defined using the <P> tag.

Unlike the other HTML tags, the paragraph tag format varied from HTML Level 1 and 2. Most browsers support both Level 1 and Level 2 tag definitions. The Level 2 format of the <P> tag is as follows:

```
<BODY>
<H1> Level 1 header /<H1>
< P >
This is a test paragraph
< / P >
</BODY>
```

Comments: It is useful to put comments into a document to help you describe the document itself and to provide some kind of status of the document. The text inside the comment tag is ignored by Web browsers. Each comment line should be individually commented. The following is an example of the comment tag:

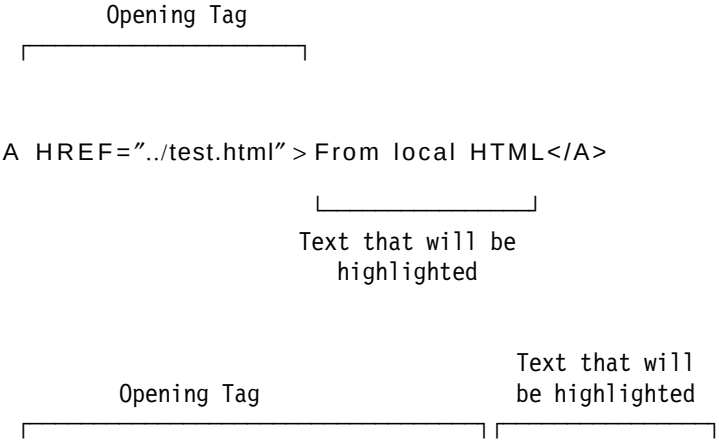
```
<!- This is a comment statement ->
```

Links: The real power of the HTML language is when you start using links to link your document with other objects on the Web. These objects include other HTML documents, images, and audio files. This allows you to reference other objects on the Web within your own document, which by simply being clicked on, can be called up locally by a Web browser. In the same light, other Web publishers can refer and place links to your objects in their documents.

To create a link in HTML, you need the following:

- The name and location of the object to which you are linking.
- The text or image which will serve as the hot spot which readers see when browsing your document. If you include images in your link, you need to take into account that some readers are using ASCII-based browsers and are not able to view your images.

To create a link you use the <A>...</A> tags. Unlike the previous tags, the <A> tag includes tag attributes. The following is a typical link tag format:



or

```
<A HREF="HTTP://ibm.com/test.html" > From HTTP site</A>

          Linked HTTP file
     └──────────────────────┘

                                          └──┘
                                        Closing Tag
```

Lists: The HTML list tags allow you create the following:

- Numbered lists <OL>
- Unordered lists <UL>
- Menu lists <MENU>
- Directory lists <DIR>
- Glossary lists <DL> and <DD>

All list tags have the following in common:

- The form is surrounded by the opening and closing tags (for example <OL> and </OL>).
- Each element within the list has its own tag <DT> and <DD> for glossary lists and <LI> for lists.

The following is an example of the list tag:

```
<BODY>
<UL>
<LI> This is list element one
<LI> This is list element two
<LI> This is list element three
<LI> This is list element four
</LI>
</BODY>
```

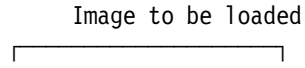
Images: HTML allows you to include both inline images and external images in your documents. Inline images are images that appear directly on the Web page. External images are images that are only downloaded at the request of the reader. Inline images can be in various graphics formats including GIF, JPEG, BMP and XBM. Inline images are specified using the HTML tag <IMG> tag. The <IMG> tag has no closing tag. Although you normally use the <A> tag to include images in your document, you can also use it to imbed the other formats. You require the necessary browser and hardware to utilize the following formats:

- AIFF sound files (.au)
- JPEG compressed image format (.jpg)
- Wave format sound files (.wav)
- MPEG audio (.mp2)
- MPEG video (.mpg)
- AVI video (.avi)

An example of an inline tag is as follows:

```
<BODY>
<IMG SRC="test.gif" >
</BODY>
```

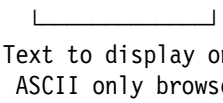
The image tags are very useful with other tag types. For example, you can imbed the images into your link tags as follows:




```
<A HREF="HTTP://ibm.com/test.html" >< IMG SRC="test.gif" >
                              Test imbedded GIF</A>
```

Images can only be viewed using GUI-based browsers. To cater for readers using an ASCII browser, you should include a text-only option in your Web page. This can be done using the ALT tag when imbedded in a <A> link tag. The ALT tag will display an alternative to the image if an ASCII browser is used. An example of the ALT tag is as follows:

```
<A HREF="HTTP://ibm.com/test.html" >
<IMG SRC="test.gif" ALT="ASCII text" >Test imbedded GIF</A>
```



Other basic HTML tags which you will encounter include:

 	A line break
<HR>	A horizontal rule
<EM>...</EM>	Emphasize text
<STRONG>...</STRONG>	Stronger emphasize your text
<B>...</B>	Bold text
<I>...</I>	Italic text
<TT>...</TT>	Typewriter text
<SAMP>...</SAMP>	Sample text

7.3.5 HTML Example

The following is an example HTML document:

```
<!doctype html public "html2.0" >
<html>
<head>
<title>Klaus Wichmanns Homepage</title>
</head>

<body>
<img src="welcome.gif" alt="IBM" >
<h1>Welcome to Klaus Wichmanns Home Page</h1>
< p >
< h 2 >
<A HREF="kw1.html" >
<img src="ME.GIF" >
About me !
< / A >
</h2>
< p >
< h 2 >
```

```
<A HREF="notes.html">
<img src="NOTES.GIF">
Lotus Notes
</A>
</h2>
<p>
<h2>
<A HREF="mailme.html">
<img src="UPS.GIF">
Mail me ...
</A>
</H2>
</body>
```

This will look like the following, when you display this document with your WebExplorer:

Figure 105. Sample Web page

7.4 Lotus Notes and the World Wide Web

Since Lotus Notes Release 4 access to the World Wide Web is supported. The InterNotes Web Navigator is a Notes Release 4 that allows you to navigate through pages on the Web directly from your Notes environment. The Web Navigator is much more than a Web browser; it combines the features of a Web browser with the powerful capabilities of Lotus Notes.

Each time you retrieve a page off the Internet, the Web Navigator translates it into a Notes document and stores it inside the database. (Whenever you retrieve a page off the Internet, Notes displays a spinning globe and status bar messages.) Then, the next time you want to read that page, you can open it directly from the database instead of retrieving it from the Internet. Opening a page that is already inside the database is much faster than retrieving it off the Internet. You can run agents that keep the pages inside the database updated or you can update the pages using the Reload button.

Once the page is in the database, you can view it, copy it into a private folder, cut and paste it into another Notes document, mail it to another Notes user, and so on. All of the capabilities of Notes are available for you to use with the pages inside the Web Navigator database.

7.4.1 Setting Up InterNotes at the Notes Server

The Web Navigator consists of a database and a server task that reside on a Notes server, referred to as the InterNotes server. The InterNotes server does the following:

- Stores the Web Navigator database
- Runs the Web Navigator server task
- Runs the TCP/IP network protocol
- Maintains either a direct or proxy connection to the Internet

To run the Lotus Notes Web Navigator the Web server task must be running on a Lotus Notes server. This task will retrieve the information from the Web and update the Web Navigator database that resides on that server. To set up the Web Navigator, the Administration document in the Web Navigator database must be configured.

The following gives you a short overview over the basic steps to set up the Web Navigator on your Lotus Notes server:

1. Start the Web server task. Enter the command:

   ```
   load WEB
   ```

   This starts the Web Navigator. The first time you start the Web Navigator at your server a Web Navigator database (Web.nsf) is created. This database contains an administration document and is set up with default values.

2. Edit your NOTES.INI file to start the Web Navigator automatically each time you start the Lotus Notes server. Add the following to the NOTES.INI file on the InterNotes server:

   ```
   ServerTasks=...WEB
   ```

3. Open the Web Navigator database at your Lotus Notes client. You get the following screen:

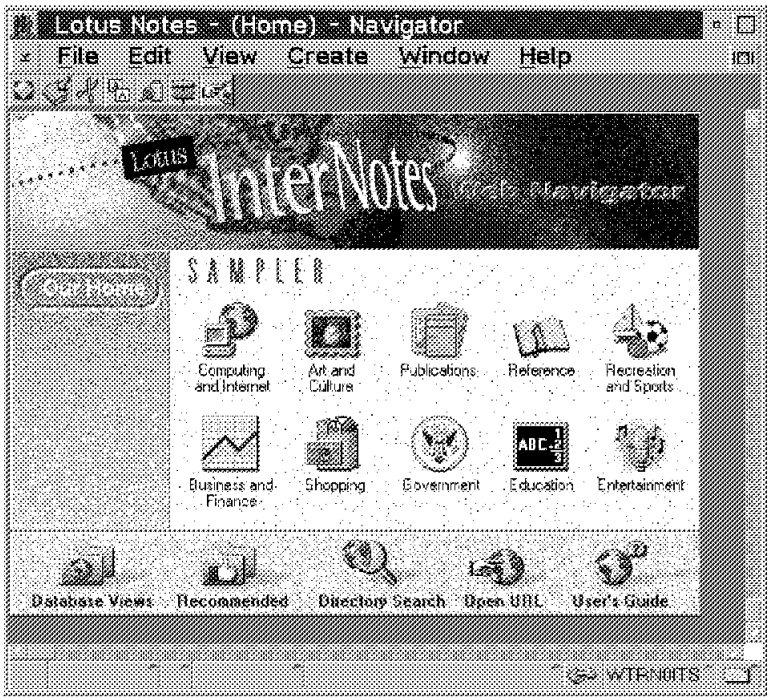


Figure 106. Web Navigator

Select **Database Views** located in the bottom left corner of this panel.

4. Select **Administration** from the Action menu. This shows you the following configuration screen:

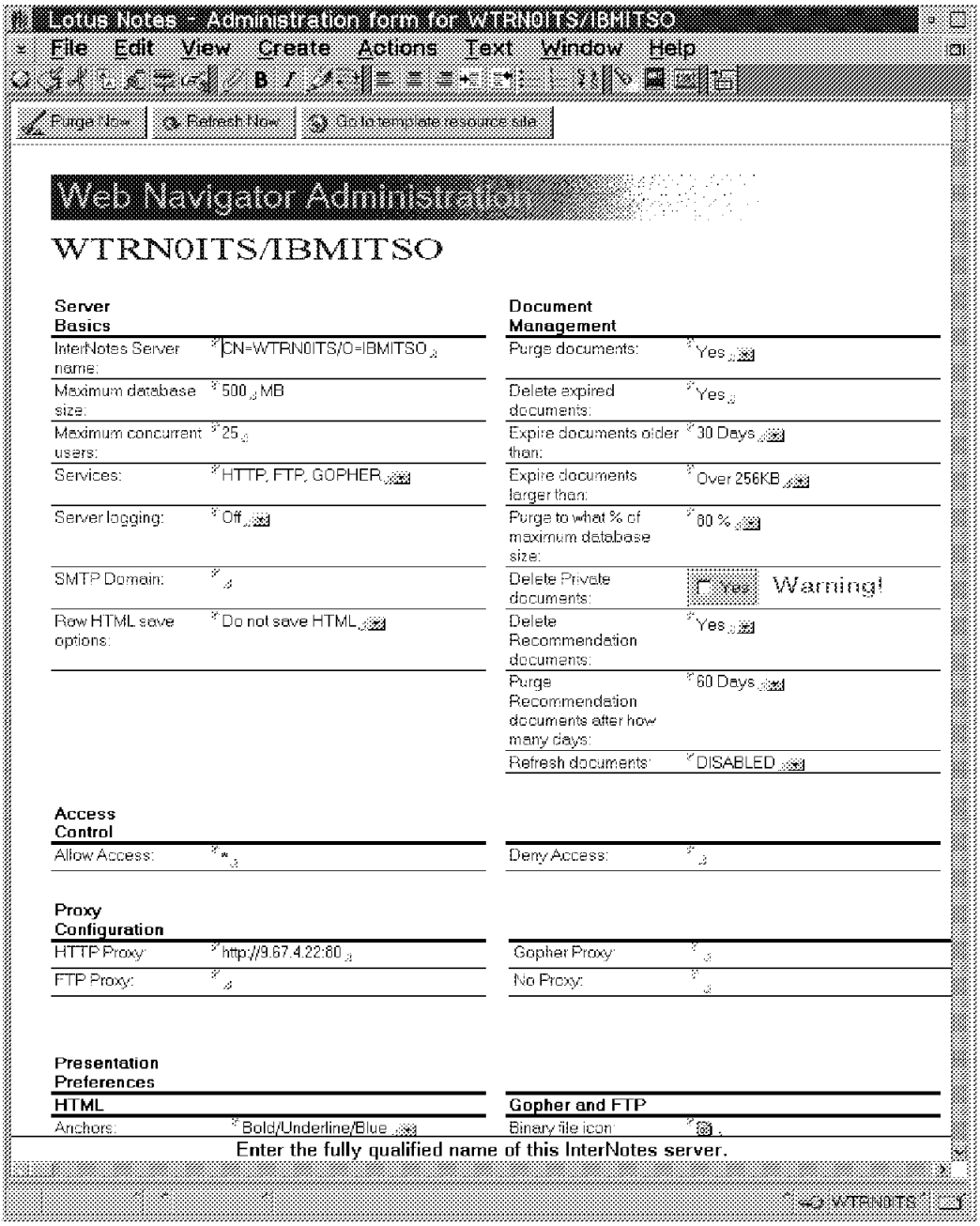


Figure 107. Web Navigator Administration

Fill in the required information as follows:

InterNotes Server name This field should already be filled in with the name of your InterNotes server. If you need to change the name in this field, enter the hierarchical name of the InterNotes server where the Web Navigator resides. This is a required field and is case sensitive.

Maximum database size Enter the maximum size in megabytes for the database. The default value of this field is 500 MB.

Maximum concurrent users Enter the maximum number of users that can use the Web Navigator database concurrently. This number depends on the system configuration for your InterNotes server.

If you find that user access is slow because the number of users specified in this field is less than the number of users attempting to retrieve pages off the Internet, increase the number. The default value of this field is 25.

Maximum documents per private folder Enter the maximum number of documents that users can store in their private folder. The private folder holds Web pages from authenticated Internet servers as well as response documents from Web fill-out forms. The default value of this field is 200.

Services Choose the Internet services you want to allow users to access. The Web Navigator supports HTTP, FTP, and Gopher. The default value of this field allows access to HTTP, FTP, and Gopher.

Server logging Choose On or Off to enable or disable the log messages being sent to the InterNotes server console and the server LOG.NSF. The default value of this field is Off.

SMTP Domain To set up routing of Notes mail to the Internet, fill in this field with the foreign domain name of your SMTP mail gateway. First you need to have an SMTP mail gateway up and running.

Raw HTML save options Choose an option to determine whether Notes saves the HTML source for Web pages. Choose **Do not save HTML** if you do not want Notes to save the HTML source after it converts the page into a Notes document. Choose **Save in field** if you want Notes to save the HTML source in this field. Choose **Save in CD record** if you want Notes to save the HTML source as a Notes CD record. The Save in CD record option is for future use.

Allow Access Groups or individual persons who can access the InterNotes server.

Proxy fields Proxy gateways for HTTP, FTP and Gopher. This field is required if you want to access the Internet from a corporate network. The following is an example:

```
http://9.67.4.22:80
```

where http is the type of gateway followed by its IP address and the port number.

All other fields are self-explanatory

5. Specify the InterNotes location in the public name and addressbook. Open the public name and addressbook and select the view **Servers**. Open the document for your Notes users' home/mail servers and enter the location of the InterNotes server in the InterNotes Server field.

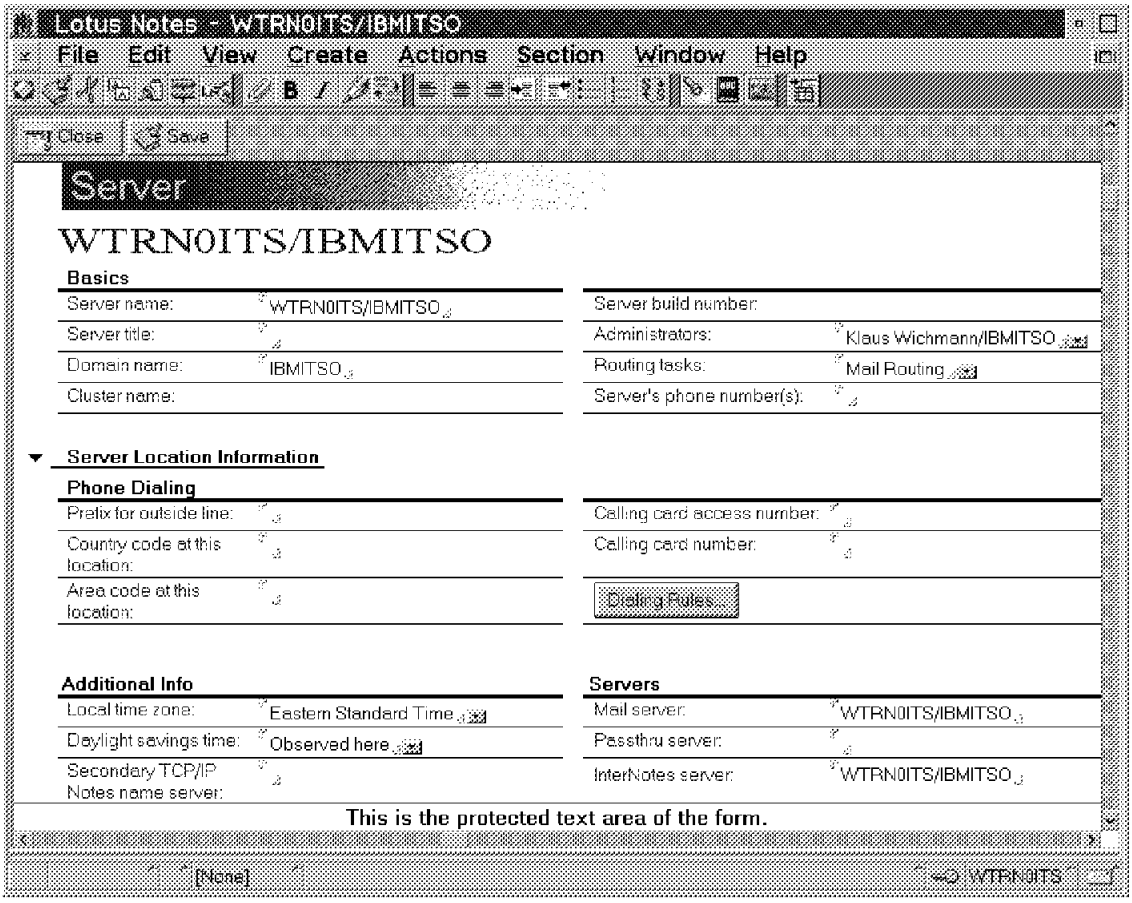


Figure 108. InterNotes Location Entry

6. Define the access rights for the Web Navigator database. Select the Web Navigator database and go to the File menu of Lotus Notes. Select **Access control...** from the Database menu. You get the following configuration screen:

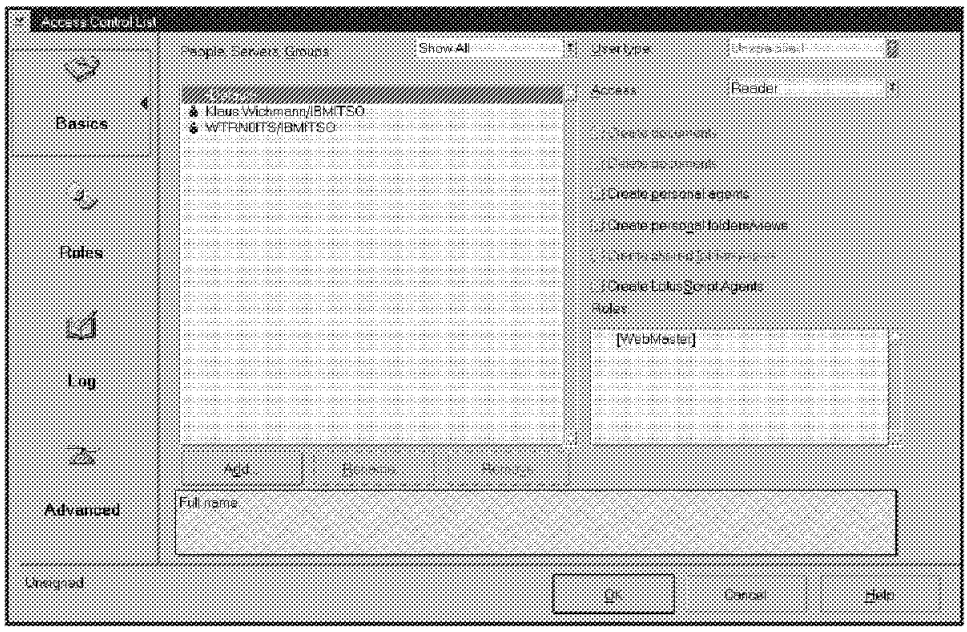


Figure 109. Access Control to the Web Navigator Database

Define the access rights for people and groups like you define them for any other database.

The configuration is now complete. Lotus Notes users can access the Web Navigator database and retrieve documents from the Internet.

7.4.2 Using the Web Navigator from the Lotus Notes Client

Each Lotus Notes R4 client can access the World Wide Web by accessing the Web Navigator database stored on the InterNotes server. Access to the World Wide Web is independent of the client's configuration and whether it is using TCP/IP or not. As long as the Lotus Notes client can access the Web Navigator database it can surf the Web.

If the location of your InterNotes server is not already defined in the Public Name and Addressbook, you have to define its location in the location document of your Private Name and Addressbook. Therefore you open your Private Name and Addressbook and select the location that you want to edit from the location view.

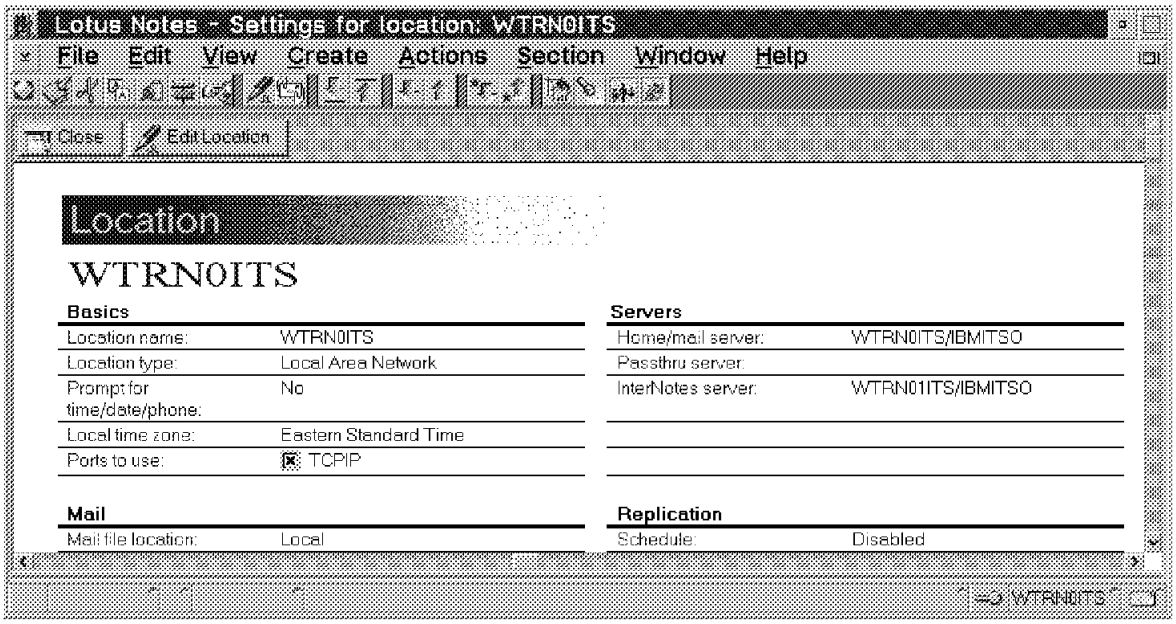


Figure 110. Updating the Location Document

Enter the location of the InterNotes Server and save the document.

To retrieve information from the World Wide Web, the Web Navigator database must be opened at the Lotus Notes client. Figure 106 on page 192 shows the home page when you open your Web Navigator database.

The ten icons in the sampler area gives you access to certain areas on the World Wide Web. Click on an icon to view the information which is behind it.

To load a certain Web page select **Open URL** and enter the address of the Web page. The following dialog box will be displayed.

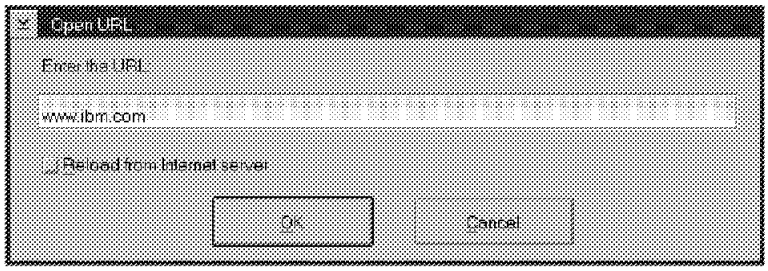


Figure 111. Entering a URL

Once you click on **OK**, InterNotes retrieves the Web page from the Internet. The following is a rotating world which is the symbol that appears in the upper right-hand corner of the screen. This symbol informs you that InterNotes is busy with retrieving and converting the document.

Figure 112. Rotating World

You can stop the loading of the Web page by clicking on the world symbol. You then return to your first Navigator page which is called your home page.

The Database View that you can select from your home page is probably the view that you will use most of the time. It shows you all the documents that are available on your InterNotes server. These documents are Web pages once received by you or by other InterNotes users.

These documents can be handled like any other Lotus Notes document. In fact, they are Lotus Notes documents. InterNotes has converted the Web pages to Lotus Notes documents.

Because they are Lotus Notes documents, these documents can be displayed using different views. The available views are as follows:

- All Documents
- By Host
- File Archive
- Recommended
  - By Category
  - By Reviewer
  - Top Ratings
- Web Tours

For example the view By Host could look like the following:

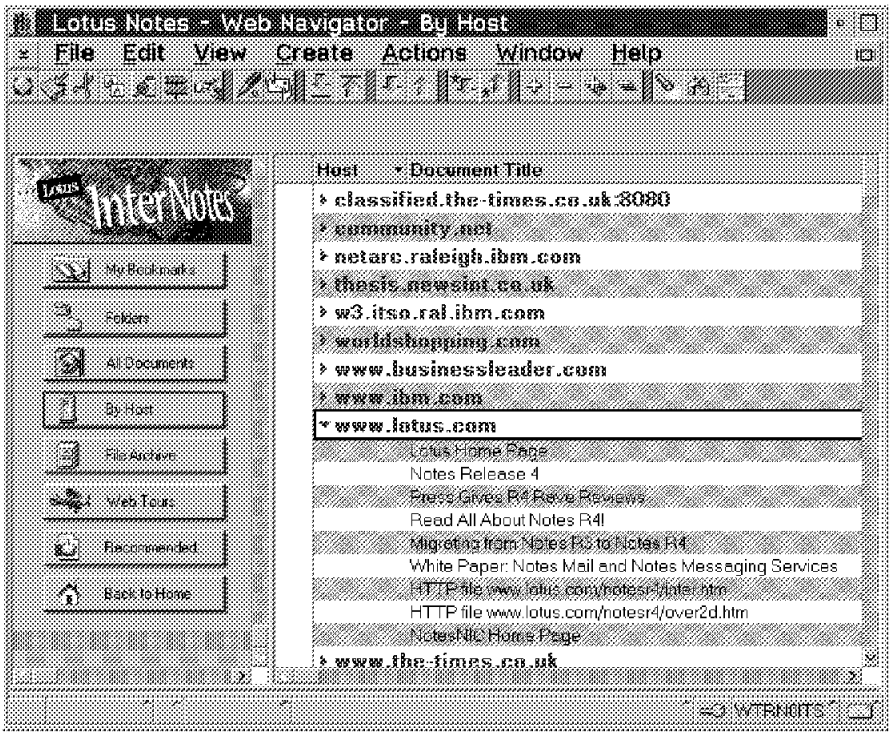


Figure 113. View by Host

To see the actual Web page you simple double-click on the document as you would do with other Lotus Notes documents. The document will then be shown,

and as you can see in the following figure, the converted document looks like the original HTML document.

Figure 114. Viewing a Web Page

On the top of the document you can see different icons with different functions. The following is a summary of these functions:

Home Brings you back to your Home document.

Open Lets you enter a URL.

History Shows you the history of viewed documents.

Reload Reloads the document from the Internet.

Recommend Lets you recommend the document to other users. You can rank and categorize the document.

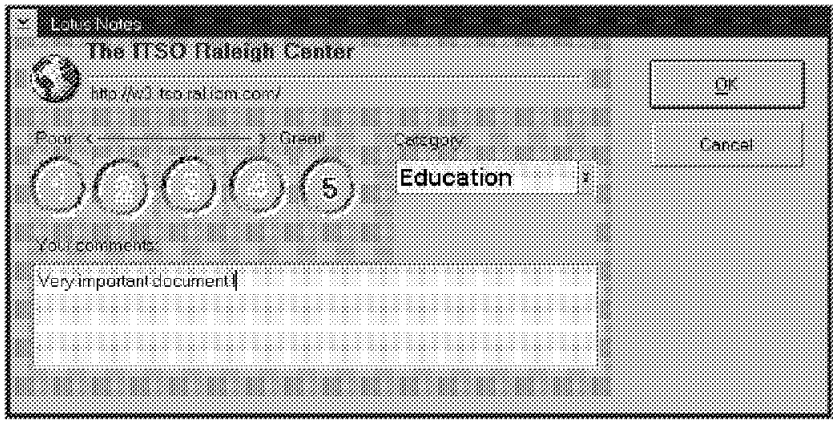


Figure 115. Recommending a Document

Click on one of the ranks from 1 to 5 to rank the document. From the list box Category you can choose a category for this document. The comments field gives you the possibility to enter a short text.

Forward Lets you forward the document to another Lotus Notes user. An address field where you can enter the address of the recipient will be inserted in the document. You can send the document to groups or to individual persons, like any other document. It is also possible to make changes to the document and enhance it with more information.

Bookmarks Lets you organize documents in different folders. You can create a new folder and add the document to that folder.

As you can see, there are various ways to organize the information that you retrieve from the Internet. The Web Navigator is more than a Web browser. The Web Navigator lets you not only view pages but also reuse the information. It lets you really work with the information, which is the greatest advantage of Lotus Notes R4.

7.5 Setting Up an Internet Connection to an IBM Service Provider

By default TCP/IP for OS/2 provides a connection into the IBM Global Network using the SLIP protocol.

To use the default IBM Global Network service provider, select any application icon in the Internet Connection for OS/2 folder, shown in Figure 116 on page 201. For example, we selected the NewsReader/2 icon.

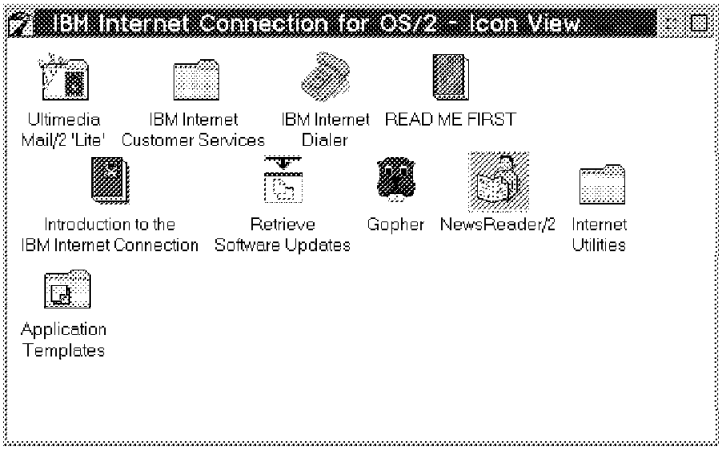


Figure 116. Internet Connection for OS/2 Folder

To set up a user ID with a service provider, you should make sure that you are not using the modem for any other applications at the same time as setting up your logon ID. After selecting the NewsReader/2 icon, you see the screen shown in Figure 117. We chose the IBM Internet Service Provider. To do so, under Dialer select the **IBM Internet** radio button, then click on **Connect**.

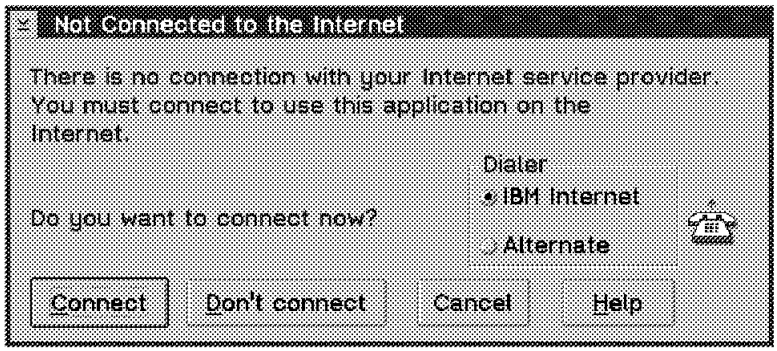


Figure 117. Not Connected to the Internet

The system now indicates that you do not have a logon ID on an IBM service provider system. To get a logon ID, you need to open a personal account with the service provider. You will see the IBM Internet Registration menu as shown in Figure 118 on page 202. Click on the **Open a personal account** button. Note that you need one of the following credit cards to register with the IBM Global Network:

- American Express
- Visa
- Diners Card
- Discover
- JCB
- MasterCard

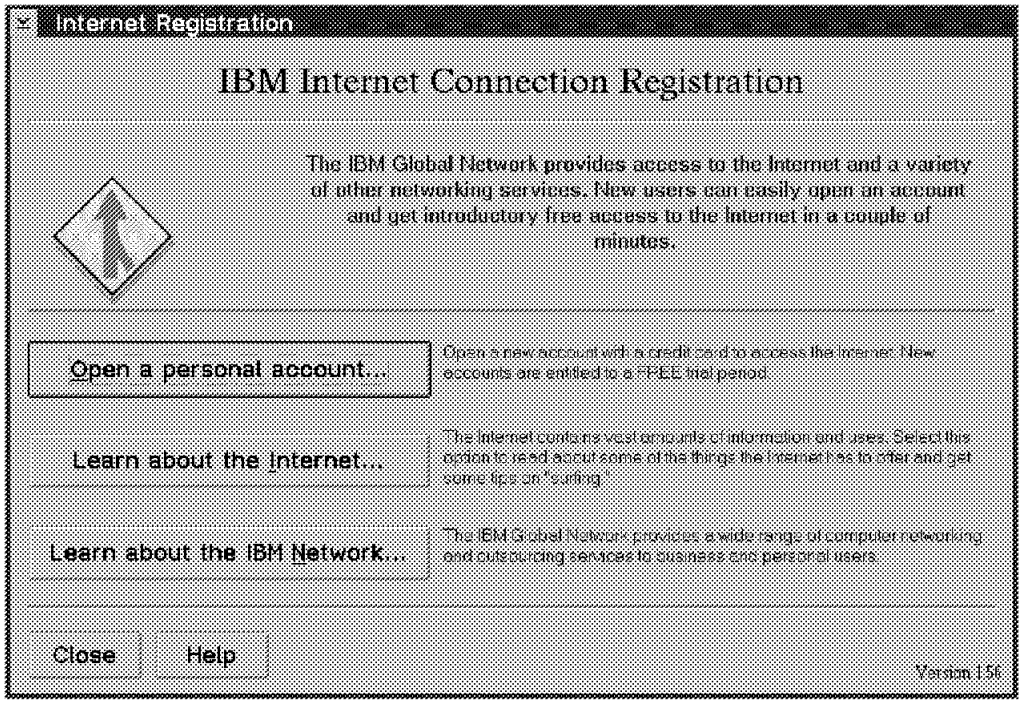


Figure 118. Internet Registration

The system will now provide you with the IBM Internet Service Terms and Conditions shown in Figure 119 on page 203. Please read these carefully before you proceed. When you have finished, click on the **OK** icon. The Terms and Conditions cover the USA terms and conditions and may not be valid in your country. Contact your local IBM representative if you have a question about your local terms and conditions.

When you are finished reading the information click on **OK**. You are then presented with an Account Owner Information screen, where you must enter your own personal account details, as shown in Figure 119 on page 203.

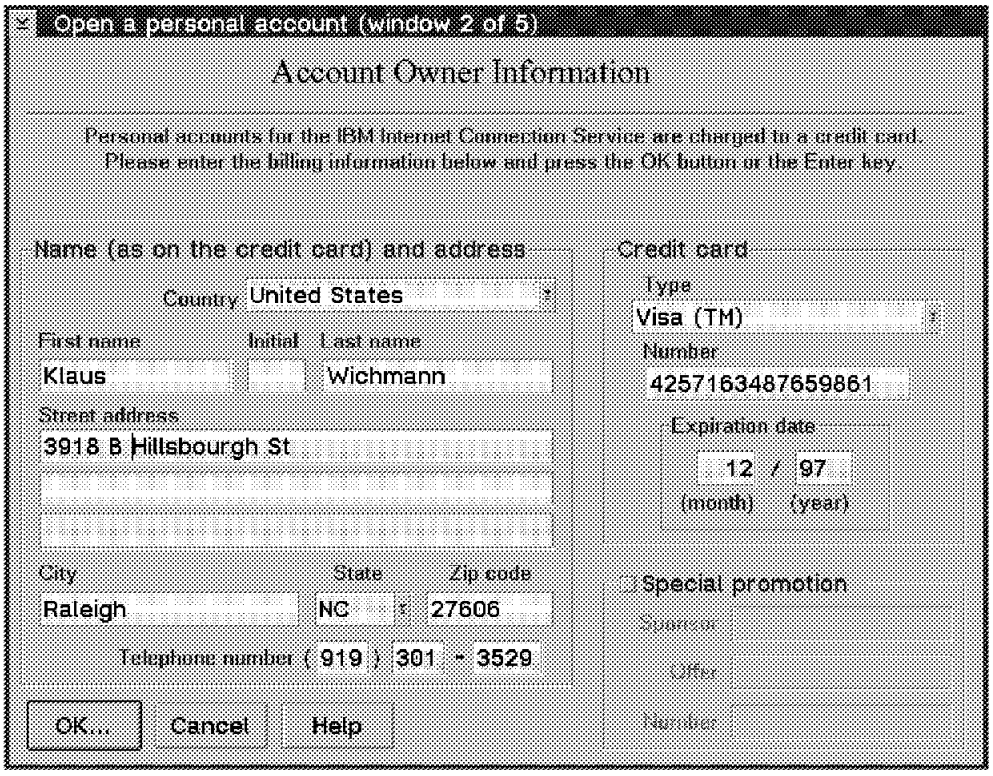


Figure 119. Account Owner Information

The system checks details such as credit card numbers for their validity. Make sure that you have selected the correct country details. Your credit card details are not sent around the currently insecure Internet. The telephone number that you dial when registering is a direct-dial telephone specifically used for registering. It does not use the Internet. You are assigned a logon account number by this registration account. The next time you dial into the system as a registered user, a different telephone number will be used, as specified when you register. You must then select the **OK** button. The system presents you with the modem and dialing details as shown in Figure 120 on page 204. This screen allows you to choose the modem that you have connected to your system. The screen shown allows you to choose your communications port settings as well as your phone line characteristics. We selected the IBM 7855 12000 bps modem.

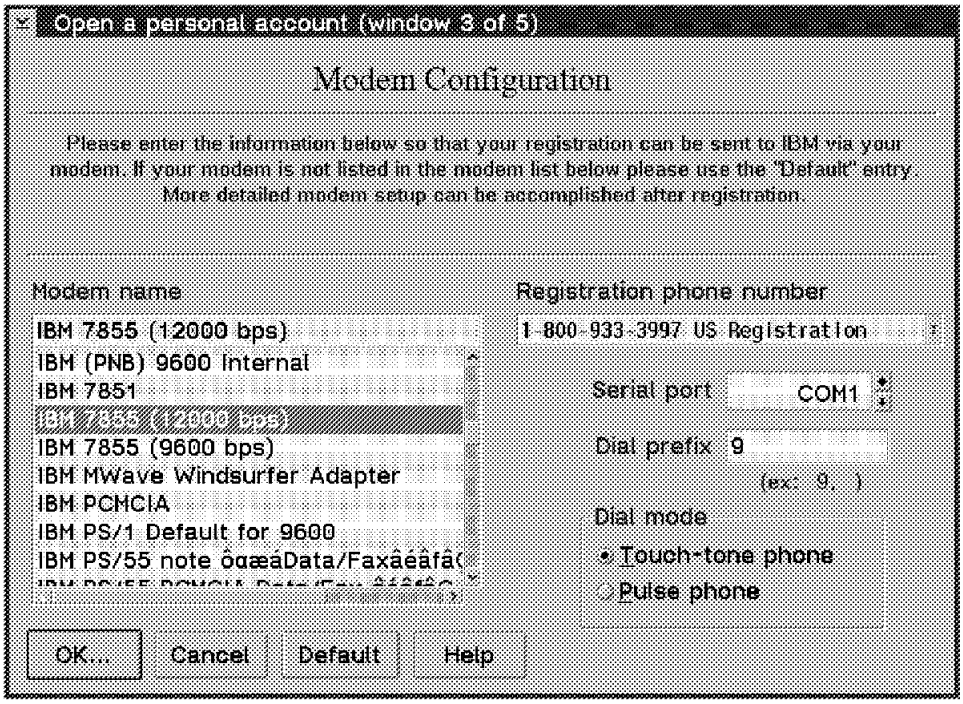


Figure 120. Modem and Dialer Information

The system assigns you a default user ID as shown in Figure 121. To allow you to receive mail and be identifiable to the Internet, the system allows you to specify your user ID. Commonly, someone else on the system may have already chosen your preferred user ID. The system creates three user ID choices. You are assigned one of these when your registration is accepted. The priority of assigning user IDs starts with your first choice of user ID. If three alternatives are not sufficient, you are registered with a number suffix after your first ID choice.

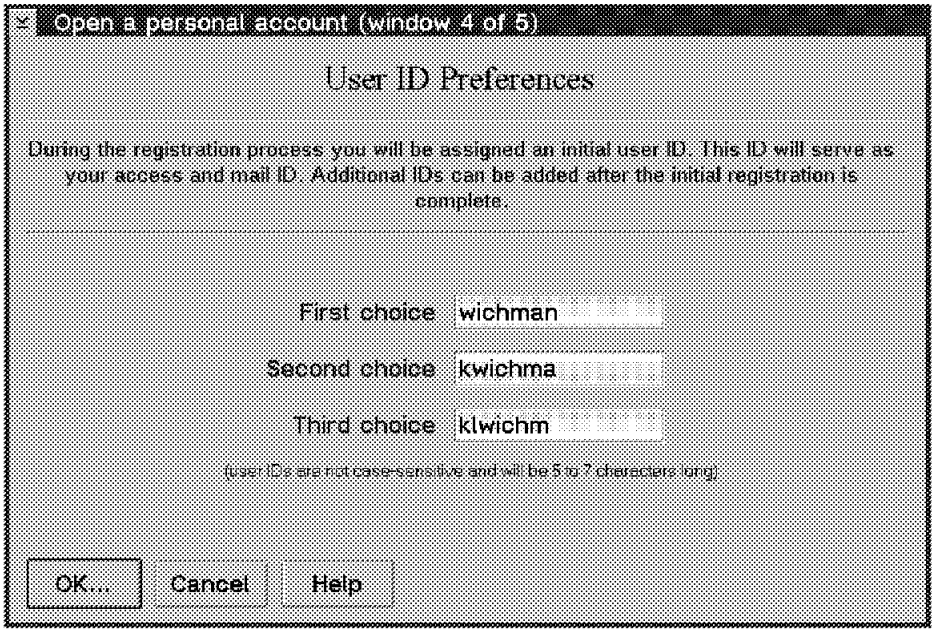


Figure 121. User ID Preferences

Click on the **OK** button. You will then be able to submit your registration to the IBM Global Network. In the next window you can send your registration to the network provider. Therefore select **Send Registration to IBM.**

Your system then starts dialing the registration telephone number. If you receive an error, check your telephone characteristics and modem type. The telephone number may be busy. Once your system is connected, your registration will be sent to the service provider.

After you sign onto the system, you will see your personal agreement price schedule, together with the service fees. Please read the terms and conditions carefully.

After registration is complete you are assigned a logon account number by the IBM Global Network service provider. You will see the following four fields:

Account Your logon account

User ID Your user ID assigned to you on this system

Password Your logon password

E-mail address Your E-mail address to which other people can send mail

You should now be signed on to the system successfully. Take note that you have a free trial period to be able to test the system. This might vary from location to location. In the US, Advantis provides you with a free trial period for three hours of use or 30 days, whichever occurs first.

Now that your registration is complete, a connection will be established automatically and the application you originally selected will start. In our case, the NewsReader/2 product will start up.

You only need to register once. The next time you select an Internet application, before a connection has been established with the service provider, you will again be asked whether you want to sign on to the Internet; however, the application will use your previous registration details, and you will not have to register again.

7.5.1.1 Updating the IBM Internet Connection Software

To ensure that you have the most current version of software, you should download the latest version. The IBM Global Network allows you to do this. To download this version, go to the IBM Internet Customer Services icon view. Select the **Customer Assistance** icon as shown in Figure 122.

Figure 122. IBM Internet Customer Services Folder

You will see the menu shown in Figure 123 on page 206. This must not be used if you are not using an IBM Global Network service provider. Select **Update Software** as shown in following figure.

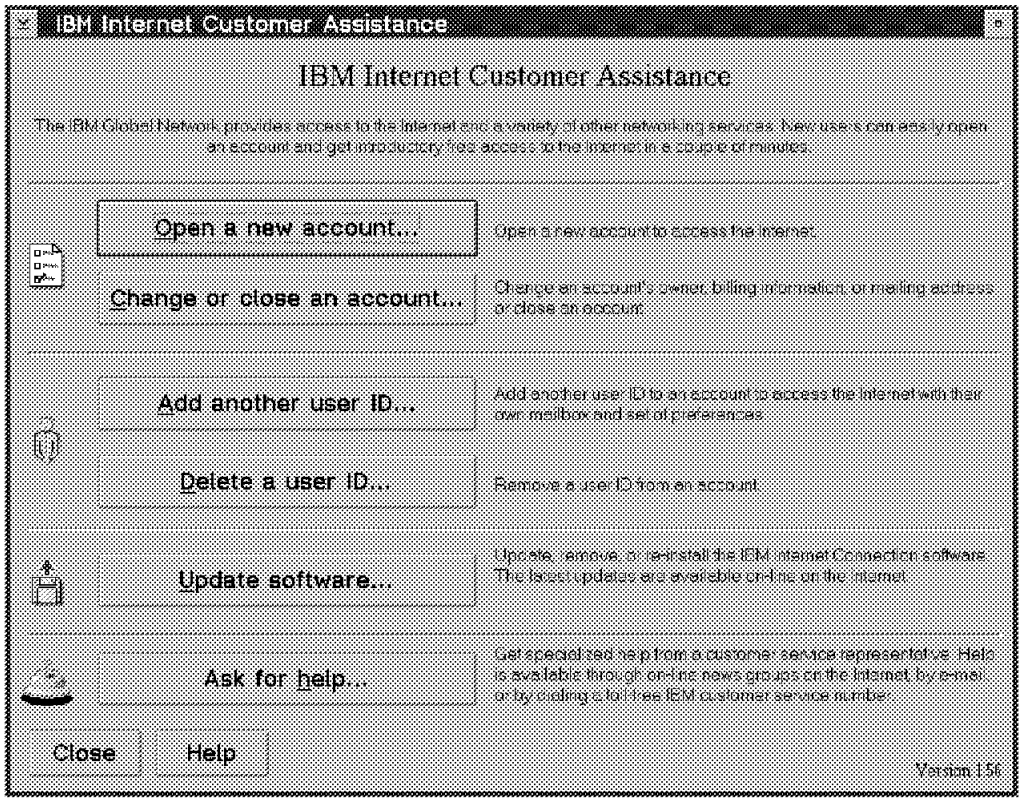


Figure 123. Customer Assistance for the IBM Internet Connection Services

You will now see Update IBM Internet Connection Service Software, as shown in Figure 124 on page 207.

On this screen you will see a number of user-selectable icons. You must select **Download latest software...**

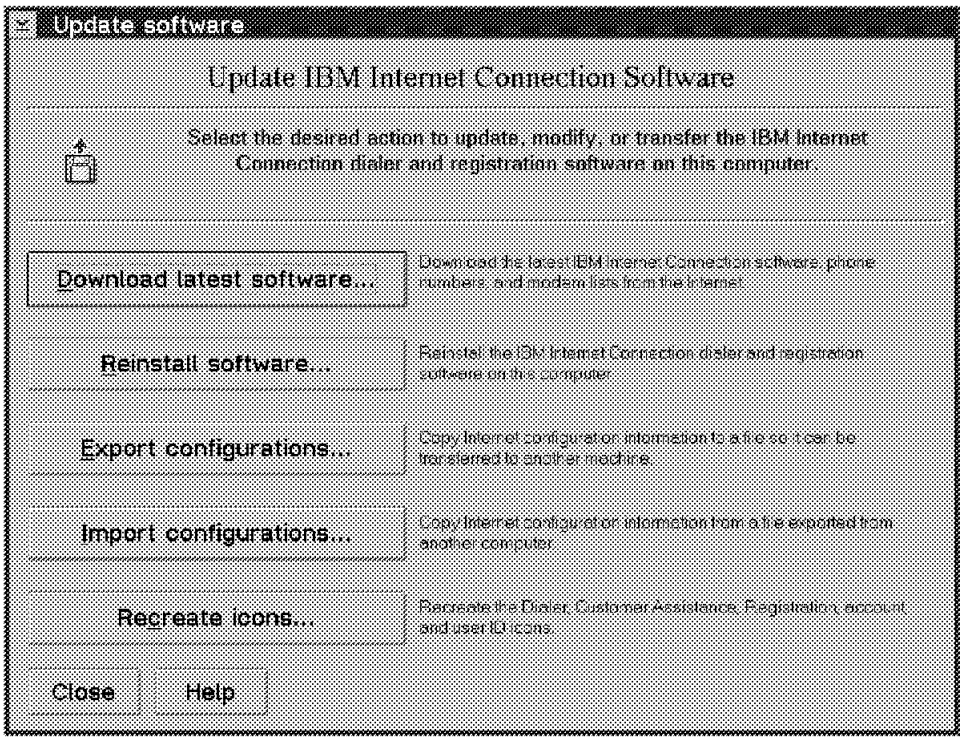


Figure 124. Download Latest Software

The Internet connection version level will now be checked on your system. The program will advise you which version of the Internet software is present on your system, as well as which version of software is available from IBM, as shown in Figure 125. This takes a couple of minutes to check. Select **Dialer & Customer Assistance programs (includes the phone and modem list)**. Select **Download**. This process takes a few minutes to complete.

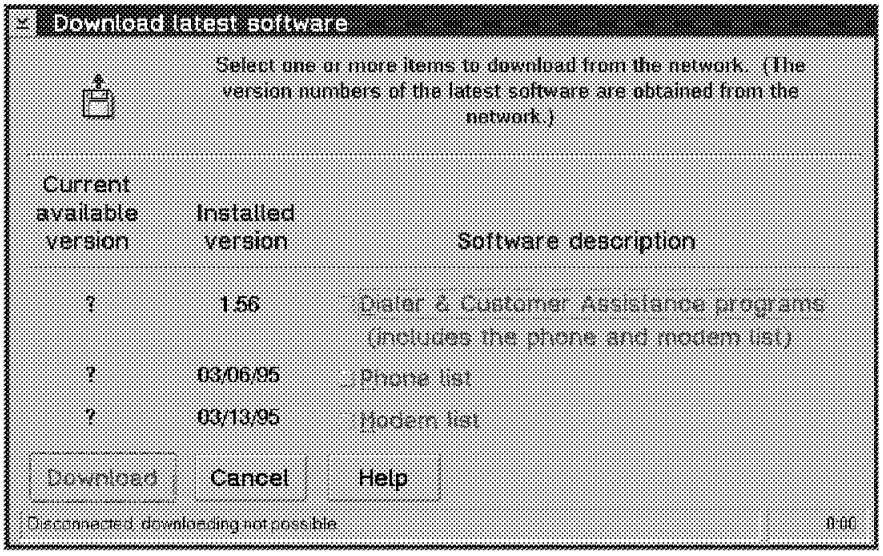


Figure 125. IBM Internet Connection Services Software Version Check

When the software has been downloaded, you have the option of installing the latest version of software on your system. You must select **Reinstall software**.

The system will therefore run the INSTALL.EXE program from your TCPIP\TMP directory.

After the reinstallation, you have to shut down and reboot your workstation for the latest version of the software to work correctly.

7.5.1.2 Deleting a User ID through the IBM Global Network

To delete a user ID, you need to first set up your Internet connection as normal. Sign on as the user you wish to modify. Open the IBM Internet Customer Services folder and double-click on the **Customer Assistance** icon. You will see the menu shown in Figure 123 on page 206. This must not be used if you are not using an IBM Global Network service provider. Select **Change an existing account**. You will see the Close Account window. The account number information will be entered automatically by the system. You need to re-enter your password and an optional reason for closing the account. Select **OK**. When next accessing the Internet system, the dialer program will automatically access this last user when trying to sign on, even though the user no longer exists. You will need to manually enter another user ID.

7.5.2 Setting Up an Internet Connection with Another Service Provider

There are several other service providers on the market. If you wish to set up a connection with another service provider, you need to set up your configuration correctly.

Select the **Network Dialer** icon from the TCP/IP icon view.

You will be presented with the IBM Dial-Up for TCP/IP panel, as shown in Figure 126 on page 209. Select the **Add Entry** icon to add a new service. When you have set up your entry, you will see your name, login ID and description. You then only have to click on the entry to dial your service provider.

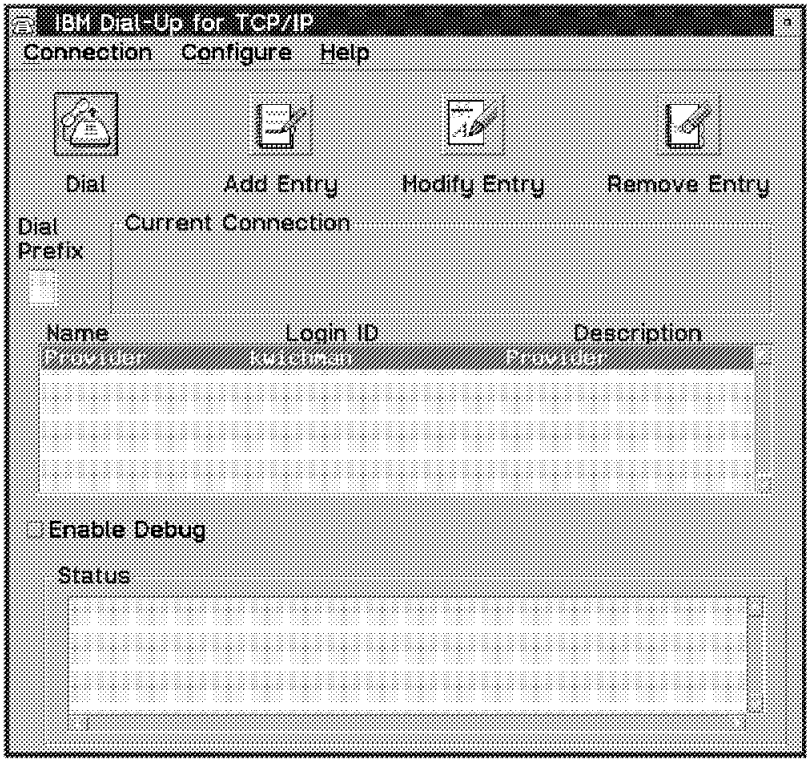


Figure 126. IBM Dial Up for TCP/IP

You must contact your service provider for the following details:

- Type of connection (SLIP or PPP).
- Your IP address (if dynamic you might have to enter 1.1.1.1).
- Your login ID and default password. You will have to change this password after you log in for the first time.
- The service provider's destination IP address.
- Netmask parameters (or leave empty if the default of no netmask will do). This will specify who you can contact on the local IP network.
- Your domain name server.
- Your hostname. (You can use your own name.)
- Your domain name (for example, ibm.net).
- Your news server.
- Your Gopher server.
- Your World Wide Web server.
- Your POP mail server, if your mail server supports the POP protocol.
- Your reply domain, if different from your domain name.
- Your mail reply ID (for example, kwichman@raleigh.ibm.com).
- Your POP login ID and password. These are normally the same as your connection ID and password.

The login configuration screens are shown in the following figures:

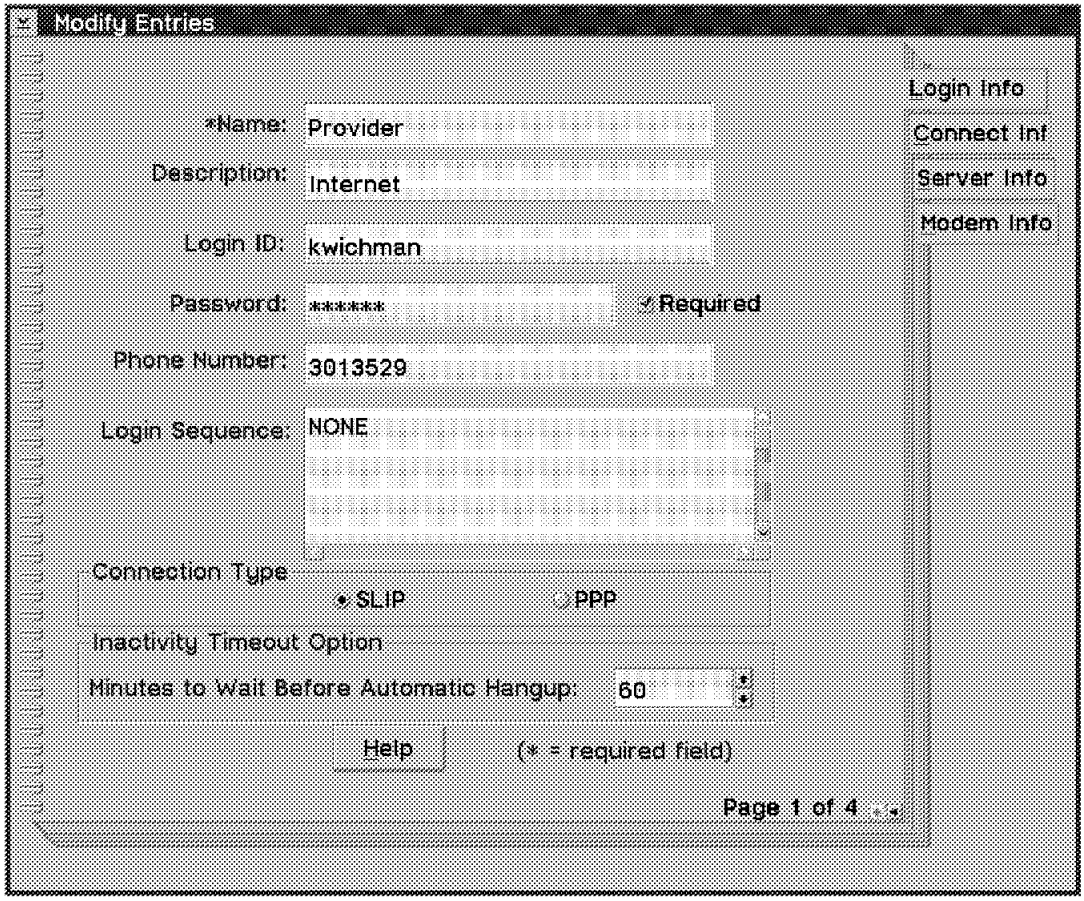


Figure 127. Login Info Configuration Screen

Fill in the login information required to connect to the service provider. The required fields are marked by an asterisk. The following gives you a short explanation of the most important fields:

Name Specify an identifier of the connection. This can be a comment such as connection to work or the name of a service provider.

Login ID Specify the user identification assigned to you.

Login Sequence Specify the login sequence that you want to use, if any. You can use a login sequence to automate a connection. To accommodate a variety of connection sequences, this field may contain the following:

- The reserved word NONE. This indicates no login sequence is required beyond the physical modem connection.
- Blank, or no entry. Let's say this field is left blank, and the Login ID and Password fields are filled in: when IBM Dial-Up for TCP/IP receives the following login sequence:

  ```
  login:
  password:
  ```

  the content of the Login ID and Password fields are sent in response.
- The name of an ASCII or REXX connection script, or response file (for example, annex.cmd). This file is executed at connection time

to negotiate the modem setup, dial the destination host, and log in to the host.

- A login sequence, which consists of a series of send-expect verbs.

Information entered in this field is stored in the TCPOS.INI file.

If you are using a service provider, each provider may use a slightly different sequence for establishing a connection. You must tailor your login sequence to match each service provider.

Connection Type Select whether you are using serial line Internet protocol (SLIP) to access the service provider or the point-to-point protocol (PPP).

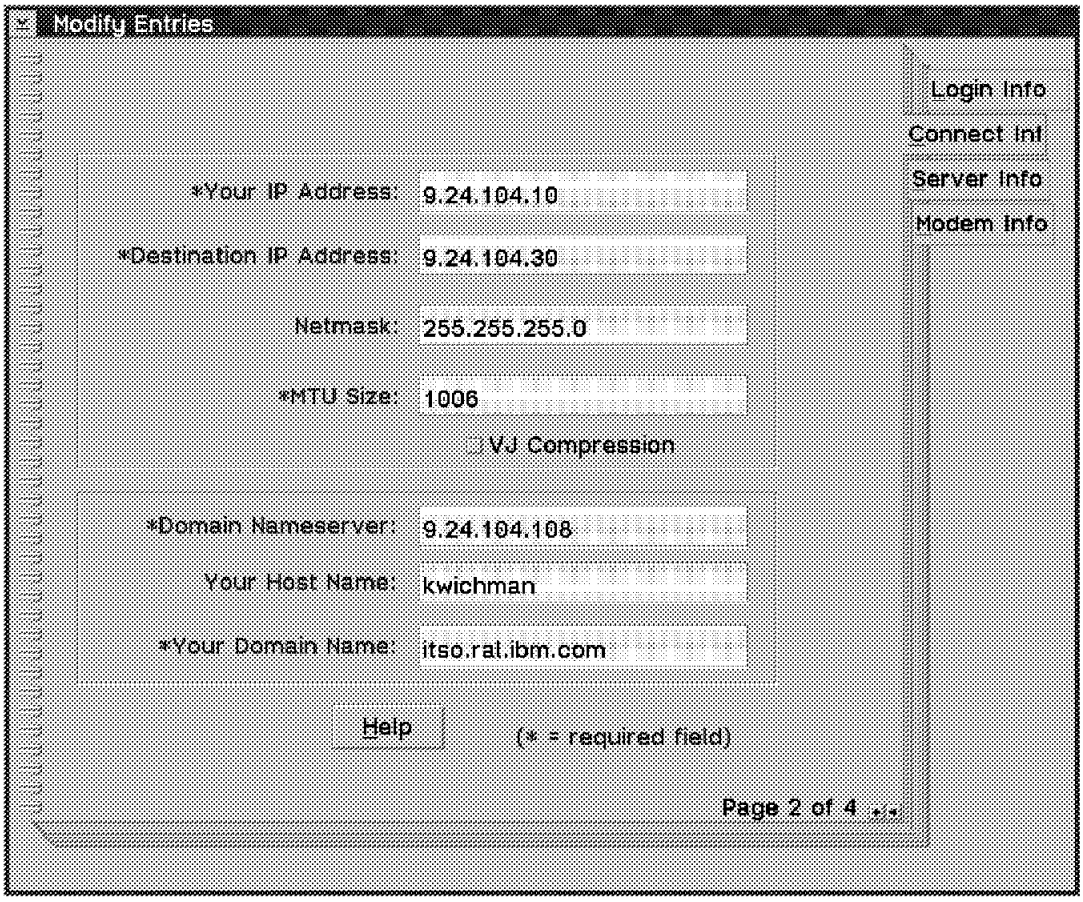


Figure 128. Connect Info Configuration Screen

Fill in the field like explained in the following:

Your IP Address Specify the 32-bit dotted decimal notation Internet Protocol (IP) address assigned to you. If you are using SLIP to access a service provider, this information should be supplied by the provider.

Destination IP Address Specify the IP address of the destination host to which you want to connect.

Netmask Specify the network mask (subnet) used to indicate which portion of your IP address represents the network address and which represents the host address.

MRU Size For SLIP, you specify the maximum transmission unit (MTU) size.

For PPP, you specify the maximum response unit (MRU) size.

Specify the MTU or MRU that your connection can handle. This is the largest possible unit of data that can be sent on a given medium in a single frame.

If you are using SLIP, the default is 1006. If you are using PPP, the default is 1500. Valid values range up to 1500.

Domain Nameserver The IP address of your domain name server.

Your hostname The name of your machine.

Your Domain Name Specify the name of the domain in which your computer resides. The domain name includes all subdomains and the root domain separated by periods.

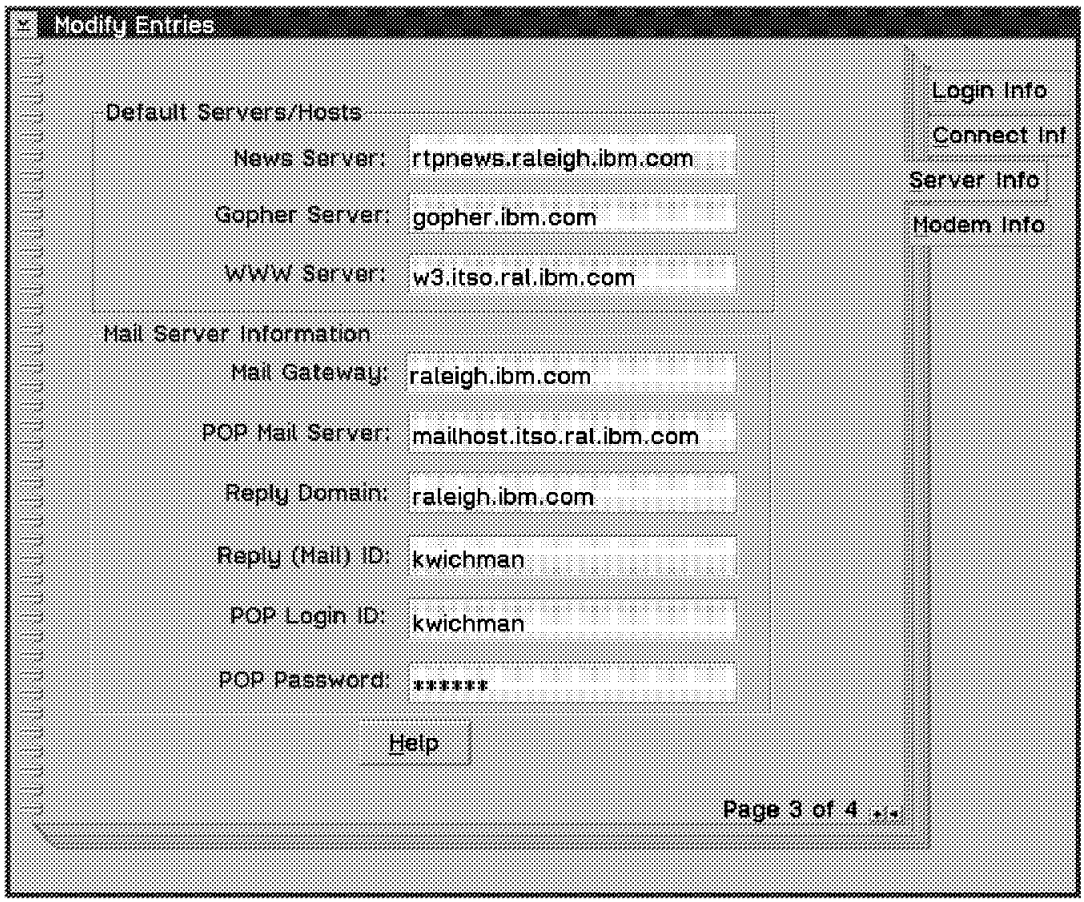


Figure 129. Server Info Configuration Screen

This page defines all servers that you want to access. The Default Server settings are the same as if you configure them in the TCP/IP Configuration. Simply fill in the names of the servers.

Below Mail Server Information you define the fields as you do when you set up UltiMail/Lite in your TCP/IP configuration folder. Please refer to the chapter about electronic mail for more information.

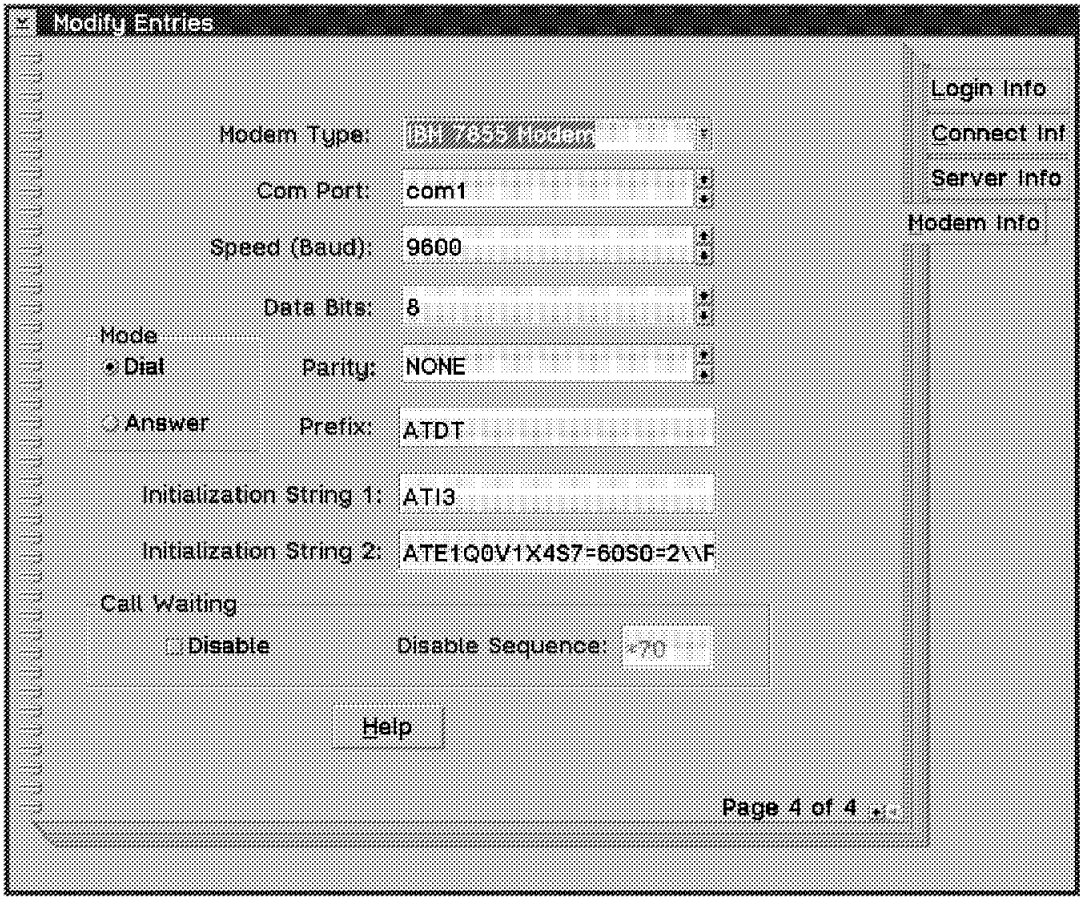


Figure 130. Modem Info Configuration Screen

This last page configures your type of modem. Simply select your modem from the modem list. If your modem is not listed, please refer to the online help for more information.

Fill in the following fields:

Com Port Port to which your modem is connected.

Prefix Fill in ATDT for tone dialing and ATDP for pulse dialing.

Call Waiting If your phone service includes call-waiting, you will want to disable call-waiting while you are using your modem. If you disable call-waiting, you must also specify a Disable sequence.

Close the configuration notebook and save the changes. The name of the service provider will be shown in the list of the IBM Dial-UP for TCP/IP window. You can configure various service providers. To connect to the provider, you double-click on the providers name or you select the service provider and click on **Dial**.

The next time you log on to your service provider, you only need to select the application you want to run (for example, IBM WebExplorer). You will see the screen shown in Figure 131 on page 214. To connect to the Internet using another service provider under Dialer select the **IBM Internet** radio button. Then click on **Connect**.

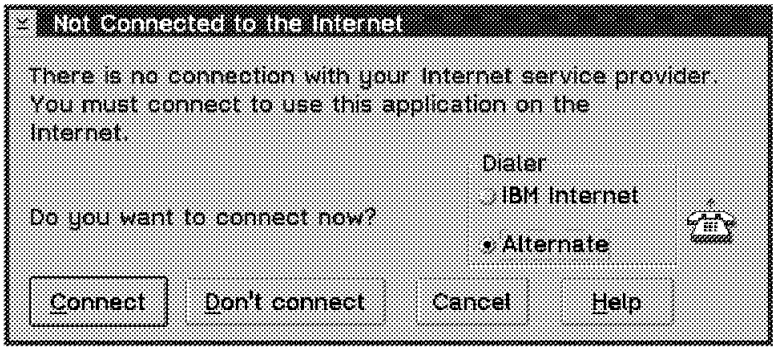


Figure 131. Connect to the Internet, Using Another Service Provider

Once you are connected to your Internet provider, you can use all of the Internet applications explained in this chapter.

7.6 Netcomber

Netcomber is a suite of Internet clients running on OS/2 Warp, designed especially for home and small business users, and for anyone who wants to use the Internet without becoming a technical expert. Netcomber aims to make using the Internet as useful and pleasurable as possible. To this end, it is designed not only with ease of use in mind, but with a high degree of integration among the applications.

Netcomber includes the following clients:

- Read Mail
- Send Mail
- Chat
- News
- Web

At the time of writing (April 1996) it is available on the WWW (IBM internal and external) as a beta-test package.

For the the latest news about Netcomber software releases and updates, new functions, and useful WWW resources see:

`http://www.raleigh.ibm.com/ncr`

7.6.1 System Requirements for Netcomber

- IBM personal computer or 100% compatible Intel(TM) 486(TM)
- 33 MHz processor, 16 MB RAM
- VGA or Super VGA graphics
- 19 MB of available hard disk space
- 14.4 Kbps modem or better
- CD-ROM drive (if installing from CD-ROM), mouse, OS/2 WARP

7.6.2 Starting Netcomber

Because Netcomber uses the dialer, there is nothing to configure in Netcomber. When your dialer application is set up correctly you can start Netcomber by double-clicking on the **Netcomber** icon in the Netcomber icon view. You will get the following screen:

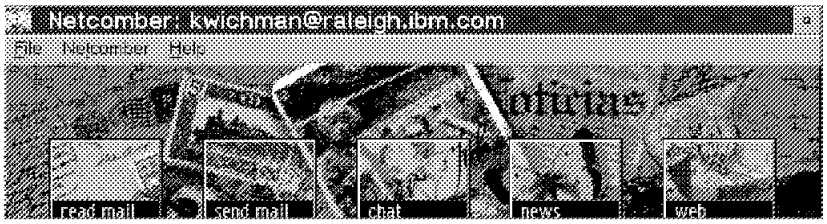


Figure 132. Netcomber Main Screen

From that screen you can start the applications mentioned above. Simply click on the icon representing the application you want to use.

7.6.3 Sending Mail

If you click on the **send mail** icon you will get the following screen to write and send a letter.

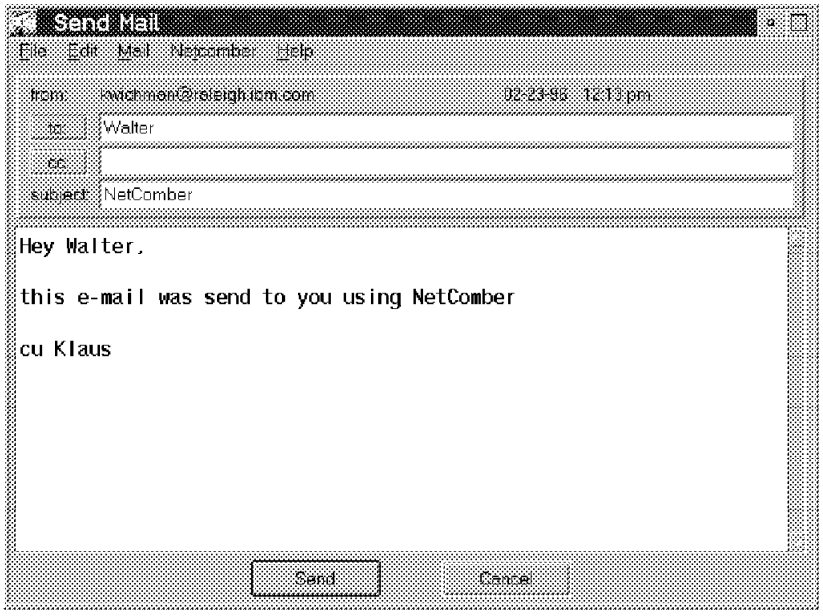


Figure 133. Netcomber Send Mail

Enter the E-mail address in the to: field, type a subject in the subject field and enter your text in the main window. If you want to send a copy to another person, fill in the cc: field. Then click on **Send** to send the letter to that person.

Another way to address a person is to use a name from the nickname list. To edit the nickname list select **Nicknames** from the Netcomber menu of your send mail application.

To add a new nickname click on **Add New** and fill in the information requested.

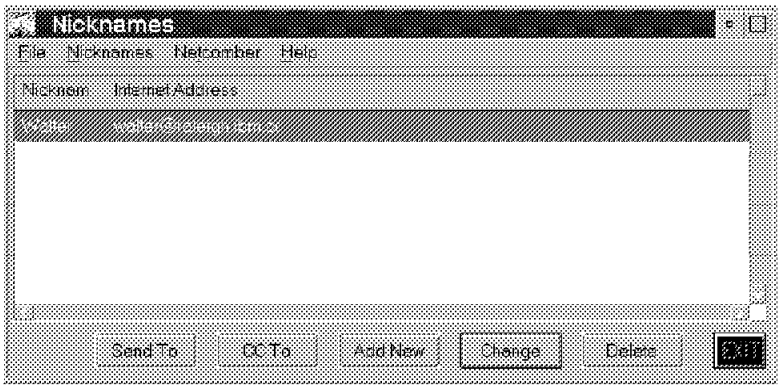


Figure 134. Netcomber Nickname List

You can then use the nickname instead of the E-mail address to address a person.

7.6.4 Reading Mail

To read your received mail you click on the **read mail** icon in NetComber. You will get a list of received mail.

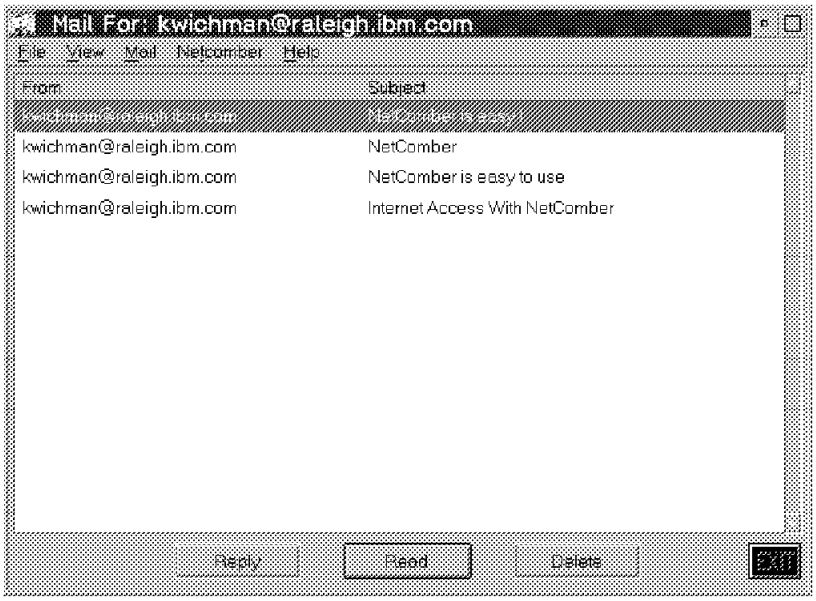


Figure 135. Netcomber Main Screen

Double-click on the mail you want to read. A window which looks like the send mail window will open and show your mail. You can reply to a message by clicking on **Reply**. A send mail window will open for you to write your reply.

7.6.5 Chat

Chat is an application that lets you talk to a group of users. In contrast to the talk command, you can talk to more than one person at the same time. Therefore you have to log on to a chat server (also known as an irc server) which handles the conversation between the people logged on.

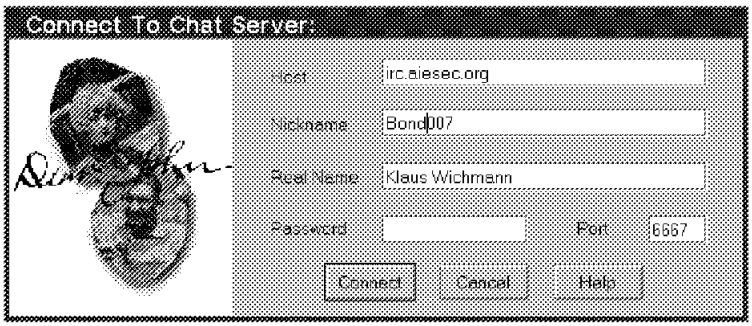


Figure 136. Netcomber Chat Logon

To log on to a server you have to specify the address of the server. In the Nickname field you enter a name that you want to use in the conversation. This name will appear in brackets when you type in a comment. The Real Name field contains your real name and can be seen by other users if they query your nickname. Enter the password for the chat server if one is required.

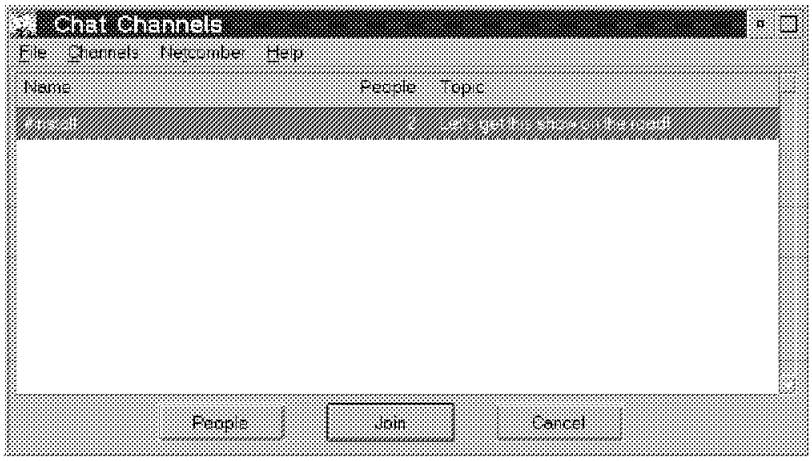


Figure 137. Netcomber Chat

Before you can start your discussion you have to join a channel or open a new channel. Click on **Join** if you want to open a new channel or if you know the name of an existing channel on that server. Type the name of the channel and click on **OK**. You are then on that channel. Another way to join a channel is to select **List Available Channels** from the chat menu. You will get a list of channels where you can choose the channel you want to join.

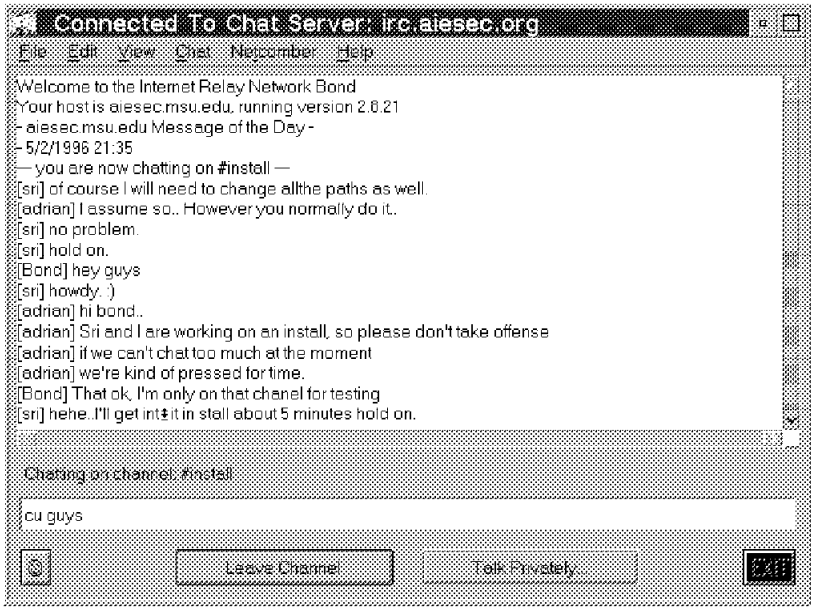


Figure 138. Netcomber Main Screen

Once you join a channel, you can type in your comments at the bottom line. You will see your comments and all comments from other persons on that channel in the above window. If you want to talk privately to a person on that channel, simply click the **Talk Privately** push button. A new window for a private conversation will come up.

7.6.6 Web

Another application that is supported by Netcomber is surfing the Web with the Netcomber Web browser. When you click on the **Web** icon, Netcomber will establish a connection to the Web server which is configured in your dialer.

Figure 139. Netcomber Web Browser

The Netcomber Web browser functions like many other Web browsers. You click on the hyperlinks to change Web pages. When you choose **Go To...** from the Web menu, you can enter an address for another Web server.

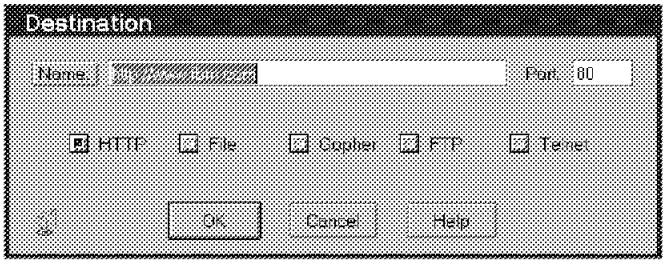


Figure 140. Connecting to a Web Server

Chapter 8. Remote Logon

This chapter describes the facilities available with the TCP/IP for OS/2 product to remotely log on to and emulate terminals on other systems on the network.

TCP/IP for OS/2 features both a Telnet server to allow other systems to access OS/2, and Telnet clients to access other systems from OS/2.

This chapter introduces the integration of Telnet clients into the OS/2 Workplace Shell, and it also discusses some of the customization requirements for the Telnet server.

8.1 TCP/IP for OS/2 Telnet Server

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. Usually, the installation program takes care of this.

The following are the DLL files:

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

To use the TCP/IP for OS/2 Telnet server, you must configure a password for a Telnet client user to specify during logon. If a password is not defined, the Telnet client user has no access to the server.

The Telnet server uses the environment variable TELNET.PASSWORD.ID as the logon password for connecting Telnet clients. However, this environment variable should not be specified in the CONFIG.SYS file because each user who has access to your CONFIG.SYS can remotely log in to your Telnet server. There are no restrictions on what a remote client can do after logging on to your workstation. One way to store this variable is to set it in a .CMD file which starts the Telnet server.

For example, the following TELNET.CMD file starts the Telnet server and sets the TELNET.PASSWORD.ID variable to the password ″secret″:

```
SET TELNET.PASSWORD.ID=secret
TELNETD -l > TELNETD.LOG
```

The -l parameter tells the Telnet server to log all client connections in the TELNETD.LOG file.

You can also use the INETD server of TCP/IP for OS/2 to start the Telnet server, but when doing so you cannot define start parameters for the Telnet server.

The following screen shows the OS/2 Telnet server:

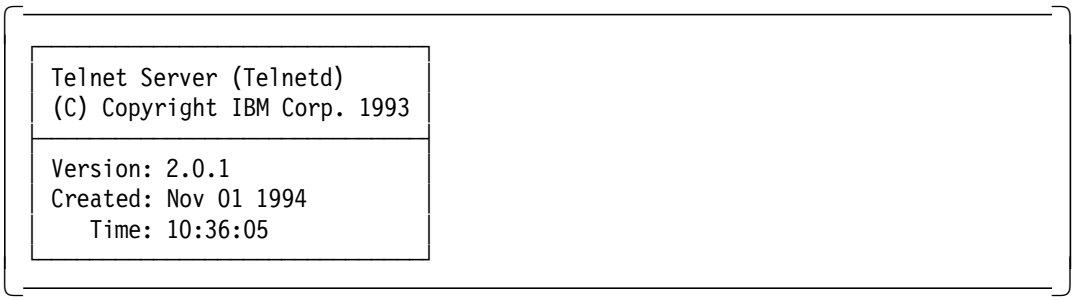


Figure 141. OS/2 Telnet Server

For a more detailed discussion of Telnet server parameters see the provided online documentation.

Another way to handle remote logins to your OS/2 workstation is to use the LOGINUNX.EXE program. Normally, the Telnet server calls LOGIN.EXE to prompt remote users for the password specified in the TELNET.PASSWORD.ID variable. You need to rename LOGINUNX.EXE to LOGIN.EXE in order to use it (don't forget to save the original LOGIN.EXE first). That will handle remote login requests similar to a UNIX system by prompting a user for user ID and password before granting access to your OS/2 system.

Using LOGINUNX.EXE does not require the TELNET.PASSWORD.ID environment variable at all, but it does require a PASSWD file in the MPTNETC directory. This file will hold the user IDs and encrypted passwords, so that no user can see somebody else's password.

Note: Since TCP/IP for OS/2 does not provide a tool to create or maintain a PASSWD file, you need to copy an existing one from a UNIX system (where it is located in the /etc directory).

From an IBM AIX system you need both the /etc/passwd and the /etc/security/passwd (holds the encrypted passwords) files to build the TCP/IP for OS/2 PASSWD file.

A PASSWD file on an AIX system may look like this:

```
root:!:0:0::/:/bin/ksh
daemon:!:1:1::/etc:
bin:!:2:2::/bin:
sys:!:3:3::/usr/sys:
adm:!:4:4::/usr/adm:
uucp:!:5:5::/usr/lib/uucp:
guest:!:100:100::/usr/guest:
nobody:!:4294967294:4294967294::/:
lpd:!:104:9::/:
martin:!:202:1:Martin Murhammer:/u/martin:/bin/ksh
miers:!:203:1:Scott Miers:/u/miers:/bin/ksh
```

For LOGINUNX.EXE to handle logons, the following PASSWD file will do:

```
root:Zw35afu3PrUu6:
```

It will allow user root to log on to the OS/2 Telnet server, with his or her unique password. All other parameters of the UNIX style PASSWD file will be ignored by LOGINUNX.EXE.

To log out from the Telnet server you must type `exit` at an OS/2 command prompt. The `logout` command found in some earlier of TCP/IP for OS/2 no longer exists.

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 are not visible when executed remotely, nor are DOS or Win/OS2 applications. The rule of thumb is that any application other than OS/2 full-screen is likely to hang the Telnet session, or worse, it may hang your system as well as the Telnet server. Even OS/2 full-screen applications may cause problems on some Telnet clients if they use colors, so be careful when accessing an OS/2 Telnet server.

8.1.1 Logon from UNIX Workstations

From AIX you will normally log on in *line* mode with the command:

```
tn host_name
```

or

```
tn IP_address
```

In the following example a command similar to above - `tn 9.24.104.77` - has already been entered. As a result you will get the following screen where the Telnet password must be entered.

Note: This example shows a login to the OS/2 Telnet server using the LOGINUNX.EXE program.

```
OS/2 Version 2.3 (walter)
login:walter
password:
```

Figure 142. Login to OS/2 Telnet Server

If the HOSTNAME environment variable in OS/2 is set, for example, to walter, you will get a command prompt in the format shown on the following screen.

```
OS/2 Version 2.3 (walter)
login:walter
password:
0

[<walter>-C:\]
```

Figure 143. Command Prompt after Login to OS/2 Telnet Server

You can change this prompt, or even create your personal greeting banner within the TELNETD.CMD file in the TCPIPBIN directory. However, you then have to start the Telnet server using that command file.

To log out and close the session with the OS/2 Telnet server you should enter the command exit.

8.1.2 Logon from 3270 Workstations

If the Telnet server is started on an OS/2 workstation, a user at an MVS or VM connected 3270 terminal can log on to OS/2 in line mode.

To do so, enter:

```
telnet
```

on a CMS or TSO command line and specify the IP address of the OS/2 host that you want to be connected to.

The following shows an example of running the OS/2 DIR *.sys command from a VM Telnet client:

```
OS/2 Version 2.3 (walter)
Enter your password:......



[<walter>-C:\]dir *.sys

The volume label in drive C is OS2.
The volume Serial Number is E720:9C14
Directory of C:\

1-23-96   6:04p     <DIR>    4238        0   CONFIG.SYS
       1 file(s)              4238 bytes used
                          20581888 bytes free


[<walter>-C:\]




                                                        Telnet
```

Figure 144. Telnet to OS/2 from VM

8.1.3 Logon from 5250 Workstations

If the Telnet server is started on an OS/2 workstation, a user at an AS/400 connected 5250 terminal can log on to OS/2 in line mode.

To do so, enter:

```
telnet
```

on an OS/400 command line and specify the IP address of the OS/2 host that you want to be connected to. You can likewise start a Telnet Session from the Access a Remote System, or the TCP/IP Administration menus of OS/400.

The following shows an example of running the OS/2 DIR T* command from an OS/400 Telnet client:

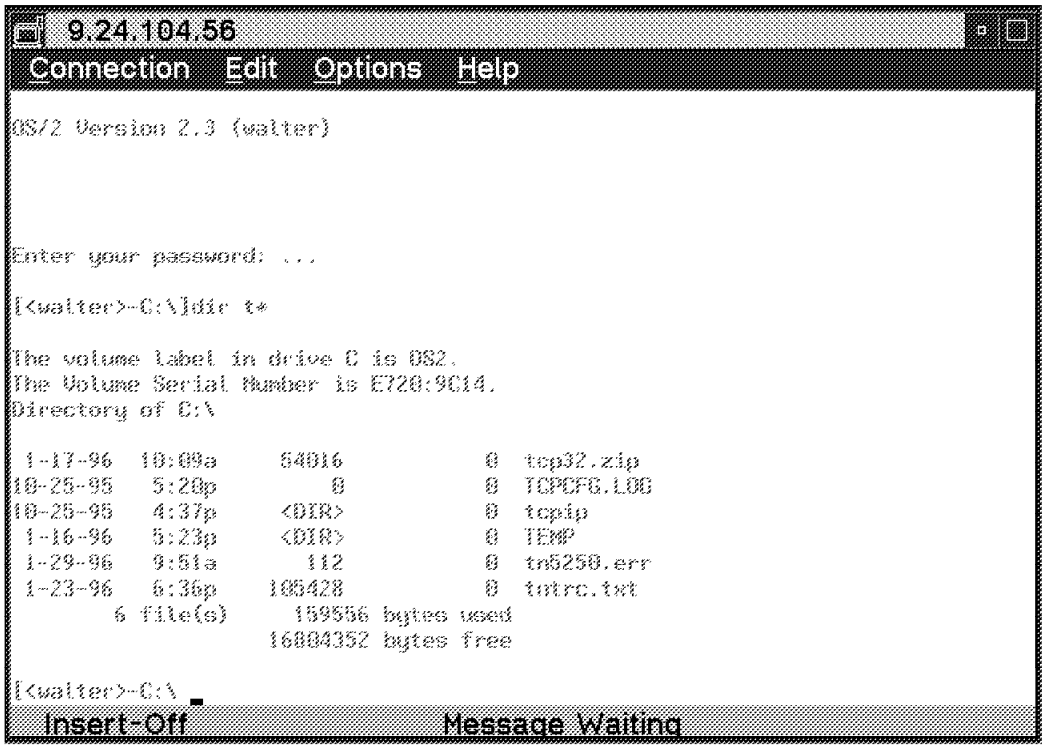


Figure 145. Telnet to OS/2 from OS/400

8.1.4 Logon from DOS Workstations

You can log on to OS/2 in line mode from your DOS workstation using a Telnet client program, for example from the IBM TCP/IP V2.1.1.4 for DOS product. To do this enter the TELNET command on a DOS command prompt, or use the Windows 3.1 user interface for Telnet named WTelnet. WTelnet can be found in the TCP/IP for DOS group in the Windows 3.1 Program Manager window. The DOS Telnet client supports the following terminal emulations:

- VT220
- VT100
- ANSI
- IBM 3278-2

The Windows WTelnet client supports the following terminal emulations in addition to the above:

- TTY
- HFT

Telnet prompts you with the Telnet menu interface, a menu-driven interface that makes it easy to supply the information needed to begin a Telnet session. After you enter the hostname or the IP address of the Telnet server, the hosts negotiate which terminal emulation to use.

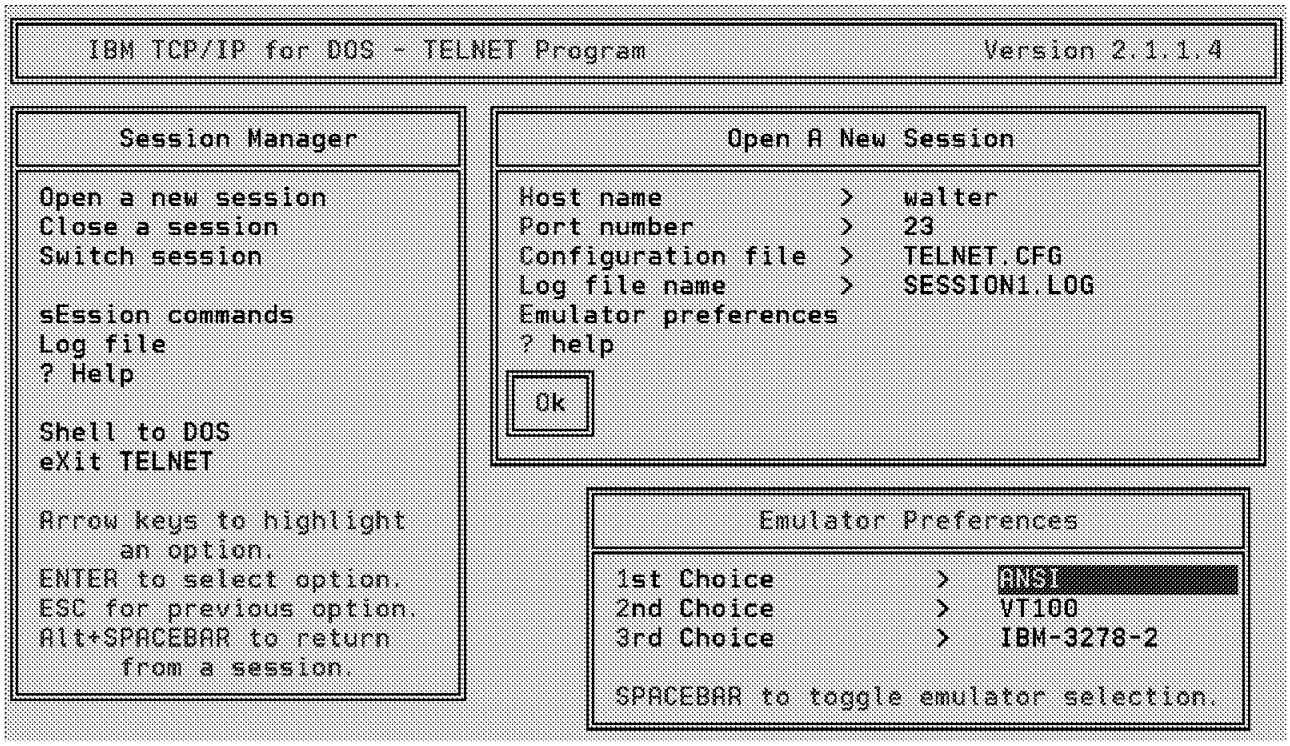


Figure 146. Telnet to OS/2 from DOS

If this negotiation succeeds, and if the HOSTNAME environment variable in OS/2 is set, for example to walter, you will get an OS/2 prompt like the following:

```
[walter:C:]
```

showing the TCP/IP hostname and the current directory.

To close the Telnet session with the OS/2 Telnet server enter exit at the command prompt. You should be careful not to execute a command file that contains an `exit` statement, because that will end the Telnet session.

For more details on the DOS Telnet clients and server, please see the IBM TCP/IP 2.1.1.4 for DOS product documentation.

8.1.5 Logon from OS/2 Workstations

There are several possibilities to remotely log on to another OS/2 system running TCP/IP for OS/2. See 8.2.2, “ASCII-Based Telnet Clients” on page 230 for a more detailed description of OS/2 Telnet clients.

The preferred terminal type between OS/2 systems is ANSI, which gives full-screen support including colors. The remote Telnet session works exactly like a local full-screen OS/2 session (including retrieve command support) with the following exceptions:

- The cursor *shape* is not handled, because it is not supported by ANSI. A full-screen editor that, for example, changes the cursor shape to a block to indicate insert mode will still show an underline cursor, although insert mode will work.

- To quit a terminal session with an OS/2 Telnet server, type `exit` at the command prompt. This will return you to the Presentation Manager window from where you started the Telnet session.
- The mouse is not supported in a remote session (for example, in the IBM OS/2 LAN Server V3.0 full-screen interface program).
- It is not possible to interrupt a program by pressing Ctrl-Break. The command `Ctrl-]` will put you at a Telnet command prompt. If you then enter `send ip`, you are put back in connected mode.

To log on to the remote OS/2 system enter:

```
telnet -t ansi hostname
```

from an OS/2 command prompt, or start a TelnetPM object and select the emulator type to be ANSI.

If the OS/2 environment variable HOSTNAME is set on the remote OS/2 system, the hostname of the remote system appears in the command prompt of your remote session to distinguish it from a local OS/2 session.

On the OS/2 Telnet server you can see logged-on clients on the OS/2 window list as Telnet Session as shown in the following:

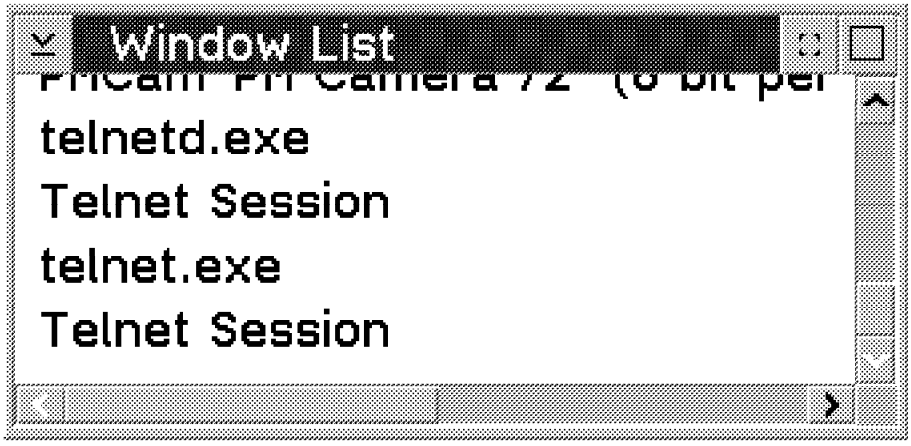


Figure 147. Window List on an OS/2 Telnet Server

To find out who is actually connected into your system, the following command is helpful:

```
[C:tcpipbin]netstat -s
SOCK      TYPE         FOREIGN          LOCAL         FOREIGN      STATE
                         PORT            PORT           HOST
==== ==========     ==========     ==========     ========== ==========
15       STREAM           1024      telnet..23      9.24.104.77  ESTABLISHED
14       STREAM      telnet..23           1024      9.24.104.77  ESTABLISHED
12       STREAM           1024      telnet..23     9.24.104.164  ESTABLISHED
11       STREAM              0      shell..514          0.0.0.0  LISTEN
10       STREAM              0       exec..512          0.0.0.0  LISTEN
9        STREAM              0          ftp..21         0.0.0.0  LISTEN
8        STREAM              0      telnet..23          0.0.0.0  LISTEN

[C:\tcpip\bin]
```

This command lists the sockets that are in use in your TCP/IP host, and what applications are using it. A name with a number in the local port column means a server on your workstation. A number only in the local port column means a client on your workstation. In this example, host 9.24.104.77 made a Telnet session to his or her own IP address, so it shows a client and server task for Telnet in addition to the listening Telnet server that is waiting for more connections.

8.1.6 Logon from DEC/VMS

The following example shows a Telnet session from a DEC MicroVAX system to the OS/2 TCP/IP host cidserver. The DEC system runs the VMS operating system which has no TCP/IP capability by itself, so the MultiNet application was installed on top of VMS.

Therefore, TCP/IP-related commands must be prefixed by the `multinet` statement:

```
$ multinet telnet 9.24.104.77
Trying... Connected.



OS/2 Version 2.3 (walter)
login:walter
password:
0

[<walter>-C:\]dir *.

 The volume label in drive C is C_DRIVE.
 The Volume Serial Number is E641:C414
 Directory of C:\

 7-14-93   1:53p     <DIR>           0  .
 7-14-93   1:53p     <DIR>           0  ..
 8-23-93   4:22p     <DIR>           0  $cidtmp$
 7-20-93   5:45p     <DIR>           0  CID
 7-14-93   2:37p     <DIR>        1317  Desktop
 7-14-93   2:33p     <DIR>           0  IBMCOM
 7-19-93  10:15a     <DIR>           0  LANLK
 7-14-93   2:52p     <DIR>           0  MUGLIB
 8-05-93   3:25p     <DIR>           0  nbtcp
 7-14-93   2:37p     <DIR>         252  Nowhere
 7-14-93   2:03p     <DIR>           0  OS2
 7-14-93   2:03p     <DIR>           0  PSFONTS
 5-11-93   8:10p     40415           0  README
 7-20-93   5:48p     <DIR>           0  SERVER
 7-14-93   2:37p     <DIR>           0  SPOOL
 7-14-93   2:34p     <DIR>           0  srvifsrq
        16 file(s)      40425 bytes used
                      8994816 bytes free

[<walter>-C:\]
```

8.2 TCP/IP for OS/2 Telnet Clients

Telnet clients give you the option to log on to and emulate terminals on remote systems. TCP/IP for OS/2 includes the following Telnet clients:

TelnetPM ASCII-based Telnet client running under Presentation Manager, as an OS/2 window, or as an OS/2 full-screen application.

Telnet ASCII-based Telnet client running as an OS/2 window or OS/2 full-screen application.

Telneto ASCII-based Telnet client running in true line mode as an OS/2 window or OS/2 full-screen application.

3270 Telnet 3270-based Telnet client running as a Presentation Manager application.

TN3270 3270-based Telnet client running as an OS/2 window or OS/2 full-screen application.

Telnet 5250 5250-based Telnet client running as a Presentation Manager application.

The ASCII-based clients can emulate one of the following types of terminals:

- VT220
- VT100
- ANSI
- HFT
- NVT

8.2.1 Workplace Shell Integration of Telnet Clients

Two of the Telnet clients that come with TCP/IP for OS/2, TelnetPM and 3270 Telnet, are implemented as objects of the OS/2 Workplace Shell. They can be found in the Templates folder on the OS/2 Desktop after TCP/IP for OS/2 has been installed.

The following shows part of a Templates folder containing the Workplace Shell objects created by TCP/IP for OS/2:

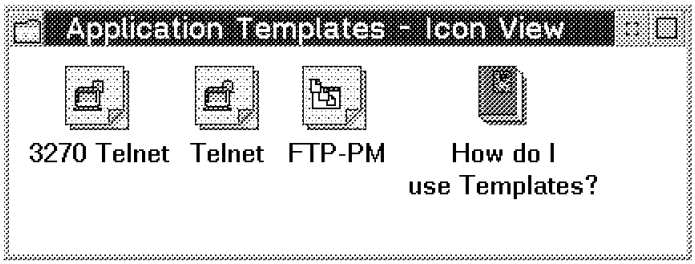


Figure 148. Templates Folder with TCP/IP Objects

You can create multiple instances of these templates by dragging a template object onto the OS/2 Desktop using the right mouse button.

To configure an instance of TelnetPM, see 8.2.3, “Configure TelnetPM” on page 231.

To configure an instance of 3270 Telnet, see 8.2.7, “Configure 3270 Telnet” on page 238.

8.2.2 ASCII-Based Telnet Clients

The TCP/IP for OS/2 ASCII-based Telnet clients are tailored to support specific terminal types. You have the option of selecting a client that best suits your communication needs.

All of the ASCII-based clients will negotiate which emulation to use in the following order:

- VT220
- VT100
- ANSI
- HFT
- NVT

If the server you want to communicate with expects another terminal type, you must set the TERM variable to the string the server is expecting. For example, you can use the following command to communicate with a SUN host that expects the terminal type name VT100-sun:

```
set TERM=vt100-sun
```

You can have several Telnet connections running concurrently to different Telnet servers. To do this, start each Telnet client at an OS/2 command prompt, or start multiple instances of TelnetPM.

The following screen shows an OS/2 Telnet client logging on to a DEC/VMS system running MultiNet:

```
******** ITSO Open Network Management Lab ********

Username: SYSTEM
Password:

    Last interactive login on Wednesday, 28-FEB-1996 10:17
    Last non-interactive login on Tuesday, 27-FEB-1996 16:55
           International Technical Support Laboratory
                            Raleigh
                             ITCSV1

%DCL-E-OPENIN, error opening PNTDIR:[EXE]SIX2SYM.COM; as input
-RMS-F-DEV, error in device name or inappropriate device type for operation
$
$ dir m*

Directory SYS$SYSROOT:[SYSMGR]

MAIL.DIR;1          MAIL.MAI;1          MARTIN.DIR;2        MARTIN.DIR;1
MCC_DNA4_EVL.LOG;14 MCC_NODE4_LOAD.TMP;1
MCC_STARTUP_BMS.LOG;16

Total of 7 files.

Directory SYS$COMMON:[SYSMGR]

MAKEROOT.COM;1      MCC_EXPORTER_BACKGROUND.COM;1
MCC_HISTORIAN_BACKGROUND.COM;1

Total of 3 files.

Grand total of 2 directories, 10 files.
$
```

8.2.3 Configure TelnetPM

To configure an instance of the TelnetPM object, double-click on the object for the first time, or select **Open Settings**. Select the appropriate tag in the notebook-style configuration window and fill in the necessary values for your Telnet session.

If you just want to create a generic TelnetPM instance in order not to overpopulate your desktop with objects, leave all fields blank. Only click on the **Create New Window** radio button on the Window menu. You can then enter all communication parameters after starting TelnetPM.

The following figures show the first two pages of the TelnetPM configuration notebook:

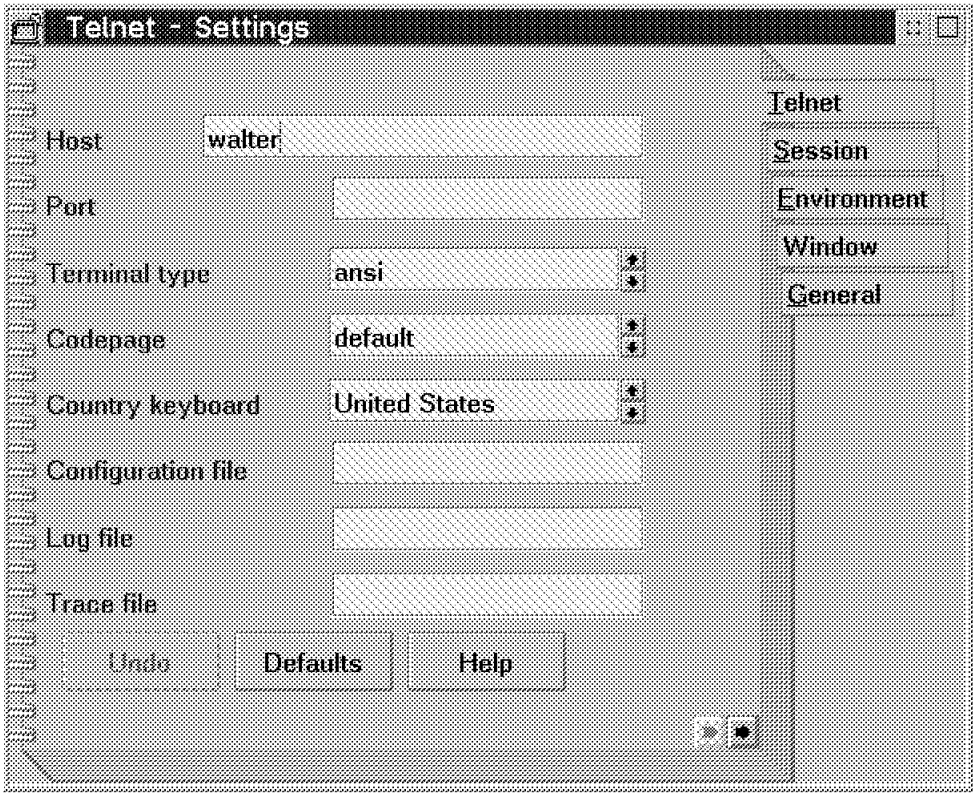


Figure 149. Telnet Page of the TelnetPM Configuration Notebook

On the Telnet page you specify necessary communications parameters.

Setting	Meaning
Host	Hostname or IP address of the Telnet server you wish to communicate with.
Port	Port number or port name to use for this session. If you leave this field blank (recommended), the Telnet port as specified in the SERVICE file in the TCPIPETC directory will be used.
Terminal Type	Can be one of the following: • VT100 • VT220 • ANSI • HFT • NVT • Default
Code page	Used, if necessary, to translate international characters.
Country keyboard	Used to emulate national keyboards.
Configuration file	Used to specify a keyboard remap file. See 8.2.10, “Keyboard Remap for Telnet Clients” on page 244 about remapping a keyboard for TelnetPM.
Log file	Specify where you want to log a TelnetPM session.
Trace file	Specify where you want to trace a TelnetPM session.

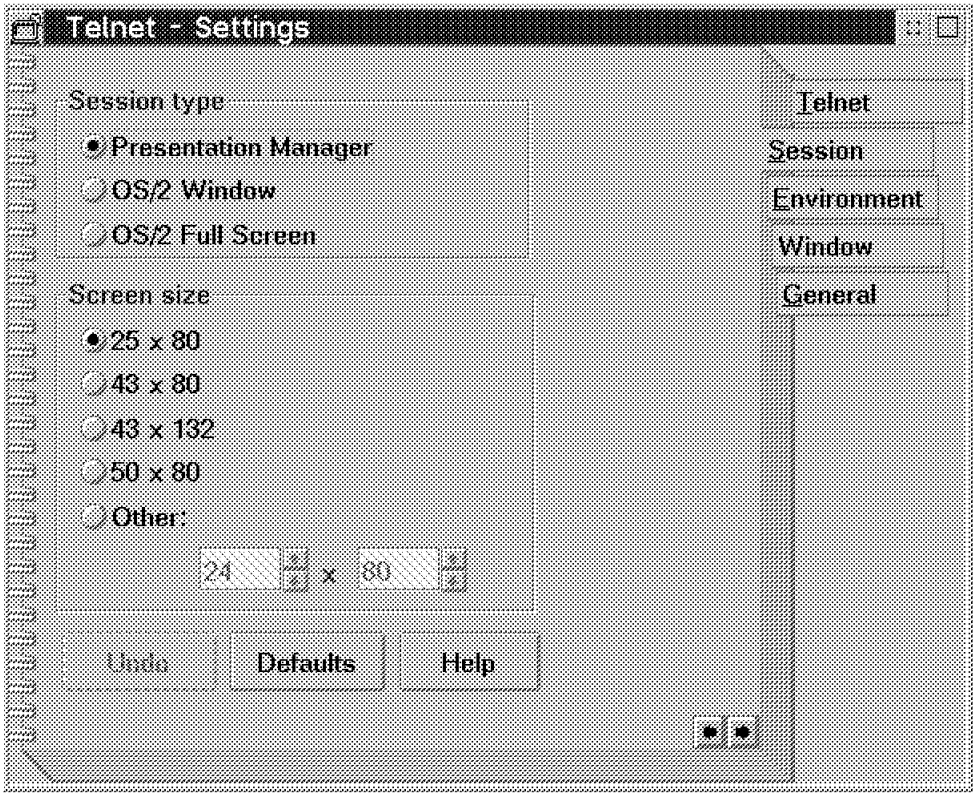


Figure 150. Session Page of the TelnetPM Configuration Notebook

On the Session page you specify how a TelnetPM session is to be represented by OS/2.

Setting	Meaning
Session type	Select the way you want that instance of TelnetPM to appear on your desktop.
Screen Size	Select the size of the screen for that instance of TelnetPM in number of rows by number of columns.

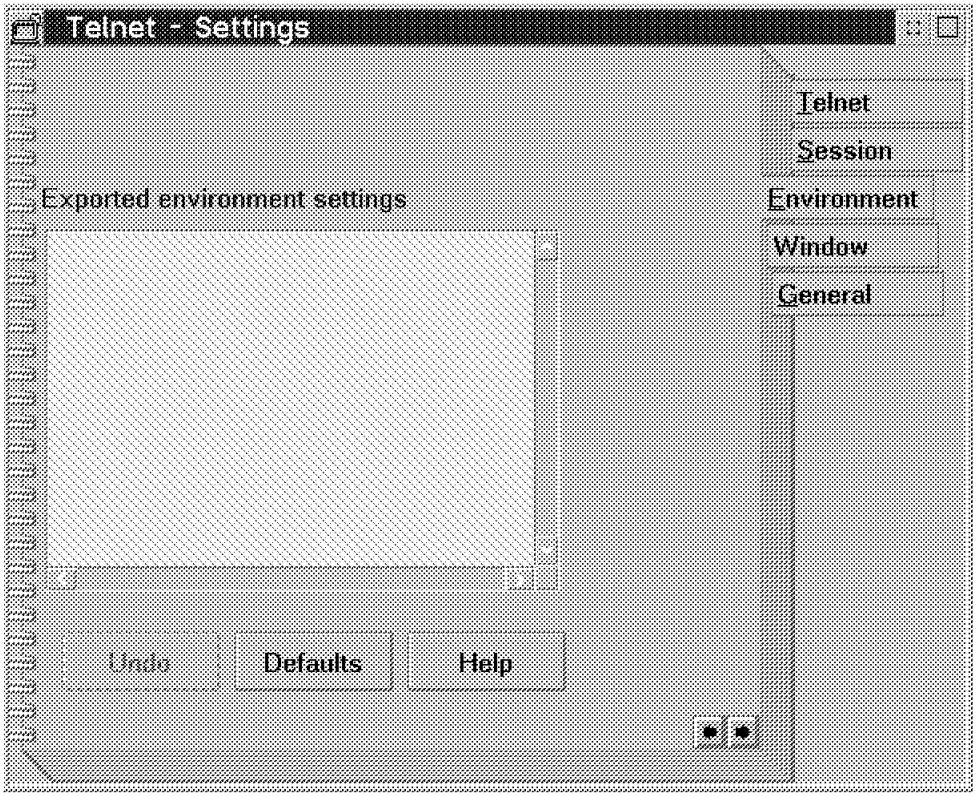


Figure 151. Environment Page of the TelnetPM Configuration Notebook

On the Environment page you specify environment variables that you want to be set for this session on the remote system. If a remote system does not support exchanging of environment variables, TelnetPM will prompt you whether or not you would like to continue without exporting any variables.

Some of the settings discussed can be changed once a TelnetPM instance is started. At this time you can also change settings that only control an active session and can therefore not be pre-set.

The following shows an active TelnetPM session with opened Options pull-down menu and the Line Mode option activated:

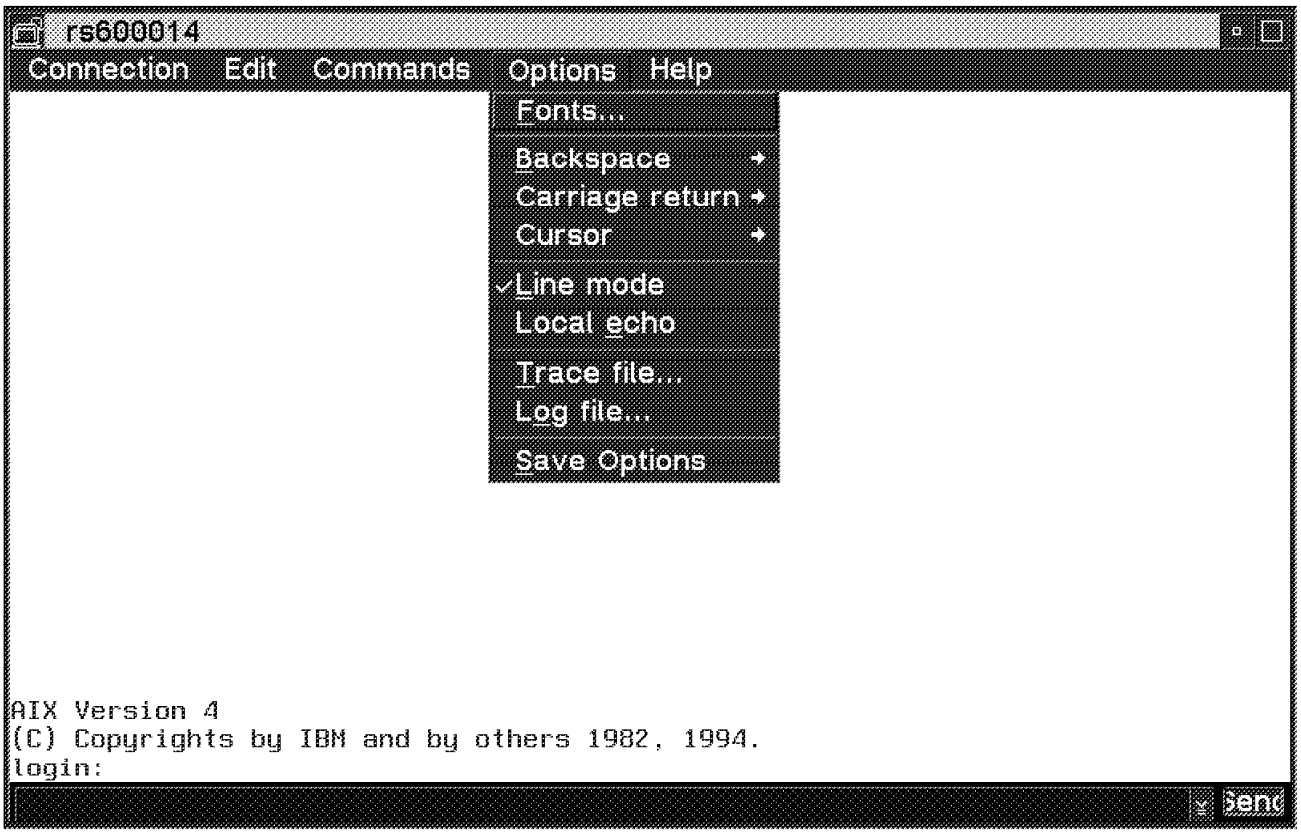


Figure 152. Configuring an Active TelnetPM Session

Menu	Control or Configuration
Connection	Used to start and stop a Telnet session.
Edit	Used to cut and paste data from or to this session.
Commands	Used to send Telnet session commands to the remote host.
Options	Used to control configuration values for this session, and to change values that have been specified in the TelnetPM object settings.

For a more detailed discussion of these settings see the online documentation.

The line mode option gives you the ability to edit a command before entering it, and to retrieve previously entered commands from a list.

8.2.4 Using New TelnetPM Features

With an earlier version of TCP/IP for OS/2, it required some work to get a 3270 emulation through HCON since the HFT terminal type and a 25-line display were not directly supported.

The following screen shows a TelnetPM session to an AIX system, using HFT terminal type, a screen size of 25x80 characters, and emulating a 3270 display using HCON:

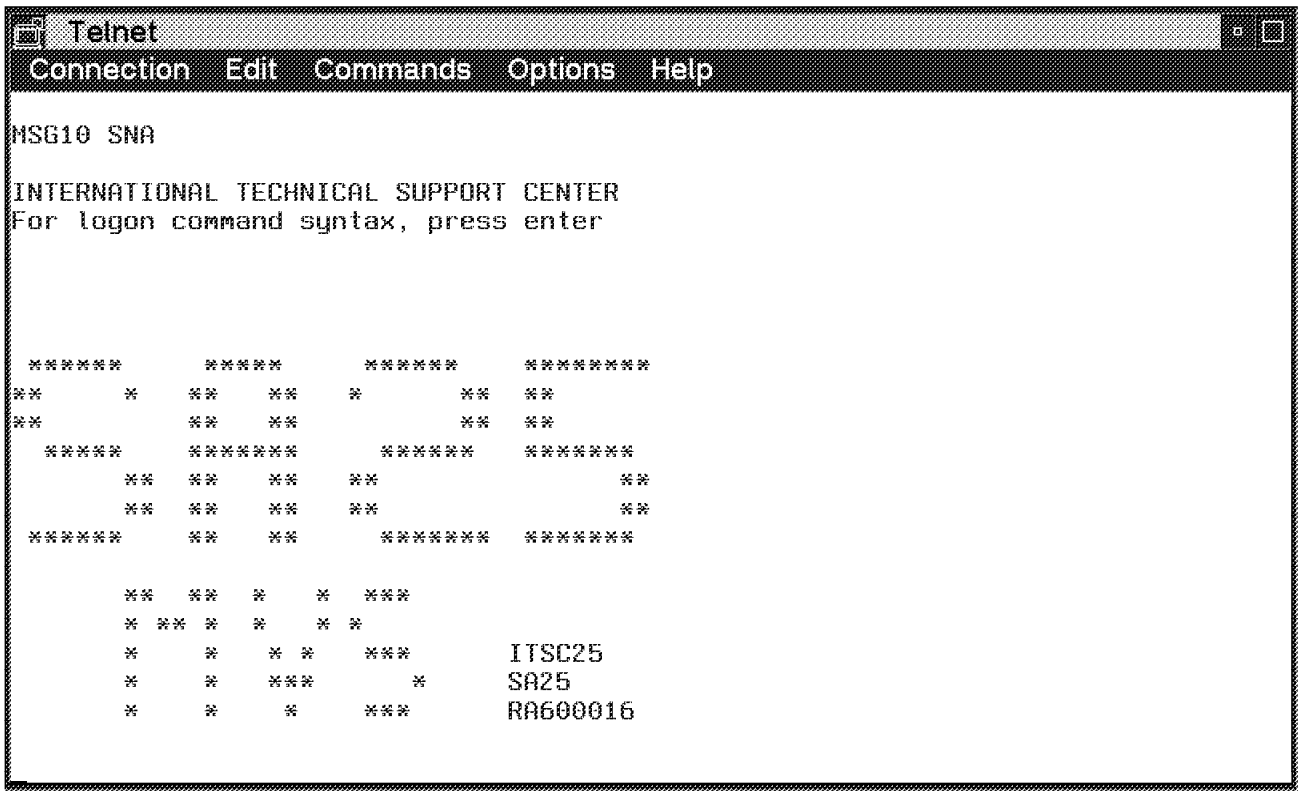


Figure 153. HCON Session with TelnetPM

8.2.5 Telneto - True Line Mode Telnet Client

You can use the Telneto program if you require true line mode for a Telnet session rather than the line mode simulation of TelnetPM. You can start a session directly by entering:

```
Telneto hostname
```

or

```
Telneto IP_address
```

When you start Telneto without the above parameters, you are taken to the Telneto command prompt. From there, you can issue Telneto subcommands such as the `open` command that starts a Telnet session, or the `quit` command that exits from Telneto.

Note: There is no keyboard remap facility for Telneto.

The following screen shows a Telneto session with a SUN SPARCStation:

```
Connected to sun.
Escape character is '¬"'.
SunOS UNIX (SPSUN009)

login: root
Last login: Thu Aug 19 11:57:02 on console
SunOS Release 4.1.3 (GENERIC) #3: Mon Jul 27 16:43:54 PDT 1992



SPSUN009#
```

For a more detailed discussion of the Telneto subcommands see the online documentation.

8.2.6 3270-Based Telnet Clients

3270 Telnet is an OS/2 Presentation Manager application and supports the mouse pointing device, whereas TN3270 is an OS/2 character-based application. We suggest that you use the TN3270 client when utilizing a serial line (SLIP) because it is somewhat faster than the 3270 Telnet client, even though 3270 Telnet offers you more functionality.

To use TN3270 or 3270 Telnet, a Telnet server must be running on a foreign host that supports a 3270 terminal emulator such as provided by the IBM TCP/IP for VM and TCP/IP for MVS programs.

Notes:

1. TN3270 and 3270 Telnet do not support HLLAPI and host graphics using GDDM.
2. There is no IND$FILE file transfer capability for TN3270 and 3270 Telnet.

You will find a brief description of the TN3270 command and the default keyboard mapping in the online documentation.

The following shows the differences between 3270 Telnet and TN3270. 3270 Telnet has a menu bar to allow configuration and control over an active 3270 Telnet session. To get to the TN3270 Main Menu press the Ctrl] keys. This will prompt you with a screen similar to the following:

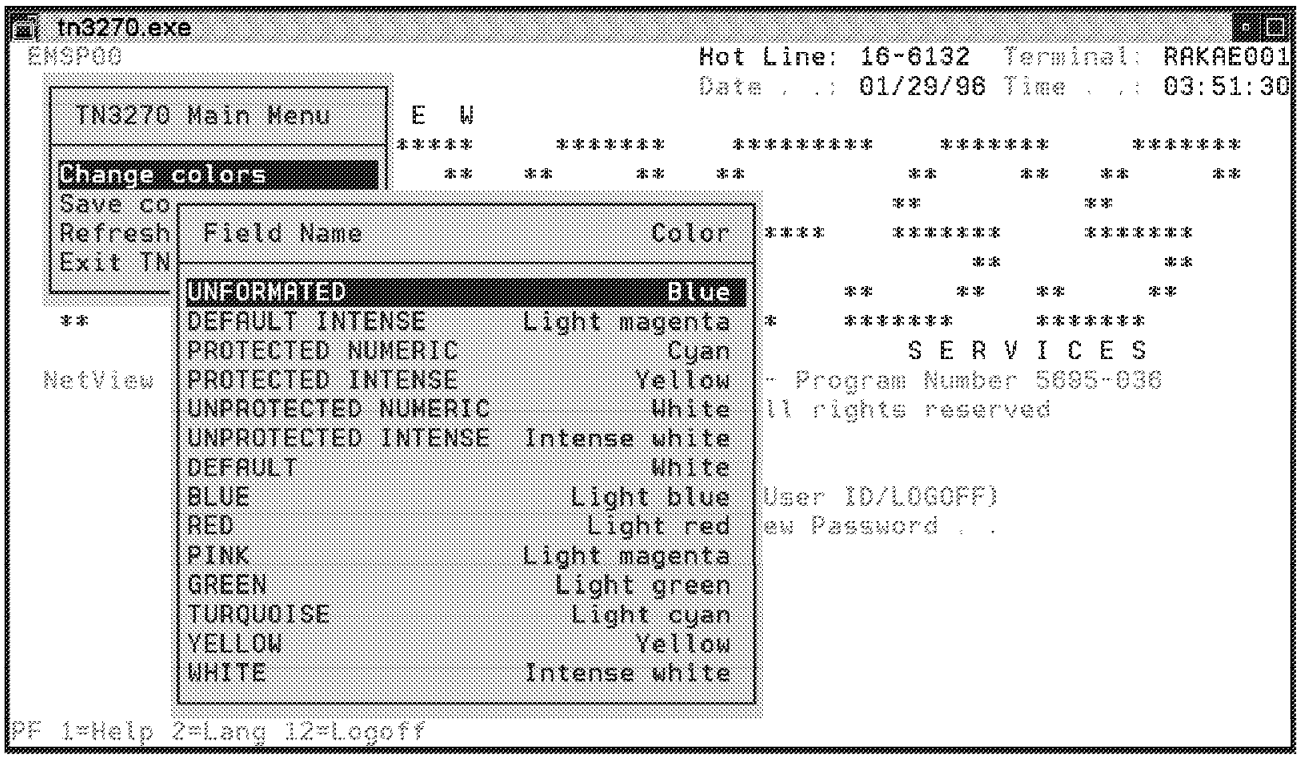


Figure 154. TN3270 Main Menu

8.2.7 Configure 3270 Telnet

To configure an instance of the 3270 Telnet object, double-click on the object for the first time, or select **Open Settings**. Select the appropriate tag in the notebook-style configuration window and fill in the necessary values for your 3270 Telnet session.

If you just want to create a generic 3270 Telnet instance in order not to overpopulate your desktop with objects, leave all fields blank. Only click on the **Create New Window** radio button on the session menu. You can then enter communication parameters after starting 3270 Telnet.

The following figures show the first two pages of the 3270 Telnet configuration notebook:

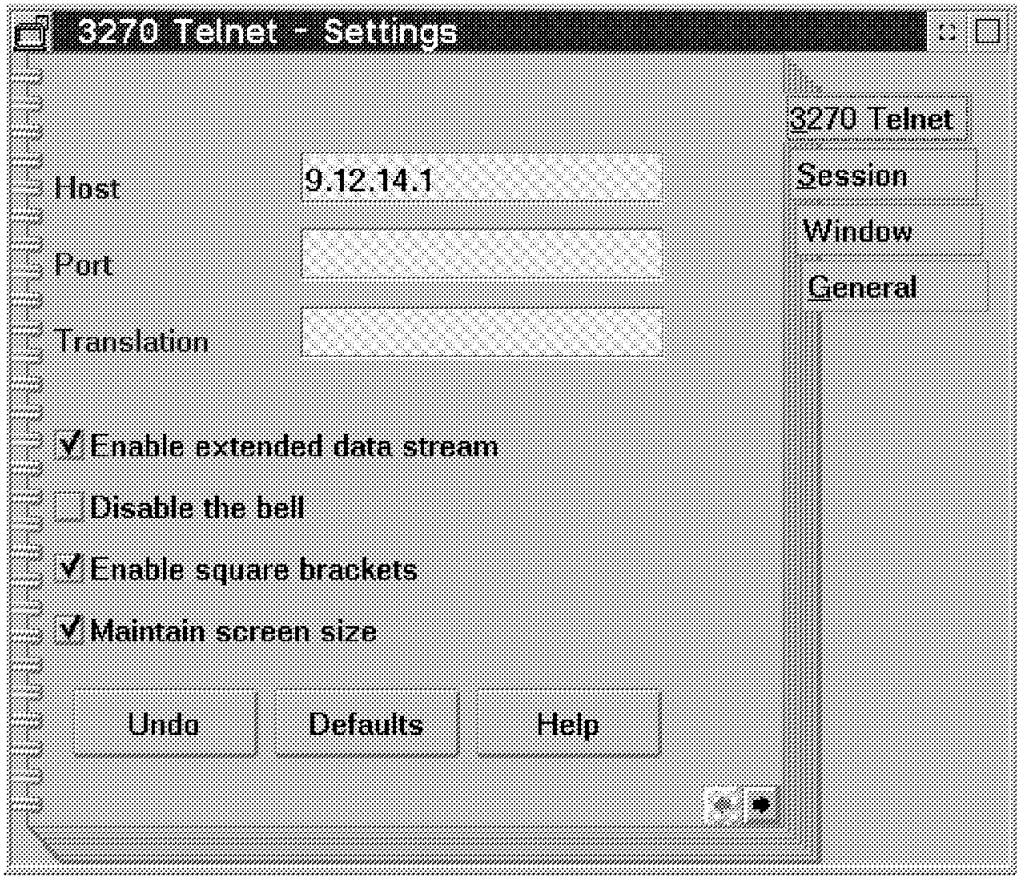


Figure 155. 3270 Telnet Page of the 3270 Telnet Configuration Notebook

On the 3270 Telnet page you specify the following necessary communications parameters:

Setting	Meaning
Host	Hostname or IP address of the 3270 Telnet server you wish to communicate with.
Port	Port number or port name to use for this session. If you leave this field blank (recommended), the Telnet port as specified in the SERVICE file in the MPTNETC directory will be used.
Translation	Specify the file name of the translation table for translating between ASCII and EBCDIC. By default, 3270 Telnet uses the file 3278XLT.TBL, located in the ETC directory, or the default US translation table.
Enable Extended Datastream	Supports extended colors, highlighting, and nonstandard displays.
Disable the Bell	Used to suppress beep signals for host screen refreshes.
Enable Square Brackets	Used to show square brackets and backslashes as they appear on the keyboard.
Maintain Screen Size	Used to prevent the emulator to change the screen size according to the host program.

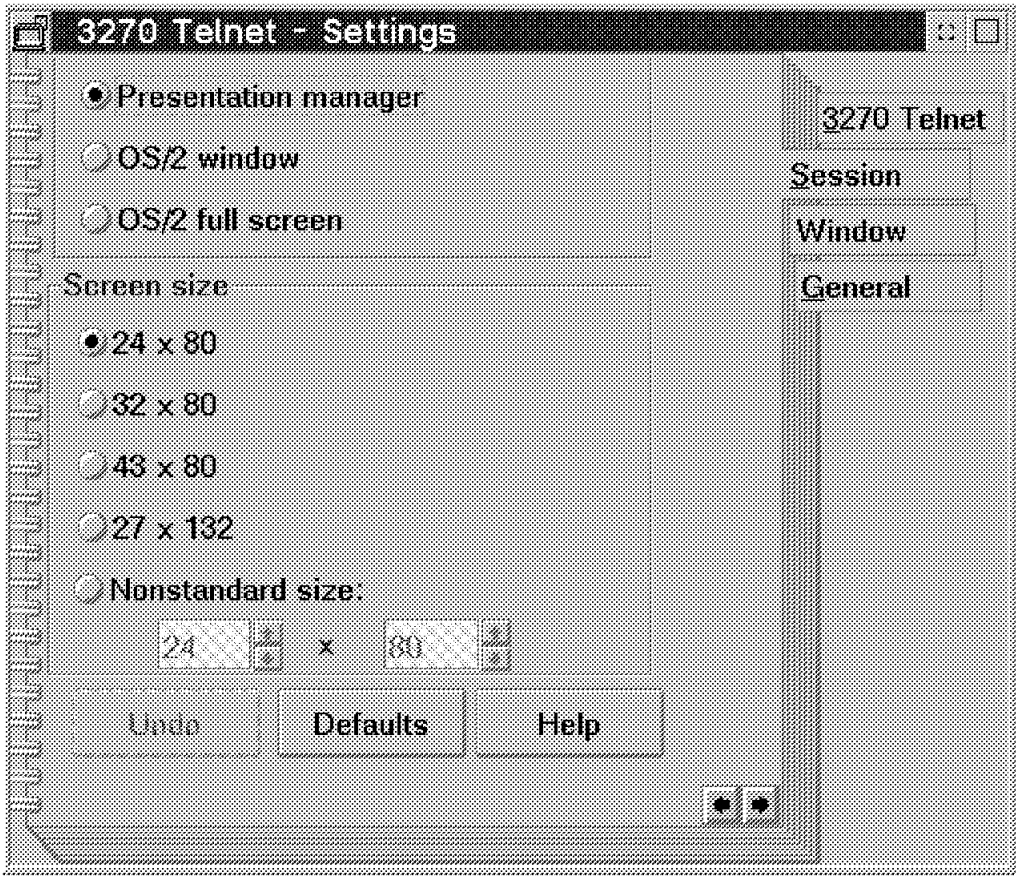


Figure 156. Session Page of the 3270 Telnet Configuration Notebook

On the session page you specify how a 3270 Telnet session is to be represented by OS/2.

Setting	**Meaning**
Session type	Select the way you want that instance of TelnetPM to appear on your desktop.
Screen Size	Select the size of the screen for that instance of TelnetPM in number of rows by number of columns.

Some of the settings discussed above can be changed once a TelnetPM instance is started. At this time you can also change settings that only control an active session and can therefore not be pre-set.

The following shows an active 3270 Telnet session with the opened Configuration pull-down menu:

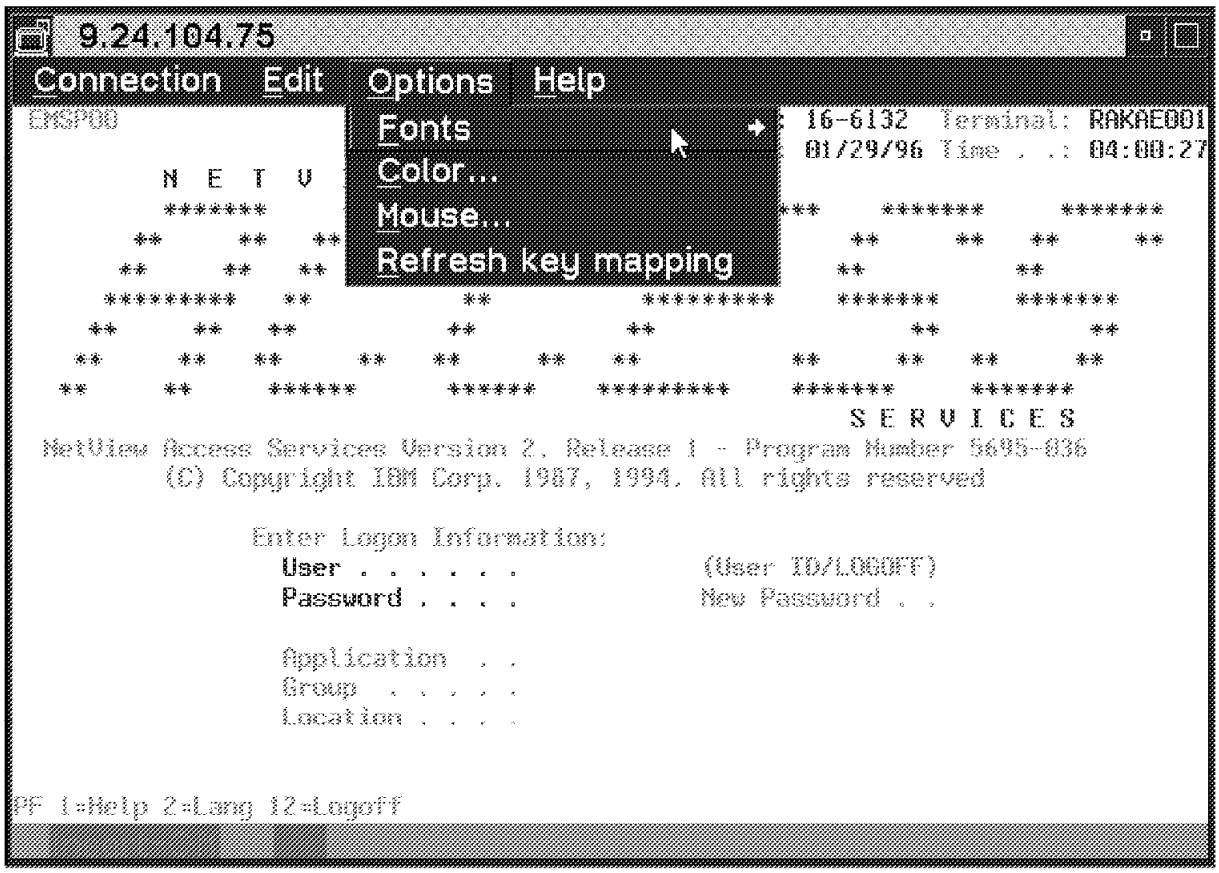


Figure 157. Configuring an Active 3270 Telnet Session

Menu	Control or Configuration
Connection	Used to start or stop a connection.
Edit	Used to cut and copy screen data from/to the OS/2 clipboard.
Options	Used to configure the following emulator settings: • Color settings • Mouse action • Keyboard remap • Fonts

The following shows the color mapping menu of a 3270 Telnet session:

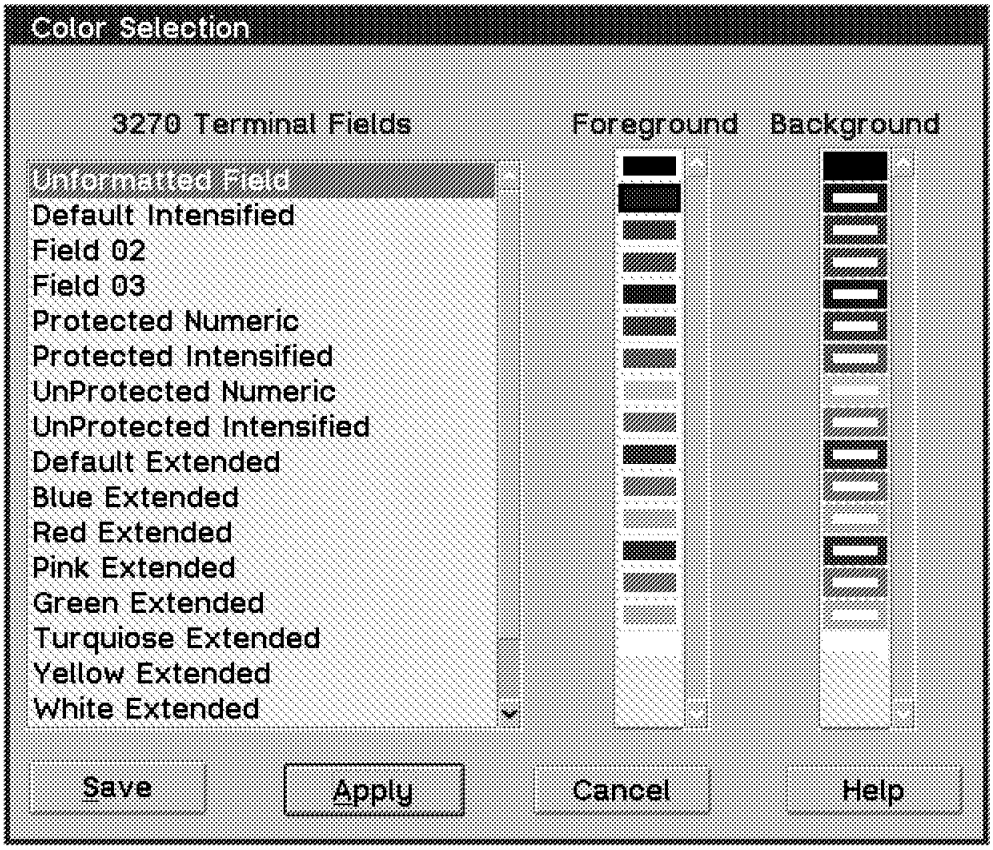


Figure 158. Color Mapping of an Active 3270 Telnet Session

For a more detailed discussion of these settings see the provided online documentation.

8.2.8 5250-based Telnet Clients

This feature allows you to log on to an AS/400 system that is running a Telnet server. You can start TN5250 either from an OS/2 command prompt or from the TCP/IP folder.

Since establishing a Telnet session to an AS/400 may take several seconds, you might get a message that tells you that session establishing is still going on. Answer `Continue` in this message box to allow TN5250 to finish to connect to the remote system.

You can change settings that control an active session once it has been established. The following shows an active TN5250 session with the opened Fonts pull-down menu:

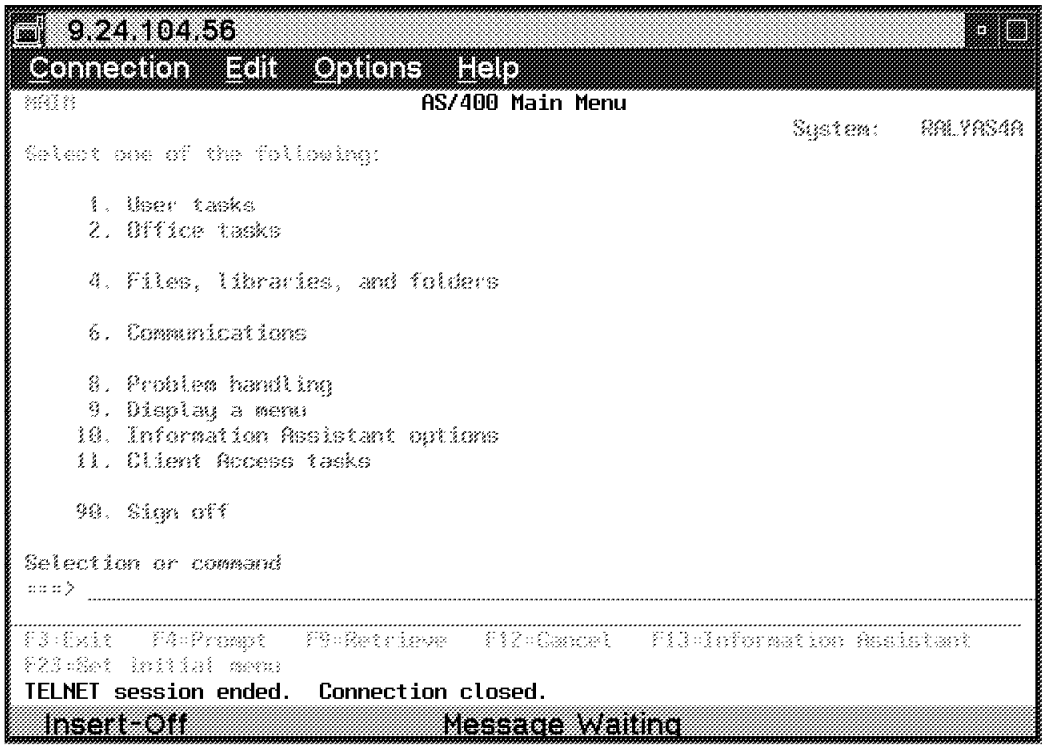


Figure 159. Configuring an Active TN5250 Session

Menu	Control or Configuration
Connection	Used to start and stop an emulator session.
Edit	Used to cut and paste screen data to or from/to the OS/2 clipboard.
Options	Used to configure the following emulator settings: • Mouse action • Keyboard remap • Fonts

For a more detailed discussion of these settings see the provided online documentation.

8.2.9 Mouse Support for 3270 Telnet and TN5250

3270 Telnet and TN5250 both support the use of a mouse to perform predefined actions in an active emulator session. You can map the right and left mouse buttons to one of the 12 PF keys or the Enter key. Mapping can be configured for both single and double-click action.

The following shows the Mouse Configuration window as it shows up from the Option pull-down menu of 3270 Telnet and TN5250:

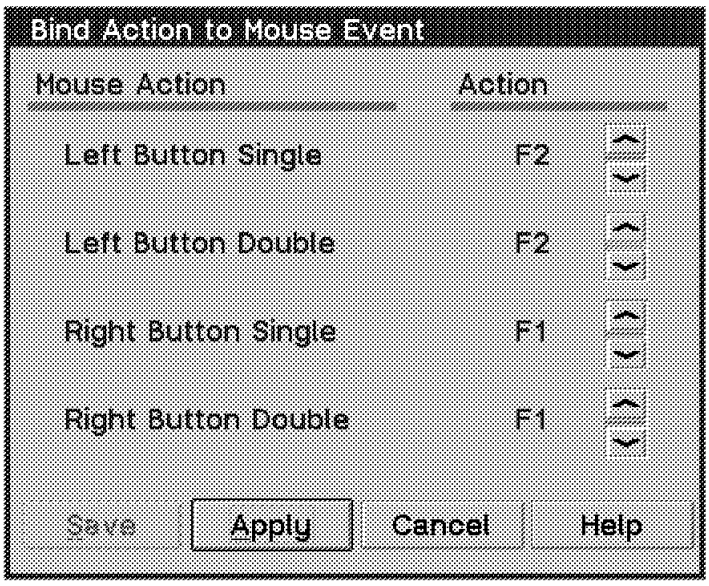


Figure 160. Mouse Configuration for 3270 Telnet and TN5250 Sessions

8.2.10 Keyboard Remap for Telnet Clients

TCP/IP for OS/2 allows you to remap the terminal keyboard for all types of emulated terminals.

8.2.10.1 The Telnet Customization Program

The Telnet Customization program is used to remap the keyboard for the ASCII-based Telnet clients TelnetPM and Telnet, including National Language keyboard layouts. It is a PM application showing the keyboard layout in an OS/2 PM window.

Telnet Customization guides you through the remap process of your terminal session keyboard by displaying an emulator keyboard layout. You can view and change keys interactively for the following types of terminal emulators:

- VT220
- VT100
- ANSI
- HFT

The following shows the remap of a German keyboard for a VT220 terminal:

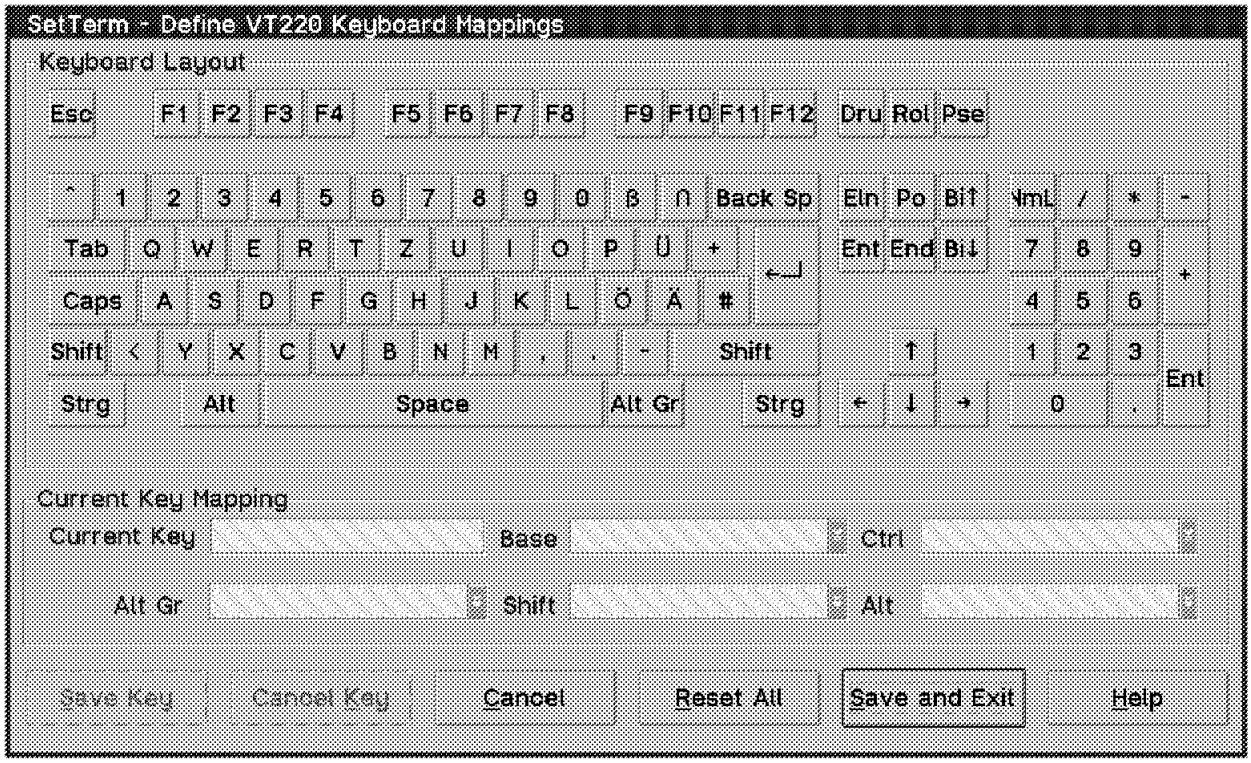


Figure 161. Telnet ASCII Emulator Keyboard Remap

The file created or modified by this utility usually has the extension *.CFG*.

8.2.10.2 3270 Telnet, TN3270, and TN5250 Keyboard Remap

To control the ASCII-to-EBCDIC translation for 3270 Telnet, the file 3278XLT.XLT is used. It is provided as a sample for the US translation table as 3278XLT.SAM, which is shown in the following example:

```
;
; ASCII-to-EBCDIC table for English (US) CECP Code Page 037
; 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
;
  40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40     ; 00 ;
  40 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40     ; 10 ;
  40 5A 7F 7B 5B 6C 50 7D 4D 5D 5C 4E 6B 60 4B 61     ; 20 ;
  F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 7A 5E 4C 7E 6E 6F     ; 30 ;
  7C C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6     ; 40 ;
  D7 D8 D9 E2 E3 E4 E5 E6 E7 E8 E9 BA E0 BB B0 6D     ; 50 ;
  79 81 82 83 84 85 86 87 88 89 91 92 93 94 95 96     ; 60 ;
  97 98 99 A2 A3 A4 A5 A6 A7 A8 A9 C0 4F D0 A1 40     ; 70 ;
  68 DC 51 42 43 44 47 48 52 53 54 57 56 58 63 67     ; 80 ;
  71 9C 9E CB CC CD DB DD DF EC FC 70 B1 80 BF 40     ; 90 ;
  45 55 CE DE 49 69 9A 9B AB AF 5F B8 B7 AA 8A 8B     ; A0 ;
  40 40 40 40 40 65 62 64 B4 40 40 40 40 4A B2 40     ; B0 ;
  40 40 40 40 40 40 46 66 40 40 40 40 40 40 40 9F     ; C0 ;
  8C AC 72 73 74 40 75 76 77 40 40 40 40 6A 78 40     ; D0 ;
  EE 59 EB ED CF EF A0 8E AE FE FB FD 8D AD BC BE     ; E0 ;
  CA 8F 40 B9 B6 B5 E1 9D 90 BD B3 DA FA EA 40 41     ; F0 ;
;
; EBCDIC-to-ASCII table for English (US) CECP Code Page 037
; 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
;
```

```
00 01 02 03 1A 09 1A 1A 1A 1A 1A 0B 0C 0D 0E 0F     ; 00 ;
10 11 12 13 1A 1A 08 1A 18 19 1A 1A 1C 1D 1E 1F     ; 10 ;
1A 1A 1A 1A 1A 0A 17 1B 1A 1A 1A 1A 1A 05 06 07     ; 20 ;
1A 1A 16 1A 1A 1A 1A 04 1A 1A 1A 1A 14 15 1A 1A     ; 30 ;
20 FF 83 84 85 A0 C6 86 87 A4 BD 2E 3C 28 2B 7C     ; 40 ;
26 82 88 89 8A A1 8C 8B 8D E1 21 24 2A 29 3B AA     ; 50 ;
2D 2F B6 8E B7 B5 C7 8F 80 A5 DD 2C 25 5F 3E 3F     ; 60 ;
9B 90 D2 D3 D4 D6 D7 D8 DE 60 3A 23 40 27 3D 22     ; 70 ;
9D 61 62 63 64 65 66 67 68 69 AE AF D0 EC E7 F1     ; 80 ;
F8 6A 6B 6C 6D 6E 6F 70 71 72 A6 A7 91 F7 92 CF     ; 90 ;
E6 7E 73 74 75 76 77 78 79 7A AD A8 D1 ED E8 A9     ; A0 ;
5E 9C BE FA B8 F5 F4 AC AB F3 5B 5D EE F9 EF 9E     ; B0 ;
7B 41 42 43 44 45 46 47 48 49 F0 93 94 95 A2 E4     ; C0 ;
7D 4A 4B 4C 4D 4E 4F 50 51 52 FB 96 81 97 A3 98     ; D0 ;
5C F6 53 54 55 56 57 58 59 5A FD E2 99 E3 E0 E5     ; E0 ;
30 31 32 33 34 35 36 37 38 39 FC EA 9A EB E9 1A     ; F0 ;
```

You might need to edit this table in order to display NLS characters properly according to the code page that you are using. You need a reference of the host code page and the OS/2 code page to do so.

To control the ASCII-to-EBCDIC translation for TN5250, the file 5250XLT.XLT is used. It is provided as a sample for the US translation table as 5250XLT.SAM.

To refresh the keyboard remap from within a PMANT session through the Options menu, a file named PMANT.KEY must exist in the TCPIPETC directory. This will be taken as the input file and can be revised using any ASCII editor.

To refresh the keyboard remap from within a TN3270 session through the Main Menu, a file named TN3270.KEY must exist in the TCPIPETC directory. This will be taken as the input file and can be revised using any ASCII editor.

To refresh the keyboard remap from within a TN5250 session through the Options menu, a file named TN5250.KEY must exist in the TCPIPETC directory. This will be taken as the input file and can be revised using any ASCII editor.

Notes:

1. PMANT.KEY can also be used as TN3270.KEY
2. You can only remap 3270 or 5250 functions to OS/2 keys, not NLS characters.
3. Use the .XLT files to remap NLS characters according to your 3270 or 5250 host code page and your OS/2 code page.

8.2.10.3 The TELNET.RC File

The TELNET.RC file in the ETC directory is a text file used to specify commands to a Telneto client that the user would normally have to type in by hand. The TELNET.RC file is made up of sections that define commands to be entered when using the line mode Telnet client to a particular host. These commands are entered on behalf of the user as if they were typed in at a particular Telneto command prompt.

Each section is identified by a line that starts with the name of the host that is being connected to. The rest of the line, and successive lines in a particular section, begin with white space and are assumed to be Telneto client commands. These commands are processed as if they had been typed in

manually at the Telneto command prompt. The last line of each section must be blank to signify the end of that section.

The following format rules apply:

1. Lines beginning with a ″#″ are comment lines.
2. Lines that begin without white space start a new host entry section.
3. A blank line signifies the end of a host entry section.
4. The hostname DEFAULT matches for all hosts.
5. All matching host entry sections will be executed.

What follows is a sample of the TELNET.RC file:

```
####################################################################
# This is a DEFAULT entry that is executed                         #
# for all servers that you connect to.                             #
####################################################################
DEFAULT         send ayt
                toggle local
                set esc ^]
                set interrupt ^c

####################################################################
# This is a host specific entry that is executed                   #
# only when telneting to the host 'klaus.itsc.raleigh.ibm.com'     #
####################################################################
klaus.itso.ral.ibm.com          toggle bs
                                set esc ^t
                                toggle wrap
                                toggle crmod
```

Chapter 9. File Transfer

This chapter describes the ways in which TCP/IP for OS/2 may transfer data between different hosts in a network.

In TCP/IP for OS/2 the following file transfer protocols are implemented:

- File Transfer Protocol (FTP)
- Trivial File Transfer Protocol (TFTP)

This chapter introduces the integration of FTP clients into the OS/2 Workplace Shell, and it also discusses some of the customization requirements for the FTP server.

If you have to make a decision regarding which of the services to use, consider that FTP has the following features, which are not available in TFTP:

- Subcommands to list files or work with directories on the foreign host.
- HPFS file names.
- User authentication.
- An FTP server supports many clients, while a TFTP server supports only one client at a time.
- FTP achieves better performance than TFTP.
- FTP uses TCP and TFTP uses UDP.

9.1 Workplace Shell Integration of FTP Clients

One of the FTP clients that come with TCP/IP for OS/2, FTPPM, is implemented as an object of the OS/2 Workplace Shell. It can be found in the Templates folder on the OS/2 Desktop after TCP/IP for OS/2 has been installed.

The following shows part of a Templates folder containing the Workplace Shell objects created by TCP/IP for OS/2:

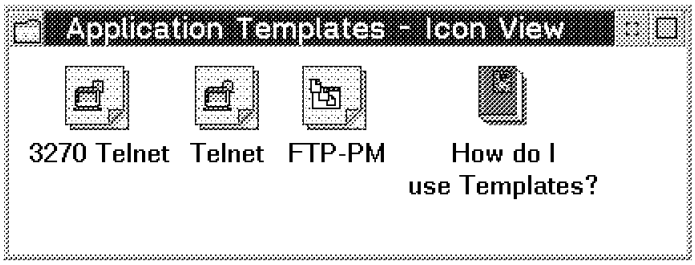


Figure 162. Templates Folder with TCP/IP Objects

You can create multiple instances of these templates by dragging a template object onto the OS/2 Desktop using the right mouse button.

To configure an instance of FTPPM, see 9.2.3, “Configure FTPPM” on page 251.

9.2 File Transfer Protocol (FTP)

This section describes the implementation of FTP in the TCP/IP for OS/2 product.

9.2.1 TCP/IP for OS/2 FTP Server

The OS/2 FTP server uses two files in order to provide user authentication, both of which are flat ASCII files that can be modified by an ASCII editor.

9.2.1.1 The TRUSERS File

The TRUSERS (trusted users) file contains information about the users that are allowed to log on to your OS/2 system, their (optional) passwords, and the directories where they have read and/or write access. It is usually stored in the MPTNETC subdirectory.

You may either create this file with an editor, or use the TCP/IP Configuration Notebook to specify the parameters that would then be stored in the TRUSERS file. A sample TRUSERS file is shown in the following example:

```
user: walter walter
rd^:
wr^:

user anonymous
rd:c:\tcpip\tmp
wr:
```

This means that the user walter with password walter has read and write access to all drives, and that user anonymous (common usage for everyone) has read access to the C:TCPIPTMP directory and subdirectories but no write access at all.

9.2.1.2 The NETRC File

The NETRC file is used for both the FTP and REXEC clients. It contains login IDs and passwords and, optionally, macro definitions or programs to run automatically when a user logs on to your OS/2 system. The file does not have to be named NETRC or reside in the TCPIPETC-directory. By using the NETRC environment variable, the user can choose to place the file under a different name and directory for some added security.

To start the OS/2 FTP server, enter FTPD on an OS/2 command prompt, or start it together with the INETD server.

The following screen shows the OS/2 FTP server:

```
****************************************
*          IBM TCP/IP for OS/2         *
*           FTP Server (FTPD)          *
*   Version: 18:29:28 on Nov 15 1995   *
* (C) Copyright IBM Corp. (1991, 1994) *
****************************************
FTPDC: spawned with socket 477
connection from 9.24.104.77 at Wed Jan 31 11:45:06 1996

FTP LOGIN FROM 9.24.104.77, grode
```

9.2.2 TCP/IP for OS/2 FTP Clients

FTP clients give you the option to log on to remote systems in order to transfer files between them and your local OS/2 system. You can use OS/2 FTP clients to connect to OS/2, DOS, MVS, VM, OS/400, or UNIX FTP servers. TCP/IP for OS/2 includes the following FTP clients:

FTPPM FTP client running as an OS/2 Presentation Manager application.

FTP FTP client running as an OS/2 window or OS/2 full-screen application.

The FTP client is implemented with the standard FTP subcommands such as:

Subcommand	Function
open	Connect to and log on at a remote host.
close	End a session with a remote host.
dir	List the remote directory.
pwd	Display the current remote directory.
lcd	Display or change the current local directory.
get	Transfer a file from the remote system.
put	Transfer a file to the remote system.
quit	End FTP.

Subcommands can be entered from the FTP command shell. You get there by entering `FTP` from an OS/2 command prompt.

You will find the following hints helpful when using FTP:

- The exclamation mark (`!`) invokes the OS/2 CMD processor. Type `EXIT` to get back to the FTP shell. The local current directory can easily be listed by typing `!dir`, while the remote directory is listed by typing `dir`.
- A simple way to list a text file on the remote host is to enter `get filename con`. By substituting `prn` for `con` the remote file will print on the user's printer. The user can actually print the file on a remote OS/2 workstation using `put filename prn` as an alternative to using the LPR command.

  Note that CON and PRN are examples of OS/2 device names and that you are not restricted to just those names in a GET or PUT command. Examples of other OS/2 device names are COM1 and LPT1.
- FTP also works with OS/2 drives redirected to an OS/2 LAN Server. That means it is possible to use FTP from another environment (for instance S/370) and directly access an OS/2 LAN Server directory. The user must have write access to the directory if he wants to copy a file to a foreign host.

For a more detailed discussion of the FTP subcommands, please refer to the online documentation.

Instead of FTP you can also use the Presentation Manager application FTPPM which is part of TCP/IP for OS/2 . The following sections describe how to configure and use FTPPM.

9.2.3 Configure FTPPM

To configure an instance of the FTPPM object, double-click on the object for the first time, or select **Open Settings**. Select the appropriate tab in the notebook-style configuration window and fill in the necessary values for your FTP session.

If you just want to create a generic FTPPM instance in order not to overpopulate your desktop with objects, leave all fields blank. Only click on the **Create New Window** radio button on the Window menu. You can then enter all communication parameters after starting FTPPM.

On the first page of the Configuration Notebook (not shown here) you can enter the hostname or the IP address of the host that you want to connect to.

The following figures show the second and third pages of the FTPPM Configuration Notebook:

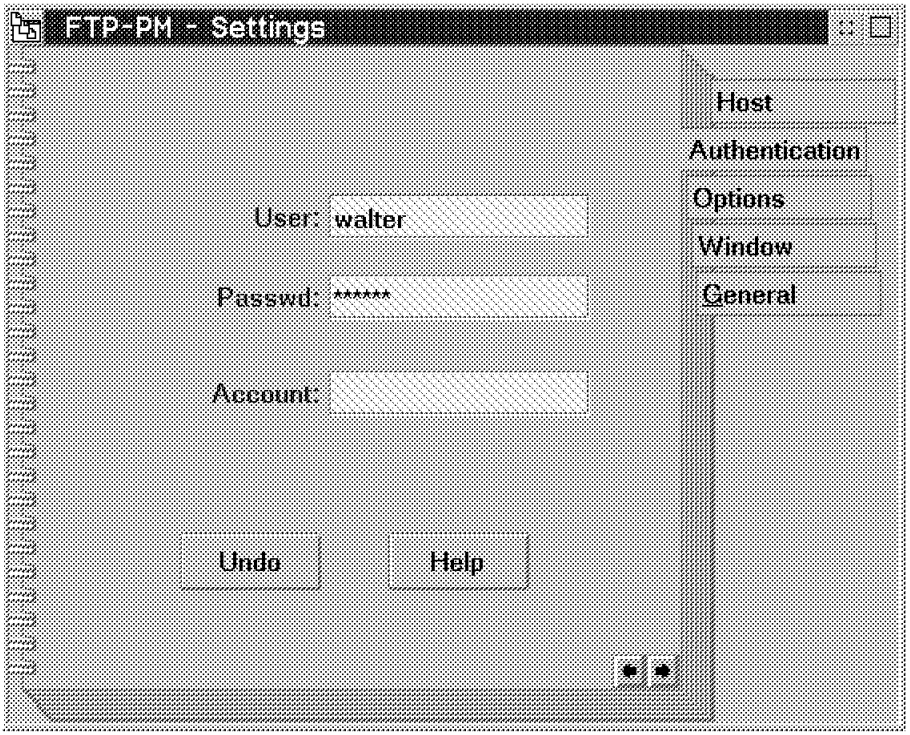


Figure 163. Authentication Page of the FTPPM Configuration Notebook

On the Authentication page you specify user information.

Setting	**Meaning**
User	The user ID that you want to log on with at the remote host.
Passwd	The password for the above user ID (if required).
Account	The account information for the above user ID (for VM and MVS, if required).

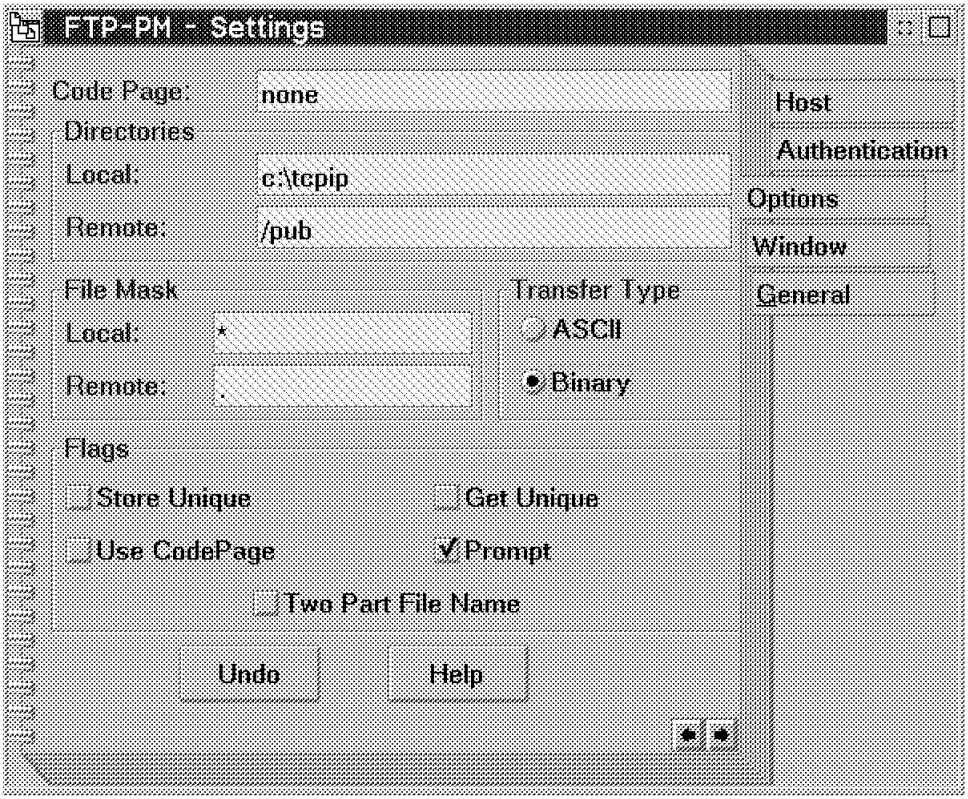


Figure 164. Option Page of the FTPPM Configuration Notebook

On the Options page you specify initial parameters for an FTPPM session.

Setting	Meaning
Code Page	Used to convert ASCII characters that are represented differently on a remote system than on the local host.
Directories Local	Name of the local directory for the file transfer.
Directories Remote	Name of the remote directory for the file transfer.
File Mask Local	Files to be listed from the local directory.
File Mask Remote	Files to be listed from the remote directory.
Transfer Type	Specify whether files to be transferred should be treated as text files (ASCII), or as program or image files (binary).
Store Unique	Store files remotely with unique file names. This will assign a new, unique, file name to an eventually existing file on the remote host.
Get Unique	Store files locally with unique file names. This will assign a new, unique, file name to an eventually existing file on the local host.
Use Code Page	Check here to use the code page named above.
Prompt	Prompt for confirmation before actually transferring files.
Two Part File Names	Selected to mark a file for transfer by just marking the name.

For a more detailed discussion of these settings please refer to the online documentation.

9.2.4 FTP to and from UNIX

File transfer with FTP or TFTP in both directions is supported between OS/2 and UNIX.

The following is an example of opening an FTP session from an AIX session to the OS/2 FTP server walter. It also shows the actual settings of the FTP by using the status subcommand:

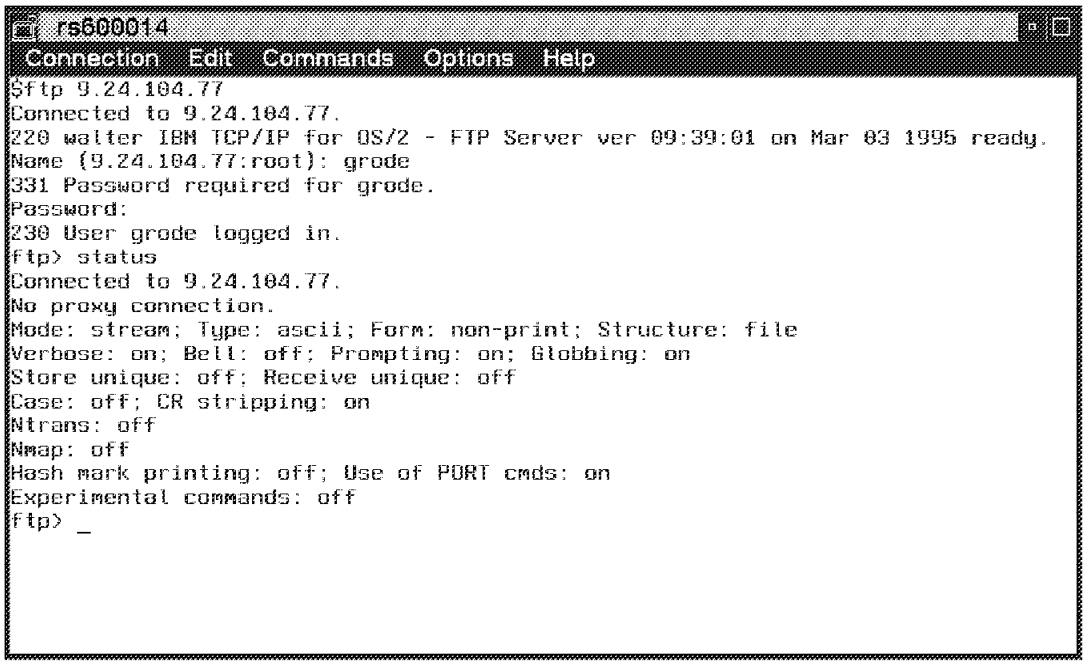


Figure 165. FTP Status on AIX Command Shell

You can do file transfers in both directions, but remember that you can only write on disks to which you have the appropriate rights depending on the user rights on AIX/UNIX or the OS/2 TRUSERS file.

The following screen shows how to use FTP subcommands (they are similarly available in AIX as they are in OS/2) to list files in the remote directory, transfer the CONFIG.SYS file, and finally list files in the local directory:

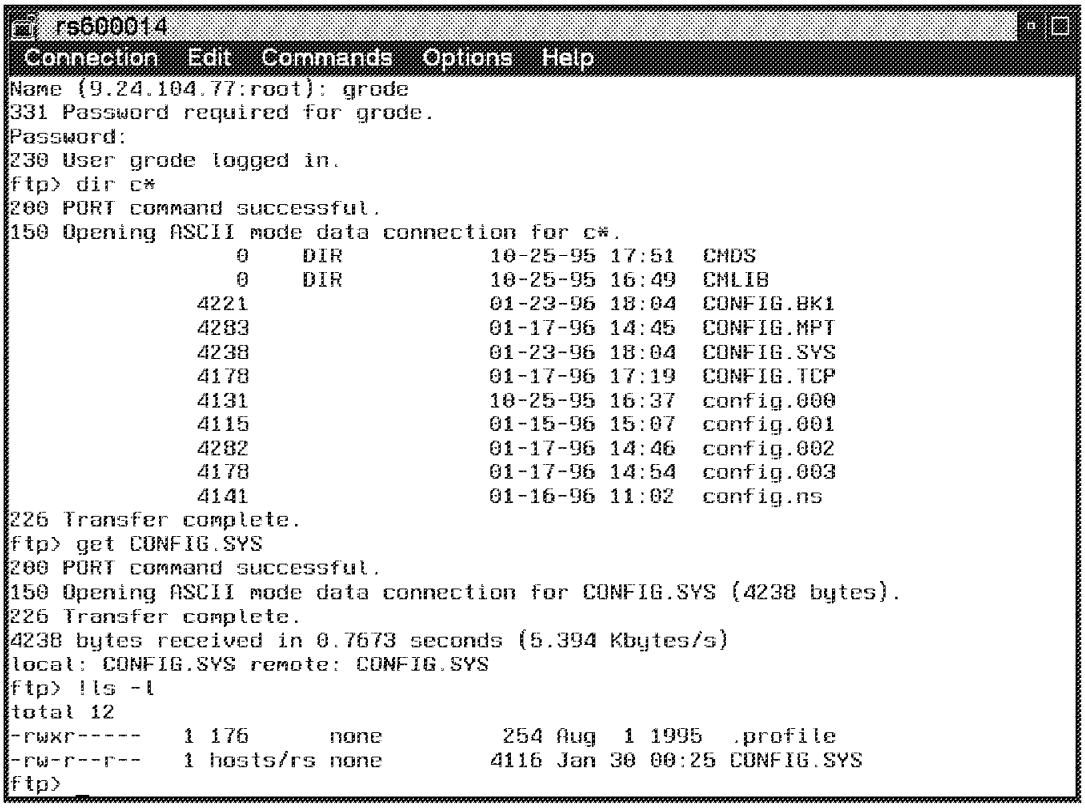


Figure 166. Transfer File from OS/2 to AIX

You see that the remote (OS/2) directory is C: and the local (AIX) directory is /u/os2. All you need to do is issue the get command to transfer the file from OS/2 to AIX. Transferring a file this way uses the OS/2 FTP server. Moreover, FTP provides information about the throughput of each file transfer in kilobytes per second.

Note: You can see that AIX distinguishes between upper and lowercase file names, whereas OS/2 actually doesn't. Though HPFS can save file names in mixed cases, it wouldn't consider CONFIG.SYS and config.sys to be different files, as AIX or UNIX would.

To close the FTP session enter the command `close` and then enter `quit` to end FTP.

The FTPPM client allows file transfer in both directions from a Presentation Manager window. To start FTPPM, double-click on an instance of the FTPPM object, or enter `FTPPM` at an OS/2 command prompt. If you didn't already specify a hostname, user ID, and password in the FTPPM object's settings, you will now be prompted to do so.

After successful logon the contents of the current local and remote directories are displayed. You are now able to choose a file for file transfer in either direction.

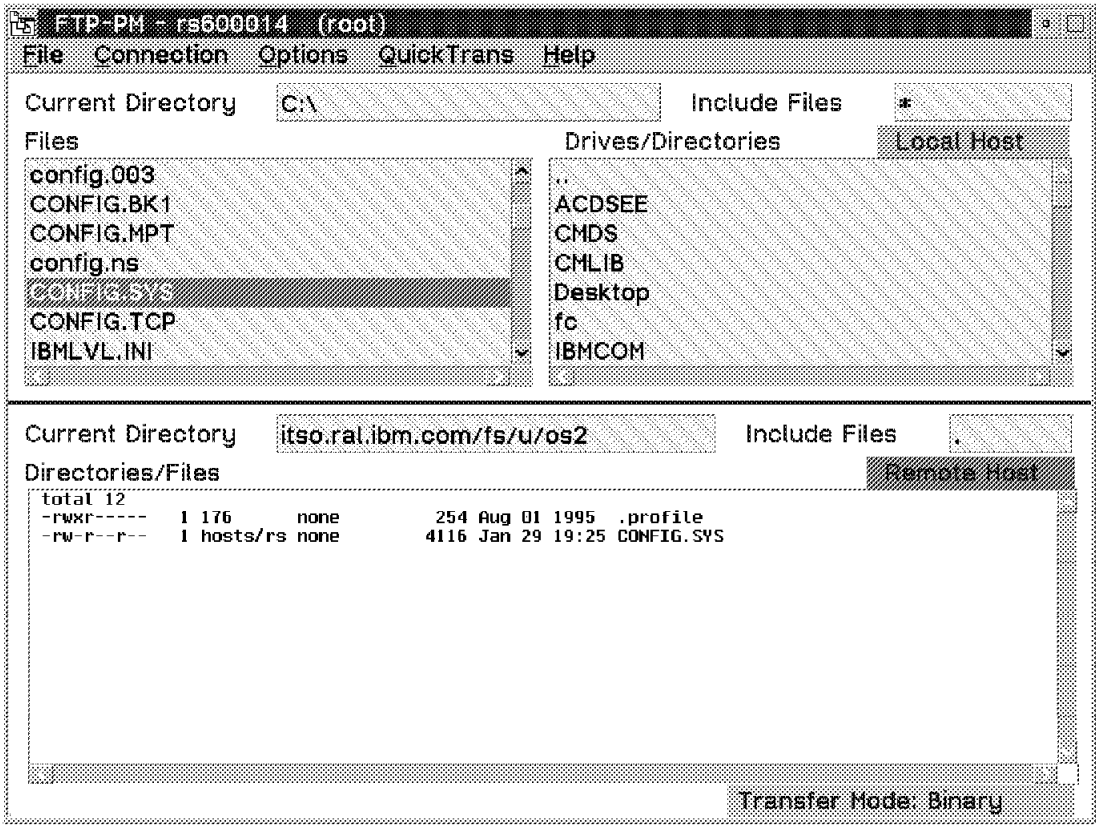


Figure 167. Using FTPPM

9.2.5 FTP to and from VM

VM directories contain CMS minidisks. The current directory after the user has issued an FTP logon is his or her A-disk, which is represented as user ID.191. If, for example, the user has a 192 disk, the command `cd user ID.192` will make the 192 disk the current directory.

Note: An OS/2 FTP user cannot access temporary CMS disks.

If the user is concurrently logged in to CMS via 3270 emulation, the FTP file transfer will have only read access to the A-disk.

The following figure shows a directory listing of a VM disk using the OS/2 FTP client:

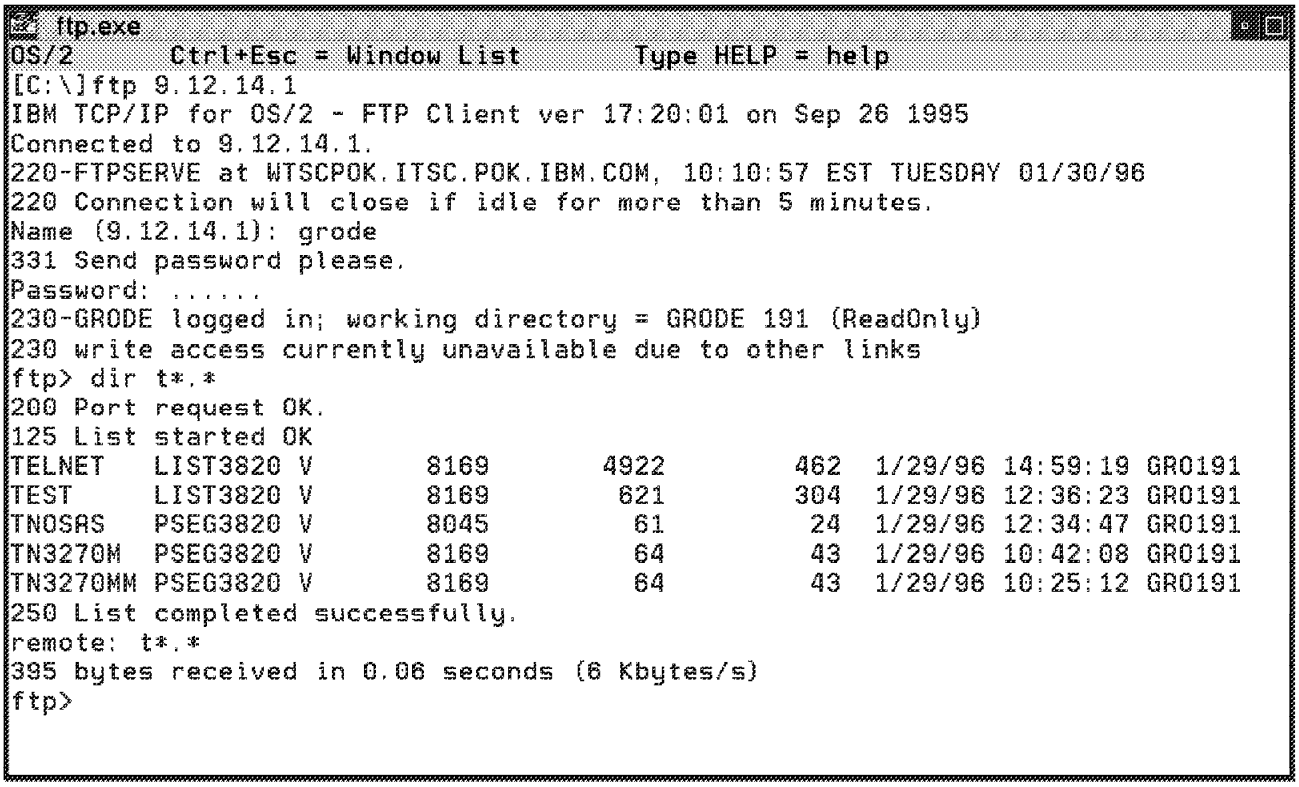


Figure 168. Listing Files in a CMS Minidisk Using FTP

9.2.6 FTP to and from MVS

For sequential files a directory is considered one or more data set name qualifiers. Use the `cd qualifier1.qualifier2...` command to descend the directory tree and `cd..` command to ascend the tree. If you do a directory listing when you have reached the fully qualified name, you will receive the message `No data sets found`.

To reach a partitioned data set (PDS), enter the fully qualified name of the PDS as the directory in the `cd` command. FTPSERVE will inform you that the working directory is now a partitioned data set. A directory listing will show the members of the PDS.

9.2.6.1 Session Example with MVS

The following example shows an OS/2 FTP session that transfers a text member in a PDS data set in MVS to a directory in OS/2. The OS/2 user in the example does the following:

1. Finds the PDS in MVS. Note that FTPSERVE in MVS informs the user that the current directory is a PDS.
2. Checks the contents of the PDS.
3. Chooses a suitable local directory and goes out to the OS/2 session to check the contents of it.
4. Transfers the PDS member and terminates the FTP session.

```
[C:]ftp 9.24.104.74
Connected to mvs18sna.
IBM TCP/IP for OS/2 - FTP Client ver 17:20:01 on Sep 26 1995
Connected to 9.24.104.74.
220-T18AFTPC IBM MVS V3R1 at mvs18.itso.ral.ibm.com, 15:11:10 on 01/31/96
220 Connection will close if idle for more than 10 minutes.
Name (9.24.104.74): tony
331 Send password please.
Password: ......
230 TONY is logged on.  Working directory is "TONY.".
ftp> cd 'TCPIP.ITSC.ASM'
250 "'TCPIP.ITSC.ASM'" partitioned data set is working directory
ftp> dir r*
200 Port request OK.
125 List started OK.
 Name     VV.MM  Created     Changed     Size  Init   Mod   Id
RAWTEST   01.03 95/12/06 95/12/06 19:00   241   240     0 ALFREDC
RDW       01.03 92/07/19 92/07/19 17:40   217    11     0 CHRISTE
REGEQUY
RIPQ      01.26 95/11/02 96/01/18 10:51   335   130     0 ALFREDC
RIPTRACE  01.24 95/11/19 96/01/30 09:11   549   467     0 ALFREDC
250 List completed successfully.
remote: r*
404 bytes received in 0.28 seconds (1 Kbytes/s)
ftp> lcd
Local directory now C:\temp
ftp> !dir

The volume label in drive C is OS2.
The Volume Serial Number is E720:9C14.
Directory of C:\temp

 1-16-96   5:23p     <DIR>           0  .
 1-16-96   5:23p     <DIR>           0  ..
        2 file(s)         0 bytes used
                       9396736 bytes free
ftp> get riptrace
200 Port request OK.
125 Sending data set TCPIP.ITSC.ASM(RIPTRACE) FIXrecfm 80
250 Transfer completed successfully.
local: riptrace remote: riptrace
23474 bytes received in 0.34 seconds (67 Kbytes/s)
ftp> !dir

The volume label in drive C is OS2.
The Volume Serial Number is E720:9C14.
Directory of C:\temp

 1-16-96   5:23p     <DIR>           0  .
 1-16-96   5:23p     <DIR>           0  ..
 1-31-96  10:11a     23475           0  riptrace
        3 file(s)      23475 bytes used
                       9372672 bytes free

ftp> quit
221 Quit command received. Goodbye.

[C:\temp]
```

VSAM data sets will be reported with DSORG VSAM in a DIR listing, but you cannot do anything useful from FTP with VSAM. FTPSERVE in MVS will report: Directory name matches a VSAM data set.

9.2.7 FTP to and from DOS

You can transfer files to and from DOS using either FTP, TFTP or FTPPM. The IBM TCP/IP V2.1.1 for DOS product offers you FTP and TFTP clients, an FTP server called FTPD, and an FTP client for Windows 3.1 called WFTP. The use of the FTPD server depends on the RAM memory your workstation has.

FTP is a full-screen FTP client which includes FTP subcommands similar to the OS/2 FTP client. WFTP is a Windows 3.1 FTP client similar to FTPPM. The FTP server (FTPD) is a dedicated (foreground process) FTP server which can be interrupted to access a DOS command shell. FTPD also uses a TRUSERS (trusted users) file in the TCPDOSETC directory similar to the OS/2 FTP server.

The following shows an FTP session between an OS/2 client and a DOS server:

```
FTP.EXE
IBM TCP/IP for OS/2 - FTP Client ver 17:20:01 on Sep 26 1995
Connected to 9.24.104.165.
220 tcpdos FTP server (2.1.1.4) ready.
Name (9.24.104.165): walter
331 Password required for walter.
Password: ......
230 User walter logged in.
ftp> pwd
257 "C:\" is current directory.
ftp> get config.sys
200 PORT command successful.
150 Opening ASCII mode data connection for config.sys (268 bytes).
226 Transfer complete.
local: config.sys remote: config.sys
268 bytes received in 0.00 seconds (0 Kbytes/s)
ftp> status
Connected to 9.24.104.165.
No proxy connection.
Verbose: on; Bell: off; Prompting: on; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on
Ntrans: off
Nmap: off
Hash mark printing: off; Use of PORT cmds: on
ftp> _
```

Figure 169. FTP Session from OS/2 to DOS

For more details about the DOS FTP client and server, please see the IBM TCP/IP 2.1.1 for DOS product documentation.

9.2.8 FTP to and from OS/2

You can use either FTP, FTPPM or TFTP to transfer files from one OS/2 workstation to another.

If you use the OS/2 FTP client and create a NETRC file it is possible to define macros within that file. The following shows the NETRC file on machine walter:

```
machine testserver login walter password walter macdef rextest
bell
prompt
status
```

The keyword macdef implies the following is a macrodefinition. The macro is called rextest and contains the commands bell, prompt and status. The following screen shows the execution of this NETRC file with the host test server and macro rextest:

```
[C:]ftp testserver
Connected to testserver.
220 testserver IBM TCP/IP for OS/2 - FTP Server ver 18:29:28 on Nov 15 1995 ready.
331 Password required for walter.
230 User walter logged in.
ftp> $rextest
bell
Bell mode on.
prompt
Interactive mode off.
status
Connected to testserver.
No proxy connection.
Mode: stream; Type: ascii; Form: non-print; Structure: file
Verbose: on; Bell: on; Prompting: off; Globbing: on
Store unique: off; Receive unique: off
Case: off; CR stripping: on
Ntrans: off
Nmap: off
Hash mark printing: off; Use of PORT cmds: on
Macros:
        rextest
ftp>
```

To execute the macro you have to enter it in the form $ftpmacro at the ftp> prompt. To make the FTP server interpret your macro correctly, you must use two consecutive CR/LF (carriage return/line feed) sequences in the NETRC file to mark the end of the macro.

9.2.9 FTP to and from OS/400

OS/400 directories contain folders. The current directory after the user that has issued an FTP logon is the QGPL library.

The following figure shows a directory listing of an OS/400 library using the OS/2 FTP client:

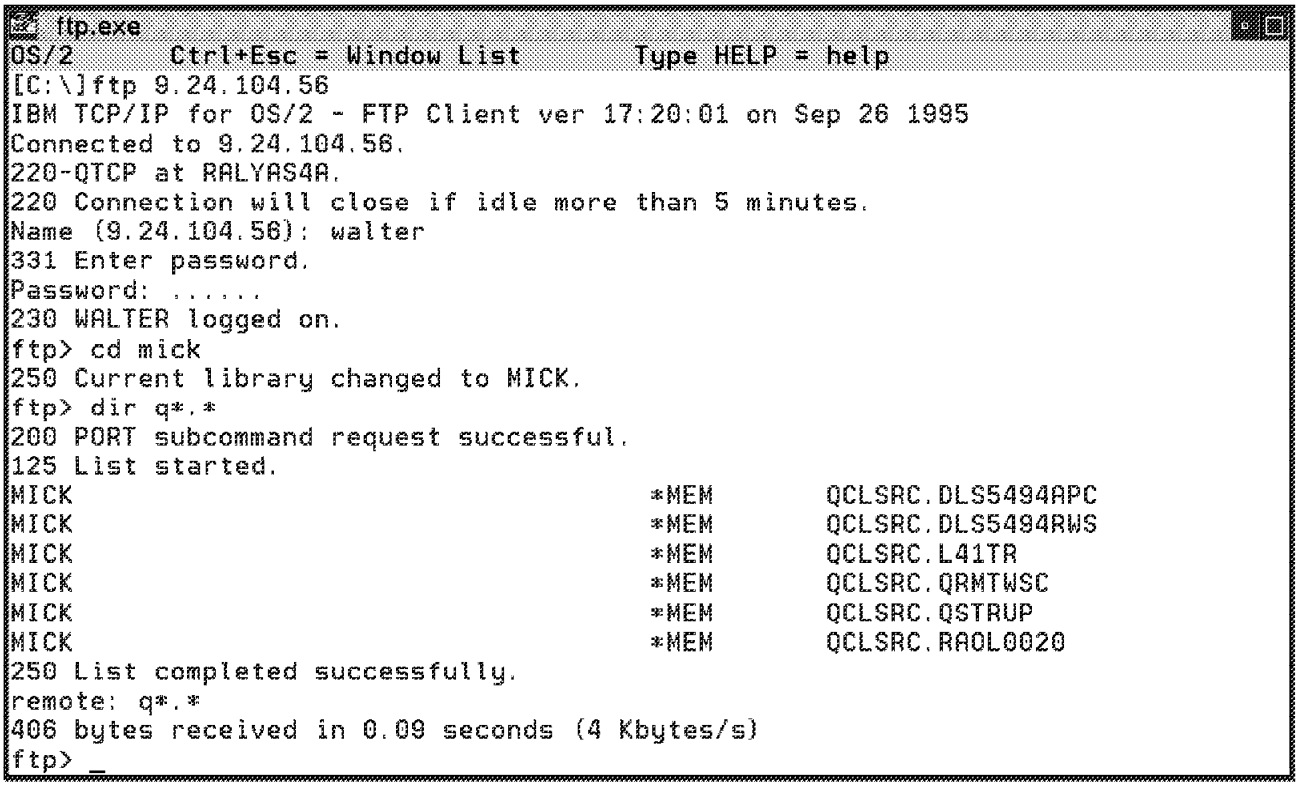


Figure 170. Displaying the Contents of an OS/400 Library Using FTP

Note: When you access an AS/400 system with FTP you only have access to OS/400 library objects, not to shared folders used by PC Support.

The following shows an FTP session from an AS/400 system to an OS/2 FTP server, listing an OS/2 directory, and transferring the PROTOCOL.INI file back from the AS/400 to OS/2:

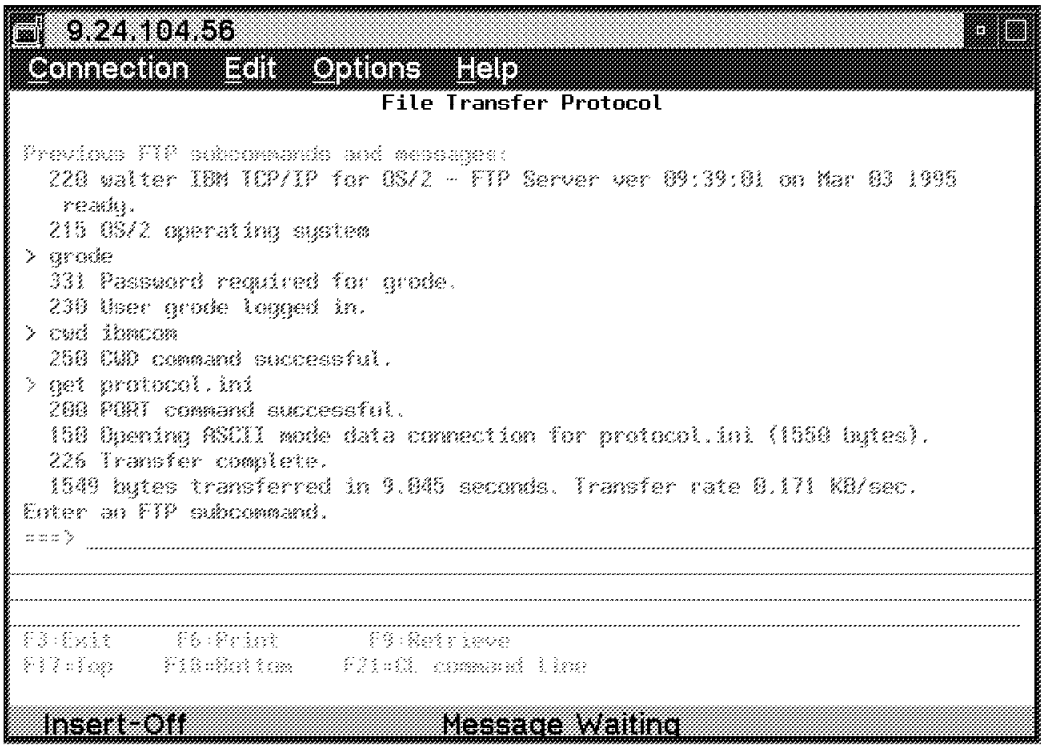


Figure 171. Transferring a File from OS/400 to OS/2 Using FTP

9.2.10 Multiple FTP Sessions from a Single OS/2 Client

It is possible to have multiple FTP sessions at the same time from one OS/2 workstation. You use FTPPM to establish all sessions and then switch back and forth among them. The following figure shows the connection list from which to choose a host for file transfer:

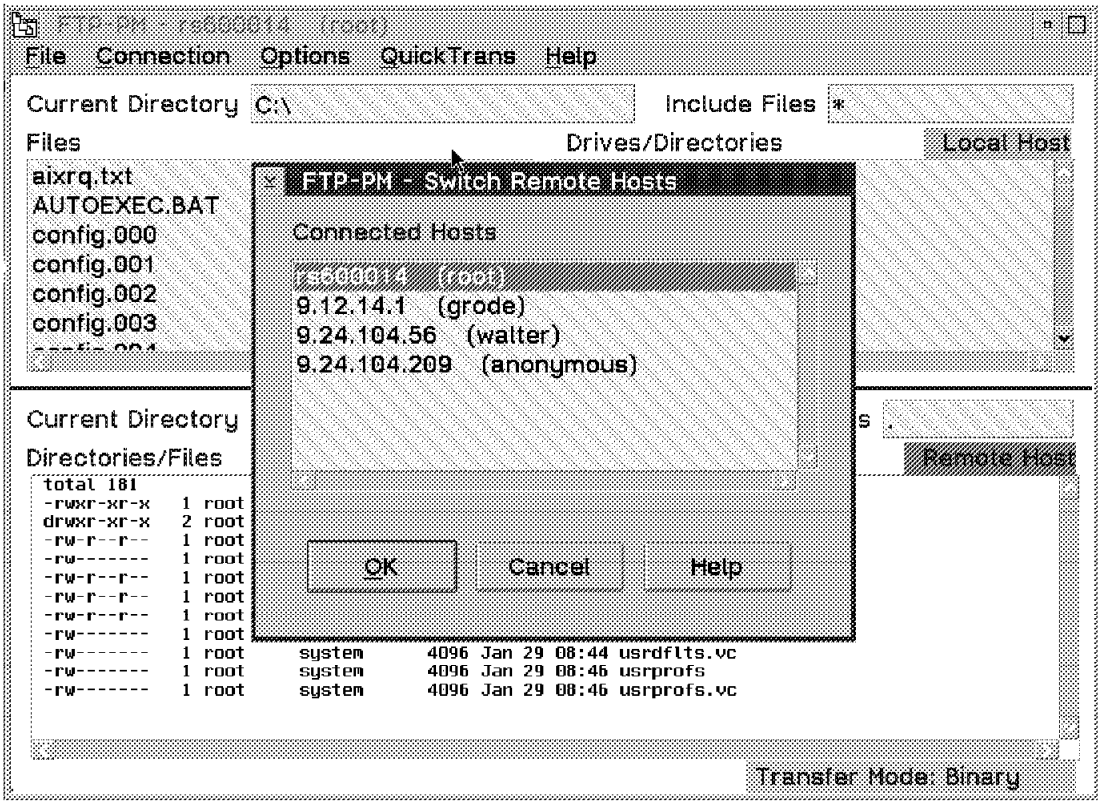


Figure 172. Multiple FTP Sessions with FTPPM

9.3 Trivial File Transfer Protocol (TFTP)

TFTP is an alternative to transferring files with FTP. TFTP is a simple file transfer protocol and does not provide all of the features available in FTP. TFTP uses User Datagram Protocol (UDP) as the underlying protocol; therefore, it is an unreliable means of file transfer.

TFTP is implemented in TCP/IP for OS/2 with both client and server support. You can start TFTP using either the INETD super server or start TFTP from an OS/2 command prompt in an OS/2 window.

Notes:

1. Only one TFTP server can run on a PC at one time.
2. There is no security for access to TFTP servers. When the TFTPD server is running on your PC, other users in the network can read, write, or even destroy the files on your machine.

The following screen shows the subcommands available for TFTP:

```
tftp> ?
Commands may be abbreviated.  Commands are:

connect         connect to tftp server
mode            set file transfer mode
put             send file
get             receive file
quit            exit tftp
verbose         toggle verbose mode
trace           toggle packet tracing
status          show current status
binary          set mode to octet
ascii           set mode to netascii
rexmt           set per-packet retransmission timeout
timeout         set total retransmission timeout
?               print help information
tftp>
```

9.4 Using FTP from OS/2 CMD Files

The following examples of OS/2 CMD files show how FTP can be called from within such a file:

FTPPUT1.CMD (Prompt for Password)

```
@echo off
rem usage: ftpput <host> <local file> <remote file>
rem password will be prompted if a NETRC-file is not used
rem no blank allowed between user name and redirect operator
rem remove next statement if a NETRC-file is used
echo walter < ftp.in
echo put %2 %3 >> ftp.in
echo quit >> ftp.in
ftp %1 < ftp.in
if errorlevel 1 goto end
echo File %2 transferred to %1
&end
erase ftp.in
```

FTPPUT2.CMD (No Prompt for Password)

```
@echo off
rem usage: ftpput <host> <local file> <remote file>
echo open %1> ftp.in
echo user walter password >> ftp.in
echo put %2 %3  >> ftp.in
echo quit >> ftp.in
ftp -n < ftp.in
if errorlevel 1 goto end
echo File %2 transferred to %1
&end
erase ftp.in
```

The following shows an example of running FTPPUT1.CMD with prompt for password:

```
  OS/2     Ctrl+Esc = Task List                                Type HELP = help
[C:\temp]ftpput1 walter ftptst.tst c:\work\test1.txt
Name (walter): Password:
File ftptst.tst transferred to walter

[C:\temp]
```

This example can also be executed from Presentation Manager by defining a program in Presentation Manager:

```
Program title:     FTPPUT
Path and filename: c:\temp\ftpput1.cmd
Parameters:        [Enter <host> <local file> <remote file>]
Working directory: c:\temp
```

Chapter 10. Remote Printing

TCP/IP for OS/2 offers you the possibility to print jobs on remote printers and receive jobs from remote systems to your printer in a TCP/IP network environment. The following modules provide remote printing functions:

Module	Remote Printing Function
lpd	Line printer daemon (server)
lpr	Line printer requester (client)
lprmon	Line printer monitor
lpq	Lists jobs in remote queues
lprm	Removes jobs from remote queues
lprportd	Enables the use of port objects in the OS/2 Workplace Shell

10.1 TCP/IP for OS/2Remote Printer Server LPD

LPD is a line printer server that allows clients to print files into a queue defined in the Print Manager. When a client sends a print request to LPD, it should specify the queue that it wants the file to be printed in. LPD will also accept a device name (for example, LPT1), and try to determine the first queue that is attached to that device.

10.1.1 File Format Types

When LPD receives a print job, the client can specify that the file is of a particular format via special codes in the control file that are passed along with the print job. LPD understands two types of formats, and the user can have either as the default. If a client notifies LPD that the file being printed is a binary file, then LPD will put exactly what is sent into the corresponding queue to be printed. If a client notifies LPD that the file being printed is a plain text file, then LPD will convert LF (line feeds) to CRLF (carriage return, line feed) pairs.

By default, if a client sends a file to LPD, and does not specify one of the known file types of binary or text, then LPD will default to assuming that the file is binary. If for some reason when you try to print a file to the OS/2 LPD, it prints diagonally across the page (that is, missing carriage returns to put the next line at the left margin), then you probably want LPD to default to assuming that the file it receives without a known type is of type TEXT (and therefore LPD will do the LF to CRLF conversion). This can be done by specifying the -f option on the LPD command line.

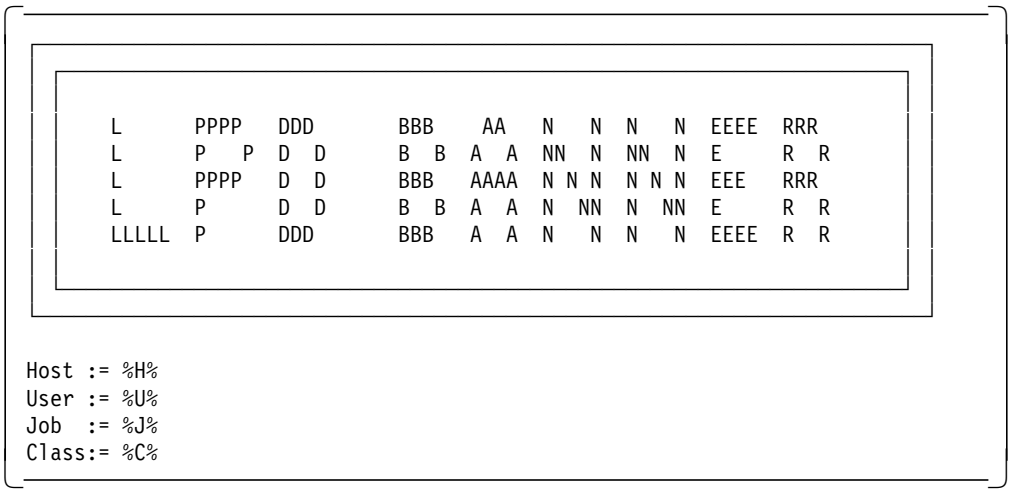
10.1.2 LPD Banner Page

Along with being able to disable the default banner page, users can also specify their own banner page. A user does this by specifying a file name along with the -b option on the LPD command line. Whenever a print request comes in, LPD will print the specified file as the banner page. If you do not specify a file name with the -b option, there is no banner printed.

The following is a list of keywords that can be specified in the banner file and that are replaced with their corresponding value upon printing:

Keyword	Description
%H%	Name of the host that originated the print job
%U%	Name of the user that originated the print job
%J%	Name of the print job
%C%	Class of the print job

The default banner page looks like the following:



10.1.3 LPD Control File

This file contains spooler, queue, and network options for each job, and some additional control parameters for the LPD server. It can optionally be printed with each job.

10.1.4 LPD Usage

The format of the LPD command is as follows:

```
lpd -? | {[-c] [-b [<banner>]] [-s] [-f]}

where    banner  - Optional file containing
                      the banner to be printed

         Options -
                 -?            This message
                 -c            Do not print control file
                 -b [<banner>]
                               If a <banner> file is supplied,
                               then use that file as the Banner.
                               Otherwise no banner is printed.
                 -s            Secure (720 < client_port < 732)
                 -f            Default to TEXT file format.
```

Note: If you use INETD to start the LPD server, you cannot define start parameters to the LPD command. If you wish to specify start parameters, you should start the LPD server in a foreground window.

The following shows the status screen of the OS/2 LPD server.

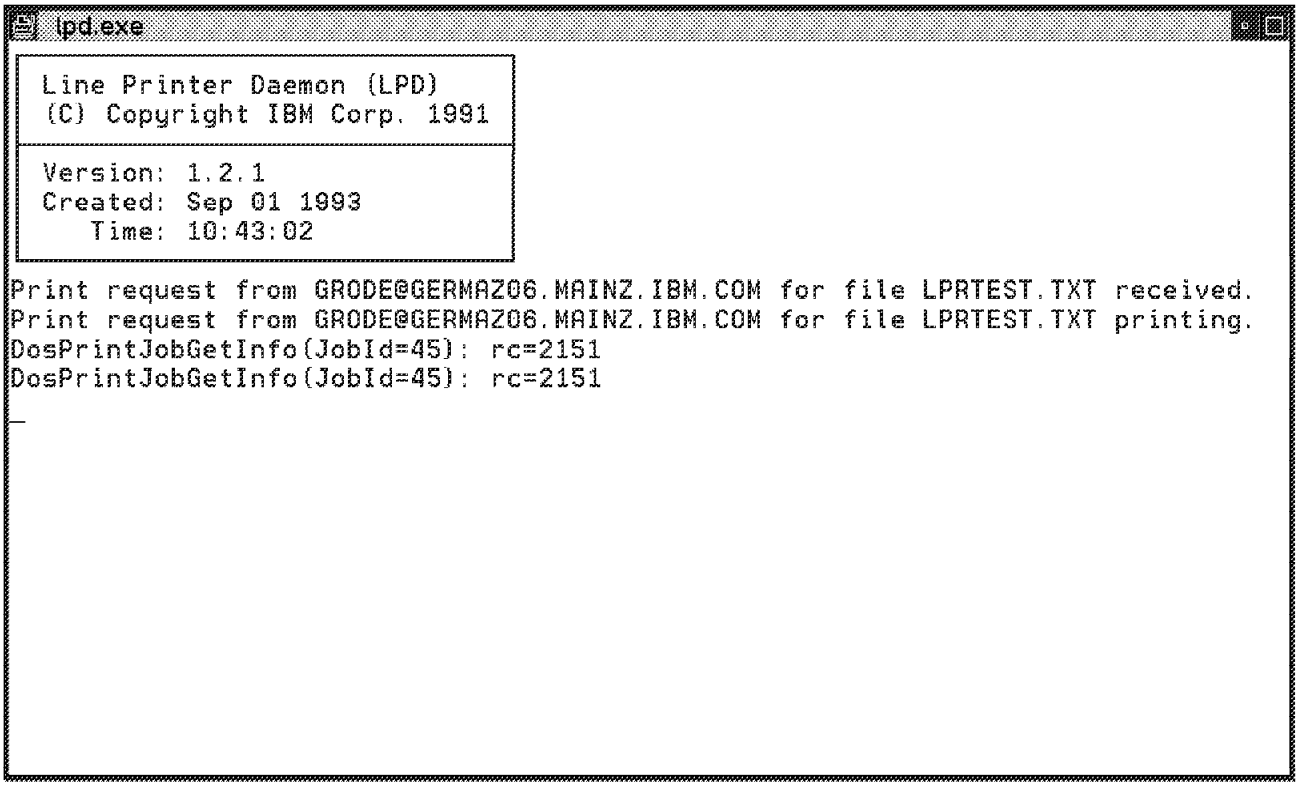


Figure 173. OS/2 LPD Server

10.2 Remote Printer Client LPR

The LPR command is used to explicitly send a file to a remote printer. If you need to send several files, you would have to issue as many LPR statements as files to be printed.

Note: Files you send to a printer can be specified using wildcards (″*″ or ″?″). LPR will accept that.

You do not have to specify the remote printer with any job if the environment variable LPR_PRINTER is set. This is usually the case if you have configured a remote print server in the TCP/IP Configuration Notebook.

The following example would send the CONFIG.SYS file to the remote printer queue WTRPRT02 at host walter:

```
[C:]lpr -p WTRPRT02 -s walter config.sys

Printing C:\CONFIG.SYS:
Trying LPD print server walter.itso.ral.ibm.com(9.24.104.77), device WTRPRT02.
Sent 4116 bytes.
The entire document was sent.

[C:\]
```

10.3 Remote Printer Monitor LPRMON

The LPRMON function allows you to redirect the output of a parallel printer port from your PC to a network host that provides the LPD server function. This allows you to print to an LPD server without an application using the line printer protocol directly.

This is very convenient for workstations without an attached printer, or for workstations that use remote printers from servers of different architectures such as, IBM OS/2 LAN Server V3.0 or Novell NetWare. To redirect all printer output on the local LPT1 port to the OS/2 printer WTRPRT02 on host walter, enter the following command:

```
start lprmon -p WTRPRT02 -s walter lpt1
```

Now, all output sent to the LPT1 device is redirected to the remote OS/2 printer queue. In this case, the command to send CONFIG.SYS is simply:

```
copy config.sys lpt1
```

The following shows how LPRMON would react on sending a file to the remote printer.

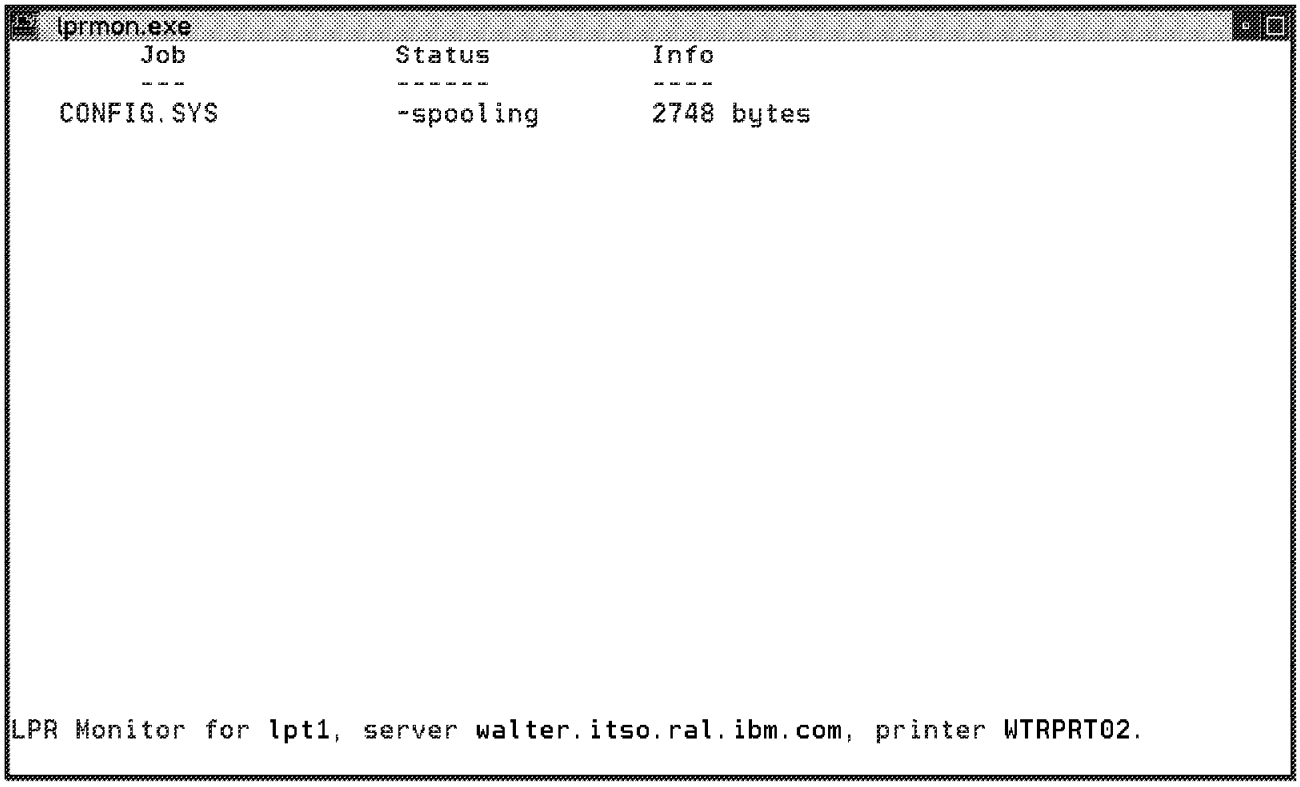


Figure 174. Remote Printing with LPRMON

Note: LPRMON only allows the redirection of parallel ports LPT1 to LPT3.

10.4 Workplace Shell Integration of Remote Printing

TCP/IP for OS/2 provides an easier method of printing than just redirecting a parallel port to a remote printer. Now you can redirect a printer object from the OS/2 Workplace Shell to a remote LPD printer. You can then select this printer from any Presentation Manager application as it appears in the list of available printers. Thus, you are no longer restricted to only three remote printers, as is the case with LPRMON. To use the Workplace Shell redirection method called the *LPR port driver*, create a new printer from the Templates folder, or open the settings from an existing printer object.

Note: You cannot use a network printer object for the LPR port driver since an LPT port does not work with the LAN-aware shell like IBM OS/2 LAN Server V3.0, or Novell NetWare Requester for OS/2.

On the TCP/IP for OS/2 Configuration Notebook setting for services, you can specify a maximum number of LPD ports. The default is 8, and the name of an LPD port is PIPELPDn, where n is the number of the port beginning with 0. These LPD ports appear in the output page of a printer object settings notebook after the parallel (LPT) and serial (COM) ports to which a printer can usually be connected.

The following shows connection icons for a Workplace Shell printer including LPR ports.

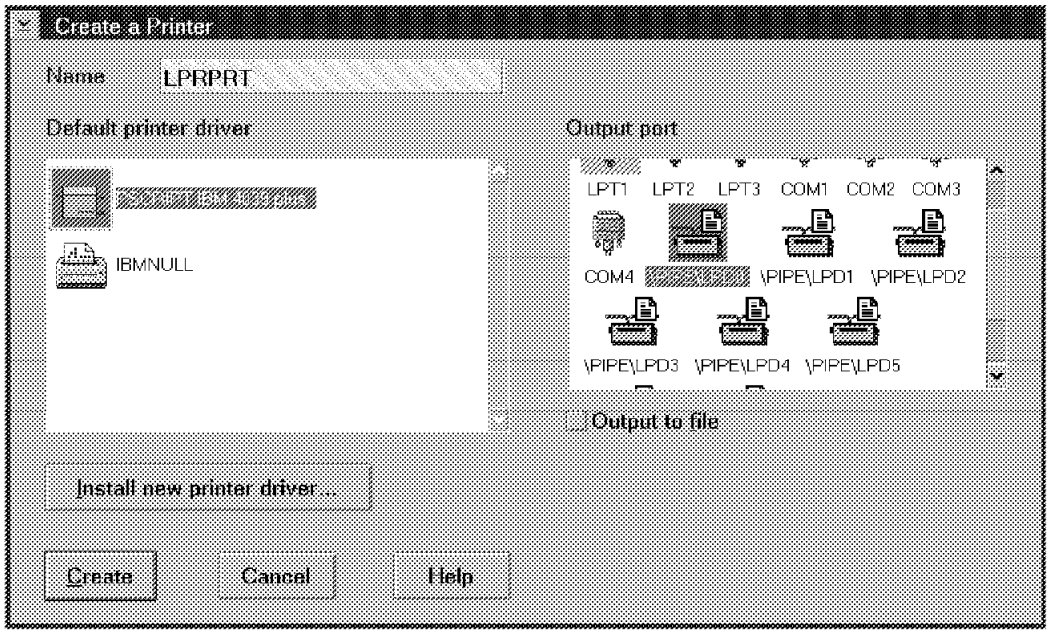


Figure 175. Connecting a Printer Object to an LPR Port

To connect an OS/2 printer to an LPR port, simply double-click on the port icon that you want to use for the printer in question.

The following shows the configuration menu for an LPR port:

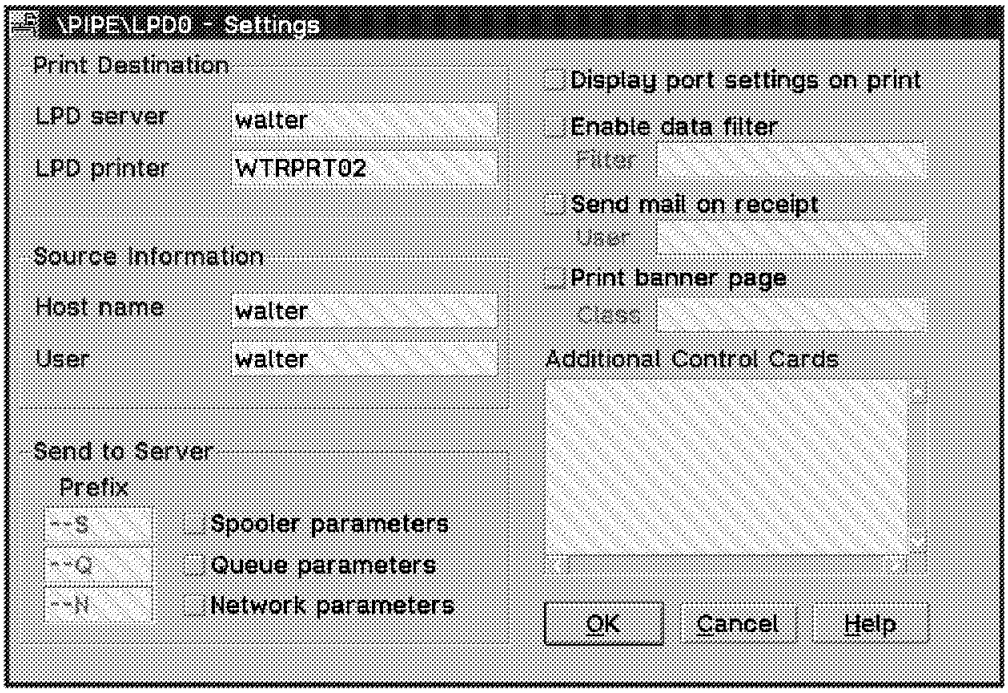


Figure 176. Configuration of an LPR Port

Menu	Setting
LPD Server	Specify the LPD server where jobs are to be sent for printing.
LPD Printer	Specify which printer to use on the above server.
Hostname	Specify the local host where print jobs originate.
User	Specify the user name at the above host.
Spooler	Send spooler parameters to the CONTROL file at the remote server.
Queue	Send queue parameters to the CONTROL file at the remote server.
Network	Send network parameters to the CONTROL file at the remote server.
Print Dialog	Display this menu every time before printing a job.
Send Mail	Send a reply when a job has finished. This option is not supported if the print destination server is an OS/2 LPD server which means that an OS/2 LPD server will not notify a user upon completion of print jobs. Other platforms may support this option.
Print Banner	Print a banner file with the optionally specified classification.
Control Cards	Send additional control information.

The following shows how to print the CONFIG.SYS file using the OS/2 Workplace Shell. The file object CONFIG.SYS is dragged from the Directory folder onto the printer object using the right mouse button. Once the button is released, the file will print to the remote printer LPR printer.

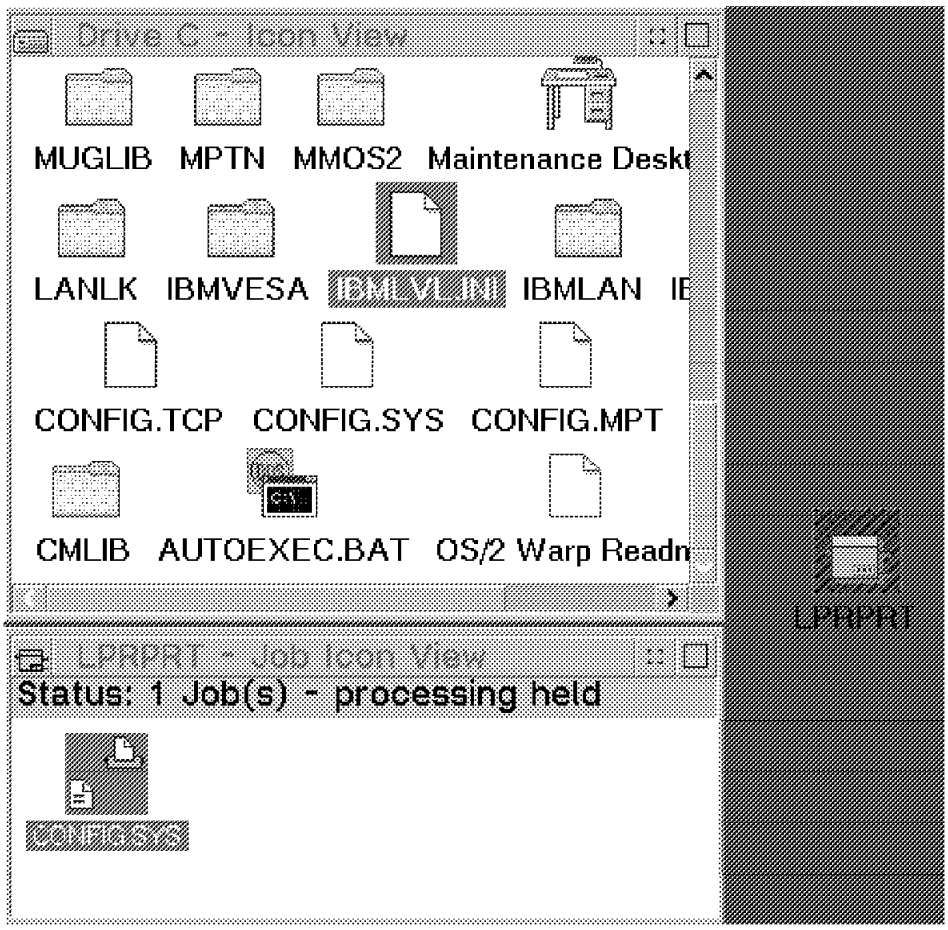


Figure 177. Printing to an LPR Port

Important Notice

Before you can actually print to printers connected to LPR ports, you must start the LPRPORTD server. See the next section about LPRPORTD.

You cannot print to LPR ports from WIN-OS2. Therefore, for WIN-OS2 applications you must still use LPRMON.

10.4.1 The LPR Port Driver

The LPRPORTD acts as the server for objects connected to remote printers via LPR port icons. This server must be started as a separate task before LPR port printers can be used. If LPRPORTD is not started you will receive a message that the remote printer is not online.

To start LPRPORTD, simply enter the following command:

```
start lprportd
```

10.5 Remote Printing from UNIX to OS/2

To use an OS/2 printer from an AIX LPR client, use SMIT to set up a remote printer queue.

Output from AIX commands can now be directed to the OS/2 printer. Note that there is no security mechanism in OS/2; any workstation can use an OS/2 LPD print server's printer(s).

The following shows how a remote print queue is defined to AIX using SMIT:

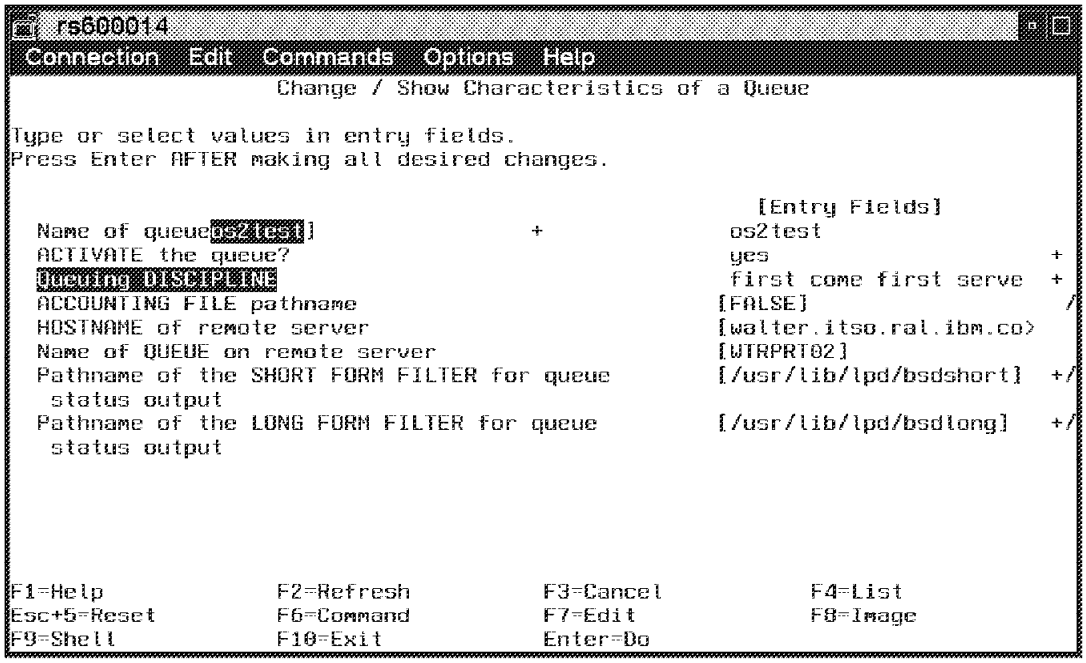


Figure 178. Configuration of a Remote Print Queue to AIX

Make sure that the OS/2 spooler and the TCP/IP server LPD are running in walter before you start printing.

The following picture shows sending a print job from AIX using SMIT. The file to be printed is /etc/qconfig on LPD server walter and its printer WTRPRT02:

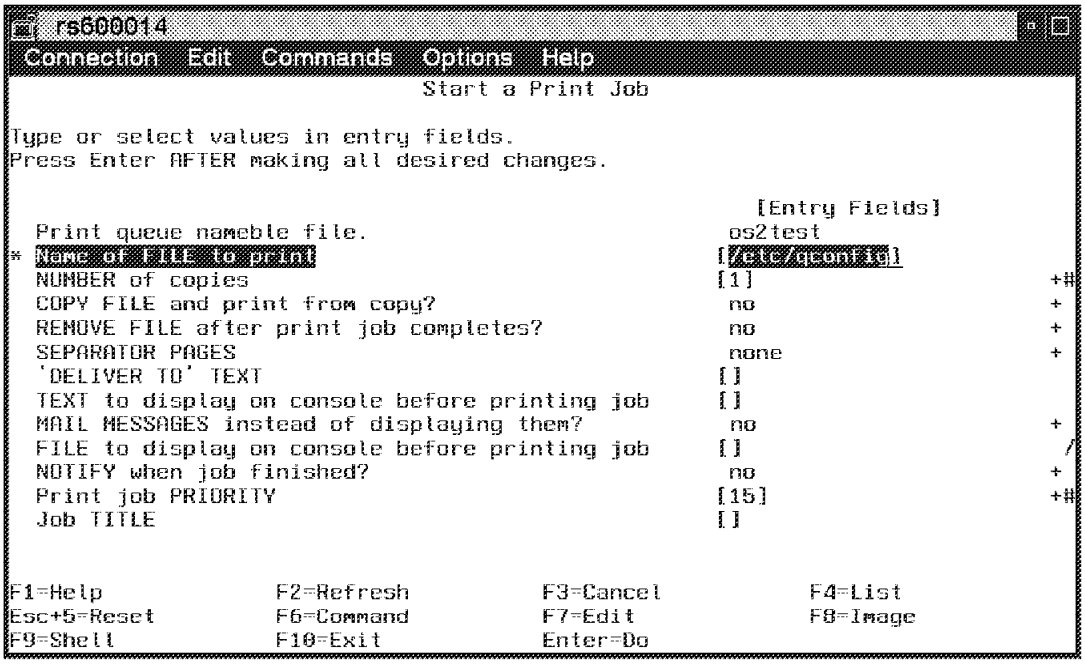


Figure 179. AIX SMIT Remote Print on OS/2 LPD Server

10.6 Remote Printing from OS/2 to UNIX

To make an AIX LPD print server accept your OS/2 print jobs, your hostname must be specified in the /etc/hosts and /etc/hosts.lpt files on the AIX system.

To print to the AIX printer, you have the following options:

- Explicitly use the LPR command
- Use LPRMON
- Use an LPR port printer

10.7 Remote Printing to/from VM

TCP/IP for VM has implemented an LPD server function. All the printers available to VM can be used from a TCP/IP LPR client.

Note: Only simple, line-oriented text (no graphics) can be printed out on such a connection.

The printer on the VM system has to be defined in the VM LPSERVE virtual machine. Our printer is available via RSCS, and the definitions are shown in the example LPD CONFIG file on System VM14.

```
SERVICE JULIO PRINTER
   RSCS SPOOL=TO RSCSV2
   TAG=PRT3287
   FILTERS f l p
   LINESIZE 132
   PAGESIZE 66
OBEY TCPMAINT
;DEBUG
```

The lpr command on OS/2 allows you to print a text document on a remote TCP/IP system that runs an LPD server. An OS/2 file can be printed on the VM printer with the following command:

```
[C:]lpr config.sys -p JULIO -s vm14

Printing C:\config.sys:
Trying LPD print server vm14.itsc.raleigh.ibm.com(9.67.32.18), device JULIO.
Sent 2008 bytes.
The entire document was sent.

[C:\]
```

To print the VM file all notebook a on the OS/2 LPD print server with the IP address 9.67.38.81, enter the command:

```
lpr all notebook a (PRINTER lpt1 HOST 9.67.38.81
Ready;
lpq (PRINTER lpt1 HOST 9.67.38.81

JobID     File Name      Rank    Size    Status
-----     ---------      ----    ----    ------
  7      ALL.NOTEBOOK     1      1071    Printing TCPMAINT@RALYESA
```

The LPQ command shows you the status of the print job in the print queue.

Note: If the port of the OS/2 LPD server machine is redirected to an OS/2 LAN Server, use the network printer queue name instead of the physical port.

10.8 Remote Print from DOS to OS/2

Set the DOS environment variables:

- LPR_PRINTER=lpt1 through lpt3
- LPR_SERVER=IP_address or host_name.

on the DOS workstation to define a default printer.

You can now print files on a printer connected to the OS/2 workstation by entering lpr file name at the DOS command prompt. If you want to use another print server in the network, you can use the lpr command as described in the online documentation. For example, to print the file AUTOEXEC.BAT on the LPT1 printer of the print server walter, enter the command:

```
lpr -p lpt1 -s walter autoexec.bat
```

10.9 The LPQ Command

To query jobs that are in the queue on a remote print server attached to the network, use the LPQ command. To see, for instance, how well the jobs from the sections before printed on printer LPT1 on host walter, enter:

```
lpq -p lpt1 -s walter
```

The results are taken from the OS/2 print manager for that queue (printer object) and may look like this:

```
JobID          File Name      Rank        Size          Status       Comment
-----          ---------      ----        ----          ------       -------
    1         CONFIG.SYS         1        4238          Queued   walter@walter
    3         startup.cmd        2          67          Queued   walter@walter
    6         /etc/qconfig       3        1516          Queued   root@rs60014
```

Notes:

1. The first job (1) was the CONFIG.SYS file dragged from the Drives folder to the printer object.
2. Job 6 was printed from an RS/6000.

10.10 The LPRM Command

This command is used to remove jobs from a remote printer queue. You can first use LPQ to see what jobs actually are in the queue, then select a job to remove by its job number (JobID). To remove, for instance, the jobs from the section above which did not print on printer lpt1 on host walter, enter:

```
lprm 1 3 -p lpt1 -s walter
```

The results may look like this:

```
JobID          File Name    Rank      Size        Status
-----          ---------    ----      ----        ------
    1         CONFIG.SYS       1      4238        -removed
    3         startup.cmd      2        67        -removed
```

Chapter 11. Remote Execution of Commands

This chapter describes the facilities of TCP/IP for OS/2 to execute commands on remote TCP/IP systems. TCP/IP for OS/2 gives you the server and client functions of:

- REXEC (Remote Execution)
- RSH (Remote Shell)

Both programs are similar because authorized clients can perform functions on a remote host. However, they differ in the authorization scheme.

The REXEC server (REXECD) serves any client that can correctly specify an existing account and the account password on the server machine. The OS/2 REXECD reads user ID and password values from the environment variables USER and PASSWD.

RSH reads a list that identifies the user on a specific remote client that has the same privileges as a particular server user. No password is required. The OS/2 RSH server (RSHD) reads the RHOSTS file in the MPTSETC directory. This file contains host names and users on those hosts that are authorized to access the server.

11.1 REXEC

The REXEC client can be used to execute commands on systems running the REXECD. Multiple commands can be entered by separating them with the “&” operator and enclosing the commands in double quotes. On VM systems, multiple commands can be entered by separating them with a semicolon (;).

11.1.1 Sample NETRC Files

To minimize the input at the command line, you can create a NETRC file. The login and password values are used by the REXEC command to log in to a foreign host using these values. The values defined in the NETRC file must match the environment variables USER and PASSWD defined on the foreign host. These values are case sensitive.

A description of the input to the NETRC file may be found in the online documentation.

We used the following NETRC on host klaus:

```
machine klaus login klaus password klaus
machine rs600014 login klaus password klaus
machine rs600011 login klaus password klaus
```

Note: This file, if it exists, is also used by the OS/2 FTP clients to log on to remote hosts automatically. Users have to make sure that login and password values in the local NETRC file match the following on the remote hosts specified by the machine tag:

- USER and PASSWD environment variables for REXEC
- User and password set for the user tag in the TRUSERS file for FTP.

If you do not want to specify these variables in the CONFIG.SYS file for security reasons, you must specify these variables at the OS/2 command prompt where you will start REXECD. You can also start the REXEC server using INETD.

The following command, issued from AIX, will change the current directory to MPTNETC, then list the contents of the directory and of the file TRUSERS:

```
rexec klaus "cd mptnetc & dir & type trusers"
```

The output from this command is displayed in the REXECD server window before it is actually sent to the client.

The OS/2 REXECD uses the environment variables USER and PASSWD to validate a request. Note that the values entered for these variables are case sensitive.

11.2 REXEC from DOS to OS/2

Set the following DOS environment variables on the DOS workstation:

USER User ID for REXECD in the OS/2 system - Contents of the OS/2 USER environment variable

PWD Password for REXECD in the OS/2 system - Contents of the OS/2 PASSWD environment variable

REXEC IP address of the OS/2 system

You can now enter `rexec <OS/2 command>` at the DOS command prompt to execute an OS/2 command on the OS/2 workstation. The following command will change the current directory to MPTNETC, then list the contents of the directory and the file TRUSERS:

```
rexec klaus "cd mptnetc & dir & type trusers"
```

The example shows the type of security issues one can face in allowing remote execution on an OS/2 workstation.

11.3 REXEC from OS/2 to VM

With the REXEC client in TCP/IP for OS/2, the OS/2 user can execute VM commands. Multiple commands can be entered in one invocation by separating them with semicolons. Normal redirection in OS/2 can be used. The following command will list the assigned VM disks and the current time on the host wtcesa:

```
[C:]rexec 9.12.14.1 -l grode -p xxxxx q disk;q t

LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL

GRO191 191  A   R/W    10 3390 4096       25       1405-78        395       1800

GDDM31 305  B   R/O    90 3390 4096      785       7328-45       8872      16200

RALDSK 5FF  C   R/O    75 3390 4096      261       4774-35       8726      13500

-      DIR  J   R/W     -    - 4096      319            -          -          -

S-DISK 190  S   R/O   100 3390 4096      489      10084-56       7916      18000

Y-DISK 19E  Y/S R/O   300 3390 4096     3058      40174-74      13826      54000

Z-DISK 19F  Z   R/O   400 3390 4096     7827      65180-91       6820      72000

TIME IS 15:13:55 EST THURSDAY 02/08/96
CONNECT= 05:34:01 VIRTCPU= 000:14.48 TOTCPU= 000:15.24
```

11.4 REXEC from MVS to OS/2

With the REXEC client in MVS, commands can be executed on an OS/2 system running the REXECD of TCP/IP for OS/2. The following command, issued on host mvs18sna, will list the contents of the C:TCPIPTMP directory on host walter:

```
 READY
rexec 9.24.104.77 dir c:\tcpip\tmp
 TCPREXO30I userid (9.24.104.77:walter)
walter
 TCPREXO29I passwd (9.24.104.77:walter)


  The volume label in drive C is OS2.
  The Volume Serial Number is E720:9C14
  Directory of C:\tcpip\tmp

 10-25-95   4:47p     <DIR>           0  .
 10-25-95   4:47p     <DIR>           0  ..
  1-20-96   3:23p      2176           0  04900000.htm
  1-20-96   4:16p       146           0  04900001.gif
  1-20-96   4:53p     12980           0  04900002.gif
          5 file(s)       1502 bytes used
                      25520384 bytes free
 READY
```

11.5 REXEC between OS/2 and UNIX

TCP/IP for OS/2 supports both the REXEC client and the REXECD server. Thus, AIX users can execute commands on an OS/2 REXECD and OS/2 users can execute commands on an AIX REXEC server. The following screen shows an example of a remote command execution at host rs600014 from an OS/2 REXEC client:

```
[C:]rexec rs600014 ls -ls

total 88
-rw-r--r--   1 root     staff       3746 Feb 10 13:07 CONFIG.SYS
-rw-r--r--   1 root     staff       3746 Feb 10 12:56 config.sys
-rw-r--r--   1 root     staff      31977 Feb 10 15:49 smit.log
-rw-r--r--   1 root     staff       2787 Feb 10 15:48 smit.scrip

[C:\]
```

If you have the X Windows Server running on your OS/2 workstation, you can open an AIX command shell window on your desktop with the following command (rather than starting it from a Telnet session):

```
rexec rs600014 aixterm -d walter:0
```

assuming that rs600014 is the AIX system where the X window is being sent from, and walter is your OS/2 system.

You may have trouble with the backslash character in OS/2, when you execute an OS/2 command from certain UNIX workstations. This can be solved by enclosing the command in double quotes. The following sample shows the input of a rexec command at an AIX command prompt to be executed at OS/2 REXEC server klaus:

```
$ rexec klaus "type c:mptnetcnetrc"
Name (klaus:klaus): klaus
Password (klaus:klaus):

machine klaus login klaus password klaus
machine rs600014 login klaus password klaus
machine rs600011 login klaus password klaus
$
```

11.6 Executing a Command on a Foreign Host - RSH

To run the OS/2 RSHD server, an RHOSTS file must exist. This file can be created manually in the directory where the ETC environment variable points to, or by using the TCP/IP for OS/2 Configuration Notebook. The RHOSTS file contains the fully qualified hostnames of the authorized client machines and if needed, the user names of the users who are allowed to use RSHD from that client machine. If no users are defined per machine, then every user that uses that machine has access to RSH services on the RSHD server.

The RSHD server compares the hostname and user field in the request frame (sent by the RSH client) with the entries in the RHOSTS file. If they are identical, the remote machine and the remote user have access to RSHD services.

The RHOSTS file on host klaus could look like the following:

```
walter klaus
```

In a network with a domain name server, the RHOSTS file would look like the following:

```
rs600014.itso.ral.ibm.com klaus
dos5.itso.ral.ibm.com
dos3.itso.ral.ibm.com
```

That means that every user on the hosts dos3 and dos5 can execute commands on host klaus. But only user klaus at machine rs600014 can obtain RSH service. The entries in this file are case sensitive.

The RSHD server can be started as an INETD subserver task or as a foreground task.

The following example shows the RSH client on AIX requesting RSHD service on the OS/2 RSHD server klaus:

```
<rs600014># rsh klaus type c:autoexec.bat

@ECHO OFF
ECHO.
PROMPT $i$p$g
REM SET DELDIR=C:\DELETE,512;D:\DELETE,512;E:\DELETE,512;
PATH C:\OS2;C:\OS2\MDOS;C:\OS2\MDOS\WINOS2;C:\;
LOADHIGH APPEND C:\OS2;C:\OS2\SYSTEM
REM LOADHIGH DOSKEY FINDFILE=DIR /A /S /B $*
REM SET DIRCMD=/A

<rs600014>#
```

A user on the AIX system rs600014 other than klaus will get the following message:

```
<rs600014># rsh klaus type rhosts

Unauthorized Request rejected.
<rs600014>#
```

If a user name is defined for a hostname in the RHOSTS file, the USER environment variable on that host has to be set to that exact value. Both values must match, even in length.

If the USER environment variable at an OS/2 RSH client is set to blank (set user=), a default user name OS2USER is used instead.

Note: There is no security when the RSHD server is running. If a remote user learns the hostnames in the RHOSTS file, that remote user can execute *any* commands on your workstation. If you have a TELNET, REXEC, TFTP, RSH, or FTP server running on your machine, and you have created a NETRC file, it provides user and password information to a foreign user who can then access other user's files.

11.6.1 OS/2 RSH Client to AIX

To execute the AIX cat command on host rs600014, the command looks like this:

```
[C:]rsh rs600014 -l passwd -n "cat .profile"

PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:$HOME/bin:/usr/bin/X11:/sbin:.

export PATH

if [ -s "$MAIL" ]      # This is at Shell startup.  In normal
then echo "$MAILMSG"        # operation, the Shell checks
fi                          # periodically.
export DISPLAY=9.24.104.191:0
stty erase ^H
set -o vi

[C:\]
```

You cannot use RSH without a command and logon to the remote system as you can with the AIX RSH client. The OS/2 RSH client requires the user to specify a command.

11.6.2 OS/2 RSH Client to VM

To execute a command on a VM RSHD server with RACF started, you have to set the USER environment variable to a valid VM user ID and pass the password by entering it with the `-l` option at the RSH command. For example, if you want to ask the time at the VM system *9.12.14.1* with the user ID tcpmaint and the password route66, the command looks like this:

```
SET USER=TCPMAINT
```

```
[C:]rsh 9.12.14.1 -l route66 "q t"

TIME IS 15:13:55 EST THURSDAY 02/08/96
CONNECT= 05:34:01 VIRTCPU= 000:14.48 TOTCPU= 000:15.24
```

If the VM RSHD server runs without RACF, you have to set the USER environment variable to a valid VM user ID and enter the RSH command without any options.

```
SET USER=TCPMAINT
```

```
[C:]rsh 9.12.14.1 "q t"

TIME IS 15:13:55 EST THURSDAY 02/08/96
CONNECT= 05:34:01 VIRTCPU= 000:14.48 TOTCPU= 000:15.24
```

Chapter 12. DOS/Windows Access

DOS/Windows Access consists of the following:

- A virtual device driver
- A set of TCP/IP Version 2.1.1 for DOS dynamic link libraries
- A set of TCP/IP Version 2.1.1 for DOS programs

The virtual device driver provides access to TCP/IP protocol stacks running under OS/2, and the dynamic link libraries (DLL) contain functions to access the virtual device driver. The DOS programs use functions in the DLLs to send and receive information to the TCP/IP protocol stack.

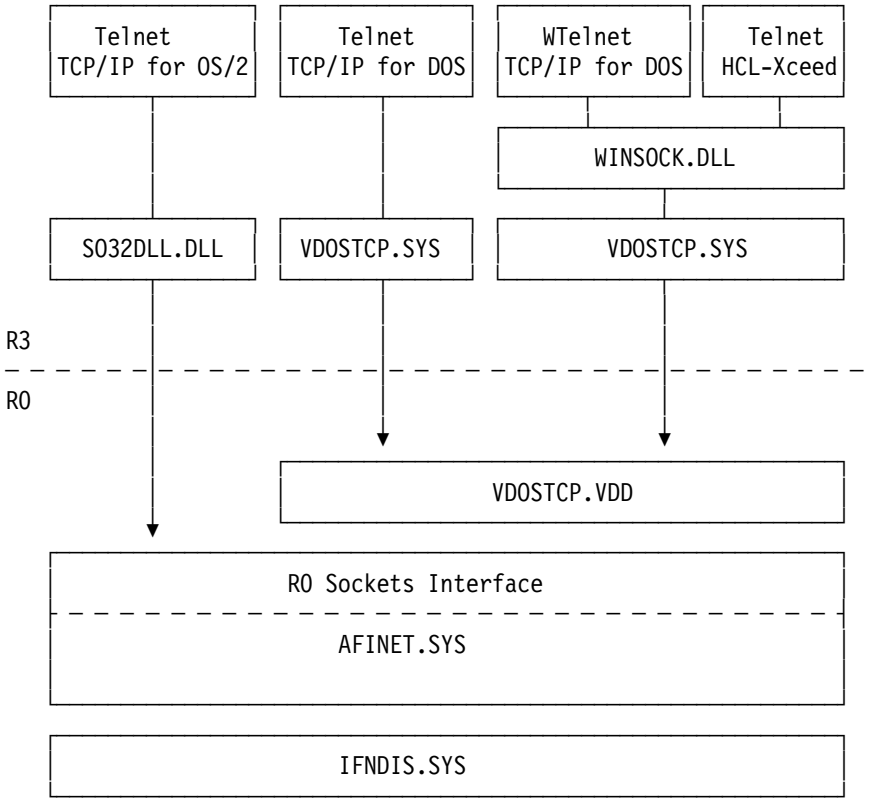


Figure 180. DOS/Windows Access Protocol Stack

DOS/Windows Access enables you to run your applications from the following:

- DOS (VDM) session
- WIN-OS2 session

The virtual device driver and dynamic link libraries have been developed to support any DOS or Windows application which has been developed using the following TCP/IP Version 2.1.1 for DOS application programming interfaces:

Library	Description
SOCKETS.LIB	Real mode small model sockets library
SOCKETL.LIB	Real mode large model sockets library

RPCS.LIB	SUN RPC small model library
RPCL.LIB	SUN RPC large model library
FTPAPI.LIB	FTP Application Programming Interface library
WINSOCK.LIB	Windows Sockets API V1.1 library
WFTPAPI.LIB	Windows FTP Application Programming Interface library
BIOS int14h	BIOS Interrupt 14 Telnet Redirector

We used and tested the following applications which meet these specifications:

- IBM TCP/IP V2.1.1 for DOS
- Hummingbird HCL-eXceed/W V3.3.3

12.1 Configuration

As described in Chapter 2, “Functional Overview of TCP/IP for OS/2” on page 19, DOS/Windows Access is an integral part of the TCP/IP V3.x for OS/2 base.

During installation, the following system modifications are made that you should be aware of:

- DOS Settings
- AUTOEXEC.BAT
- Directory Structure
- ETC Environment Variable
- Name Resolution

12.1.1 DOS Settings

During installation the settings of DOS and WIN-OS2 objects on your system are modified to include the following device statement:

```
DOS_DEVICE C:TCPIPBINVDOSTCP.SYS
```

12.1.2 AUTOEXEC.BAT

During installation, your AUTOEXEC.BAT will be modified to include the following:

- C:TCPIPDOSBIN in your path statement
- SET ETC=C:TCPIPDOSETC

12.1.3 Directory Structure

During installation, the following directory structure is set up for DOS/Windows Access.

```
TCPIPDOSETC Configuration and Name Resolution Files
\TCPIP\DOS\BIN\ Programs and DLLs
```

12.1.4 ETC Environment Variable

Both OS/2 and DOS TCP/IP applications use the ETC environment variable to find the path to the TCP/IP configuration and name resolution files. The OS/2 TCP/IP installation program initializes the ETC variable by creating the following entry in CONFIG.SYS, on the boot drive:

```
SET ETC=<path>ETC
```

DOS/Windows Access initializes the ETC variable by creating the following entry in in the AUTOEXEC.BAT file on your boot drive:

```
SET ETC=<path>DOSETC
```

The two different directories are used because some of these files have different formats in OS/2 and DOS systems.

With DOS/Windows Access installed, the following files are stored in the <path>DOSETC directory:

File	Description
RESOLV	Domain name server name resolution
HOSTS	HOSTS file name resolution
SERVICES	Service name to port number mapping
PROTOCOL	Protocol name to protocol ID mapping

12.1.4.1 Name Resolution

The DOS TCP/IP programming interfaces use name resolution files like those used by OS/2 TCP/IP. These files are as follows:

File	Description
RESOLV	Domain name server name resolution
HOSTS	HOSTS file name resolution

With DOS/Windows Access installed the ETC environment variable is set (in the AUTOEXEC.BAT file on the boot drive) to the directory that contains the name resolution files. This is usually the <path>DOSETC subdirectory. If you change the ETC path in AUTOEXEC.BAT, or if you would like to use a HOSTS file, you must update one or both of the RESOLV and HOSTS files pointed to by the ETC path in AUTOEXEC.BAT. In most cases, you can just copy the corresponding files from the ETC path specified in CONFIG.SYS to the ETC path specified in AUTOEXEC.BAT.

If you do not modify the ETC environment variable in AUTOEXEC.BAT (to point to a path other than the path set during installation), the configuration notebook (TCPIPCFG) will update the RESOLV file when you change the configuration notebook pages that correspond to the RESOLV file. The configuration program does this by copying the RESOLV file from the ETC path specified in CONFIG.SYS to the ETC path specified in AUTOEXEC.BAT after it makes any changes.

12.2 Using DOS/Windows Access

This section describes how to use DOS and Windows TCP/IP applications under OS/2 with TCP/IP for OS/2 and DOS/Windows Access.

12.2.1 DOS Programs supplied with TCP/IP for OS/2

PING is the only DOS TCP/IP program supplied with the DOS/Windows Access. This can be run from a DOS window:

```
C:\>ping rs600014 10 1
PING rs600014.itso.ral.ibm.com (9.24.104.191): 56 data bytes
64 bytes from 9.24.104.191: icmp_seq=0 ttl=255 time=30 ms
64 bytes from 9.24.104.191: icmp_seq=1 ttl=255 time=30 ms
64 bytes from 9.24.104.191: icmp_seq=2 ttl=255 time=0 ms
64 bytes from 9.24.104.191: icmp_seq=3 ttl=255 time=30 ms
64 bytes from 9.24.104.191: icmp_seq=4 ttl=255 time=30 ms
64 bytes from 9.24.104.191: icmp_seq=5 ttl=255 time=30 ms


--- rs600014.itso.ral.ibm.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0/25/5000 ms
```

12.2.2 TCP/IP Version 2.1.1 for DOS

TCP/IP Version 2.1.1 for DOS can be installed on a system already using TCP/IP for OS/2. This will allow you to run many of the DOS and Windows applications provided with TCP/IP Version 2.1.1 for DOS in DOS and WIN/OS2 sessions of OS/2. The following are examples of these applications:

WTELNET.EXE
TELNET.EXE

Although these applications will work with the DOS/Windows Access, this is not the primary purpose of DOS/Windows Access, as most of these applications are provided with TCP/IP for OS/2 in OS/2 versions. The main purpose of DOS/Windows Access is to allow you to run other DOS and Windows applications which are not already available for OS/2.

You should note that the protocol stack and application programming interfaces provided with TCP/IP Version 2.1.1 for DOS should not be used with a machine already equipped with TCP/IP for OS/2 and DOS/Windows Access since this provides the same functions through virtual device drivers and a set of its own dynamic link libraries.

12.2.2.1 TCP/IP for DOS Configuration for DOS 5.01

We have included this configuration so that you can see the differences between the configurations required for the DOS and TCP/IP for OS/2 environments. Listed below is the configuration information for a machine running TCP/IP V2.1.1 for DOS configuration on DOS 5.01.

CONFIG.SYS

```
shell=c:doscommand.com /e:2000 /p
files= 50
DEVICE=C:\DOS\SETVER.EXE
DEVICE=C:\WINDOWS\HIMEM.SYS
DOS=HIGH
COUNTRY=001,,C:\DOS\COUNTRY.SYS
DEVICE=C:\WINDOWS\SMARTDRV.EXE /DOUBLE_BUFFER
DEVICE = C:\TCPDOS\BIN\PROTMAN.DOS /I:C:\TCPDOS\ETC
DEVICE = C:\TCPDOS\BIN\DOSTCP.SYS
DEVICE = C:\TCPDOS\BIN\IBMTOK.DOS
STACKS=9,256
device=c:\dos\ansi.sys
lastdrive=z
```

AUTOEXEC.BAT

```
@echo off
c:\tcpdos\bin\NETBIND
C:\WINDOWS\SMARTDRV.EXE
SET ETC=C:\TCPDOS\ETC
SET COMSPEC=C:\DOS\COMMAND.COM
SET TCPBASE=C:\TCPDOS
@ECHO OFF
PROMPT $P$G
PATH C:\DOSLAN;C:\WINDOWS;C:\DOS;C:\TCPDOS\BIN;C:\UTIL;C:\CMD;
SET TEMP=C:\DOS
KEYB US,,C:\DOS\KEYBOARD.SYS
doskey /reinstall
CALL TCPSTART
@ECHO OFF
YNPROMPT Y N 30 Start DOS LAN Requester (Y/N)?
IF ERRORLEVEL 1 GOTO NODLR
NET START
IF ERRORLEVEL 1 GOTO NODLR
CALL INITFSI.BAT
:NODLR
```

TCPSTART.BAT

```
:=-------------------------------------------------------------------------=:
:=- TCPSTART batch file                                                   -=:
:=- begins by determining that the necessary environment variables are set  -=:
:=-------------------------------------------------------------------------=:
@echo off
IF %ETC%.==. GOTO ETCHELP
INETCHK
IF ERRORLEVEL 1 GOTO INET_DOWN
GOTO INET_UP
:INET_DOWN
:=-------------------------------------------------------------------------=:
:=- install the Protocol Stack                                             -=:
:=-------------------------------------------------------------------------=:
   IF %INET%.==. inet
   IF NOT %INET%.==. inet -d %INET%
   if errorlevel 1 GOTO done
:=-------------------------------------------------------------------------=:
:=- Assign our IP address and set NETMASK and default ROUTE                -=:
:=-------------------------------------------------------------------------=:
route -f
arp -da
```

```
REM slipdial
ifconfig nd0 9.24.104.77
netmask 255.255.255.0 broadcast 9.24.104.255 up
route add -hopcount 1 -mtu 1496 default 9.24.104.1
PING -c1 9.24.104.1>nul
REM ftpd -b
REM if errorlevel 0 echo ..... FTP Daemon Started
REM lpd -b -c
REM if errorlevel 0 echo ..... LP Daemon Started
nbtcp
if errorlevel 0 echo ..... NetBIOS Started
REM routed
REM if errorlevel 0 echo ..... ROUTED Started
   echo.
   echo TCP/IP is enabled.

:=-------------------------------------------------------------------------=:
REM  USER CUSTOMIZATION SECTION
REM
REM  WARNING:  Do not remove the lines, 'REM  Begin_User_Customization'
REM            and 'REM  End_User_Customization' because these are used
REM            to protect any user-added commands from being deleted by
REM            the CUSTOM program
REM
REM  Begin_User_Customization
REM  End_User_Customization
   goto DONE
:=-------------------------------------------------------------------------=:
:INET_UP
   ECHO TCP/IP is already up!
   goto DONE
:=-------------------------------------------------------------------------=:
:=- Provide any needed HELP                                              -=:
:=-------------------------------------------------------------------------=:
:ETCHELP
   ECHO You MUST set the· 1;33;40m ETC··0;37;40m  variable
   ECHO before attempting to start TCPPLUS
:DONE
```

12.2.2.2 TCP/IP for DOS Configuration with DOS/Windows Access

You will notice that there is very little setup required to run DOS or Windows applications with DOS/Windows Access. The following are examples of configurations:

CONFIG.SYS: You should ensure that this device driver is included in the DEVICE area of your DOS Settings of the VDM:

```
DOS_DEVICE C:TCPIPBINVDOSTCP.SYS
```

AUTOEXEC.BAT: You should ensure that your AUTOEXEC.BAT has the necessary path information to your TCP/IP Version 2.1.1 for DOS applications.

```
@ECHO OFF
ECHO.
PROMPT $i$p$g
REM SET DELDIR=C:\DELETE,512;
PATH=C:\OS2;C:\OS2\MDOS;C:\OS2\MDOS\WINOS2;C:\;C:\TCPIP\DOS\BIN;C:\TCPDOS\BIN
LOADHIGH APPEND C:\OS2;C:\OS2\SYSTEM
SET TMP=C:\
```

```
REM LOADHIGH DOSKEY FINDFILE=DIR /A /S /B $*
REM DOSKEY EDIT=QBASIC/EDITOR $*
REM SET DIRCMD=/A
SET TEMP=C:\OS2\MDOS\WINOS2\TEMP
SET ETC=C:\TCPIP\DOS\ETC
```

TCPSTART.BAT: You do not need to execute this batch file. These functions are provided by TCP/IP for OS/2.

12.2.2.3 Restrictions

Although the DOS/Windows access supports all of the documented TCP/IP Version 2.1.1 for DOS APIs, there are some applications, shipped with TCP/IP Version 2.1.1 for DOS, that are not supported by the DOS/Windows Access because they manage the DOS TCP/IP protocol stack (INET). Because the DOS/Windows Access lets you run on the OS/2 TCP/IP stack, there is no DOS stack to manage. The following table lists DOS TCP/IP applications that are not supported in a VDM, and the corresponding (suggested) OS/2 TCP/IP applications.

Table 22. Restrictions

DOS Application	OS/2 Application	Description
ifconfig	ifconfig	Used to manage network interfaces
arp	arp	Used to manage ARP table entries.
netstat	netstat	Used to query internal structures
route	route	Used to manage static routes
routed	routed	RIP support
snmpd	snmpd	SNMP Agent support
inet -d <flg>	vdebug <flg>	Internal protocol stack tracing

12.2.3 HCL-eXceed/W

We installed HCL-eXceed/W V3.3.3 on a system with the following software components already installed:

- OS/2 Warp Server
- TCP/IP for OS/2
- DOS/Windows Access

HCL-eXceed/W provides a suite of TCP/IP applications such as the following:

- Telnet
- FTP
- PING

It also provides X client and X server functions, so that you can run X Windows applications in a WIN-OS2 session. It does not provide any transport layer functions. HCL-eXceed/W supports several products as transports, for example:

- IBM TCP/IP for DOS
- Novell LAN Workplace for DOS
- Microsoft DOS TCP/IP

Normally, you need to install one of these products to provide the transport layer functions to HCL-eXceed. However with our setup, the transport layer is provided by TCP/IP for OS/2 and DOS/Windows Access. You should set up your CONFIG.SYS and AUTOEXEC.BAT as shown in 12.2.2, “TCP/IP Version 2.1.1 for DOS” on page 288.

During the installation process of HCL-eXceed/W, you should select WINSOCK API as your TCP/IP software. You should also ensure that XPORT.DLL and WINSOCK.DLL are included in your DOS SEARCH PATH before you start WIN/OS2.

We were able to use HCL-eXceed/W as a full-screen and seamless WIN-OS2 session. These are the steps that we had to perform in order to use HCL-eXceed/W seamless:

1. Create a program called Program Manager on your OS/2 desktop with the following properties:

Parameter	Value
Path and file name	PROGMAN.EXE
Working Directory	C:OS2MDOSWINOS2
Session	WIN-OS2 window

2. Start the Program Manager from your OS/2 desktop.
3. Select **Window** from the action bar of the Program Manager.
4. Select **eXceedW** from the pull-down menu.
5. Double-click on the object eXceed/W in your Program Manager. This will start the X server.
6. Double-click on the object **Telnet** in your Program Manager.
7. Type in the name of the host on which you would like to establish this Telnet session.
8. Once the Telnet session is established you must type in your user ID and password.
9. You will see the greeting messages from the host on the Telnet session. We connected to a RS/6000 running AIX with an address of 9.24.104.191.
10. We started an X Window terminal session by running this command:

    ```
    aixterm -d 9.24.104.191:0 &
    ```

11. Select the X Windows terminal session on your desktop.
12. We started several X Windows applications.
13. The OS/2 desktop should look similar to Figure 181 on page 293.

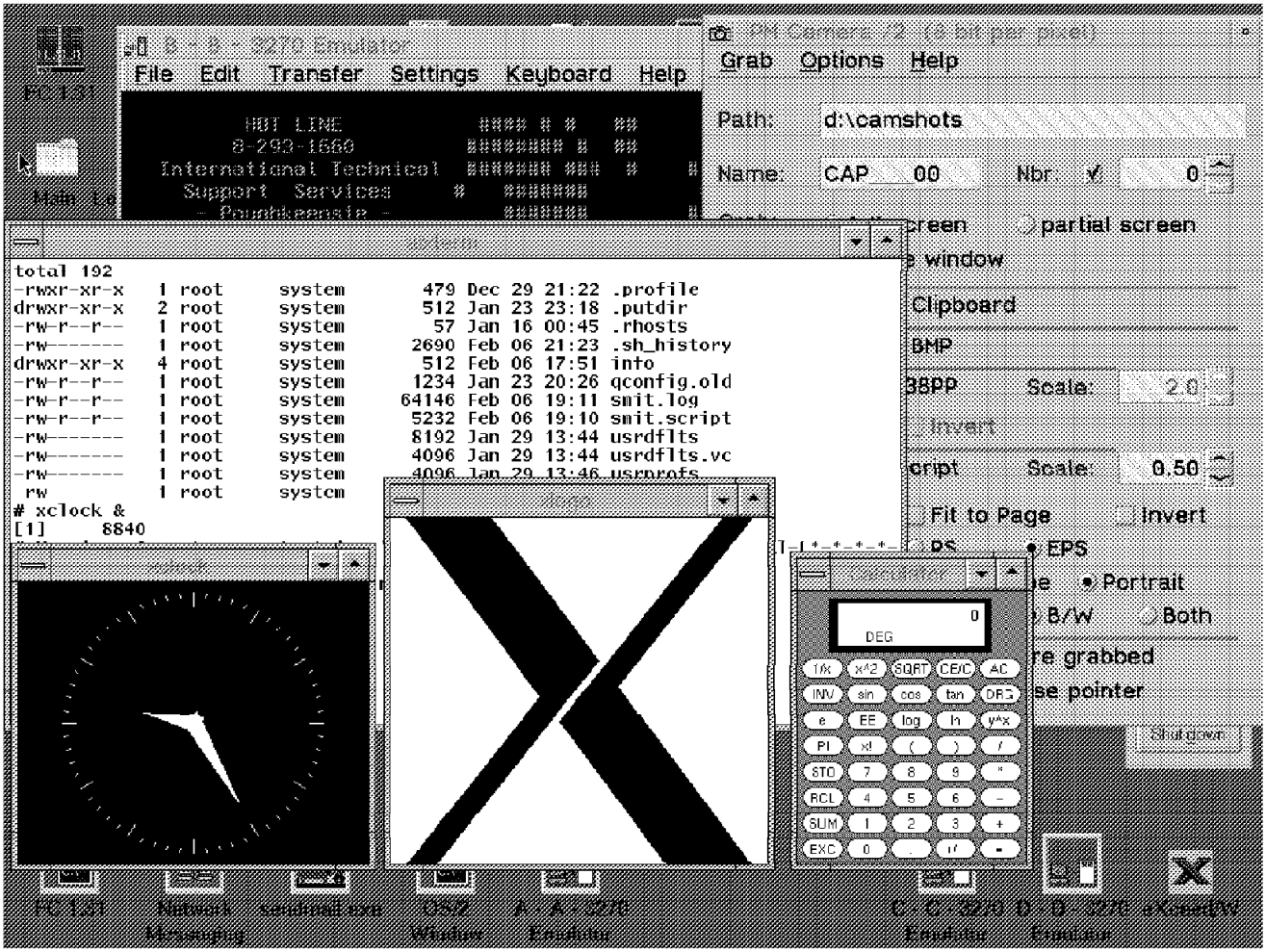


Figure 181. HCL-eXceed/W Example

14. We have just created several sessions in the preceding steps, all of which are WIN/OS2 sessions. You can check this by examining your OS/2 task list.

In the example described above, we only used the X Server functions of HCL-eXceed. But you can also use HCL-eXceed/W as an X Client.

12.3 Summary

The following tables show which types of TCP/IP applications you can use with DOS/Windows Access:

Table 23 (Page 1 of 2). Summary of TCP/IP Functions Available in DOS, Windows and OS/2

	DOS Applications Installed using DOS/Windows Access		
OS/2 TCP/IP Client	**None**	**IBM TCP/IP V2.1.1 for DOS**	**HCL-eXceed**
Telnet	O	ODW	OW
PM-ANT,TN3270	O		
FINGER	O	OD	O
TALK	O	OD	O

Table 23 (Page 2 of 2). Summary of TCP/IP Functions Available in DOS, Windows and OS/2

OS/2 TCP/IP Client	DOS Applications Installed using DOS/Windows Access		
	None	IBM TCP/IP V2.1.1 for DOS	HCL-eXceed
FTP	O	ODW	OW
TFTP	O	OD	
LPR/LPRMON	O	OD	O
REXEC	O	OD	OW
RSH	O	OD	OW
SENDMAIL	O	OD	O
SNMP	O	O	O
NFS	O	OD	O
SOCKETS	O	OD	OW
RPC	O	OD	OW

Notes:

Abbrev	Description
O	OS/2 application
D	DOS application
W	Windows application

Table 24. OS/2 Server Interoperability with Other IBM Clients

OS/2 TCP/IP Server	DOS Applications Installed using DOS/Windows Access		
	None	IBM TCP/IP V2.1.1 for DOS	HCL-eXceed
TELNETD	O	O	O
X Windows	O	O	OW
TALKD	O	O	O
FTPD	O	O	O
TFTPD	O	O	O
LPD	O	OD	O
REXECD	O	OD	O
RSHD	O	OD	O
SENDMAIL	O	ODW	O
SNMPD	O	O	O
NFS	O	O	O

Notes:

Abbrev	Description
O	OS/2 application
D	DOS application
W	Windows application

Chapter 13. X Window System Server (PMX)

The X Window System Server (PMX) kit enables you to display and control X Window System client applications in one or more multiple OS/2 Presentation Manager (PM) windows. PMX is an implementation of the X11R5 Version of the X Window System and offers features such as backing-store (a feature designed to significantly reduce network traffic) and pseudo-color support using PM palette manager.

You can expand your TCP/IP V3.x for OS/2 with PMX services by installing the originally available X Window System Server kit for TCP/IP V2.0 for OS/2, and apply the latest corrective service.

This chapter describes the installation of PMX and its customization and usage to run X Window System client applications from OS/2 and AIX operating systems.

13.1 The X Window System

The X Window System is a distributed, window-based graphics system developed at Massachusetts Institute of Technology (MIT) in 1984. Today the X Window System is owned by the X Consortium Inc. which is an independent, not-for-profit membership corporation. The purpose of the X Consortium is to foster the development, evolution and maintenance of a comprehensive set of vendor-neutral, system architecture-neutral, network-transparent windowing and user interface standards. The X Window System Server kit supports Version 11 Release 5 (X11R5) of the X Window System server (X server) function.

The X server is a dedicated program that provides display services on a graphic terminal at the request of an X client program. An X client is the actual application program that sends its output to an X server in order to provide a graphical user interface. X clients send their output to remote X servers using TCP/IP as a transport. X clients can also run on an operating system different than the one running at the X server to where they send their data. The following shows the basic concepts of the X Window System. It illustrates where the necessary steps are performed to run the Xcalc application:

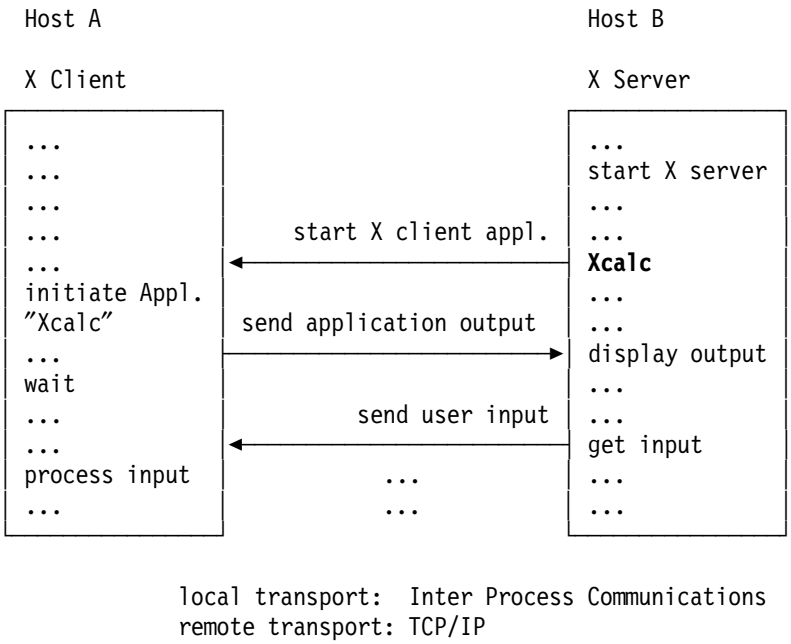


Figure 182. The X Window System Concept

Since many X clients can compete for an X server to display their data, an X Window manager is employed to resolve service conflicts.

PMX uses OS/2 Presentation Manager as the X Window manager and supports all of the keyboard, display, and pointer devices that are supported by OS/2 PM, and it can also use native PM fonts (but not DBCS fonts). 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 MOTIF) cannot act as the window manager for the OS/2 X server.

The X Window System server support comprises the following components:

- X Window System Server (PMX.EXE)
- X Window System Font Support
- X Window System Utilities
- National Language Support for Keyboards
- XDMCP support
- DHCP support for control of fontpath and XDMCP parameters

13.2 Installing and Configuring the X Window System Server

The X Window server component is contained in the originally available X Window System Server kit for TCP/IP V2.0 for OS/2. In order to use it with TCP/IP V3.x for OS/2, it requires that you also install a corrective service diskette (CSD) for the PMX kit. At the time of writing the latest PMX CSD is UN86625. The functional examples we document in this chapter are based upon this level of the PMX code.

To install the X Window System Server kit from product diskettes, insert the X Window System Server kit installation diskette into your diskette drive A: and enter the following command from an OS/2 command prompt:

```
A:TCPINST
```

Since the PMX kit complies with IBM′s Configuration, Installation, Distribution (CID) architecture (which provides for unattended, remote installation of programs and applications from code servers to client workstations) you can install the PMX kit remotely, from another workstation.

Notes:

1. The TCP/IP V2.0 for OS/2 Base kit or TCP/IP V3.x for OS/2 is a prerequisite for the X Window System Server kit.
2. The X Window System Server kit requires 11.7 MB of disk space.
3. The X Window System Server kit requires a large amount of memory. You should have at least 8 MB of memory on your machine and about 10 MB of swapping space on your hard disk, though the real amount of memory required by PMX depends on the X client applications that you are using.

After installing the X server support, you can use the TCP/IP Configuration Notebook or use PMX Configurations, Initial/Current Settings to customize the X Window System server to your requirements.

13.2.1 Configure PMX Using the TCP/IP Configuration Notebook

The Configuration Notebook is invoked by selecting the **TCP/IP Configuration** icon in the TCP/IP folder on your OS/2 desktop. It allows you to configure all TCP/IP components and services via a menu interface rather than manually changing several configuration files.

All the settings from the PMX part of the TCP/IP Configuration Notebook are saved to PMX.INI file in the directory specified by the ETC environment variable.

You can also get to the configuration pages for PMX from the running X server by selecting **Commands** followed by **Configuration** and then **Settings**.

The following pages briefly explain the nine configuration pages for PMX:

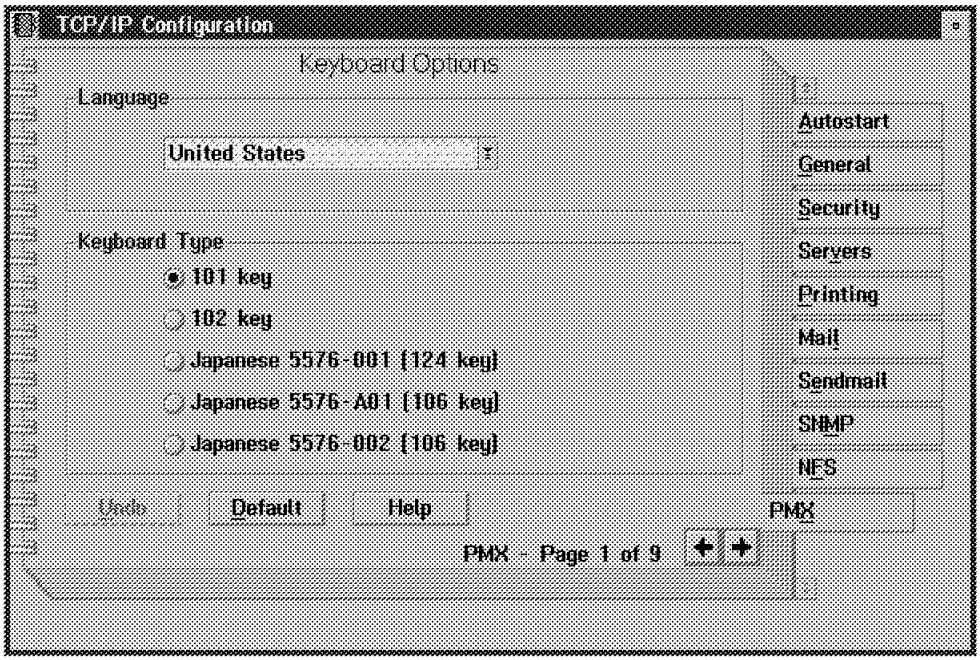


Figure 183. TCP/IP Configuration Notebook for PMX, Page 1

The first PMX configuration page is used for the following keyboard parameters:

Setting	**Meaning**
Language	Specify the keyboard mapping to use.
Keyboard Type	Specify the type of keyboard that is attached to your workstation.

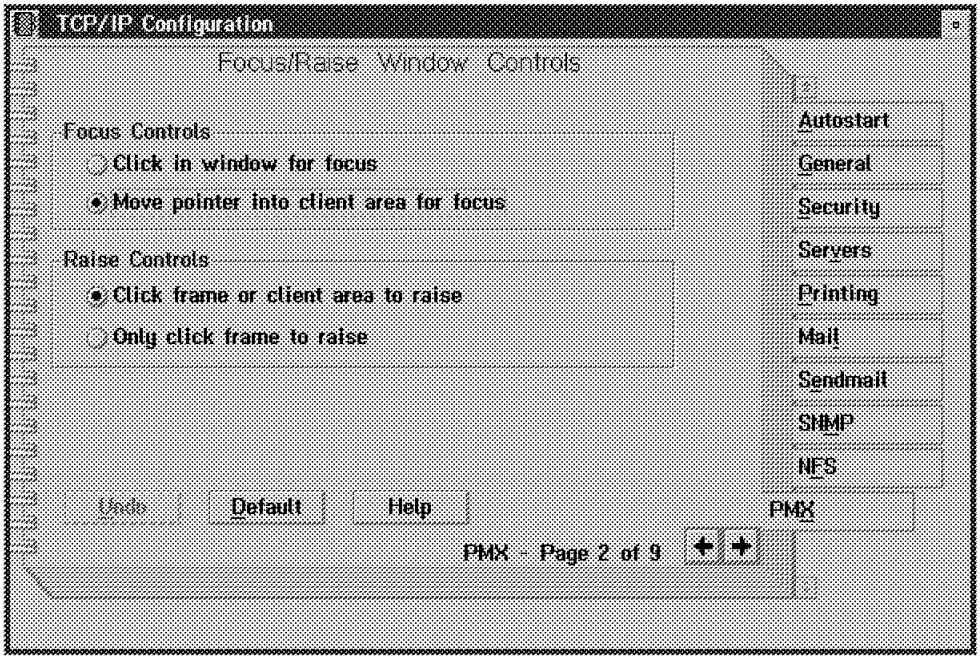


Figure 184. TCP/IP Configuration Notebook for PMX, Page 2

The second PMX configuration page is used for the following focus and raise control parameters:

Setting	Meaning
Focus Controls	Specify the preferred action to bring the focus to a PMX window on your OS/2 desktop. The normal behavior of OS/2 Presentation Manager is ″Click in window for focus″, whereas the default behavior of the X Window System is ″Move pointer into client area for focus″.
Raise Controls	Specify the preferred action to raise a PMX window to the top of the OS/2 desktop.

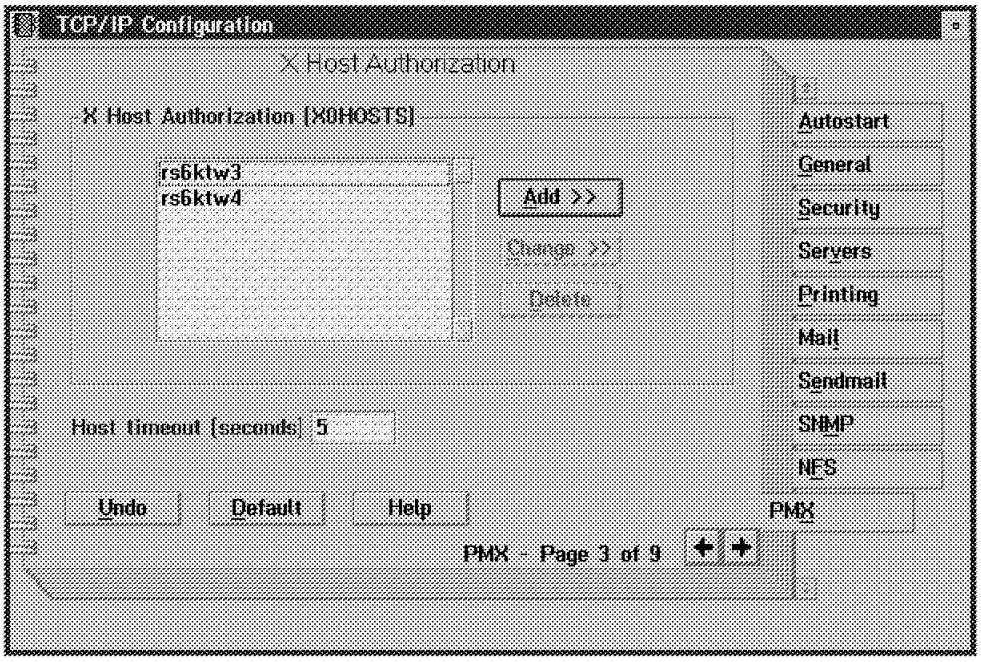


Figure 185. TCP/IP Configuration Notebook for PMX, Page 3

The third PMX configuration page is used for the following authorization parameters:

Setting	Meaning
X Host Authorization	Specify a list of hostnames that are allowed to send X client data to your X server. The entries are saved in the X0HOSTS file which is in the directory specified by ETC environment variable.
Host timeout	Specify the number of seconds to wait when initializing entries in the X0HOSTS file.

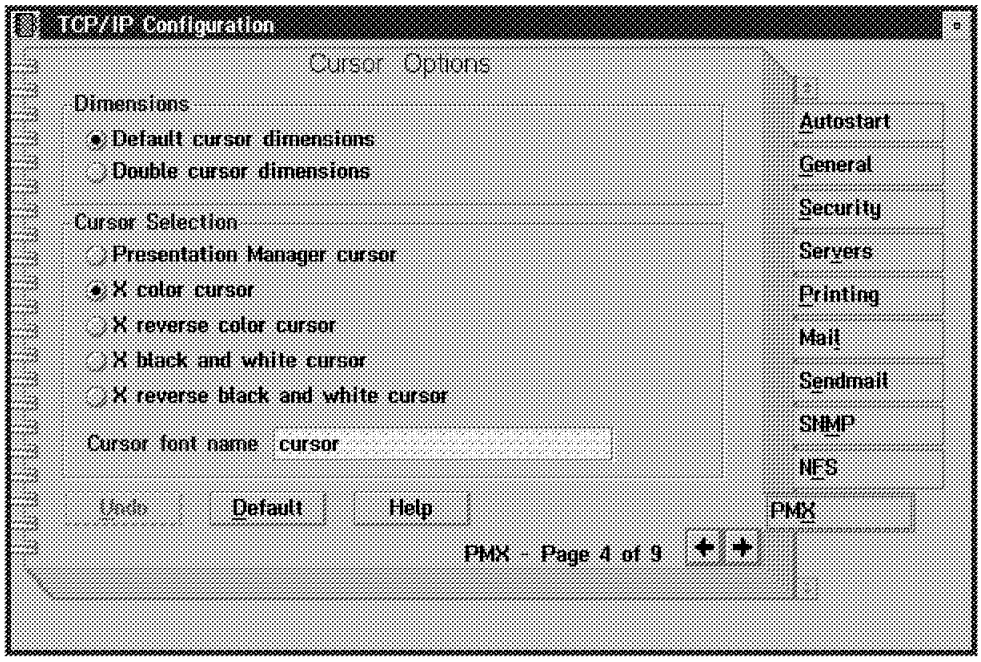


Figure 186. TCP/IP Configuration Notebook for PMX, Page 4

The fourth PMX configuration page is used for the following cursor parameters:

Setting	Meaning
Cursor Dimensions	Specify the size of a cursor.
Cursor Selection	Specify the type of cursor that you want the X server to use when displaying X client windows, and its color options. (The PM cursor is an arrow tilted towards the upper left corner; the X Window System cursor is an arrow tilted towards the upper right corner.)
Cursor Font name	Specify the cursor font to use for PMX.

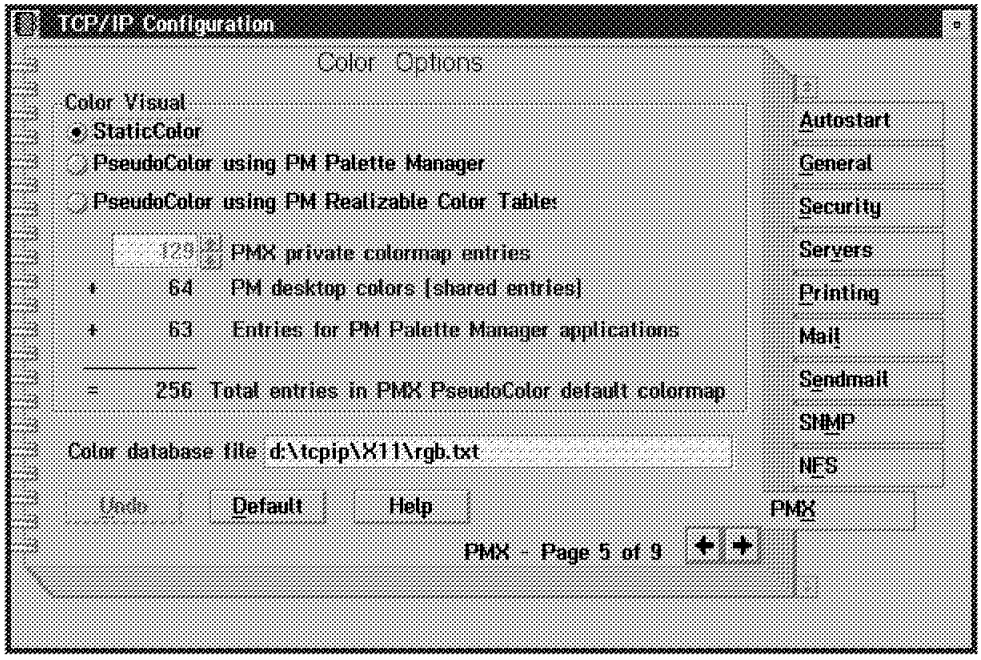


Figure 187. TCP/IP Configuration Notebook for PMX, Page 5

The fifth PMX configuration page is used for the color parameters:

Setting	Meaning
Color Visual	Specify how you want PMX to handle colors for X clients.
Color Database file	Specify the file that holds the color database.

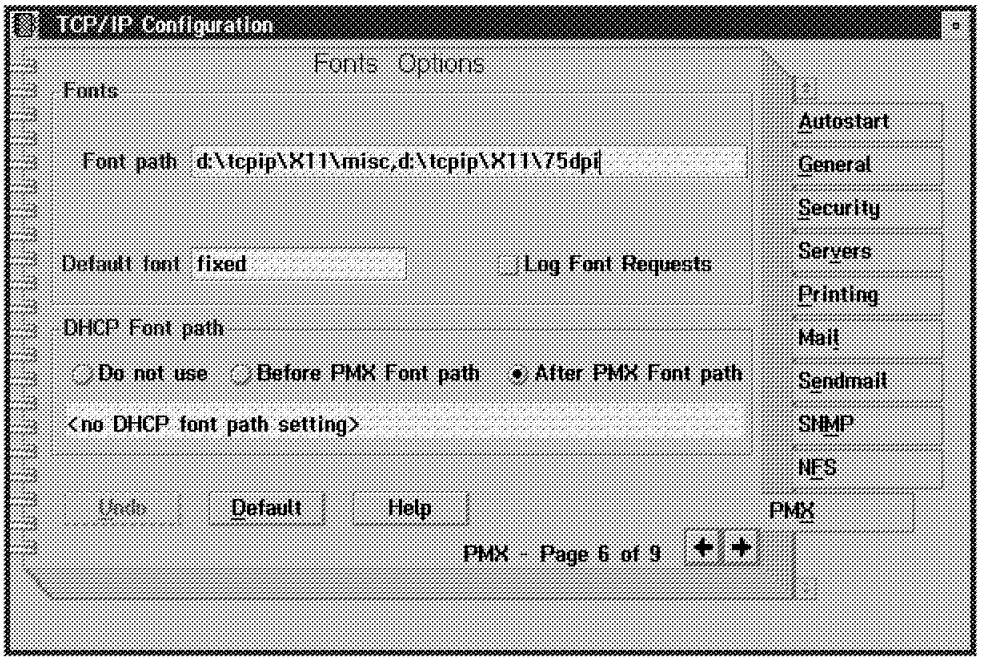


Figure 188. TCP/IP Configuration Notebook for PMX, Page 6

The sixth PMX configuration page is used for the following fonts parameters:

Setting	Meaning
Font path	Specify where PMX is to find the default fonts that it will use to display X clients' data.
Default font	Specify the default font that PMX is using.
DHCP Font path	Select one of the radio buttons to have the DHCP font path searched before or after the PMX font path or to not use it.

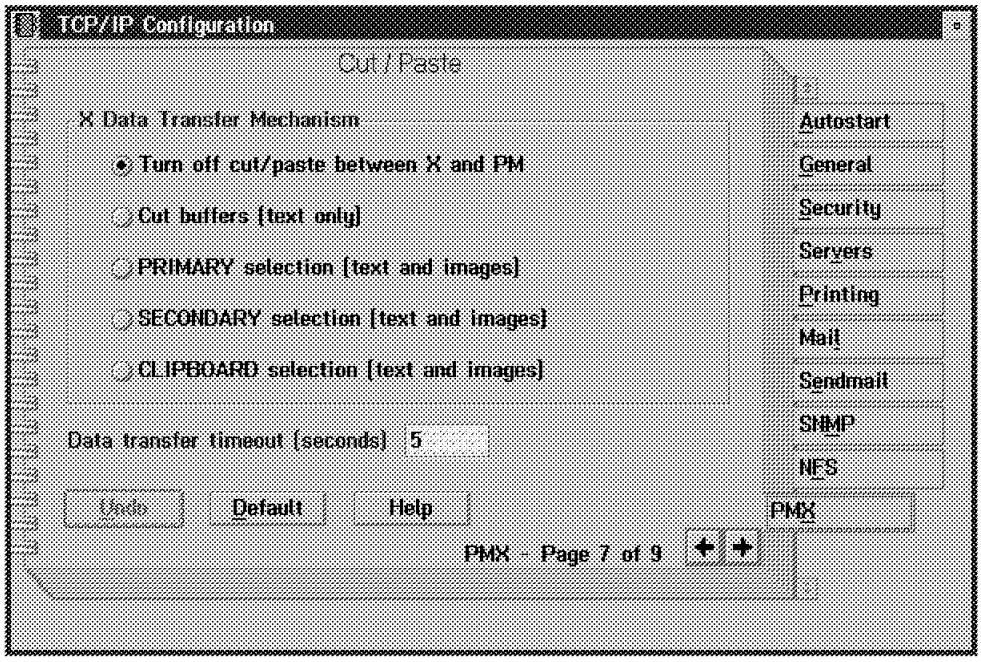


Figure 189. TCP/IP Configuration Notebook for PMX, Page 7

The seventh PMX configuration page is used for the following cut and paste parameters:

Setting	Meaning
X Data Transfer Mechanism	Specify whether and how you want to transfer data between OS/2 Presentation Manager and X Window System client applications.
X Data Transfer Timeout	Specifies how long PMX should wait for data that it has requested from an X client for data transfer.

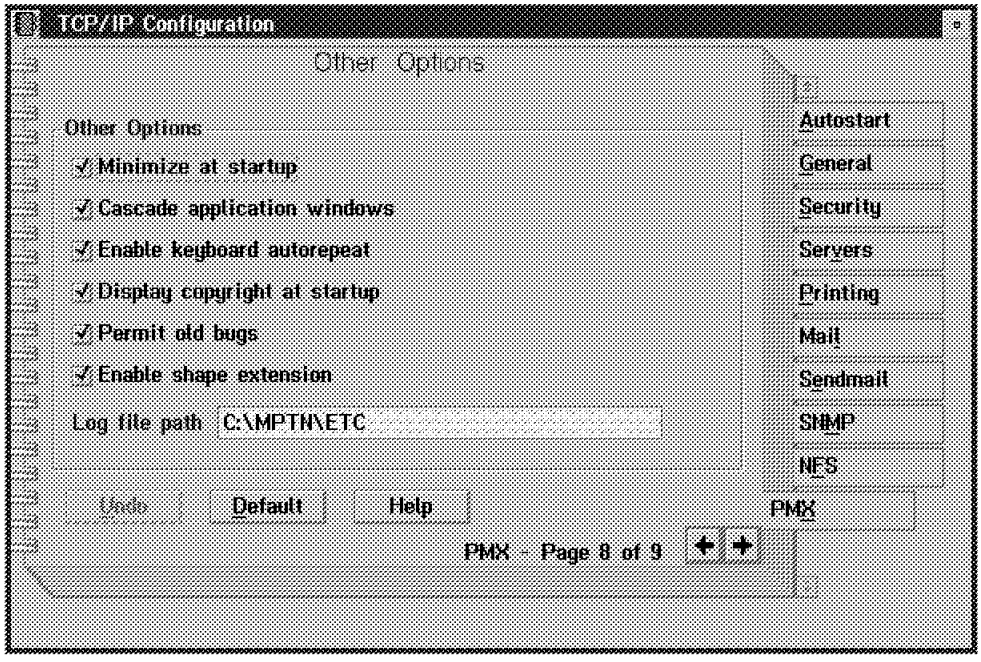


Figure 190. TCP/IP Configuration Notebook for PMX, Page 8

The eighth PMX configuration page is used for the following miscellaneous parameters:

Setting	Meaning
Minimize	Specify whether you want PMX to start minimized on your desktop.
Cascade	Specify whether PMX should cascade X client windows.
Autorepeat	Specify whether PMX should turn on automatic repetition of keys.
Copyright	Specify to display copyright information when PMX starts.
Old bugs	Specify to permit certain old broken clients to function properly.
Shape extension	Specify to start PMX with shape extension installed. This will allow non-rectangular windows to be created by clients, with the exception of top-level windows.
Log file path	Specify where to write the PMX.LOG file.

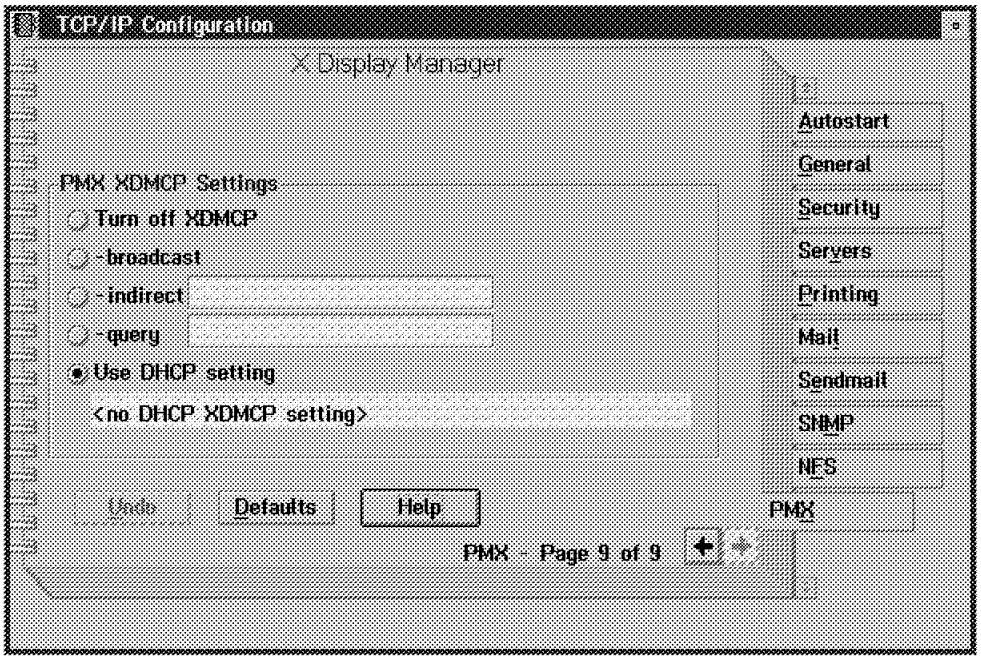


Figure 191. TCP/IP Configuration Notebook for PMX, Page 9

The ninth PMX configuration page is used for the following X Display Manager parameters:

Setting	Meaning
Turn off XDMCP	Specify to disable XDMCP when PMX starts up.
-broadcast	Specify to use the broadcast method of acquiring a display management host.
-indirect	Specify a host to search the network for a display manager host. This indirect host will do an XDMCP broadcast on your behalf and will display a ″chooser″ dialog with a list of hosts on the network that are willing to manage your PMX X session.
-query	Specify a host to be the display manager for your PMX session.
Use DHCP setting	Specify to let the DHCP administrator specify the XDMCP options for your PMX startup.

For further descriptions of PMX configuration settings, see the PMX online reference.

13.2.2 Starting the X Window System Server

To start the OS/2 X Window System server, you have the following three options:

1. Type XINIT.CMD.
2. Type PMX.EXE.
3. Double-click on the **PMX** icon..

XINIT.CMD is a REXX command file that checks your environment prior to starting the X server, and it sets the correct NLS keyboard and other configuration attributes. The PMX command starts the X server explicitly.

Note: If there are problems loading PMX, check if you have a RESOLV2 file created and if the name server defined there is up and running. If you do not have a name server in your network, delete the RESOLV2 file.

Both commands take the settings described in the previous section as default or optional parameters, and they both rely on the following OS/2 environment variables to control server functions:

Variable	Explanation
XFILES	If you allow your CONFIG.SYS file to be updated, the installation program sets the environment variable 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.

DISPLAY	This variable is required by the X Window System utilities. It is set automatically to point to your hostname or IP address and has the format:
	`hostname:0`
	or
	`IP_address:0`
	where 0 denotes the first X server on your host (you will normally have only one).

LANG	This environment variable selects the proper NLS keyboard for X client applications, if Keyboard type and Language settings cannot be found in PMX.INI file.
ETC	This environment variable is already set by TCP/IP for OS/2 (usually to MPTNETC). It tells PMX where to look for the X0HOSTS file.
PMXUNIX	This environment variable is used to supply the value for the -unix parameter for XINIT.CMD.
PMXKEYBOARD	This environment variable is used to supply the value for the -k parameter for XINIT.CMD.
PMXFLAGS	This environment variable is used to supply additional parameters for XINIT.CMD.

Once the X server is started and has finished initializing the X Window System, it runs as an OS/2 Presentation Manager application on your desktop. It will display a Ready for Clients message during normal operation. The following figure shows the PMX main window with open commands menu:

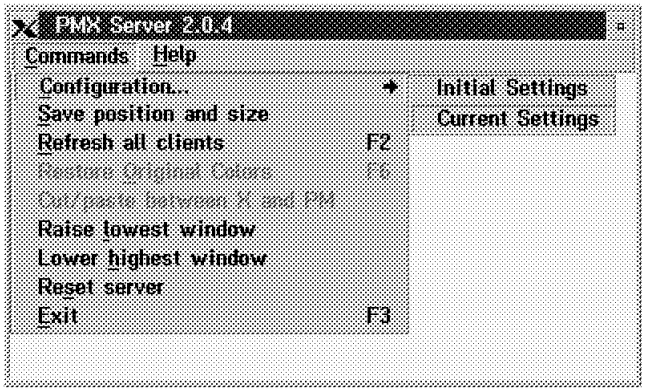


Figure 192. PMX Server Main Window

Setting	Meaning
Configuration	Leads you to the PMX Configuration Notebook. You can change either the initial PMX settings (all) or settings that would influence the currently running server (some).
Save position and size	Saves the position and size of the PMX main window on the OS/2 desktop.
Refresh all clients	Repaints all PM windows which includes X client applications.
Restore Original Colors	Restores the original color table of all running PM applications.
Cut/Paste	Transfers data between Presentation Manager and X Window System clients.
Raise lowest window	Brings the lowest X client window to the top.
Lower highest window	Brings the top X client window to the background.
Reset server	Closes all X applications without shutting down server.
Exit	Ends PMX.

For further descriptions of the environment variables used by PMX, see the PMX online reference.

13.3 National Language and Keyboard Support for PMX

PMX supports national language keyboard definitions as well as different keyboard layouts.

PMX uses the XMODMAP utility to set up your keyboard for the language of the different countries. You can use XINIT.CMD and specify the -lang option to start the X Window System server to select the proper keyboard mapping to use. The -lang option will override the Language setting in the PMX.INI file. If you don't use the -lang option, and the Keyboard Language settings could not be found in the PMX.INI file, XINIT.CMD will look for the LANG environment variable to select the proper NLS keyboard.

The following shows the -lang option being used to start the X Window System server:

```
[C:]xinit -lang de_DE

[C:\]pmxwait -q 0
PMX Server is not ready! rc(1)

[C:\]set DISPLAY=nways2:0

[C:\]START PMX.EXE   -k 102

[C:\]pmxwait -v 60
Waiting for 60 seconds.
PMX Server is ready!

[C:\]xmodmap d:\TCPIP\X11\DEFAULTS\XMODMAP\DE_DE\keyboard

[C:\]
```

To select a different country's keyboard, the following table shows the available variables for the appropriate language:

Table 25. Keyboard Languages Supported by PMX

Language	**Keyboard**	**Language**	**Keyboard**
Belgian	nl_BE	Belgian French	fr_BE
Canadian French	fr_CA	Danish	da_DK
Dutch	nl_NL	Finnish	fi_FI
French	fr_FR	German	de_DE
Greek	el_GR	Icelandic	is_IS
Italian	it_IT	Japanese	ja_JP
Japanese English	en_JP	Latin American Spanish	es_LA
Norwegian	no_NO	Portuguese	pt_PT
Spanish	es_ES	Swedish	sv_SV
Swiss French	fr_CH	Swiss German	de_CH
Turkish	tr_TR	United Kingdom	en_GB
United States	en_US		

PMX supports different physical types of keyboards apart from NLS layouts. The following are supported keyboard types:

- 101 keys (for example, US keyboard)
- 102 keys (for example, German keyboard)
- 106 keys (Japanese 5576-A01 or 5576-002)
- 124 keys (Japanese 5576-001)

13.4 Using the X Window System Server on OS/2

In order to be able to send data to your X server, the host systems where the X client applications are run need to be authorized at your host. This can be done via the X0HOSTS file statically, or with the XHOST utility dynamically.

You can disable X client authorization by uncommenting the line:

```
/* 'xhost +' */   /* DISABLE PMX access control */
```

in your XINIT.CMD file, thus allowing any host to send X client data to your X server.

X Window clients can be executed from a Telnet session with a remote host. The DISPLAY environment variable must either be defined in the user profile on that system for the current user, or it must be specified as an option to the X client command, if the applications supports that. The following command starts an X client window from an RS/6000 under AIX:

```
aixterm -d 9.24.104.91:0 &
```

By specifying the display variable, you can send an X client's output to any host on the network. You will get an error message if that host cannot be reached, or if your host is not authorized to use the designated X server.

Another possibility to start an X client is via the REXEC command instead of a Telnet session. For example:

```
rexec rs600014 -l shlee -p shlee xcalc -display 9.24.104.91:0
```

or if you created the appropriate NETRC file to support the REXEC command:

```
rexec rs600014 xcalc -display nways2:0
```

The NETRC file on host 9.24.104.91 has the following entry to execute remote commands on host rs600014:

```
machine rs600014 login shlee password shlee
```

In the sample below we started an AIX X terminal from an RS/6000 using REXEC:

```
rexec rs600014 aixterm -d nways2:0
```

Since the display variable points to the OS/2 PMX server (-d nways2:0), the graphic output will be displayed on the OS/2 server nways2.

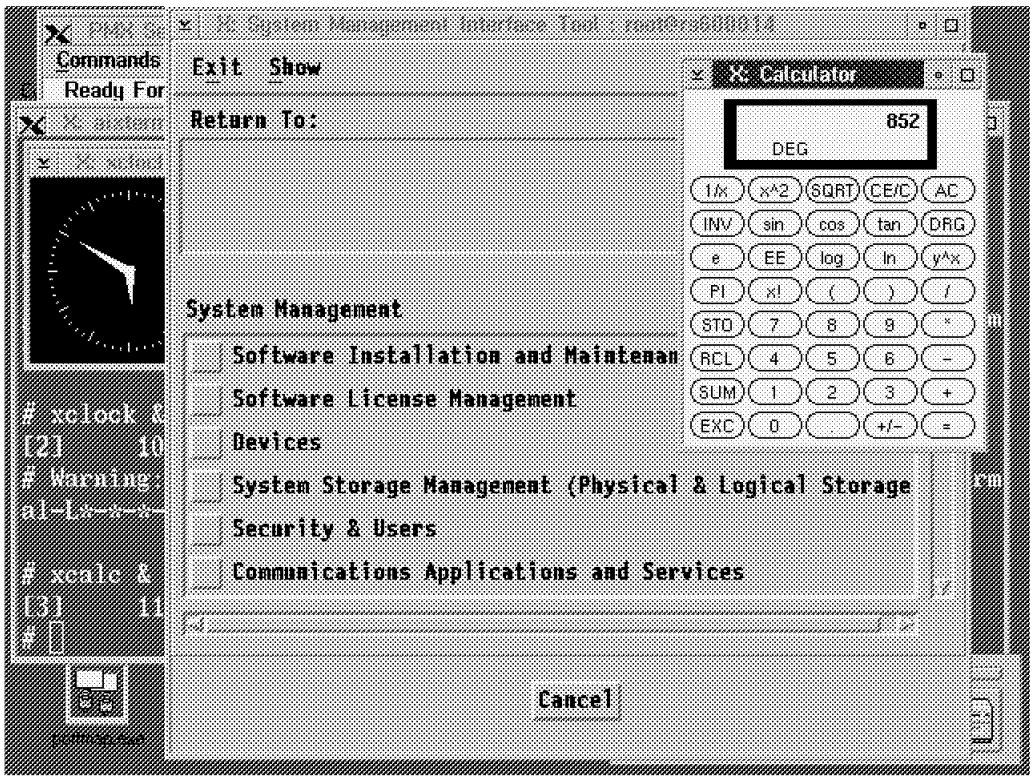


Figure 193. Execute AIX X Window System Clients on OS/2

13.5 X Window System Utilities

The X Window System Server kit provides the following utilities to control and configure the X server environment:

Utility	Function
PMXWAIT	Tests whether PMX is ready to accept client connections within a specified time period.
XEV	Creates a test window to see how PMX responds to user input.
XFD	Creates a window in which the characters of a font are displayed.
XHOST	Dynamically adds or deletes hosts from the list of hosts that are allowed to use your X server. It can also be used to disable X client authorization entirely.
XLSFONTS	Displays information about the fonts used by PMX.
XMODMAP	Displays or changes the keyboard map and table of key synonyms for the X Window System. This utility can also be used for keyboard remap.
XPROP	Displays the window and font properties of an X window.
XSCOPE	Displays the X Window System protocol activity between X clients and PMX.
XSET	Dynamically changes the behavior of PMX
XSTDCMAP	Selectively defines the color map properties for X clients.

XWININFO Displays information about X client applications.

For further descriptions of using the PMX server utilities, see the PMX online reference.

13.6 PMX Clipboard Support

The OS/2 X server supports copying data between OS/2 Presentation Manager and X Window System applications using cutting, copying, and pasting. You can specify the way you want data transfer between PM and PMX to be handled on page 7 of the PMX Configuration Notebook:

Transfer Option	Description
Turn Off PMX Cut/Paste	Disables data transfer between PM and X Window System applications.
Cut Buffer (Text only)	Allows you to share text data between PMX and PM. PMX monitors the contents of an X Window System buffer called CUTBUFFER0. If an X client cuts or copies data, that goes into that buffer. Once the contents of the cut buffer change, PMX transfers that data to the PM clipboard where it is then available to OS/2 applications. When an X client wants to paste data from CUTBUFFER0, PMX checks the PM clipboard. If there is text data there, PMX transfers it into the cut buffer so that the X client can use it.
Selection Names	Allows you to share any type of data between PMX and PM based on a transaction-oriented mechanism. When PMX starts, it becomes the owner of one of the following selection names: 1. PRIMARY 2. SECONDARY 3. CLIPBOARD When an X client cuts or copies data using the selection mechanism, it takes ownership of the selection name from PMX. PMX then requests the data from the X client and places it on the PM clipboard. When an X client wants to paste data using the selection mechanism, it requests data from the owner of that selection name, which in our case is PMX. PMX then transfers data from the PM clipboard to the X client, provided that it is there and matches the requested data format.

For further descriptions of using the PM clipboard with PMX, see the PMX online reference.

13.7 PMX Color Table Support

The OS/2 X server supports the modification of physical colors on the screen by applications. While Presentation Manager applications usually do not do that, X client applications may need this capability to function properly. In fact, this is very common within the X Window System environment.

OS/2 Presentation Manager supports color tables (which cannot be modified), and color palettes (which can be modified). The X Window System supports color maps (the same as color tables and color palettes in PM) and color visuals. Color visuals are specific types and depths of color maps that are supported by the combination of an X server and a display adapter. The depth of a visual is the number of bits available to represent a color or grey shade; a visual with a depth of 5 would have 32 shades of grey, a color visual with a depth of 8 would have 256 colors.

The default color table (color database) that PMX uses is the RGB.TXT file in the TCPIPX11 directory. It is a plain ASCII file holding values for the red, green, and blue components of a color, and a name for that color. You can make changes to this file or create your own color database, but be aware that PMX reads this file at startup and does not dynamically adapt to changes. PMX maps these values into PM color tables and palettes.

PMX supports the following color visuals:

Visual	Explanation
StaticColor	In this visual, colors cannot be changed. Only one color table is available which contains the colors of the physical color table of the display. If an application asks for a color (an RGB value) that is not in the table, it is pointed to the color that is closest to the one requested. Presentation Manager supports such a non-modifiable color table for all displays, hence PMX does as well.
PseudoColor	In this visual, PM uses modifiable color palettes rather than a static color table. These are called PseudoColor maps in the X Window System. If an application selects such a table, it can allocate colors in the default PseudoColor table until there is no more room for colors, or it can create tables and control the contents of those. PMX queries PM to see if the display has support for the Palette Manager before it starts using the PseudoColor visual.

The depth of either color visual is dependant upon the display driver that is used with Presentation Manager, as shown in the following table:

Table 26. PM and PMX Color Support

Display	Depth	Colors	PseudoColor
VGA	4	16	no
8514/A	8	256	yes
SVGA	8	256	yes
XGA-2	8	256	yes
Image Adapter /A	8	256	yes

When you use PseudoColor, the color table being used is that of the application being brought to the top of your OS/2 desktop. If that application happens to modify the color table, this will affect the colors of all applications running on your desktop at that time. To restore colors to all applications, click on the **Refresh all Clients** option from the PMX Command menu.

By default, PMX determines which color visual to use by querying PM's ability to support the Palette Manager. If an application wants to modify color tables but the display driver does not support that, an error is displayed and the application is likely to end abnormally.

For further descriptions of how PMX handles color visuals, see the PMX online reference.

13.8 PMX Font Support

PMX no longer uses the obsolete SNF bit-mapped fonts. After CSD UN68122 (or subsequent PMX CSDs) have been installed, PMX can use native Presentation Manager fonts, which display text significantly faster than PCF fonts. The following are the types of fonts which PMX can use:

BDF source fonts Bit-mapped Display Fonts (BDF) format is an ASCII source code format for defining X Window System fonts. The PMX server cannot use BDF format fonts directly. Fonts in BDF format must first be compiled to either XFN (using the BDFTOPM utility) or PCF (using the BDFTOPCF utility) formats.

PCF bit-mapped fonts Portable compiled format (PCF) font files are binary encodings of BDF fonts created by using the BDFTOPCF utility. As the name of this format states, these binary files are portable between X Window System Servers on different platforms. PCF fonts compiled with the BDFTOPCF utility on any operating system can be used with any X Window System server that is based on X11R5 or later.

Speedo scalable fonts Speedo fonts are scalable fonts supplied by Bitstream Inc. PMX is able to use these fonts.

Public PM fonts These fonts can be used by any process in the system, use system resources, and are usually loaded at startup time and cannot be deleted until after system shutdown. OS/2 public fonts can be in either Image or Outline format.

Private PM fonts These fonts are located by a specific process, and can be used only by that process. OS/2 private fonts are in Image format. PMX private fonts are created using the BDFTOPM or PCFTOPM utilities, each generating font files with .XFN extensions. However, by renaming the extension from .XFN to .FON, you can install the resulting .FON font file as a PM public font using the Font Palette application. If there are other OS/2 PM .FON private fonts that you want to make available to PMX, install them as OS/2 public fonts, and specify `pmpublic` on the PMX font path.

Notes:

1. Only PM public outline fonts can be scaled.
2. PMX does not support double-byte character set (DBCS) PM fonts. If a DBCS font must be used, continue to use the .PCF format.
3. Public (image and outline) fonts, and private image (bit-mapped) fonts in PMX format, are now supported.

The following are the utilities for converting X Window fonts to OS/2 PM fonts:

BDFTOPCF The BDFTOPCF utility compiles a font source file (.BDF) into a portable compiled font (.PCF).

BDFTOPM Use the BDFTOPM utility to convert Bit-mapped Distribution Format (BDF) fonts to Presentation Manager fonts.

PCFTOPM Use the PCFTOPM utility to convert Portable Compiled Format (PCF) fonts to Presentation Manager fonts.

ALLFONTS The ALLFONTS command is a REXX program that converts fonts from .BDF or .PCF format to .XFN (OS/2 PM private) format.

MKFONTDR The MKFONTDR command builds a font directory file of all the font files that it can find in an OS/2 directory. A font directory maps a font name to an OS/2 file name and is accessed by PMX to pick the right font.

13.8.1 Using Font Servers

PMX can access one or more font servers. Any place where a directory can be specified in a font path, you can specify a font server as well. The font server specification has the form:

```
tcp/<hostname>:<port number>
```

where port number is 7000 by default, and must be 7500 for AIX. For example, to specify the host named rs6ktw3 as your PMX fonts server, start the X Window server by using XINIT.CMD command like the following:

```
xinit -fp tcpipx11misc,tcpipx1175dpi,tcp/rs6ktw3:7500
```

Note: You may need to start the font server on your host system. On some systems, the font server is not started automatically.

PMX supplies three font server utilities:

FSINFO The FSINFO command executes a utility for displaying information about an X font server. Use this command to examine the following:

- Capabilities of a particular server
- Predefined values for various parameters used in client/server communications
- Font catalogs and availability of alternate servers

FSLSFNTS The FSLSFNTS command lists the fonts from a font server that match a given pattern.

FSTOBDF The FSTOBDF command converts a font server font to .bdf format. This command is useful for testing servers, debugging font metrics, and reproducing lost BDF files.

The following screen shows examples of using these font server utilities:

```
[D:]fsinfo -server tcp/rs6ktw3:7500
name of server:     tcp/rs6ktw3:7500
version number:     2
vendor string:      International Business Machines Corp.
vendor release number:      5001
maximum request size:       16384 longwords (65536 bytes)
number of catalogues:       1
    all
Number of alternate servers: 0
number of extensions:       0

[D:\]fslsfnts -server tcp/rs6ktw3:7500 -fn *sung*
-ibm_aix-sung-medium-r-normal--0-0-100-100-m-0-cns11643.1986-1
-ibm_aix-sung-medium-r-normal--0-0-100-100-m-0-cns11643.1986-2
-ibm_aix-sung-medium-r-normal--0-0-100-100-m-0-ibm-sbdtw
-ibm_aix-sung-medium-r-normal--0-0-100-100-m-0-ibm-udctw
-ibm_aix-sung-medium-r-normal--0-0-100-100-m-0-iso8859-1
-ibm_aix-sung-medium-r-normal--27-170-100-100-m-130-iso8859-1
-ibm_aix-sung-medium-r-normal--27-170-100-100-m-260-cns11643.1986-1
-ibm_aix-sung-medium-r-normal--27-170-100-100-m-260-cns11643.1986-2
-ibm_aix-sung-medium-r-normal--27-170-100-100-m-260-ibm-sbdtw
-ibm_aix-sung-medium-r-normal--27-170-100-100-m-260-ibm-udctw

[D:\]fstobdf -server tcp/rs6ktw3:7500 -fn -ibm_aix-sung-medium-r-normal--
0-0-100-100-m-0-iso8859-1 > e:\sung.bdf

[D:\]
```

13.9 XDMCP Support

XDMCP, the X display management protocol, is now implemented in PMX. XDMCP is a protocol between X servers and X session managers. It allows a server to contact a session manager on another machine, when the server is started. The session manager can then start X applications for the user automatically.

Use the XINIT.CMD with the -broadcast option to start the X Window server, that enables broadcasting of BroadcastQuery packets to the network. The first responding display manager will be chosen for the session with your PMX. The following is an example:

```
xinit -broadcast
```

If there is a host willing to manage your PMX X session, you will be presented with a login window like the following:

Figure 194. The X Login Window

Or you can use the -query option to start the X Window server, which enables XDMCP and sends query packets to the specified host display manager to start a session with your PMX. For example, to specify the host named rs6ktw3 to manage your PMX X session, start the X Window server by using the XINIT.CMD command as following:

```
xinit -query rs6ktw3
```

The -indirect option enables XDMCP and sends IndirectQuery packets to the specified host. This option asks the host display manager to start sessions between your PMX and other hosts. For example, to specify the host named rs6ktw3 to list the hosts on the network who are willing to manage your PMX X session, start the X Window server by using the XINIT.CMD command as follows:

```
xinit -indirect rs6ktw3
```

After the X Window server has started, it will display a ″chooser″ dialog with a list of hosts on the network who are willing to manage your PMX X session. The following shows the chooser dialog window:

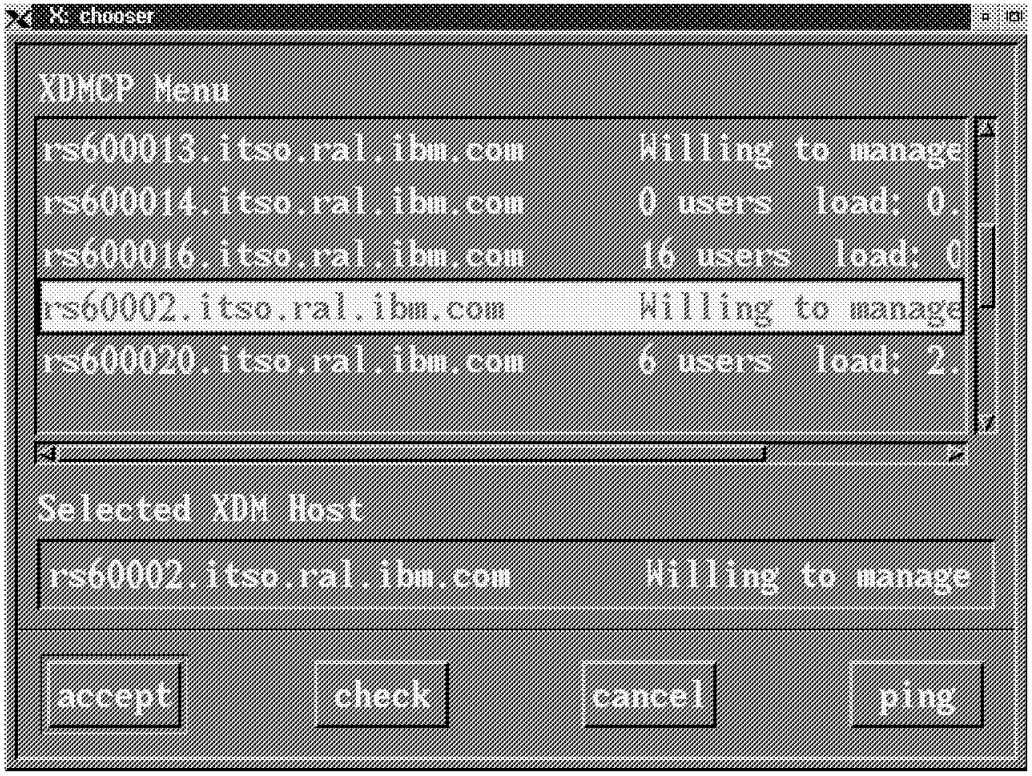


Figure 195. The Chooser Dialog Window

Note: Normally XDMCP is enabled by using either -query, -indirect, or -broadcast. The X Display Manager (usually xdm) must be running on the host machine. The system administrator for that machine also should have it set up so that an X session to your machine is possible. For example, at minimum, you will be presented with a login window from a session manager, if it can contact your PMX. Your PMX host-access list should include the host that your PMX will have the session with. The host-user ID that you use to log in should have its DISPLAY environment variable set to use your PMX. You may need to have an .xsession file in your host home directory to set up and start the X applications that you wish to use, if the default startup for the host is not what you want.

13.10 Using DHCP with PMX

DHCP (Dynamic Host Configuration Protocol) is a proposed standard of the Internet Engineering Task Force (IETF) as described in Request for Comment (RFC) 1541. It is a client/server protocol that allows you to centrally allocate and distribute configuration information.

PMX allows the system administrator to configure both the font path and XDMCP settings. The system administrator might set up X Window System font servers and X Display Managers on the network. These hostnames can be automatically configured in PMX using DHCP.

The DHCP client can insert the font path and XDMCP settings into the PMX.INI file. The font path setting goes under application name DHCP and key name defaultFontServer. The XDMCP settings are inserted using application name DHCP and key name xdmcpHosts. Both these settings can be seen in the TCP/IP

configuration notebook PMX pages and also in the PMX run-time Current Settings or Initial Settings notebook pages.

To provide the site-specific options for the X Window System default font path and the XDMCP parameters from our DHCP server, we had to add the following option lines to the DHCP server's configuration file:

```
option 207       tcp/porsche.tw.ibm.com:7500     #.name 207 X Window font path
option 208       -broadcast     #.name 208 X Window Display Manager par
```

Both these options are provided to modify the PMX.INI file in the DHCP client.

In this case the DHCP server's configuration file is named dhcpsd2.cfg. Option 207 is to specify that the DHCP client should use tcp/porsche.tw.ibm.com:7500 as its X Window font path. Option 208 is to specify that the DHCP client should use -broadcast as its XDMCP parameter.

You can edit the DHCP server's configuration file either manually or by using the DHCP server configuration program.

You can then start the DHCP server by entering the following command from an OS/2 command prompt:

```
dhcpsd -v -f dhcpsd2.cfg
```

At the DHCP client workstation, to enable the DHCP client to get the configuration parameters from the DHCP server, you should uncomment the following two lines:

```
#option 207 exec "dhcpibm.cmd 207 %s"     # Default X Font Server
#option 208 exec "dhcpibm.cmd 208 %s"     # Default X System Display Manager
```

in the DHCP client's configuration file which is named dhcpcd.cfg in the directory specified by the ETC environment variable.

Then shut down the DHCP client workstation and restart it again. After it has restarted and has obtained the configuration parameters from the DHCP server successfully, start the TCP/IP Configuration Notebook. You should find that Page 6 and Page 9 of PMX has been modified with DHCP options. For examples refer to Figure 196 on page 317 and Figure 197 on page 317.

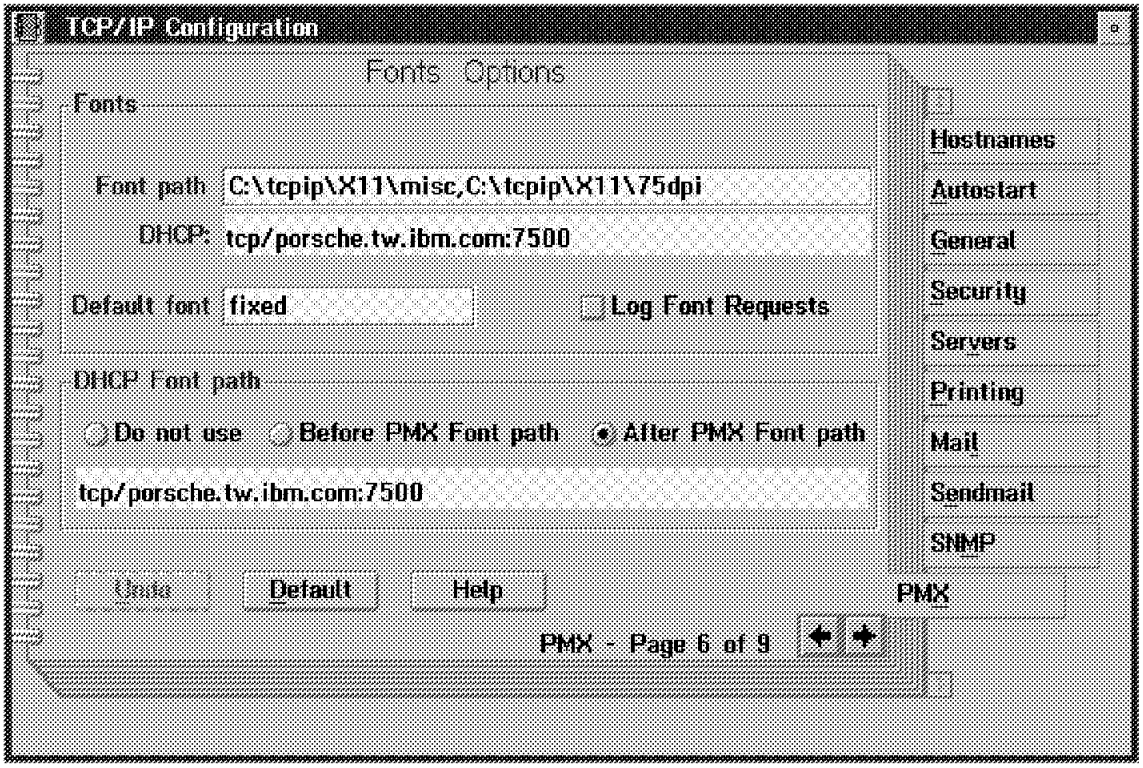


Figure 196. TCP/IP Configuration Notebook for PMX, Page 6 Modified with DHCP

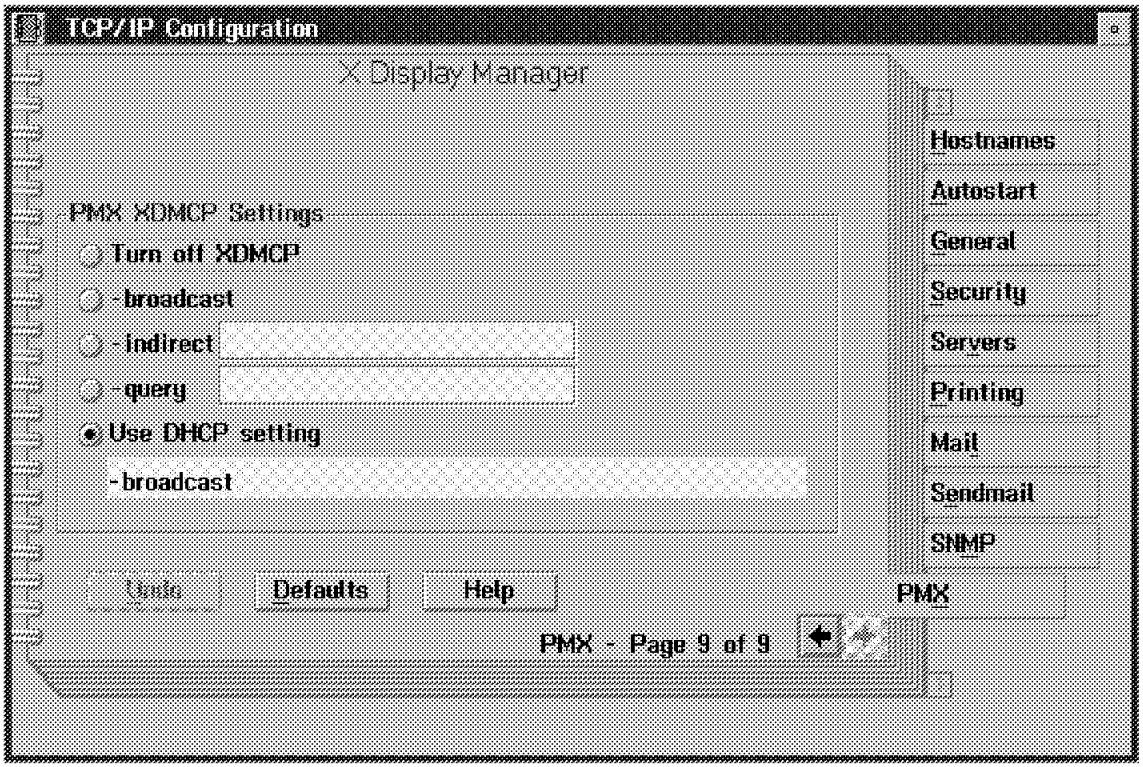


Figure 197. TCP/IP Configuration Notebook for PMX, Page 9 Modified with DHCP

Then start the X Window server on the DHCP client by typing:

```
xinit
```

We have configured to use the DHCP option to enable XDMCP with -broadcast, therefore the X Login Window of the first responding X Display manager on the network will be presented.

Figure 198. Another Login Window

13.11 Execute Sun Microsystems X Clients on OS/2

You can execute the Sun Microsystems X Window System Client applications on your OS/2 PMX server. If you would like to use XDMCP to manage your X display session, start your PMX server by entering the following command from an OS/2 command prompt:

```
xinit -query sun.itso.ral.ibm.com
```

In this case, the Sun Microsystems workstation sun.itso.ral.ibm.com will present an X Login Window on the PMX server as follows:

Figure 199. Login to Sun Microsystems from OS/2 PMX Server

You can also start an X terminal by entering the following command under the Sun Microsystems command shell:

```
xterm -display 9.24.104.91:0 &
```

The following shows the running Sun Microsystems X Client applications on OS/2 PMX server:

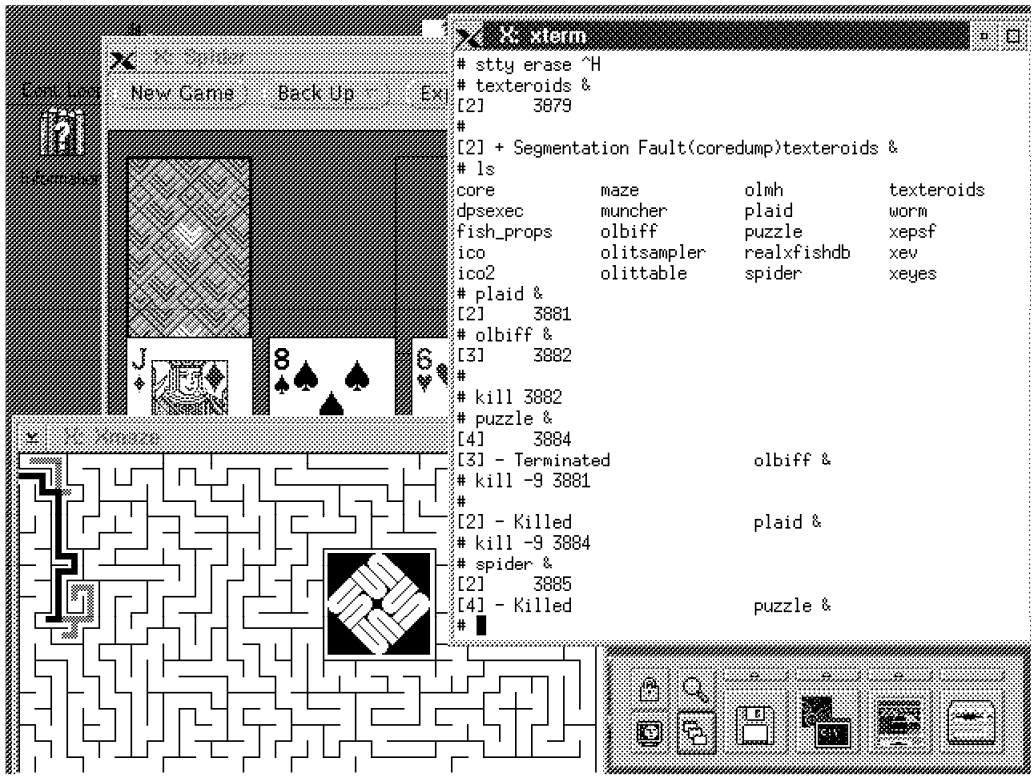


Figure 200. Execute Sun Microsystems X Clients on OS/2

Chapter 14. X Window System Client and OSF/Motif

The X Window System is a distributed graphical user interface (GUI) that uses the X protocol. The protocol is communicated between the client or application and an X server over a reliable bidirectional byte stream. This byte stream is provided by the TCP/IP communication protocol.

In an X Window System environment, the X server distributes user input to, and accepts requests from, various client programs located either on the same system or elsewhere on the network. The X server and the X client communicate using the X protocol.

In OS/2, the X Window System client support consists of an application program interface (API) that creates the X program. This API lets you create an application that uses the TCP/IP sockets system functions to communicate with an X Window System server. As an application writer, you need to be concerned only with the client API in writing your application.

The communication path from the OS/2 X Window System application to the server involves the client code, the X Window System library, and the TCP/IP library. The application program that you create is the client part of a client/server relationship. The X server provides access to the resources that are shared among many X applications, such as the screen, keyboard, mouse, fonts, and graphics contexts.

You can expand your TCP/IP V3.x for OS/2 with X Window System client support by installing the originally available X Window System Client kit for TCP/IP V2.0 for OS/2, and apply the latest corrective service. This chapter describes the installation of X Window System client and OSF/Motif kits, and their usage to run X Window System clients on the OS/2 Warp system.

14.1 X Window System Client Kit

The X Window System client for OS/2 supports the X Window System Version 11 Release 5. It has the following two purposes:

- Develop X Window applications to run on an OS/2 system
- Run X Window applications on an OS/2 system

In the X Window System, the notion of client and server is somewhat reversed. Normally you think of yourself, or the program you are running on your PC, as a client, which accesses shared resources like disk storage or printers on a server in another location.

However with the X Window System, the resource that you share is the display, which is attached to your workstation. So, your workstation must function as a server in order to share the display among several X client applications.

The X Window System Client kit provides you with the ability to develop X Window applications for the OS/2 platform and then also run them on the same platform. It includes the following two parts:

- X client run-time services
- X client programmer's toolkit

The X client run-time services contains the executable sample X client applications, and can be installed on either a FAT or an HPFS file system. The X client programmer's toolkit contains all the libraries, the header files, and the source files for the sample applications, that you need to create your own X client applications. The programmer's toolkit must be installed on an HPFS file system.

Note: If you want to develop or execute an OSF/Motif application, you need the OSF/Motif kit. The OSF/Motif 1.2 widget set is supported. User interface language (UIL) is also supported. The Motif window manager (MWM) is not supported.

14.2 X Window Structure

The X Window System Client kit includes the following layers shown in Figure 201.

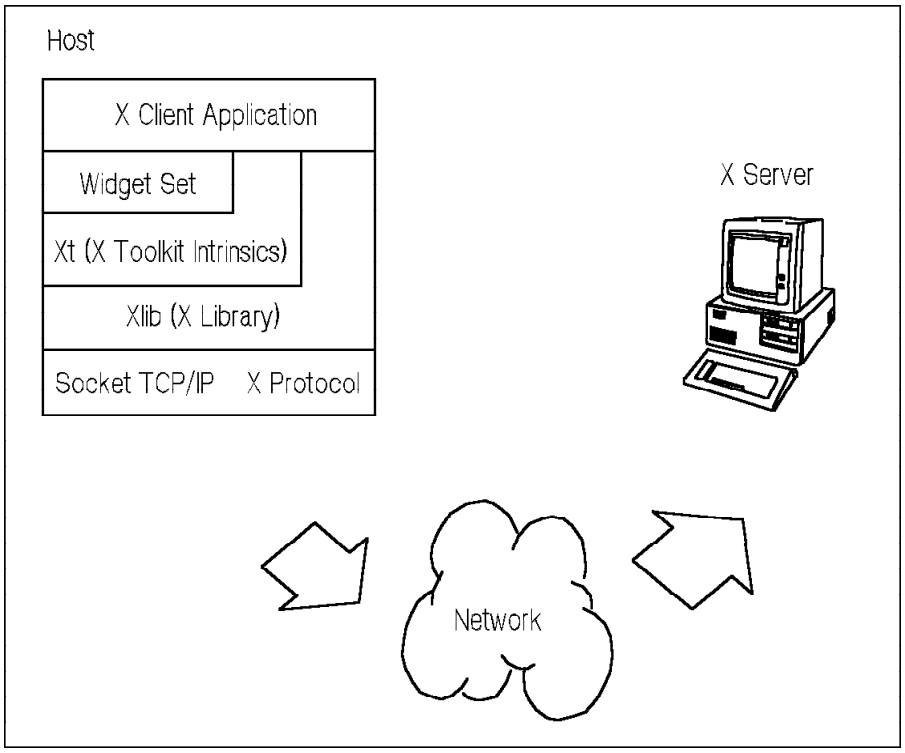


Figure 201. X Window Application Layers

14.2.1 The X Library (Xlib)

The X Window System Client kit contains the X library (Xlib), a set of low-level application functions that provide access to, and control of, the display, its windows, and the input devices. Xlib is the fundamental layer that supports the intrinsics and widgets that are included in the X Window System client kit.

14.2.2 The X Toolkit (Xt) Intrinsics Library

The X Window System Client kit contains the X toolkit library (Xt) intrinsics library, on the top of the X library, that allows you to simplify the design of applications by providing an underlying set of common user interface functions. Xt provides an improved approach to GUI programming. It creates a general mechanism for producing reusable user interface components, and provides routines for creating and using user interface components called widgets. For more information on the X Toolkit Library read the *X Window System Client Guide, SC31-7087*.

14.2.3 The Widget Sets

The X Window System Client kit provides you with the Athena widget set.

You can obtain the OSF/Motif widget set that is separately available in the TCP/IP for OS/2 OSF/Motif kit.

A widget set is a collection of separate widgets. The widget is the fundamental data type in the X Window System client kit. Widgets generally provide a user interface component, such as scroll bar, a text-entry field, or a menu. Widgets are allocated dynamically and contain state information. Each widget belongs to a widget class that is allocated statically and initialized. The widget class contains the operations allowed on widgets of that class.

14.2.4 OSF/Motif Widget Set

You can separately order the TCP/IP for OS/2 OSF/Motif kit. It contains the OSF/Motif widget set, which implements user interface components, such as scroll bars, menus, and buttons. You can combine the OSF/Motif widget set with the Xt intrinsics and Xlib to construct a Motif application. For more information about the OSF/Motif widget set, refer to the *X Window System Client Guide, SC31-7087*.

14.3 Installing X Window System Client and OSF/Motif kits

The X Window client component is contained in the originally available X Window System Client kit for TCP/IP V2.0 for OS/2. Before you install X Window System Client kit, ensure that the required software is installed and running on the workstation.

14.3.1 Requirements to Use X Window System Client Kit and the OSF/Motif Kit

The X Window System Client kit requires that the following be installed and running on your OS/2 Warp workstation:

- IBM OS/2 Warp Connect Version 3.0 or higher with TCP/IP V3.x for OS/2 Warp (TCP/IP Version 3.0 for OS/2 Warp is a part of Warp Connect)
- For development of X Window System applications, the TCP/IP for OS/2 Warp V3.0 Programmer's Toolkit
- For development of X Window System applications, the high-performance file system (HPFS)
- To run and display X Window System applications locally, the X Window System Server kit (PMX)

In addition to the above, the OSF/Motif kit requires that, in particular, the X Window System Client kit be installed on the OS/2 workstation.

The following table shows the latest CSDs for these kits:

Table 27. X Window System Client, Server, OSF/Motif Kits: CSDs

TCP/IP kit	**CSD**
X Window System Client kit	UN59374
OSF/Motif kit	UN59376
X Window System Server kit (PMX)	UN86625

Notes:

1. If you want to develop X Window System applications, install the X Client Programmer's Toolkit in an HPFS file system.
2. If you want to use the X Window System Client kit just to enable your OS/2 workstation to run X Window System applications, you can install X Client Run-Time Services on a file allocation table (FAT) file system or on an HPFS file system.

The following table shows the disk space required for this kit:

Table 28. Disk Space Requirements

TCP/IP kit	**Disk Space (MB)**
X Window System Client kit	4.0
OSF/Motif Kit	2.3

14.3.2 Installing X Window System Client kit

You can install the executable sample X client programs, the X client programmer's toolkit, or both. However, the programmer's toolkit can be installed only to a disk formatted with the HPFS option.

14.3.2.1 Installing the Executable Files

To install only the X Window System Client kit executable files, either to a FAT or to an HPFS file system, do the following:

1. Insert the X Window System Client kit diskette in your diskette drive A:.
2. At the OS/2 command prompt, enter `a:tcpinst`.
3. The TCP/IP installation window is displayed.
4. Select **X-Client Run-time Services** and select **Install**.

14.3.2.2 Installing the Programmer's Toolkit

To install the X client programmer's toolkit to an HPFS file system, do the following:

1. Insert the X Window System Client kit diskette in your diskette drive A:.
2. At the OS/2 command prompt, enter `a:tcpinst`.
3. The TCP/IP installation window is displayed.
4. Select **X-Client Programmer's Toolkit** and select **Install**.

14.3.3 Installing the OSF/Motif Kit Files

You can install the executable sample Motif programs, the Motif programmer′s toolkit, or both. However, the programmer′s toolkit can only be installed to a disk formatted with the HPFS option.

14.3.3.1 Installing the Executable Files

To install just the executable sample Motif executable files, either to a FAT or HPFS file system, do the following:

1. Insert the OSF/Motif diskette in your diskette drive A:.
2. At the OS/2 command prompt, enter `a:tcpinst`.
3. The TCP/IP installation window is displayed.
4. Select **OSF/Motif Run-time Services** and select **Install**.

14.3.3.2 Installing the Programmer′s Toolkit

To install the OSF/Motif programmer′s toolkit to an HPFS file system, do the following:

1. Insert the OSF/Motif diskette in your diskette drive A:.
2. At the OS/2 command prompt, enter `a:tcpinst`.
3. The TCP/IP installation window is displayed.
4. Select **OSF/Motif Programmer′s Toolkit** and select **Install**.

Since these kits comply with IBM′s configuration, installation, and distribution (CID) architecture, which provides for unattended, remote installation of programs and applications from code servers to client workstations. Therefore you can install these kits remotely, from another workstation.

14.4 Running X Window Client Applications

There are several sample OS/2 X Window System applications provided with the X Window System Client. In this section, we describe how to set up an environment to run these applications:

1. Start the X Window System server from the TCP/IP folder on your OS/2 desktop. This provides access to the resources that are shared among many X applications, such as the following:
   - Screen
   - Keyboard
   - Mouse
   - Fonts
   - Graphics contexts
2. Ensure that you have set your display environment variables. Whenever you start an X Window System application, it will look up the variable DISPLAY in your OS/2 system environment. You can set this variable in the Configuration Notebook or from an OS/2 command prompt with this command:

   ```
   SET DISPLAY=9.24.104.91:0
   ```

   In this case, 9.24.104.91 is the IP address of our X Window System server and 0 is the screen on which we want to display the information.
3. Many of the commonly used X Window System functions are stored in dynamic link libraries which are called by X Window applications at run time.

These dynamic link libraries are normally copied to your TCPIPDLL directory at installation. The following are the files that contain the dynamic link libraries:

- Xaw.dll
- oldX.dll
- Xext.dll
- Xmu.dll
- Xlib.dll
- Xt.dll

Please ensure that these files are on your system.

4. This is a list of the sample X client executable programs provided in the X Window System Client kit.

Xant.exe	A 3270 Terminal Emulator program
Xcalc.exe	A calculator application
Xclock.exe	A time-keeping application
Xedit.exe	A simple text editor
Xeyes.exe	Eyes that watch your mouse pointer
Xlogo.exe	Displays the X logo

You can start any of these applications from an OS/2 command prompt. For example, to start the calculator use the following command:

```
xcalc
```

The following shows the calculator program running on the Workplace Shell:

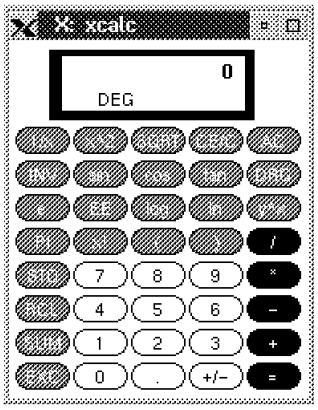


Figure 202. Xcalc (OS/2 X Window Client Application)

If you have access to a S/390 processor running TCP/IP for MVS or VM, then you should try running the X Window 3270 emulator supplied with the X Window System Client kit. For example, to start an X Window 3270 emulator to host wtsc.itsc.pok.ibm.com enter the following command:

```
xant wtscpok.itsc.pok.ibm.com
```

The following shows the Xant 3270 terminal emulator:

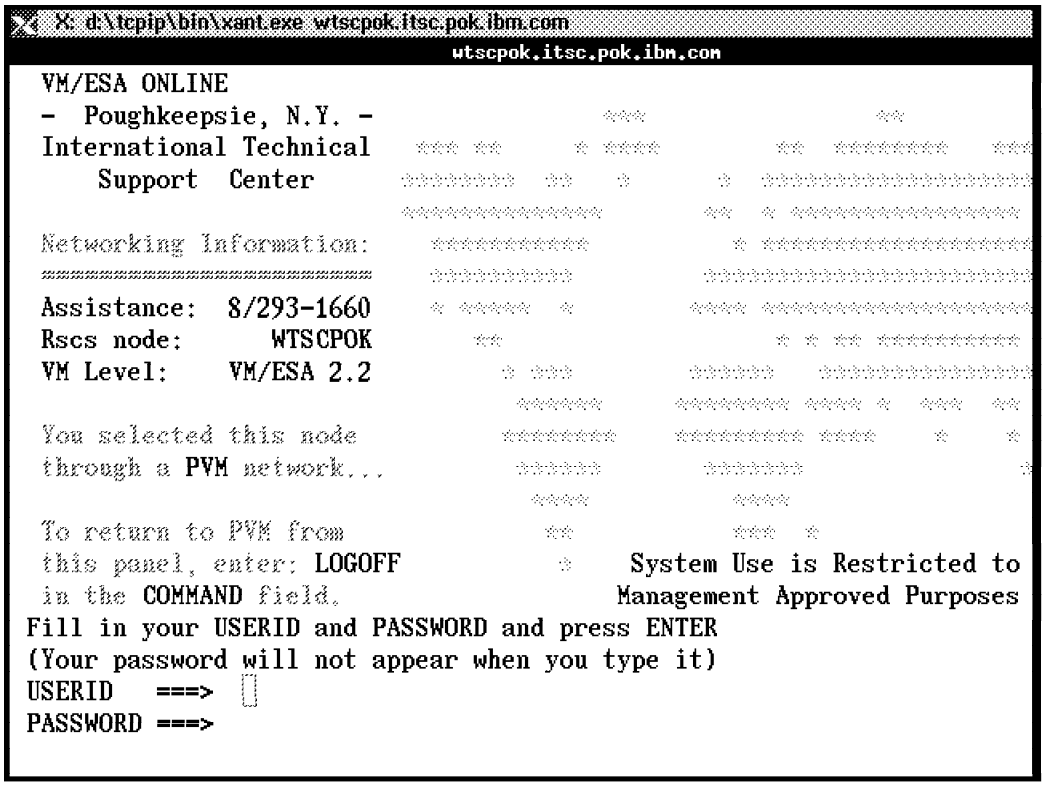


Figure 203. Xant (OS/2 X Window Client Application)

14.5 Development of X Window Client and OSF/Motif Applications

X Window System Client kit provides a set of application programming interfaces that enable you to create an X client program that will use the X protocol to send and receive information to and from an X server for presentation on a screen. The X client program communicates to an X server using sockets.

The X Window System Client kit provides the following components for development of an X client application for OS/2:

Component	Subdirectory
Library files	LIB
Sample Source Code	SAMPLESX11
X Header Files	INCLUDEX11
X Bitmap Files	INCLUDEX11BITMAPS

The library files consist of the following:

File	Description
Xlibi.lib	X
Xti.lib	X Intrinsics
Xexti.lib	X Extensions
Xmui.lib	X Miscellaneous utilities

oldXi.lib X10 Compatibility routines

Xawi.lib X Athena widgets

The X Window System Client kit requires that the following be installed and running on the OS/2 workstation for application development:

- TCP/IP for OS/2 Programmer's Toolkit
- High performance file system (HPFS)
- Any IBM 32-bit C compiler for OS/2, including:
  - VisualAge C++
  - C Set++

If you intend to develop OSF/Motif applications, you need the OSF/Motif kit, which includes the following:

- Motif library (Xm)
- Motif header files
- Motif resource manager library (Mrm)
- Motif resource manager header library
- A proof of licence

Note: Currently the sample programs provided by X Window System Client kit are compiled and linked by using IBM C Set++. If you would like to use IBM VisualAge C++ to compile and link these sample programs, you might have to modify the makefile and source files. Replace the statement:

```
LINK = link386 $(LFLAGS)
```

in makefile with the following:

```
!if defined(VACPP_SHARED)
LINK = ilink /NOFREEFORMAT $(LFLAGS)
!else
LINK = link386 $(LFLAGS)
!endif
```

You should comment out the following lines:

```
extern long time();
extern rand();
```

in these sample source files if you encounter errors associated with redeclaration of these functions.

For more information on how to develop X Window System and OSF/Motif applications, please refer to *X Window System Client Guide, SC31-7087-01* and online *TCP/IP for OS/2 Warp Programmer's Toolkit Reference*.

14.5.1 Tips for Porting Applications from UNIX

When porting an application from a UNIX system to an OS/2 system, you should consider the following important recommendations:

1. OS/2 does not have the same level of signal handling as a UNIX system. You may have to rewrite or remove some signal-related codes.
2. UNIX system calls like pipe and fork have no exact OS/2 equivalent. For fork, there is not a very good OS/2 match. Operations such as spawn work, but

since this typically produces a new process ID, the environment of the parent is not matched exactly. Using threads is probably the closest approximation.

3. Makefiles must be written to be compatible with nmake under OS/2. Sample makefiles are provided. Please consult these when creating your own makefile.
4. Often various UNIX applications will force you to include header files when you try to compile under OS/2. Examples include the following:
   - process.h (getpid)
   - time.h
   - types.h (caddr_t)
   - file.h
   - sys.h
   - signal.h
   - utils.h (bzero)
5. If the application calls sleep, the following line should be used:

   ```
   #define sleep(x) DosSleep(((long)(x))*1000L)
   ```

6. The static storage class cannot be used with functions declared at the block scope. This happens in fragments like the following:

   ```
   static void Whatever(Widget w)
   {
     static int new_time();
   ```

   You can solve this problem by moving the declaration outside of the current function.
7. I/O like:

   ```
   if ((fd = open(fileptr, O_RDONLY)) < 0)
   ```

   can work under OS/2, but you must include the following:

   ```
   #ifdef OS2
   #include <io.h>
   #include <fcntl.h>
   #include <sys\stat.h>
   #endif /* OS2 */
   ```

   Otherwise, the application compiles without errors, but fails at run time.
8. Some UNIX applications read a file by doing something like:

   ```
   if ((file == fopen(string, open_mode)) != 0) {
   ```

   and determine the number of bytes in the file using the following:

   ```
   (void) fseek(file, 0L, 2);
   length = ftell(file);
   .
   .
   .
   if (fread(local_str, sizeof(unsigned char), length, file) != length)
     printf("Error reading file!\n");
   ```

   At run time, the fread fails. The problem is that in OS/2 ftell returns the number of bytes in the file including carriage returns. However, the fread returns the number of bytes without carriage returns. A possible solution is to substitute the fragment below for the fseek and ftell statements.

```
int ch;
length = 0;
while ((ch = getc(file)) != EOF)
  length++;
```

The same problem occurs for open and lseek and to make matters worse, the suggested solution seems to fail when using open and read. To work around this problem, use fopen, getc, and fread along with the fragment above.

9. Change statements such as:

   ```
   fd = creat(name, 0666)
   ```

   to:

   ```
   fd = creat(name, S_IREAD | S_WRITE)
   ```

   or whatever is appropriate. The C Set manuals have examples in the sections regarding migration.

10. If an application calls the system function, beware of attempts to execute commands such as cp, lp, or ls, which have different counterparts in OS/2.

11. Avoid opening a directory to see if it has read or write privileges. This is not legal in OS/2 since you can′t open directories like files and check or assign permission.

12. Limit your file names to eight characters and three letter extensions. While you can′t do development with FAT drives, you can install the X/Motif run-time and run applications on systems with FAT drives.

13. If the application is writing binary data to a file, it is very important to open the file using the binary flag. Otherwise, OS/2 will insert carriage returns in the data. This will corrupt the binary data and make it unusable.

14. You should be able to port and run any Motif application under OS/2. However, when you are using PMX, there are the following limitations:

    - There is no X window manager. PMX handles all of the functions performed by an X window manager like the Motif window manager.
    - Drawing on the root window is prohibited in OS/2. The root window for an X application is what you would call the desktop in OS/2. Unfortunately, if an X client tries to draw on the root window in OS/2 nothing happens and there are no error messages.

15. Applications that follow mouse movements constantly may not receive good information unless they have focus in OS/2. For example, when you run xeyes using PMX, it will follow the pointer perfectly as long as the PM xeyes application has focus. If it does not have focus, it may not follow the pointer.

16. The current release of OS/2 Motif only has support for C xlocale (USA).

Chapter 15. Remote File Systems

Network File System (NFS) enables you to share disk drives or directories located at an NFS server across TCP/IP networks as if the resources were local to an NFS client. It uses Remote Procedure Calls (RPC) for communication between clients and servers.

You can expand your TCP/IP V3.x for OS/2 with NFS services by installing the originally available Network File System kit for TCP/IP V2.0 for OS/2, and applying to it the required corrective service.

The Network File System kit for OS/2 will provide you with the following functions:

- NFS client
- Mounting NFS drives from remote hosts, including UNIX, MVS, OS/2, and other systems
- NFS server
- PCNFSD support for the NFS server
- Query an NFS server for exported directories
- Create and maintain a PASSWD file for PCNFSD

This chapter describes the installation of NFS, its customization as a server and its usage as an NFS client to mount a remote NFS server for OS/2, AIX, VM, MVS, and OEM operating systems.

15.1 Installing NFS

The NFS component is contained in the originally available Network File System kit for TCP/IP V2.0 for OS/2. In order to use it with TCP/IP V3.x for OS/2, you may have to apply the following additional corrective services or program fixes which are:

- To run an NFS client with TCP/IP V3.x for OS/2 requires that you also install the corrective service diskette (CSD) package UN57064 of the NFS kit.
- To run an NFS server with TCP/IP V3.x for OS/2 requires that you install CSD UN57064 and that you additionally apply the APAR PN69745 program fix.

Note: A program fix that provides DHCP support for the NFS kit will soon be available. With the DHCP enabled, your OS/2 NFS server can operate in a DHCP environment. That is, an OS/2 NFS client can mount a directory exported, even if the client's decimal IP address has changed. For more information, please refer to the document provided with the fix.

To install the Network File System kit from the product diskette, insert the NFS kit installation diskette into your diskette drive A: and enter the following command from an OS/2 command prompt:

```
A:TCPINST
```

Since the NFS kit complies with IBM's Configuration, Installation, Distribution (CID) architecture, which provides for unattended, remote installation of

programs and applications from code servers to client workstations, you can install the NFS kit remotely, from another workstation.

The OS/2 NFS client is built on the OS/2 Installable File System (IFS) base, and therefore an IFS statement must exist in the CONFIG.SYS file of the host on which the client runs. If you allow the installation and configuration process to modify your CONFIG.SYS file, that statement is added for you.

15.2 Configuring NFS Services

The Configuration Notebook allows you to enable your workstation to act as an NFS client, an NFS server (NFSD), or both. It may be accessed by selecting the **TCP/IP Configuration** object in the TCP/IP folder on your OS/2 desktop or you can also enter the following command to start the configuration:

```
tcpcfg
```

The following shows the first configuration page for NFS:

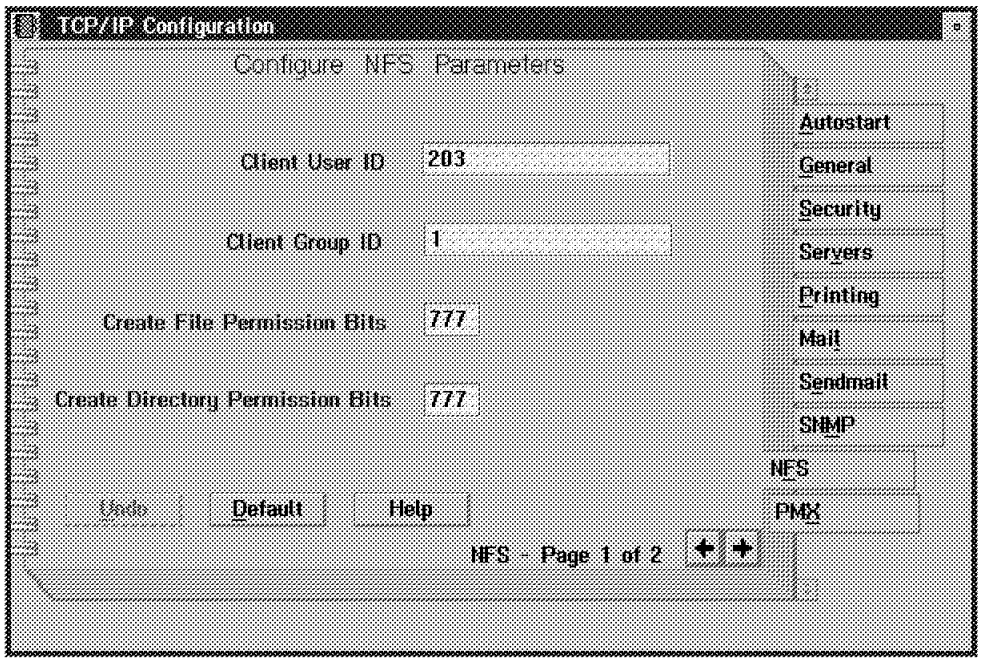


Figure 204. NFS Configuration First Page

The first NFS configuration page is used for the following parameters:

Setting	Meaning
Client UID	The user ID when mounting from a UNIX NFS server. It is a numeric value in the range from -2,147,483,648 to 2,147,483,647, inclusive.
Client GID	The group ID when mounting from a UNIX NFS server. It is a numeric value in the range from -2,147,483,648 to 2,147,483,647, inclusive.
File Permission	The bits to determine read/write and execute permission bits are set for: 1. File owner 2. Group of file owner 3. Everyone

and can have a value from 0 to 7, depending on any combination of:

Read	Value of 4
Write	Value of 2
Execute	Value of 1
No permission	Value of 0

Directory Permission The bits to determine read/write and execute permissions on directories you create on a UNIX NFS server.

The following shows the second configuration page for NFS:

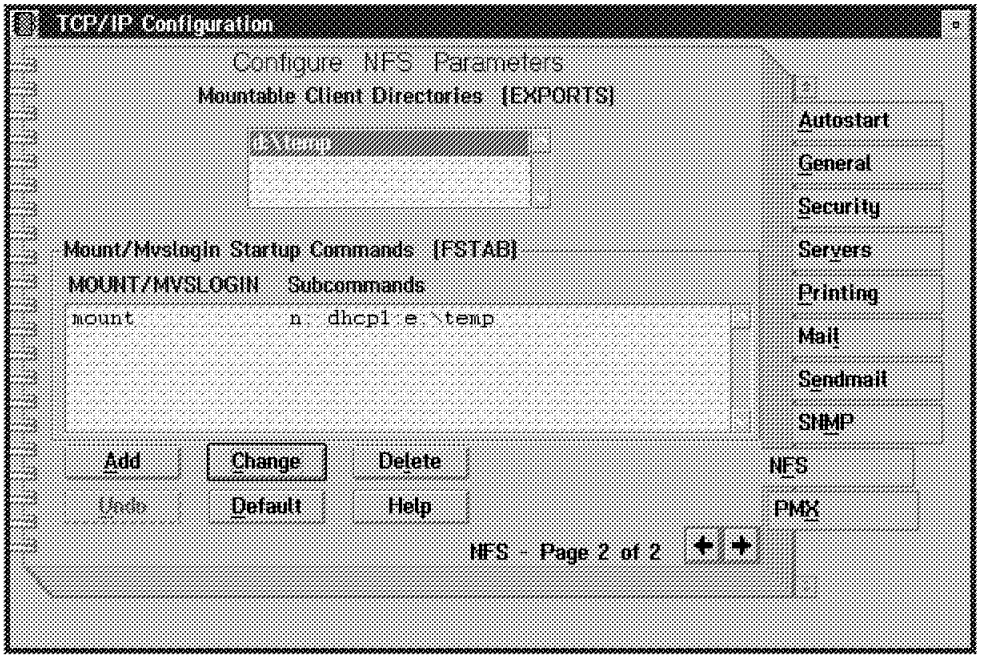


Figure 205. NFS Configuration Second Page

The second NFS configuration page is used for the following parameters:

Setting	Meaning
EXPORTS	Directories that are to be exported by the NFS server to NFS clients
FSTAB	MOUNT and MVSLOGIN commands to be executed when the NFS client is started, in order to automatically mount remote file systems

The following shows the Configure Automatic Starting of Services page:

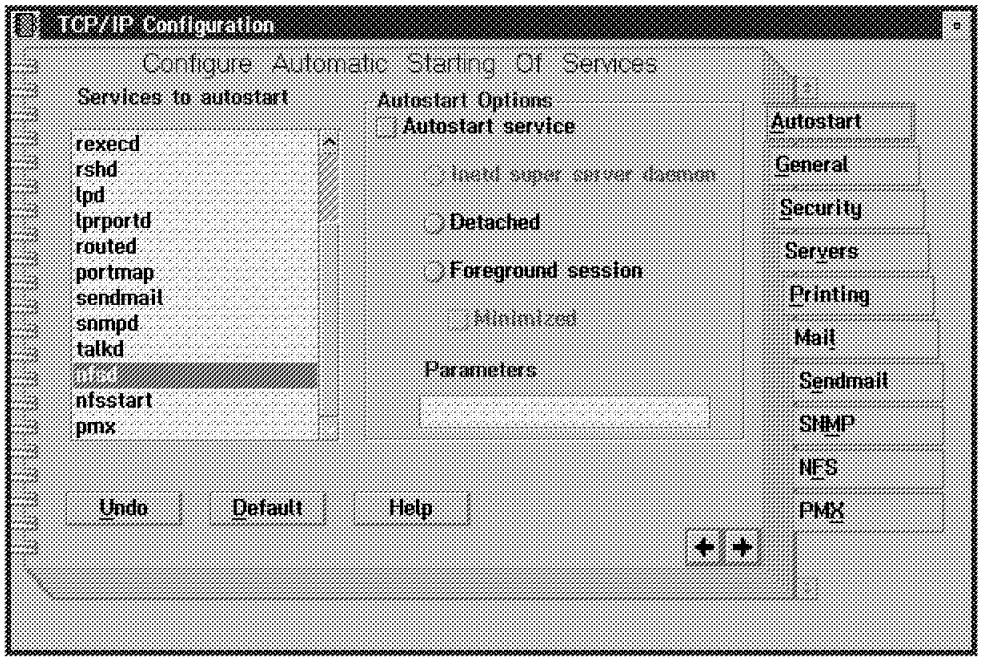


Figure 206. Configuration Automatic Starting of Services Page

If you would like your workstation to automatically function as an NFS client when TCP/IP starts, select the **nfsstart** service to autostart.

If you would like your workstation to automatically function as an NFS server when TCP/IP starts, select the **nfsd** service to autostart.

15.3 OS/2 NFS Client

If you select to enable the NFS client from the TCP/IP Configuration Notebook, the TCPSTART.CMD file will be updated to start the NFS client automatically, by adding an NFSSTART statement. If not, you can start it by entering:

```
NFSSTART
```

NFSSTART invokes the NFS client control program NFSCTL which communicates with the NFS IFS driver. The program NFSCLEAN is loaded (which unmounts drives that are still attached) then the program QMOUNT (to query the information about mounted drives), and finally, mount entries defined in the MPTNETCFSTAB file are processed.

NFSCTL must be running to mount a remote resource as a local drive. The following shows the screen when OS/2 NFS Control Program is running.

```
IBM TCP/IP OS/2 NFS Client Release (May 11 1994)
Copyright (c) IBM Corp. 1993.  All rights reserved.
Buffer size:     8192
RPC timeout:     1 seconds.
No. of retries:  5
No. of Biods:    4
Priority:  class:4 level:1
Parallel Read requests:  on
Parallel Write requests:         off
Respect case when creating files/directories:    off
Case sensitive comparisons:      off
File creation permission bits: 777, directory creation permission bits: 777
UMASK for accessing files: 600
NFS BIOD 1 running
NFS BIOD 2 running
NFS BIOD 3 running
NFS BIOD 4 running

NFS Control Program Running.
```

Figure 207. OS/2 NFS Control Program

15.3.1 Mounting an OS/2 NFS Server

Before mounting a directory on an OS/2 NFS server, you can use the SHOWEXP command to see whether your OS/2 NFS client can mount the requested directory and with what type of access rights. For example, if you want to see which host has which access rights to the NFS server dhcp1, enter the command:

```
showexp dhcp1
```

and you get the following listing:

```
export list for dhcp1:
e:\temp             porsche.tw.ibm.com 9.24.104.91
```

To mount the E:TEMP directory on OS/2 NFS server running on dhcp1 host, as your N: drive, enter the command:

```
mount -u -g n: dhcp1:e:temp
```

This results in the following messages:

```
mount: dhcp1:e:temp

NFS Drive 'n:' was attached successfully.
```

If you do not enter the -u and -g options, and the environment variables UNIX.UID and UNIX.GID are not set, you are prompted to enter UID and GID. Since the OS/2 NFS server does not use these values, you may just press the Enter key.

If the PCNFSD is running on the OS/2 NFS server, you are prompted to enter user name and password, even if you enter the -u and -g options. The OS/2 NFS server checks with the PASSWD file.

A way to automatically mount remote NFS server resources is to create an FSTAB file in your MPTNETC directory. This can be useful if you always have

to mount the same NFS servers. Assuming you want to automatically mount the E:TEMP directory of OS/2 NFS server dhcp1 as your logical drive N:, your FSTAB file would look like this:

```
mount -u -g n: dhcp1:e:temp
```

You also can use the OS/2 Configuration Notebook to edit the FSTAB file. NFSSTART looks for an FSTAB file at start time and executes any MOUNT commands that are contained.

If you want to see which hosts in your network have already mounted resources at the OS/2 NFS server dhcp1, enter the command:

```
showmoun dhcp1
```

This will show you a display similar to the following:

```
rs6ktw3 : e:temp
nways2.itso.ral.ibm.com : e:\temp
```

After you mount a resource at the NFS server you will see the resource as an icon in the Drives Icon View on OS/2 Warp.

To see if and what kind of remote file systems are attached to your OS/2 system, enter the QMOUNT command, for example:

```
[C:]qmount

Type    Name    FSDName         FSAData
Local   C:      HPFS
Local   D:      HPFS
Local   E:      FAT
Remote  H:      LAN             \\GRODEX85\EDRIVE
Remote  I:      LAN             \\GRODEX85\CDROM
Remote  M:      NFS             rs6ktw4:/tmp/cd
Remote  N:      NFS             dhcp1:e:\temp

[C:\]
```

The example above shows locally and remotely attached file systems:

- Local drive with HPFS
- Local drive with HPFS
- Local drive with FAT
- Remote OS/2 LAN Server drives (LAN)
- Remote OS/2 LAN Server drives (LAN)
- Remote NFS drives (NFS)
- Remote NFS drives (NFS)

15.3.2 Mounting a VM NFS Server

The VM NFS server allows an NFS client to mount a VM/CMS minidisk. This can be done basically in two modes, binary or with ASCII-to-EBCDIC translation. In binary mode, there is no translation at all done. This option is useful if you use the CMS minidisk only as additional disk storage for OS/2. The data stored on the CMS minidisk will probably be useless to an ordinary VM/CMS user.

If you want to share text files among NFS clients and the CMS users, you need to use ASCII to EBCDIC translation mode. The NFS server does the necessary translation.

The following example shows how to mount a CMS minidisk in text mode.

```
[C:]mount -v v: wtscpok.itsc.pok.ibm.com:shlee.191,ro,user=shlee,record=nl
mount: wtscpok.itsc.pok.ibm.com:shlee.191,ro,user=shlee,record=nl
Enter password: *******

NFS Drive 'v:' was attached successfully.

[C:\]
```

The following screens show an example of copying a file from VM/CMS to an OS/2 disk.

Note: VM/CMS and OS/2 interpret their data differently; VM/CMS uses EBCDIC, and OS/2 uses ASCII. The data in a CMS file is arranged in records, whereas an OS/2 file is a stream of data using a special character combination (CR,LF) to indicate record boundaries. The mount option record=nl requests the data conversion between EBCDIC and ASCII. The data on the VM/CMS minidisk is meaningful for regular VM/CMS users as well as for OS/2 users. The files stored should also meet the VM/CMS file system restrictions. If you want to store executable OS/2 modules (.COM or .EXE files) and share these programs with other OS/2 network users, do not use the record=nl option. The data should not be translated; rather it should be stored in binary format.

```
[C:]dir v: /w

The volume label in drive V is NFS.
The Volume Serial Number is 001E:0000.
Directory of V:\

ctcbinst.doc         epqprof.maclib       epruprof.file
lasting.globalv      nfs.script           toolcat.log
nfs00100.pseg3820    nfs00101.pseg3820    nfs00102.pseg3820
ofsmail.ofsdata      po-kip-c.history     profile.exec
profile.ovmlp        profile.xedit        read_me.first
shlee.netlog         shlee.synonym        nfs.list3820
       18 file(s)     839277 bytes used
                     6344704 bytes free

[C:\]
```

The TYPE command to the read_me.first file shows the following result:

```
[C:]type v:read_me.first
Welcome to WTSCPOK!!!  Getting information is easy on WTSCPOK if you
                                                                know the cor
rect tools to use.   Your profile exec was customized to
                                                      remind you of the key ex
ecs (applications or tools) to use if you need
                                                 to get in formation.  2INFO is the
 most important.

                  ENTER:   2INFO
```

As you see, there is no order in the records of the file shown at the screen. But if you use the UNIX2OS2 command you will get the result shown in the next screen:

```
[C:]unix2os2 < v:read_me.first
Welcome to WTSCPOK!!!  Getting information is easy on WTSCPOK if you
know the correct tools to use.   Your profile exec was customized to
remind you of the key execs (applications or tools) to use if you need
to get in formation.  2INFO is the most important.

ENTER:   2INFO

[C:\]
```

To copy the file to your local workstation disk, enter the command in the following manner:

```
[C:]unix2os2 < v:read_me.first > test.os2
```

This all looks very complicated and not user friendly. However, the reality proved to be less complicated than expected. The OS/2 editor and some other editors can handle the missing CR without problems. If you want to edit remote files with an OS/2 editor, you have to use the -c option in the MOUNT command, in order to suppress the writing of the CR character.

15.3.3 Mounting an MVS NFS Server

To use an MVS NFS server, the client must use the MVSLOGIN program to authorize access to the mounted directory.

You have to use the MVSLOGIN command only once to access files on a particular MVS NFS server, even if you mount this MVS NFS server multiple times.

When you finish accessing files, or have unmounted an MVS mounted NFS drive, use the MVSLOGUT command to quit the NFS session with your MVS NFS server.

To mount an MVS NFS file system it is required to set UNIX.UID and UNIX.GID environment variables. The following screens show the activities to be done before mounting and to mount an MVS NFS file system.

```
[D:]set  unix.uid=-2
[D:\]set  unix.gid=-2
[D:\]mvslogin mvs20 terreld
 Password required
 Enter MVS password:
 terreld logged in ok
[D:\]MOUNT  z: mvs20:tcpip.itsc,text,crlf
mount: mvs20:tcpip.itsc,text,crlf

NFS Drive 'z:' was attached successfully.
```

To show text files in an orderly manner, you have to set the options `text` and `crlf` at the `MOUNT` command.

Note: The `MVSLOGIN` and `MVSLOGUT` commands are not required if the MVS system runs the PCNFSD server. To determine whether it does so, run the `rpcinfo -p` command against it.

The next screen shows the content of the Z: drive.

```
[D:]dir z:

The volume label in drive Z is NFS.
The Volume Serial Number is EAE6:0002
Directory of Z:\

7-15-92   9:29a     <DIR>           0  .
7-15-92   9:29a     <DIR>           0  ..
7-15-92   9:29a      3914           0  dig.help
7-15-92   9:29a      5159           0  etc.services
7-15-92   9:29a      2976           0  ftp.data
7-15-92   9:29a      3526           0  hosts.local
7-15-92   9:29a      2495           0  lpd.config
7-15-92   9:29a     22960           0  mib@desc.data
7-15-92   9:29a      1736           0  nslookup.help
7-15-92   9:29a       192           0  pw.src
7-15-92   9:30a      6208           0  ralvsmv6.tcpip
7-15-92   9:30a      9668           0  ralvsmv6.tcpip.distrib
7-15-92   9:30a      8072           0  smtp.config
7-15-92   9:30a       193           0  smtpnje.hostinfo
7-15-92   9:30a        75           0  snmptrap.dest
7-15-92   9:30a     23974           0  standard.tcpkjbin
7-15-92   9:30a       768           0  standard.tcpxlbin
7-15-92   9:30a      5326           0  tcpip.data
7-15-92   9:30a       768           0  telnet.tcpxlbin
7-15-92   9:30a       768           0  telnetse.tcpxlbin
       20 file(s)      99834 bytes used
                    61440000 bytes free
[D:\]
```

To copy the file snmptrap.dest to the local drive D: the destination file name must conform to the destination file system restrictions. The following shows the copy of the file snmptrap.dest to the local drive D: on host paul:

```
[D:]type z:snmptrap.dest
9.67.38.93  UDP           ;  PHILIPPE
9.67.38.96  UDP           ;  MVS20

[D:\]copy z:snmptrap.dest d:\snmptrap.dst
        1 file(s) copied.

[D:\]type d:\snmptrap.dst
9.67.38.93  UDP           ;  PHILIPPE
9.67.38.96  UDP           ;  MVS20
```

15.3.4 Mounting an AIX NFS Server

On the AIX NFS server, you must export the file system to your OS/2 machine. To do this, add an entry to the /etc/exports file. 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 following shows how to export a directory from an AIX system using SMIT:

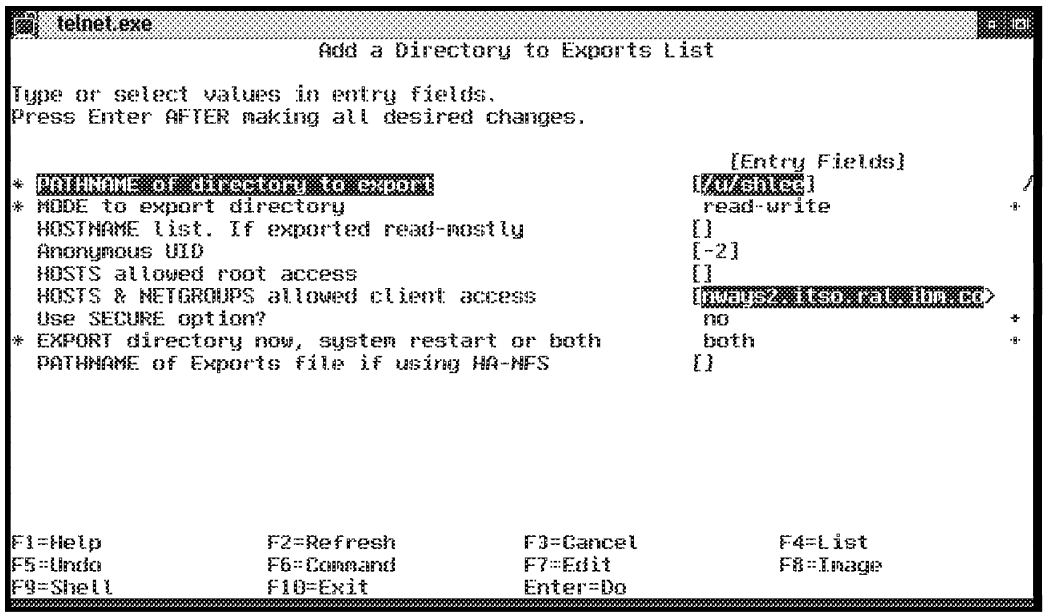


Figure 208. Exporting a Directory for NFS from AIX

The UID specifies a user ID (UID) on UNIX systems, and the GID specifies a group ID (GID) on UNIX systems. These variables are defined when a user ID is created. If you want to mount an UNIX NFS server, you have to enter these variables. You get the values of these variables if you enter the following command at a UNIX command prompt:

```
$ grep shlee /etc/passwd
shlee:!:213:999::/u/shlee:/bin/ksh
```

In this example, 213 is the UID and 999 is the GID.

Note: The considerations above do not apply if the server is running PCNFSD.

The NFS server can share files based on the FAT (File Allocation Table) or HPFS (High-Performance File System) system.

Before mounting an AIX NFS server make sure all NFS daemons are started. You can do this by entering the RPCINFO command at an OS/2 command prompt. The following screen shows you the result of this command:

```
[C:]rpcinfo -p rs6ktw3
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100024    1   udp    648  status
    100024    1   tcp    650  status
    300082    1   udp    653
    300082    1   tcp    655
    100005    1   udp    628  mountd
    100005    1   tcp    630  mountd
    100021    1   tcp    906  nlockmgr
    100021    1   udp    908  nlockmgr
    100021    3   tcp    911  nlockmgr
    100021    3   udp    913  nlockmgr
    100020    1   udp    916  llockmgr
    100020    1   tcp    918  llockmgr
    100021    2   tcp    921  nlockmgr
    150001    1   udp    779  pcnfsd
    150001    2   udp    779  pcnfsd
    150001    1   tcp    783  pcnfsd
    150001    2   tcp    783  pcnfsd

[C:\]
```

Notes:

1. The RPCINFO command allows you to query the registered RPC programs on a remote host. The result shows that the important programs (100000, 100003, 100005) are running on this RS/6000.
2. The program 150001 is the pcnfsd server component, which is a user verification method for NFS.

The following example shows how to mount an RS/6000 NFS exported directory to the logical drive N: on host rs6ktw3:

```
[C:]showexp rs6ktw3
export list for rs6ktw3:
/u/shlee              nways2.itso.ral.ibm.com

[C:\]mount -s n: rs6ktw3:/u/shlee
mount: rs6ktw3:/u/shlee
user name: shlee
password:

NFS Drive 'n:' was attached successfully.
[C:\]dir n:

The volume label in drive N is NFS.
The Volume Serial Number is 001E:0000
Directory of N:\

 1-31-96   9:12a     <DIR>           0  .
 1-29-96   3:53a     <DIR>           0  ..
 6-06-94  11:25p       424           0  .profile
 6-08-94   3:13a     <DIR>           0  ods
 5-12-94   6:42a       401           0  infor.profile
 2-05-96   1:28p      2088           0  .sh_history
 6-06-94   1:18a     <DIR>           0  leetest
 2-05-96   1:35p      3950           0  smit.script
 5-25-95   3:02a       994           0  mbox
 7-28-94   1:33a     <DIR>           0  tpc.dbs
 2-05-96   1:27p     78683           0  smit.log
 5-25-95  10:11p        34           0  big5.txt
 5-25-95  10:34p     15537           0  README
 8-28-95  11:20p        60           0  test
       14 file(s)     102171 bytes used
                     4988928 bytes free

[C:\]
```

Notes:

1. Using the SHOWEXP command, we get a list of authorized users.
2. The -s parameter for the MOUNT command tells the NFS server that this client requests record locking according to SUN's NLM protocol. Therefore, the NFS server must run a file locking daemon, LOCKD, which, as shown in the above example for RPCINFO, our RS/6000 does.

   If a server does not run LOCKD, a mount request with the -s parameter will fail.
3. Because the PCNFSD program is running on the RS/6000 we are asked for a user ID and password instead of UID and GID in the mount command.

15.3.5 Mounting a Sun Microsystems NFS Server

You can mount resources from a Sun Microsystems NFS server by using the OS/2 NFS client. The following screen shows an OS/2 NFS client mounting a drive from a Sun Microsystems workstation after having verified that a drive is actually exported:

```
[C:]showexp sun.itso.ral.ibm.com
export list for sun.itso.ral.ibm.com:
/export             everyone

[C:\]mount s: sun.itso.ral.ibm.com:/export
mount: sun.itso.ral.ibm.com:/export

NFS Drive 's:' was attached successfully.

[C:\]s:

[S:\]dir

The volume label in drive S is NFS.
The Volume Serial Number is 001E:0000.
Directory of S:\

 1-04-96   1:21p     <DIR>           0  .
 1-04-96   1:21p     <DIR>           0  exec
 1-04-96   1:21p     <DIR>           0  share
        3 file(s)       1536 bytes used
                    73452544 bytes free

[S:\]
```

15.4 OS/2 NFS Server

Before starting the OS/2 NFS server, verify that PORTMAP (PORTMAPPER server) is running. If PORTMAP is not running, enter the following command in an OS/2 window to start it:

```
portmap
```

Then, to start the OS/2 NFS server, enter the following command in a different OS/2 window:

```
nfsd
```

If you are using the OS/2 TCP/IP Configuration Notebook to select your workstation to function as an OS/2 NFS server, your TCPSTART.CMD file is updated to start the PORTMAPPER server and the NFS server automatically. The PORTMAPPER server is used to dynamically assign port numbers to RPC programs.

If you would like your OS/2 NFS server to use the PCNFSD support, you can set up PCNFSD to autostart, or enter the following command from an OS/2 command prompt:

```
pcnfsd
```

Note: Because TCPCFG does not currently recognize PCNFSD, every time you run TCPCFG, it will remove PCNFSD from TCPSTART. Therefore, you should start PCNFSD from the TCPEXIT.CMD file located in TCPIPBIN. If this file exists, TCPSTART.CMD executes it as its final action.

To allow clients to mount a directory on an OS/2 NFS server, you must export the OS/2 file system. To do this, add an entry in the MPTNETCEXPORTS file on the NFS server.

Notes:

1. If you use the NFS server on a FAT drive, file names are restricted to the FAT 8.3 format and must be in uppercase.
2. 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 lowercase).
3. The OS/2 NFS server does not support SUN's NLM protocol for record locking.
4. NFS does not support HPFS extended attributes. If you need to copy files with extended attributes to an NFS mounted drive, make sure that you extract all extended attributes first using the OS/2 EAUTIL command. EAUTIL will extract extended attributes from HPFS files and place them in a separate file that needs to be copied along with the files. The same command can later be used to attach extended attributes back to their original files.

For the OS/2 NFS server dhcp1 the following EXPORTS file was created:

```
#DIRECTORY                    CLIENTS                          COMMENT
e:\temp              rs6ktw3 9.24.104.91
```

Note: The hostname field of the NFSD EXPORTS file now accepts dotted-decimal IP addresses as well as hostnames. Dotted-decimal addresses have the advantage of not requiring the resolver during initialization.

When you edit the EXPORTS file by using the TCP/IP Configuration Notebook, you will see the following panel:

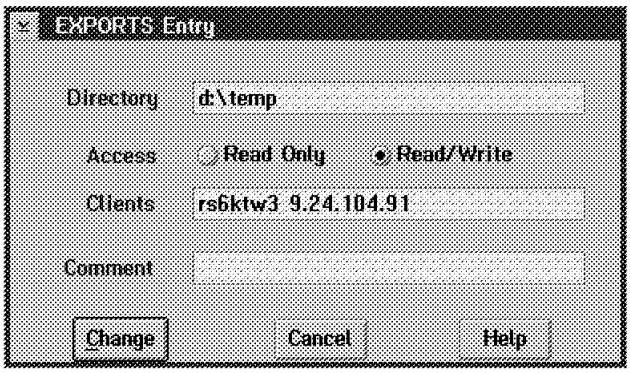


Figure 209. EXPORTS File Configuration

Setting	Meaning
Directory	Directory to be exported
Access	Access permission for exported directory
Client	Hosts authorized to access exported directory
Comment	Optional comment for exported directory

The following screen shows the OS/2 NFS Server:

```
NFSD    IBM OS/2 NFS Server Version 1.2 (Feb 06 1995)

Reading the exports file...
Registering MOUNTD with portmap...
Registering NFSD with portmap...
NFS: File ownership set to uid 0, gid 0.
NFSD: Initialization complete.  Server running.
```

15.4.1 PCNFSD

The OS/2 NFS server also provides security by enforcing authentication through PCNFSD. PCNFSD is a server which maps a valid login name and password to a uid (user ID).

When the OS/2 NFS client issues a MOUNT command to access an NFS server's directories, the MOUNT command checks to see if PCNFSD is available. If it is, the OS/2 MOUNT command prompts you for your login name and password which it then passes to PCNFSD. If PCNFSD authenticates the login name and password, it passes back the associated uid. The OS/2 NFS client then uses that uid for each NFS request, ignoring any previously specified uid. If PCNFSD reports that the login name or password is invalid, the MOUNT command is terminated and access to the server's files is denied.

PCNFSD uses the PASSWD file, located in the directory specified by the ETC environment variable. Each line in the PASSWD file contains the following user information:

- Login name
- Password
- User ID (uid)
- Group ID (gid)
- Full name
- Home directory
- Login shell

As additional security, the password is encrypted in the file using a randomly chosen encryption key. Because each machine uses a different key, a user can use the same password in different machines with no loss of security. The user's full name, home directory, and login shell fields are provided only for UNIX compatibility, and are not used by OS/2.

Notes:

1. You can update the PASSWD file using the PASSWD command. You should not edit the PASSWD file manually because of its special syntax.
2. If the PASSWD file contains path information in one or more of its fields, the colon normally used as part of the OS/2 path will appear in the PASSWD file as a semicolon. This is normal, and is due to the fact that PASSWD uses the colon character to separate the fields.

The syntax of PASSWD command is as follows:

```
passwd [-a] [-f] [-s] username
```

-a Add a new user to the password file
-s Change the shell for a user
-f Change the full name for a user

The following screen shows how to use the PASSWD command to add a user:

```
[C:]passwd -a user01
Login name: user01
Password? ******
Verify password? ******
UID?
GID?
Full name? (spaces are allowed)User for NFS testing
Home directory?
Shell?

[C:\]type \mptn\etc\passwd
shlee:aFRCSnnGSFUW2:0:0:S.H. Lee:d;\temp:
user01:UeN/Oky8sV.xw:0:0:User for NFS testing::
[C:\]
```

After you start PCNFSD, all error messages and warnings are logged to SYSLOGD (if running). Whenever a client's authentication fails, a message is logged to SYSLOGD. If a client repeatedly fails to authenticate, you should consider investigating the problem, because it could be an unauthorized attempt to access your data.

You can start SYSLOGD from an OS/2 command line, or you can set up SYSLOGD to autostart. The SYSLOGD syntax is:

```
syslogd [-t target]
```

-t target Specifies the target host, where target can be:

- A fully qualified path name
- An IP address
- An IP hostname

If target specifies a fully qualified file name, SYSLOGD will append all messages to that file. If target is an IP address or IP hostname, SYSLOGD forwards all messages to that host's SYSLOGD. If the -t option is not specified, SYSLOGD defaults to a file called SYSLOG.MSG in the directory specified in the ETC environment variable.

Notes:

1. You should inspect the target file occasionally, because the file will grow depending on which applications are running and how many messages they record. In addition, you should monitor all recorded program errors (for example, authentication failures).
2. Because TCPCFG does not currently recognize SYSLOGD, every time you run TCPCFG, it will remove SYSLOGD from TCPSTART. Therefore, you should start SYSLOGD from the TCPEXIT.CMD file located in the TCPIPBIN subdirectory. If this file exists, TCPSTART.CMD executes it as its final action.

The following screen shows how to start SYSLOGD:

```
[C:]syslogd
syslogd version 1.0
output going to "C:\MPTN\ETC\syslog.msg."
syslogd running...
```

The following screen shows how to view the file which SYSLOGD logged:

```
[C:]type c:mptnetcsyslogd.msg
96/02/05 15.06: pcnfsd: user 'user01' authentication error

[C:\]
```

15.4.2 Mounting from an IBM TCP/IP for DOS NFS Client

The IBM TCP/IP for DOS NFS client has the following commands available:

MOUNT	MVSLOGIN	MVSLOGUT
NFSDOWN	NFSPING	NFSPRINT
NFSSET	NFSSTAT	QMOUNT
SHOMOUNT	SHOWEXP	TODOS
TOUNIX	UMOUNT	

Before you can use NFS, you need to load, enable, and configure the DOS NFS terminate and stay resident (TSR) program. To do this you can use the CUSTOM utility to customize your DOS TCP/IP environment. Once NFS is started you can enter NFS commands.

To mount the E:TEMP directory of OS/2 NFS server dhcp1 as your F: drive enter the following command:

```
mount -i -u -g f: dhcp1:e:temp
```

Now you can use dhcp1's E:TEMP directory as your F: drive, subject to any access rights stated by the NFS server.

15.4.3 Mounting from an AIX NFS Client

To run the MOUNT command on an AIX NFS client, you must have root user authority or be a member of the system group. This is because mounting an NFS or other file system affects the file system for all users on the AIX machine.

Considering the OS/2 NFS server's read and write buffer size, the AIX mount command should be entered with the following options:

- rsize=4096
- wsize=4096

The mount of host dhcp1's E:TEMP directory to the /u/shlee/mount mount point is shown on the following screen using SMIT:

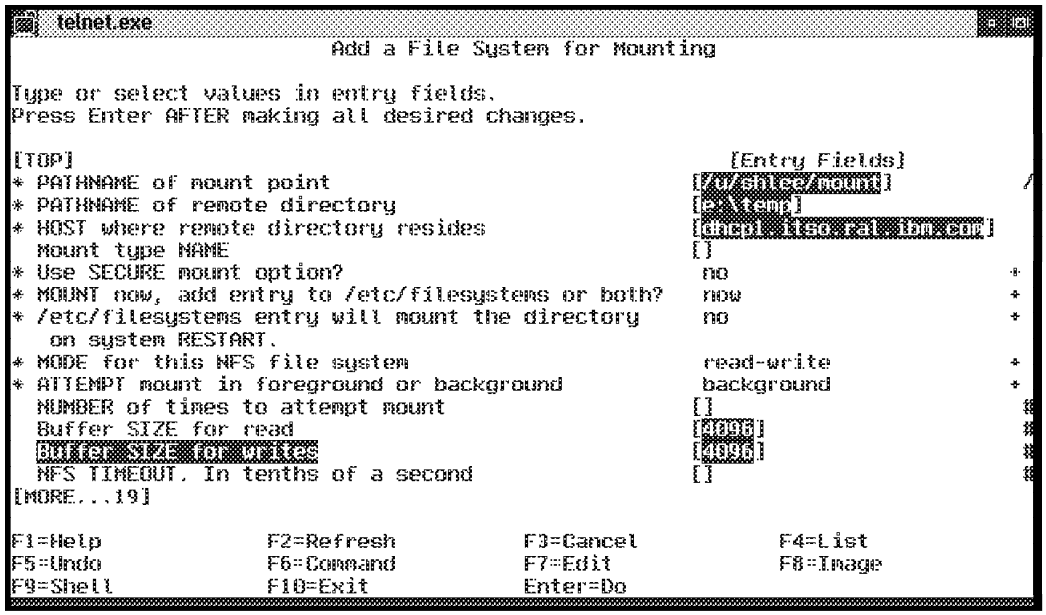


Figure 210. Exporting a Directory for NFS from AIX

The following shows how an NFS mounted OS/2 drive's files can be listed under AIX:

```
$ pwd
/u/shlee
$ cd mount
$ pwd
/u/shlee/mount
$ ls -la n*
-rw-rwSrw-   1 demohp   staff      91824 Feb  1 01:22 mount/nfs00100.EPS
-rw-rwSrw-   1 demohp   staff     133932 Feb  1 01:44 mount/nfs00101.EPS
-rw-rwSrw-   1 demohp   staff     130610 Feb  1 01:45 mount/nfs00102.EPS
$
```

To unmount a remote file system enter the umount command. For example to unmount all mounted file systems from OS/2 NFS server dhcp1.itso.ral.ibm.com enter at the AIX command prompt:

```
umount -n dhcp1.itso.ral.ibm.com
```

Chapter 16. Extended Networking

This chapter describes the extended networking capabilities of TCP/IP for OS/2 to connect to a wide area network. The X.25 and SNALINK connectivity is provided by the Extended Networking kit of the product.

16.1 Connecting TCP/IP for OS/2 to an X.25 Network

An X.25 interface allows you to connect to an X.25 packet switching network on which you can use the TCP/IP protocols for communications. You can use most of the same TCP/IP applications over an X.25 network as if your host were connected to a LAN.

To connect your OS/2 system to an X.25 network in order to use it for TCP/IP communications, you need the following:

- X.25 Interface Co-Processor Adapter (Micro Channel or ISA)
- Appropriate X.25 attachment cable
- Subscription to an X.25 network (obtain from your local PTT)
- Attachment to an X.25 network (obtain from your local PTT)
- OS/2 Warp
- TCP/IP V3.x
- IBM Communications Manager/2 V1.11
- X.25 Interface Co-Processor Adapter Option Diskette

Notes:

1. The subscription to an X.25 network lists the necessary parameters that need to be specified in the configuration files for Communications Manager/2 V1.11 and TCP/IP for OS/2 as shown later in this chapter.
2. The attachment (modem) to an X.25 network (DCE; Data Circuit Terminating Equipment) determines the type of cable you need to attach the DCE to the X.25 Co-Processor.
3. The X.25 Co-Processor acts as DTE (data terminal equipment).
4. Communications Manager/2 V1.11 is required to provide the X.25 subsystem APIs; these are not provided with TCP/IP for OS/2.

The following figure shows how TCP/IP for OS/2 uses the X.25 subsystem of Communications Manager:

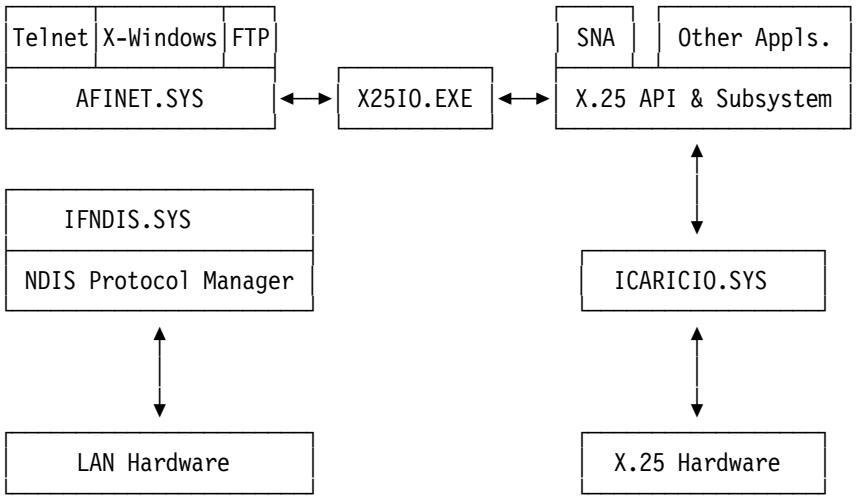


Figure 211. Communications Manager X.25 Subsystem Used by TCP/IP for OS/2

16.1.1 X.25 Installation and Configuration

OS/2 allows multiple X.25 adapters, but TCP/IP for OS/2 allows you to assign only one IP address to an X.25 interface.

The X.25 interface for TCP/IP for OS/2 is part of the Extended Networking kit of the product. To install it, insert the Extended Networking kit diskette into drive A: and enter:

```
A:TCPINST
```

You can install the Extended Networking kit from a remote drive as well.

Since you need Communications Manager/2 V1.11 to provide the X.25 subsystem API used by TCP/IP for OS/2, you first need to create a Communications Manager configuration file for X.25. To simplify that process, TCP/IP for OS/2 provides the following sample configuration files:

Sample	Purpose
X25CM1.CFG	Configuration file for a local workstation using Communications Manager/2 V1.11.
X25CM2.CFG	Configuration file for a remote workstation using Communications Manager/2 V1.11.

In order to use the sample configuration files, you need to copy them from the MPTNETC directory to the CMLIB directory. The following figures guide you through that process using Communications Manager/2 V1.11 and one of the sample configuration files provided with the Extended Networking kit (X25CM1.CFG):

- Select **Communications Manager Setup** from the Communications Manager Group on your OS/2 desktop. Select a configuration file that you want to work with (in our case X25CM1.CFG), and reply with Yes to use that configuration file for this workstation. On the CM Configuration Definition, select **X.25** from the Workstation Connection Type list, and **X.25 APIs** from the Feature or Application list:

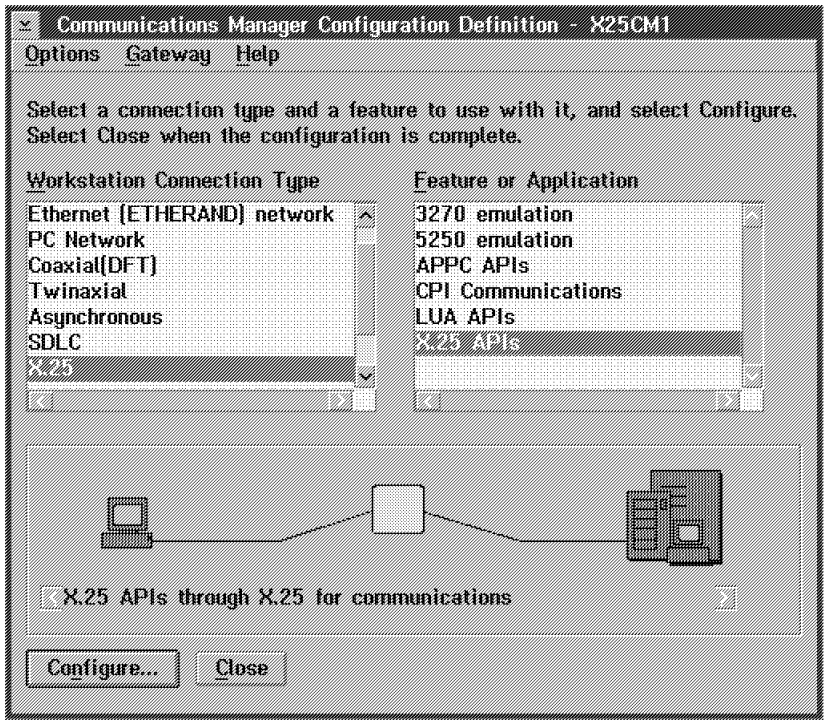


Figure 212. CM Configuration Definition for X.25

- This leads you to the Profile List Sheet for X.25:

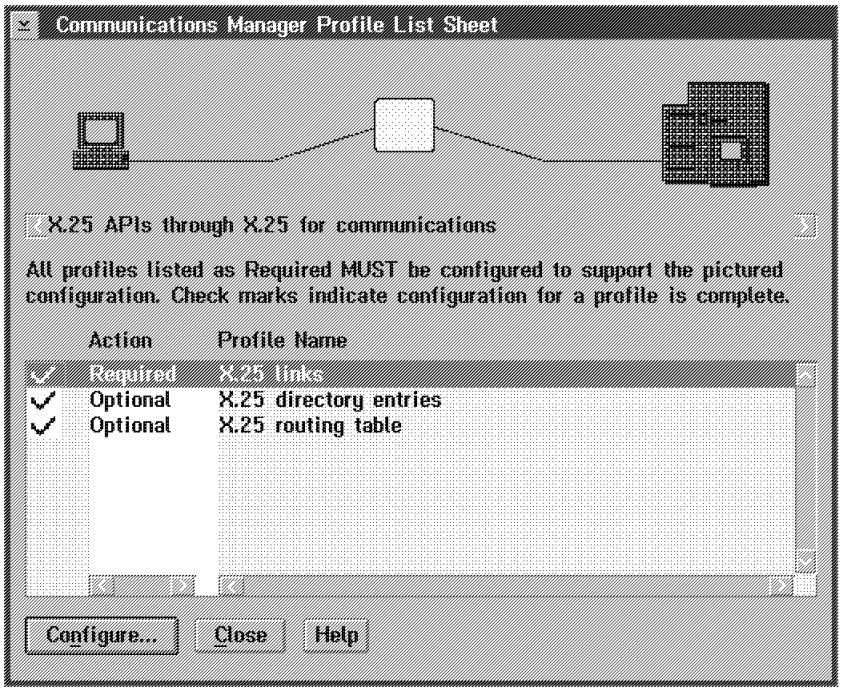


Figure 213. CM Profile List Sheet for X.25

- The first to configure is an X.25 link. One link has already been created in the sample configuration file, but you need to adjust definitions and parameters to meet the specifications of your X.25 subscription. If you use a different configuration file or create one from scratch, you must select **Create** on the X.25 Links menu, otherwise select **Change**:

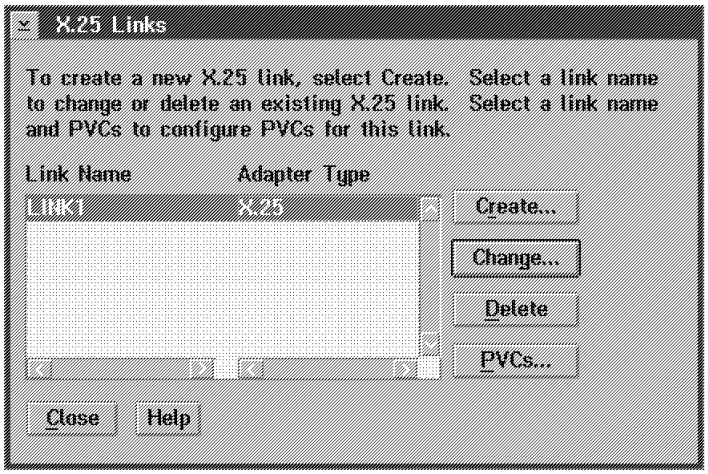


Figure 214. Configure X.25 Links

- Configure the X.25 link parameters according to your X.25 subscription.

  This selection allows specification of both link connection and network-related information. Link connection information specifies characteristics of the link itself, while network-related information must be specified before the link profile can be used.

  Setting **Meaning**

  Link Name The name of the link profile to create. Duplicate names are not allowed.

  Adapter Type Select **X.25**.

  Slot number The actual PS/2 slot number where the X.25 adapter is installed.

  Network type Specify the characteristics of the packet switching network that X.25 is connected to. Communications Manager provides a Help panel giving the appropriate value to enter for the main X.25 networks in the world. Most networks have the characteristics of network type 1; the ones that have different values are the following:

Table 29. X.25 Network Types

Network	Type
Austria DATEX-P	5
Germany DATEX-P	6
Japan ISDN	9
United Kingdom PSS basic	2
United Kingdom PSS extended	3
United States TELENET	4

  Enter this value if your network is not listed.

  Local CCITT compliance This item specifies whether the X.25 network complies with the 1980 or 1984 CCITT recommendation.

  Link set up mode This item specifies whether, on the network, the DTE or DCE is responsible for initiating the link connection. Select **DTE** to indicate that the DTE should actively poll the network by sending an SABM (Set Asynchronous Balance Mode) at intervals. Select **DCE** to indicate

that the network actively polls the DTE by sending an SABM at intervals. In this case, the DTE will remain passive until an SABM command is received.

Local DTE address This item specifies the address that was assigned to the DTE when subscribing to the network. Up to 15 numeric characters are allowed.

Note: If the Data Network Identification Code (DNIC, which specifies the country and the telecommunications service within the country) is required, the first four characters are used for this purpose. The DNIC is not always required. Local calls or a private network use their own addressing method.

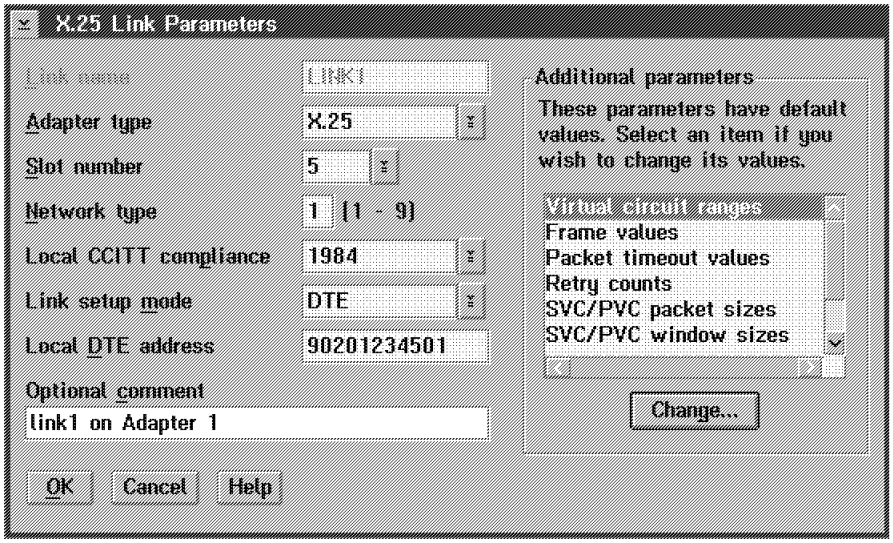


Figure 215. Configure X.25 Link Parameters

- The following selection allows specification of the range of logical channel numbers that can be assigned to PVCs and SVCs. This information can be obtained from the form that the network provider returned when access to the network was requested.

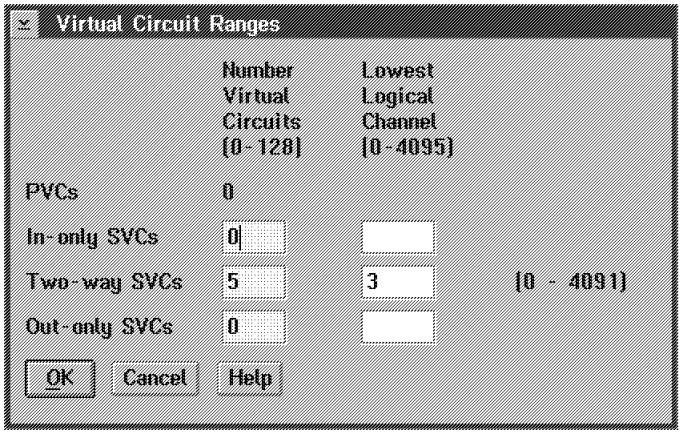


Figure 216. Configure Virtual Circuit Ranges for X.25

- The following selection allows specification of information related to the X.25 frame level. This includes the frame sequence modules, window size, retry count, and timeout values. These items are related to the link (frame) level, and care should be taken not to confuse them with similar named items in the SVC/PVC Window Sizes menu.

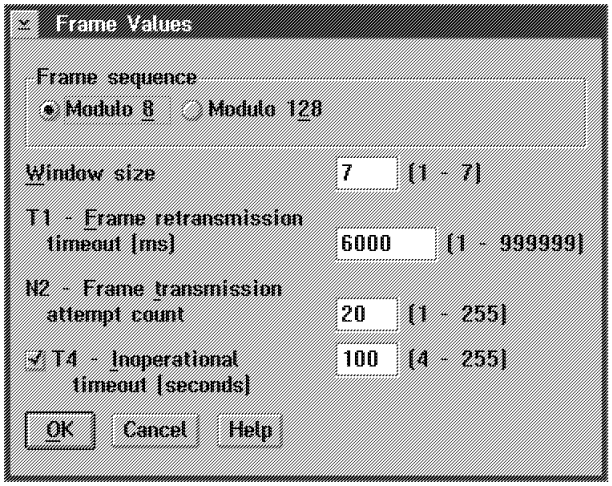


Figure 217. Configure Frame Values for X.25

- The following selection allows specification of timeout values for the X.25 packet level.

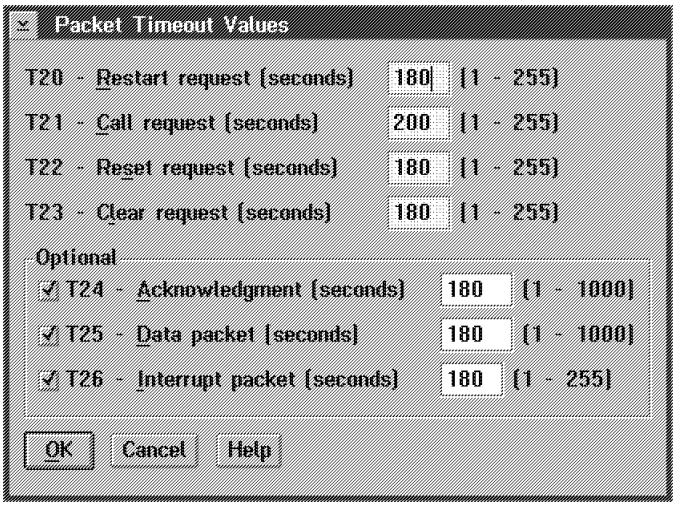


Figure 218. Configure Packet Timeout Values for X.25

- The following selection allows specification of retry counts for the X.25 packet level.

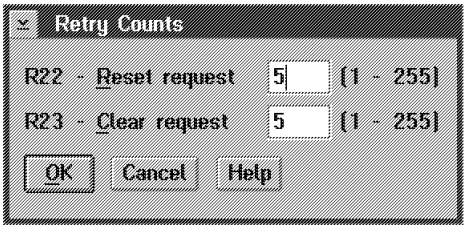


Figure 219. Configure Retry Counts for X.25

- The following selection allows specification of information related to the link's virtual circuits. The SVC packet sizes for TCP/IP should be set to 1024.

  Note: Packet sizes and window sizes can only be negotiated if subscribing to the flow control parameter negotiation facility.

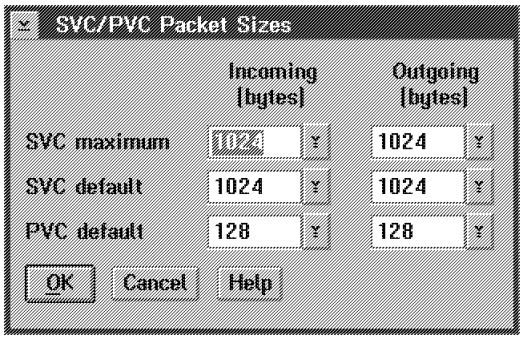


Figure 220. Configure SVC/PVC Packet Sizes for X.25

- The following selection allows specification of information related to the link's virtual circuits. It consists of the packet sequence number modules, and window sizes.

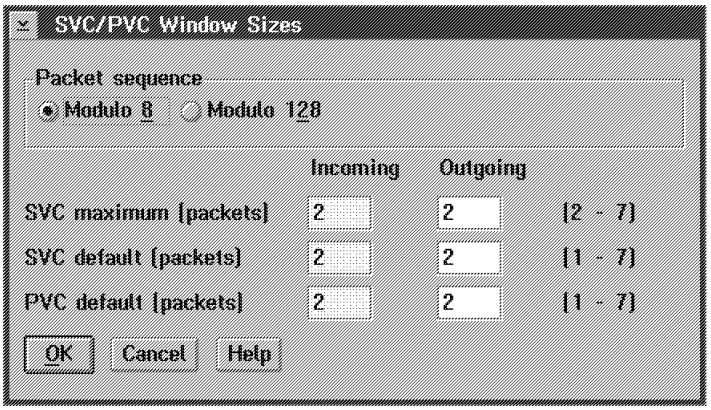


Figure 221. Configure SVC/PVC Window Sizes for X.25

- The following selection allows specification of information related to your modem:

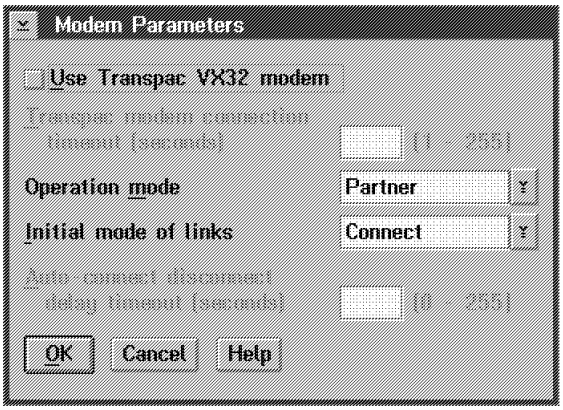


Figure 222. Configure Modem Parameters for X.25

Setting **Meaning**

Use Transpac V.32 modem This item specifies whether the link is connected through a VX32 modem provided by the French Transpac network. The VX32 modem supports either one incoming SVC or up to eight outgoing SVCs. If an incoming call is in progress, no outgoing calls can be placed. If an outgoing call is in progress, no incoming calls can be received. No PVCs are supported. Additionally, you are prompted for a Connection timeout, indicating the amount of time the modem will attempt to connect to the network before discontinuing.

Notes:

1. See Transpac VX32 modem documentation for details of the link profile parameter values that are needed when using this equipment. The semi-automatic dialing option, manual dialing option, and SVC assignments are not supported.
2. If you configure X.25 over ISDN, the Transpac VX32 modem cannot be selected.

Operation mode This item specifies whether the PS/2 should operate in DTE or Partner mode. Select DTE when connecting a PS/2 to a network. Partner mode allows connection of a PS/2 to another X.25 DTE directly, for example, through a modem eliminator or a pair of modems without an intervening network. When in partner mode, a PS/2 acts like a DCE. This allows a high-speed point-to-point connection with another X.25 DTE.

Note: When connecting two PS/2s back-to-back (using two modems or one modem eliminator), one must be configured as DTE and the other as partner. Except for the DTE addresses, all other link parameters configured for the two machines must be identical.

Initial mode of links This item specifies how the X.25 support will start the link. Select **Auto-connect** or **Disconnect**. In the auto-connect mode, and if the link is not already connected, the X.25 support attempts to connect the link (at the frame level) whenever an application requires the link to:

- Make a call
- Allocate a PVC
- Receive an incoming call

If auto-connect is selected, a prompt will ask for the amount of time that the X.25 support delays disconnecting an auto-connected link.

Note: If using the Transpac VX32 modem, it is recommended that a 55-second disconnection delay be specified. Similarly, the X.25 support disconnects a link in auto-connect mode when it is not being used by any application, namely, when:

- All SVCs are cleared.
- All PVCs are freed.
- No application is listening for incoming calls.

In the Disconnect mode, the link is disconnected and will not be connected by the X.25 support until **Connect X.25 Physical Link** or **Autoconnect X.25 Physical Link** is selected from a Communications Manager X.25 Link Profile menu or API.

- You cannot configure ISDN parameters if you are using a standard X.25 connection, only if you are configuring for X.25 over ISDN.
- The next to configure are the X.25 directories. You need one local and one remote directory entry for TCP/IP. One of either entries is already contained in the sample configuration file but may require changes.

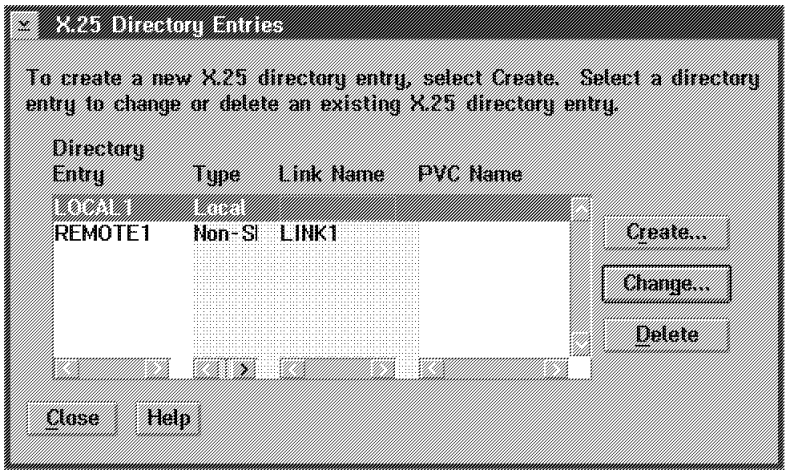


Figure 223. Configure X.25 Directory Entries

- The local directory entry allows assignment of a name to a local DTE. By referring to this name when coding X.25 API verbs, it is possible to write applications that are independent of the local DTE address. Since a PS/2 workstation can be connected to multiple X.25 networks, each local directory entry allows specifying a different network address (and address extension) for each of up to eight different network connections. Note that local directory entries are not required (or used) by SNA applications or SNA APIs.

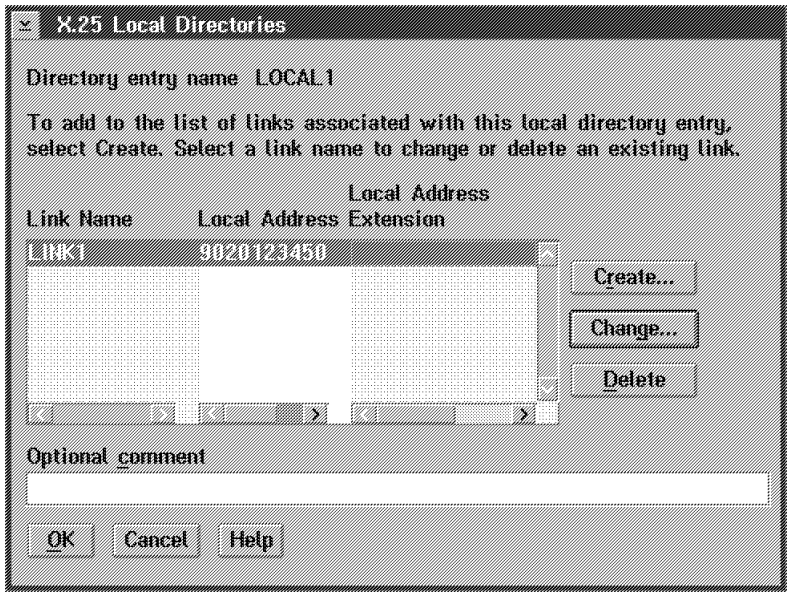


Figure 224. Configure X.25 Local Directories

- This entry is the content of the X25IP file. If you have more than one local entry for different SVCs or X.25 adapters, the names of the entries must also be in the X25IP file.

Figure 225. Configure Local Directory Entry for X.25

- The remote directory entry (SVC) allows the association of a name with a remote DTE when communicating using an SVC. This name is the entry to the X25DIR file, which also contains the corresponding IP address of the remote host. The X25DIR file contains each remote directory entry.

Figure 226. Configure Non-SNA SVC Directory Entry for X.25

- Finally, you need to define an X.25 routing table for TCP/IP. The sample already contains one that might need to be changed to suit your actual configuration. The X.25 feature can run up to 40 applications concurrently. When it receives an incoming call packet from the network, it must decide which application to route the call to. The X.25 feature uses the routing table to make this decision. Calls are routed to a particular application by the process of the X.25 support matching fields in the incoming call packet with corresponding fields in the routing table entries that the application specifies.

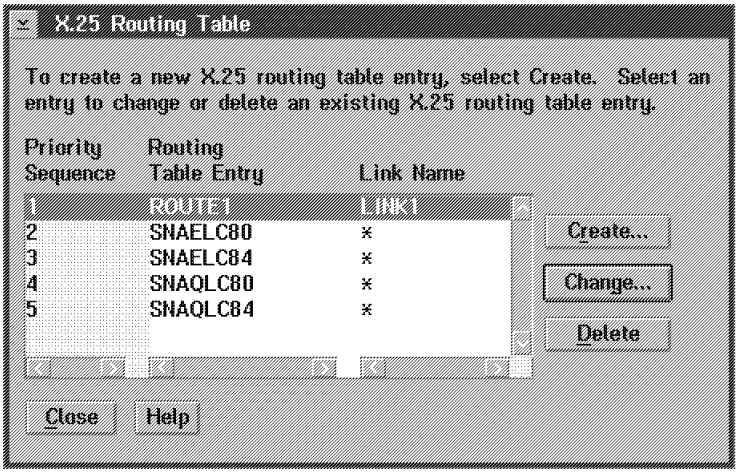


Figure 227. Configure X.25 Routing Tables

- The name of the route (ROUTE1) is the entry to the X25RTE file, which contains all routes to the corresponding remote hosts.

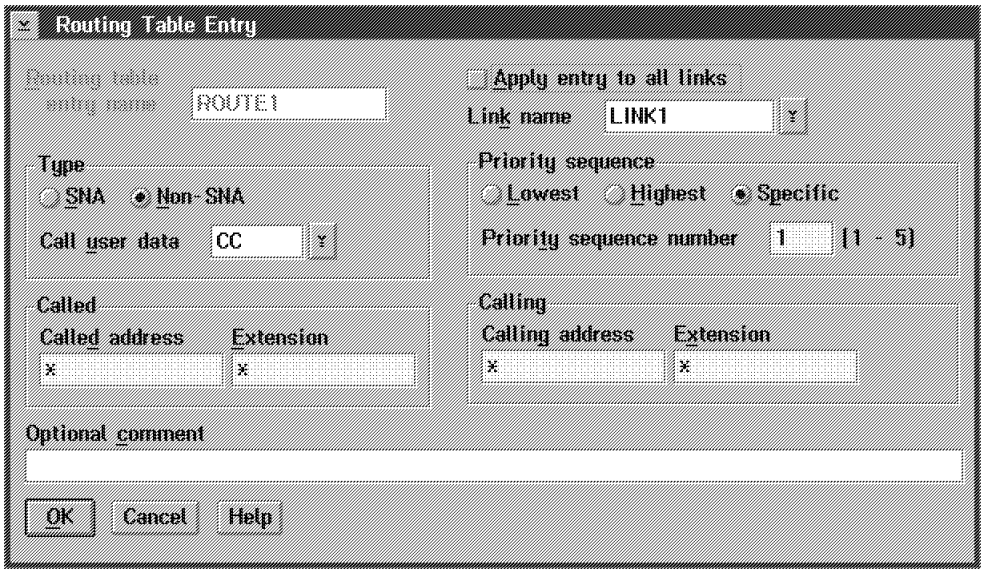


Figure 228. Configure Routing Table Entry for X.25

Setting	Meaning
Link name	Name of the X.25 link that this routing table entry applies to
Type	Type of application using this routing table. Select **Non-SNA** for TCP/IP.
Call user data	Enter CC for TCP/IP.

In addition to the configuration file, you need to copy the ICAAIM.COM file from the X.25 Co-Processor option diskette to the CMLIB directory. When Communications Manager/2 V1.11 is configured for X.25, it places the following statement in the CONFIG.SYS file:

```
DEVICE=d:CMLIBICARICIO.SYS
```

where d is the drive where Communications Manager is installed. This device driver will actually download the ICAAIM.COM file into the X.25 Co-processor at startup time of your OS/2 workstation.

After a Communications Manager configuration file has been created for X.25, you have to configure the X.25 connection for TCP/IP for OS/2. This can easily be done from the Configuration Notebook contained in the TCP/IP folder on your OS/2 desktop.

The following shows the first page of the X.25 Configuration Notebook:

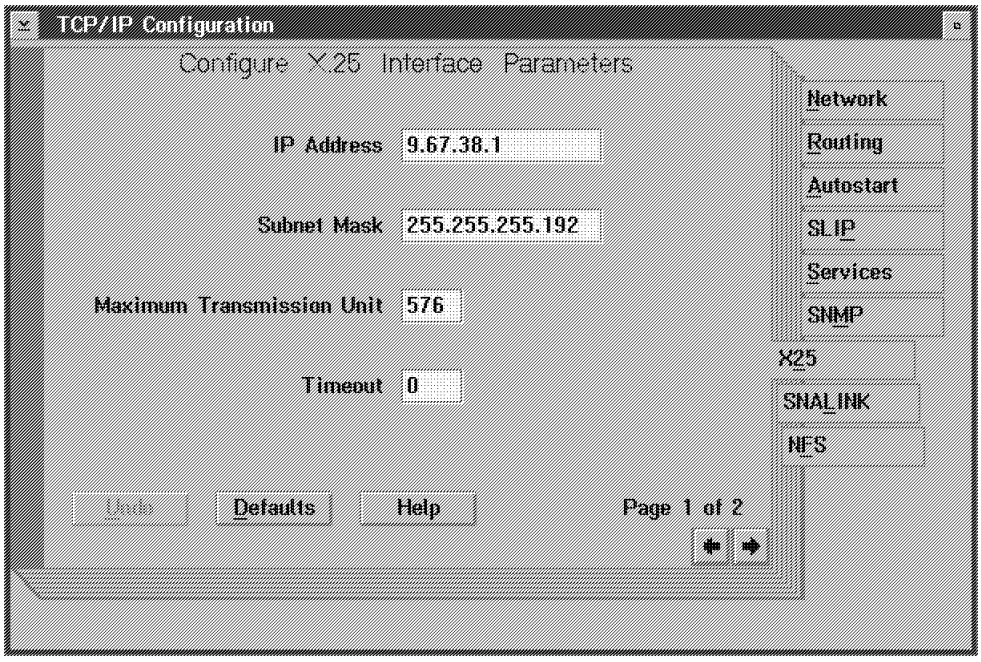


Figure 229. TCP/IP Configuration Notebook for X.25, Page 1

Setting	Meaning
IP Address	Your local IP address that will be assigned to the X.25 interface by the IFCONFIG command.
Subnet Mask	The subnet mask that will be assigned to the X.25 interface by the IFCONFIG command.
MTU	The maximum transmission unit size for the local X.25 interface. A value of 576 is recommended.
Timeout	The period of time before an X.25 SVC will close due to inactivity. A value of 0 means no timeout.

The following shows the second page of the X.25 Configuration Notebook:

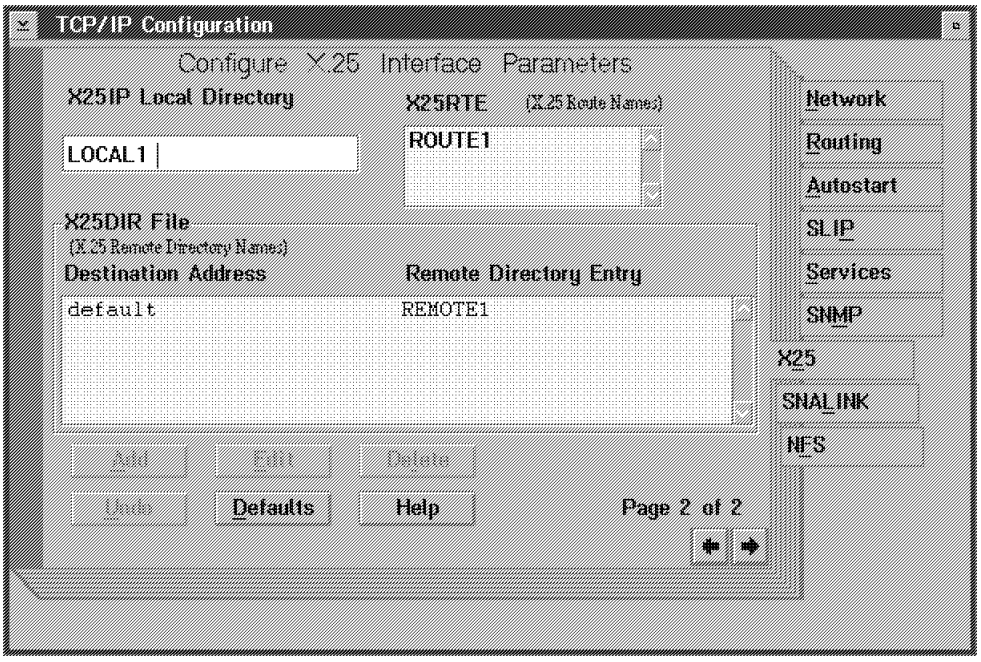


Figure 230. TCP/IP Configuration Notebook for X.25. Page 2

Setting	Meaning
X25IP	Local directory that identifies a local DTE address with an X.25 link.
X25RTE	Routing table that determines which incoming X.25 calls are to be routed to the TCP/IP application.
Destination Address	Associate a remote DTE address to an IP address.
Remote Directory	Specify the directory to direct outgoing IP packets to a remote DTE address.

The Configuration Notebook will make changes to the following files:

File	Purpose
MPTNETCX25IP	Contains the directory entry name of the local entry point with its link to the local DTE address.
MPTNETCX25RTE	Contains the routing table entry name.
MPTNETCX25DIR	Contains the directory entry name of the remote entry.
TCPIPBINX25.CMD	Command file that starts and initializes the X.25 interface for TCP/IP. It executes the following programs: • X25IO.EXE • XIOWAIT.EXE • IFCONFIG.EXE

The X25IO program is the driver that connects the TCP/IP interface to the X.25 interface. It must be started after Communications Manager and before IFCONFIG, and it must stay up as long as the TCP/IP connection on the X.25 interface is required. The following shows an active X25IO window:

```
X25: Available using INET interface unit #1.  X25IO version 2.0
```

The XIOWAIT program is used to make sure that IFCONFIG does not try to initialize the X.25 interface before X25IO has finished connecting to the X.25 API.

The IFCONFIG statement assigns an IP address to the X.25 interface on your workstation. To configure the X.25 interface, you must enter the IP address, the subnetmask parameter is optional. The maximum transmission unit (MTU) size should be set to 576 which is the default.

The following shows the X25.CMD on host routx25:

```
start x25io.exe
xiowait
ifconfig x25 9.67.38.1 netmask 255.255.255.192 mtu 576
```

To start the X.25 interface automatically, you need to call X25.CMD from a command file that is executed at startup time of your workstation, for instance STARTUP.CMD. When you do that, please remember the following:

Notes:

1. Communications Manager must be running before X25.CMD.
2. The X.25 link must be up, or X25IO will fail.
3. If you use static route statements on your X.25 TCP/IP connection, run X25.CMD before SETUP.CMD since the ROUTE command will fail on an interface that has not been initialized.

The following shows the STARTUP.CMD file on host routx25:

```
start c:cmlibcmstart.exe x25cm1
cmwait -s e
call c:\tcpip\bin\x25.cmd
call c:\tcpip\bin\setup.cmd
exit
```

The CMWAIT statement causes any program that wants to access Communications Manager APIs to wait until it is up and host session E is started. This may be different in your configuration and is no guarantee that the X.25 link has been successfully activated, but it helps to automate communications startup.

When you have finished configuring Communications Manager and TCP/IP for OS/2, restart your workstation.

For a more detailed discussion of how to configure Communications Manager, TCP/IP, and the X.25 interface, please refer to the online *TCP/IP for OS/2 Extended Networking Guide*.

16.1.2 X.25 Limitations for TCP/IP for OS/2

You should consider the following recommendations when using X.25:

- X.25 Permanent Virtual Circuits (PVCs) are currently not supported.
- A maximum of 16 active switched virtual circuits (SVCs) are supported. An SVC will become inactive after a specified period of inactivity.
- The MTU size is not negotiated with the remote host.
- The Department of Defense 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.

16.1.3 Starting the X.25 Connection

Before you initialize the X.25 interface for TCP/IP, Communications Manager must be running to provide you with the X.25 API interface.

Usually, an X.25 link is established when both ends are activated at the same time. This may, however, not always be feasible, or the link may drop during operation for several reasons. In that case, the Subsystem Management of Communications Manager/2 V1.11 can be used to activate or re-activate an X.25 link.

The following shows the Subsystem Management window of Communications Manager/2 V1.11:

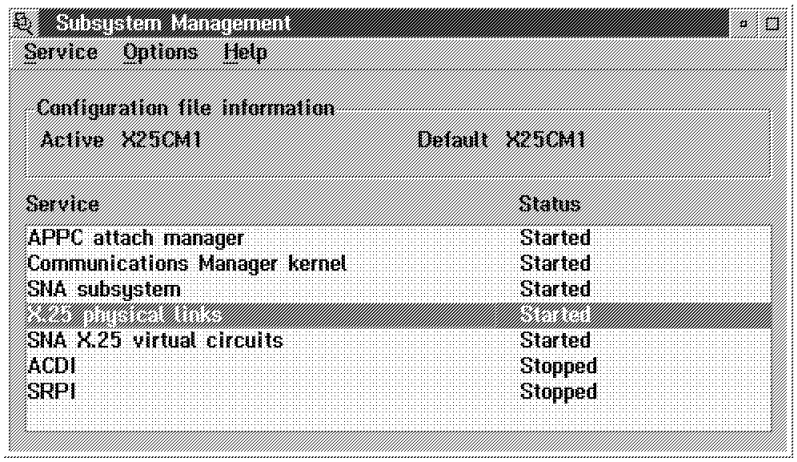


Figure 231. Communications Manager Subsystem Management

The following figure shows an X.25 link used for TCP/IP. It is active and connected, and there is no line activity at the moment:

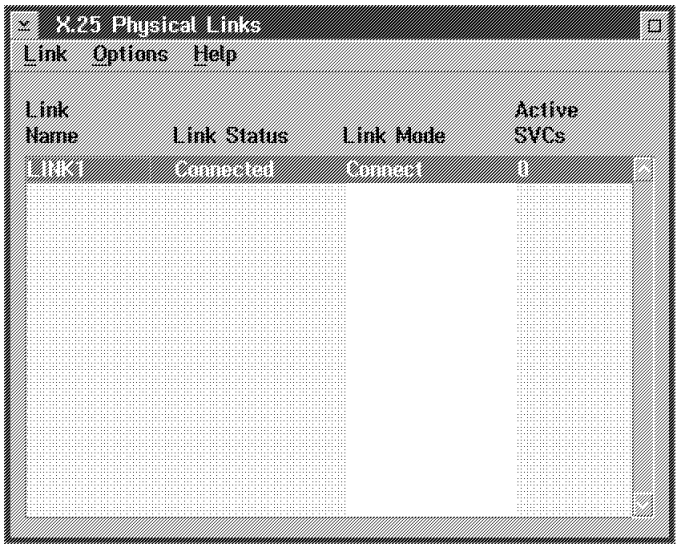


Figure 232. Manage X.25 Physical Links

If that still fails, you do need somebody on the other side of the X.25 connection in order to bring the link back up on either side simultaneously.

After the X.25 interface is successfully started you can start TCP/IP for OS/2. If you start it before, initialization will fail and also starting of TCP/IP services.

16.1.4 Using the X.25 Connection

Make sure that the X.25 interface on the other end of your link is also initialized using IFCONFIG before you start using this connection. To verify this, try to PING the remote address on the link.

In most cases, a workstation with a connection to a wide area network is used to route traffic from a LAN to remote destinations. You can route TCP/IP across the X.25 link using either the ROUTED daemon or static route statements. Static routes should be preferred, if:

- The bandwidth of the X.25 link is narrow.
- You want to use the X.25 link cost-effectively.
- Your partner on the X.25 link does not support the RIP protocol.

ROUTED can be used if the above points do not apply to your configuration, and if you want to use dynamic routing. ROUTED uses the Route Information Protocol (RIP) to exchange routing information to other hosts running ROUTED. It transmits its routing table every 30 seconds on all links that your host is attached to and therefore causes network traffic that is not directly related to TCP/IP applications.

16.2 Connecting TCP/IP for OS/2 Across an SNA Network Using SNALINK

This version of TCP/IP for OS/2 includes an SNALINK interface, which allows you to connect TCP/IP networks across an SNA network.

The Extended Networking kit provides an APPC program which will send and receive TCP/IP packets to another SNA node. This APPC program is designed to

run on Communications Manager/2 V1.11, which provides the base SNA functions.

There are many files supplied with the Extended Networking package to provide SNALINK functions. Most of these files are example configurations. Instead of using these example configurations, we document the steps that we took to create a working SNALINK from scratch, and describe our choices in the configuration so that you may understand them fully. The main files shipped with the Extended Networking kit for SNALINK are the following:

File/Directory	Description
BINSNALIO.EXE	An APPC program which sends and receives TCP/IP packets over an SNA link.
BINSNALWAIT.EXE	A utility which waits until SNALIO is running.
BINSNALIO.ICO	Icon for SNALIO.EXE
ETCSNALIP.CFG	Text file created by the Configuration Notebook program which contains APPC values that will be used by SNALIO.EXE.

This figure illustrates how TCP/IP packets are sent over the SNA Link:

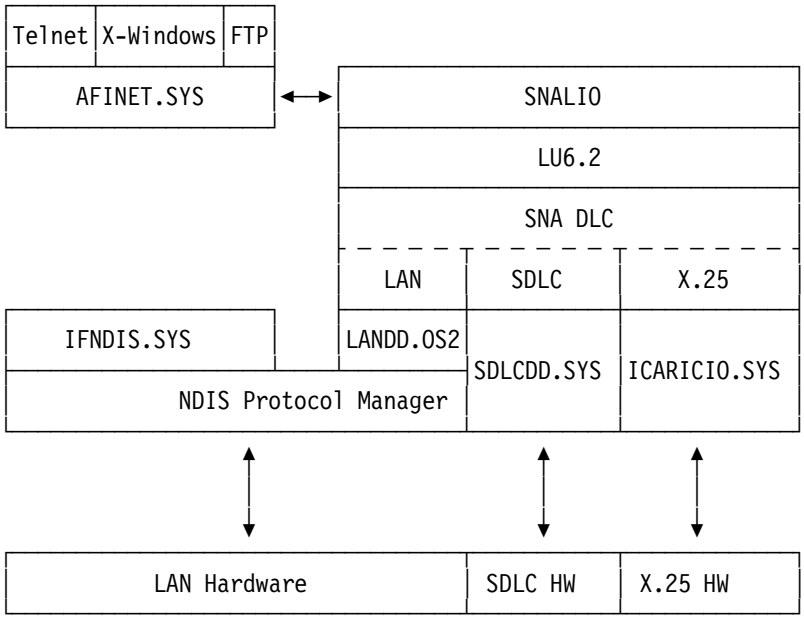


Figure 233. TCP/IP over SNALINK Protocol Stack

In this diagram, we show that all TCP/IP packets are sent through SNALIO. But you can also send TCP/IP packets to IFNDIS.SYS at the same time. This means that you can have a conventional IP link and SNA link working on a machine concurrently.

Your SNALINK connection can be established over any medium supported by Communications Manager for LU6.2 communications. For example:

- LAN
- SDLC
- X.25

16.2.1 Configuring Communications Manager

In this section, we provide instructions to configure Communications Manager for these example scenarios:

- OS/2 workstation to OS/2 workstation
- OS/2 workstation to OS/2 workstation via VTAM

The steps to set up these configurations are cumulative. We explain the OS/2 to OS/2 workstation scenario, and then the changes necessary for each of the other scenarios.

Each of the scenarios have been set up in the USIBMRA SNA network.

16.2.1.1 OS/2 Workstation to OS/2 Workstation

In this scenario we set up a direct SNA link between two OS/2 workstations that are physically connected using token-ring. This type of connection does not require any changes to the VTAM system controlling the SNA network. However we use naming conventions recommended by our network administrator, so that we may easily change our configurations for the other scenarios described later in this section.

We set up each machine with identical hardware and software. Each machine is equipped with a token-ring adapter and has these software packages installed:

- OS/2 Warp
- TCP/IP for OS/2 Extended Networking kit
- Communications Manager/2 V1.11

Figure 234 illustrates the SNA connections required for this scenario. This is an explanation of the names used in this figure and how they relate to conventions in Communications Manager Setup.

Name	Description
Alias	Alias of Logical Unit Sessions (LU Name)
LU	Logical Unit Sessions (LU Name)
SSCP	SSCP VTAM ID (Partner Network Node for connections through VTAM)
CP	CP Name (Network Node Name)
LAA	Locally Administered Address (Token-Ring Address)

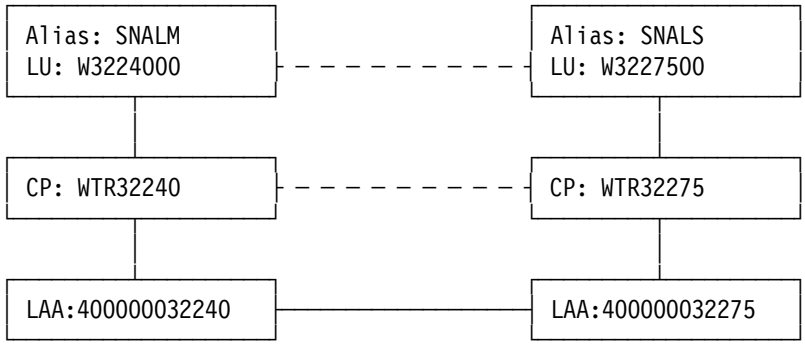


Figure 234. OS/2 to OS/2 Workstation SNALINK Configuration

Once these configurations are set up, the SNALIO program supplied with the Extended Networking kit will send and receive information to and from the LUs.

Here are the steps that we used to configure Communications Manager/2 for this scenario. We fully illustrate the set up of machine WTR32275, and also describe the parameter values that should be used when configuring machine WTR32240:

1. Start Communications Manager Setup from the Communications Manager/2 folder on your OS/2 desktop.

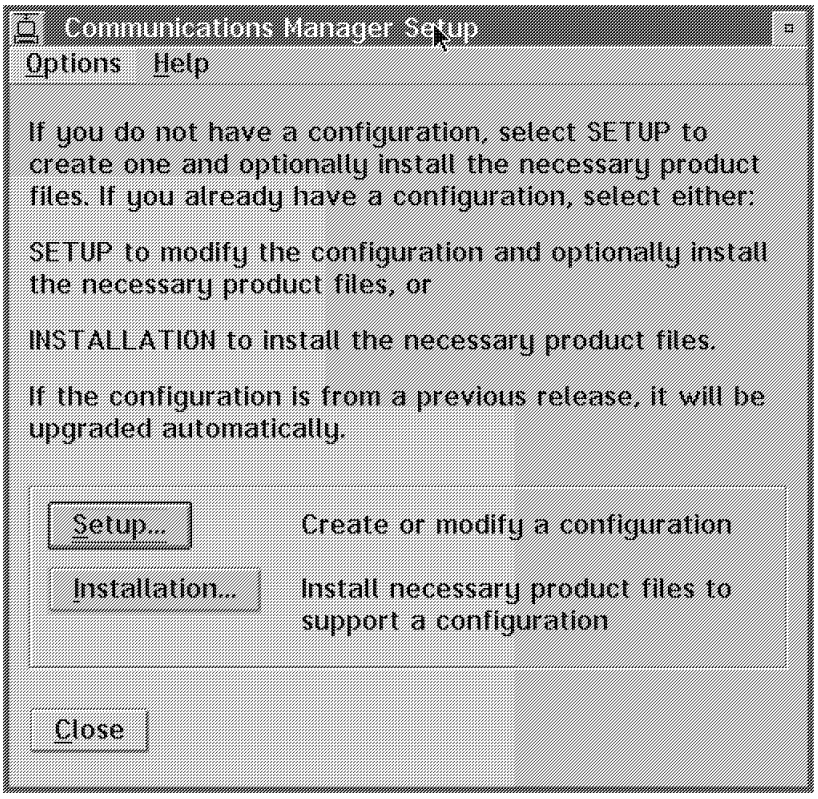


Figure 235. Communications Manager Setup

2. Click on **Setup**.
3. Type in the name and description of the configuration.

   Parameter Value

   Configuration SNALINK2

   Description For TCP/IP over SNALINK

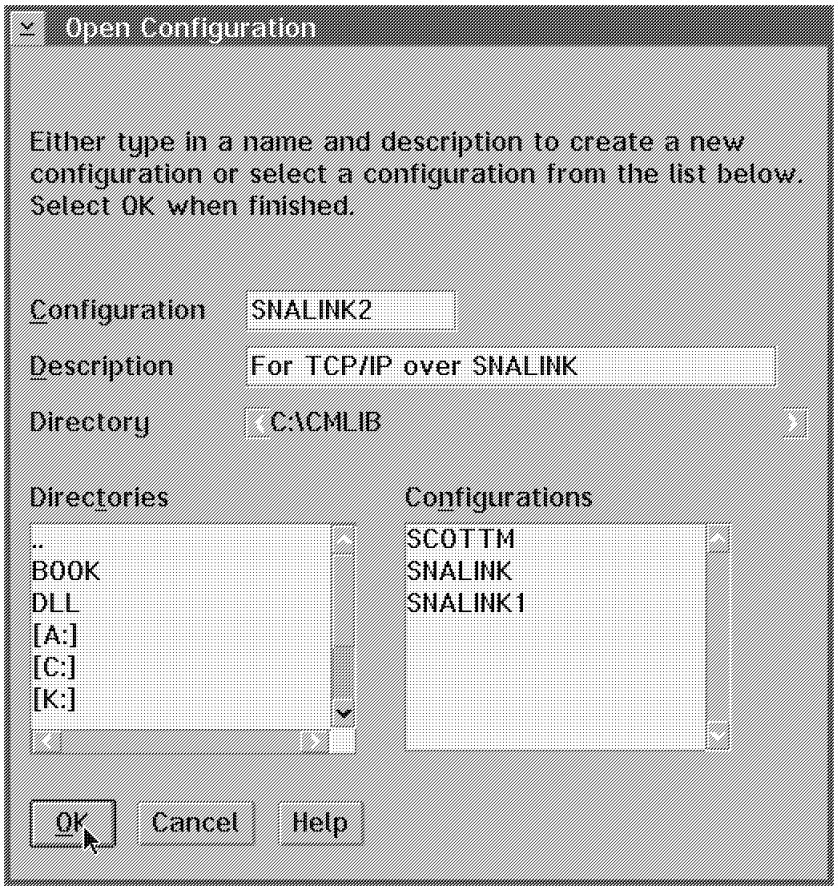


Figure 236. Open Configuration

Click on **OK**.

4. We are creating an entirely new configuration:

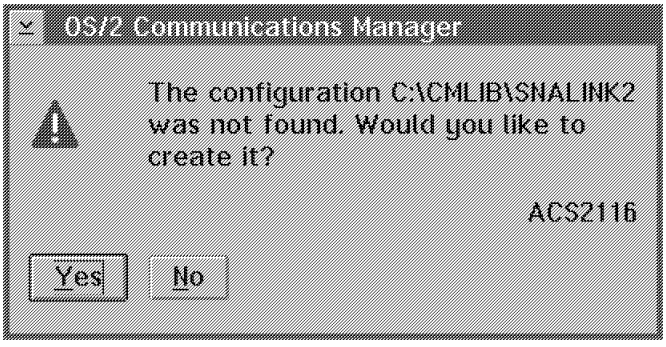


Figure 237. Create a New Configuration

Click on **Yes**.

5. We recommend that you use the Advanced Configuration option for this configuration. Click on **Options**, **Use advanced configuration** and **On**:

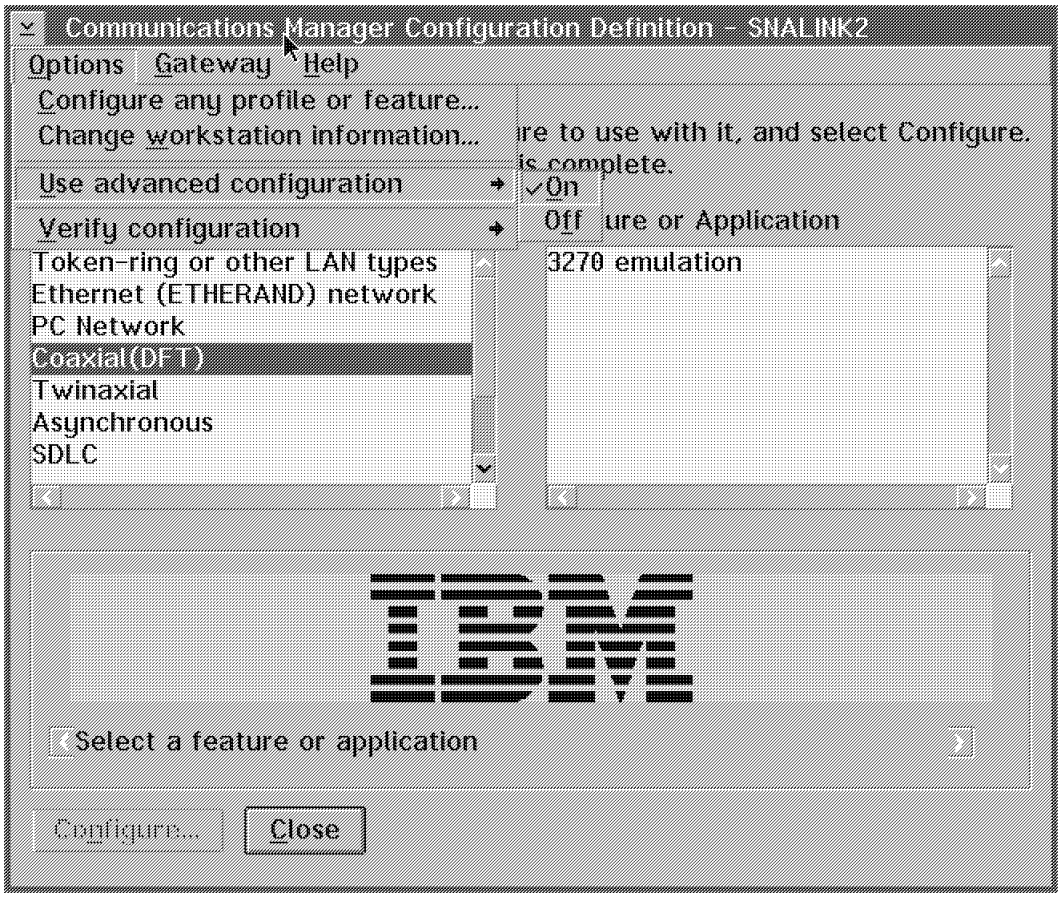


Figure 238. Communications Manager Configuration Definition - SNALINK2

6. In this example, TCP/IP for OS/2 will use an APPC link across a token-ring interface, so you should select **Token-ring or other LAN types** as your Workstation Connection Type and **APPC APIs** as your Application.

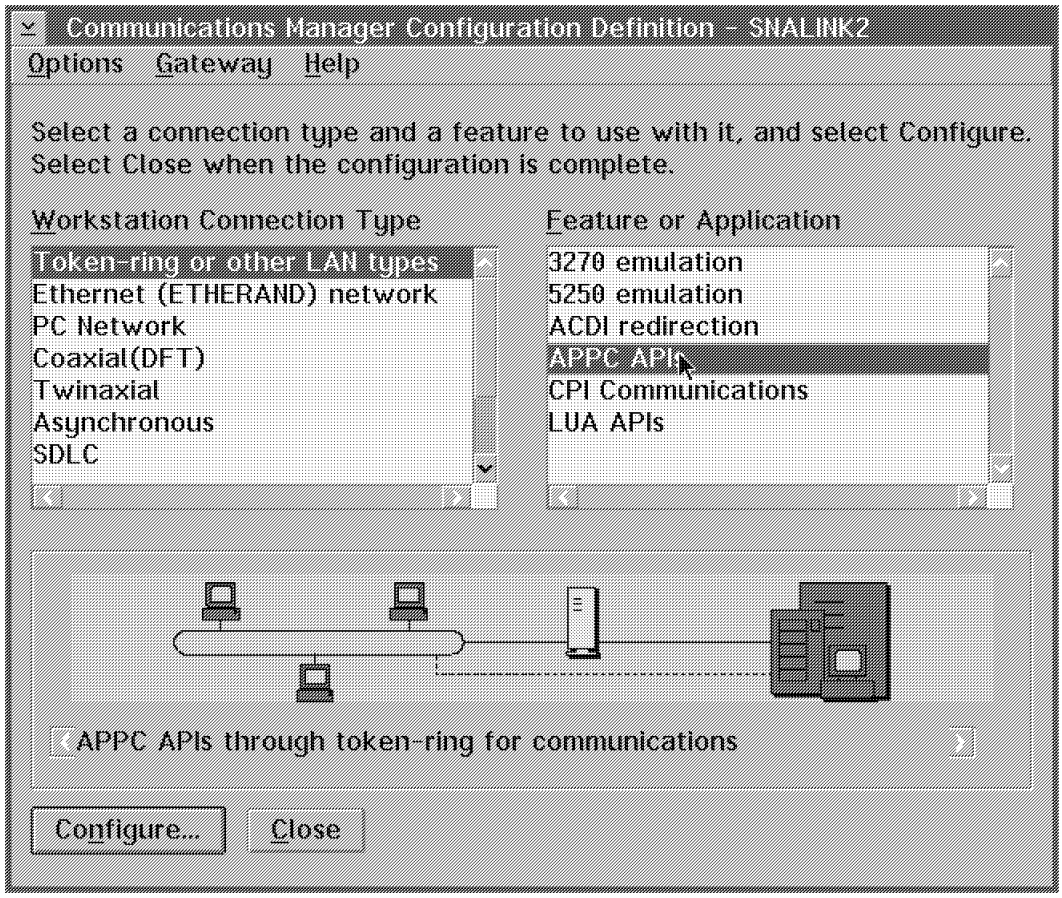


Figure 239. Communications Manager Configuration Definition - SNALINK2

Click on **Configure**.

7. You will now configure each profile in the Communications Manager Profile List Sheet:

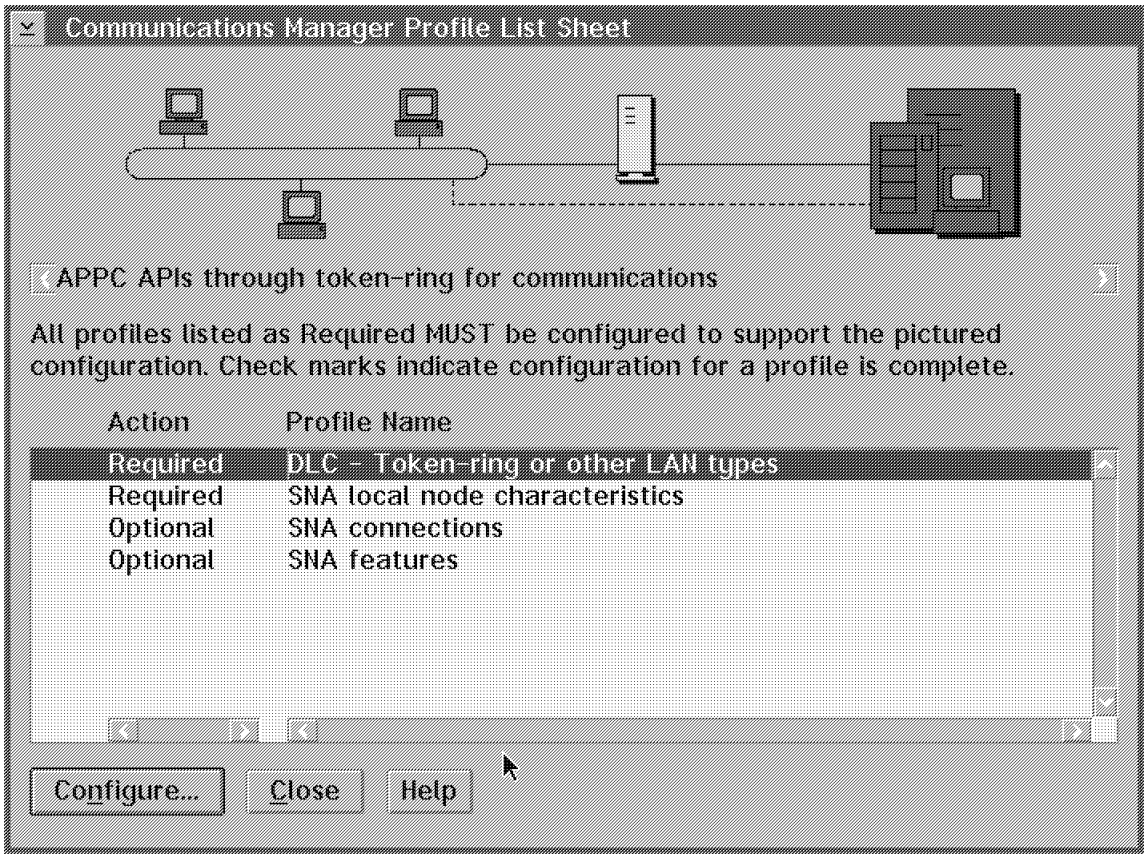


Figure 240. Communications Manager Profile List Sheet

Select **DLC - Token-ring or other LAN types**, and click on **Configure**.

8. This panel allows you to configure the interface between the SNA protocol layers in Communications Manager/2 and the token-ring adapter on your machine. Type in your C&SM LAN ID. This is usually the same as your Network ID. We used USIBMRA.

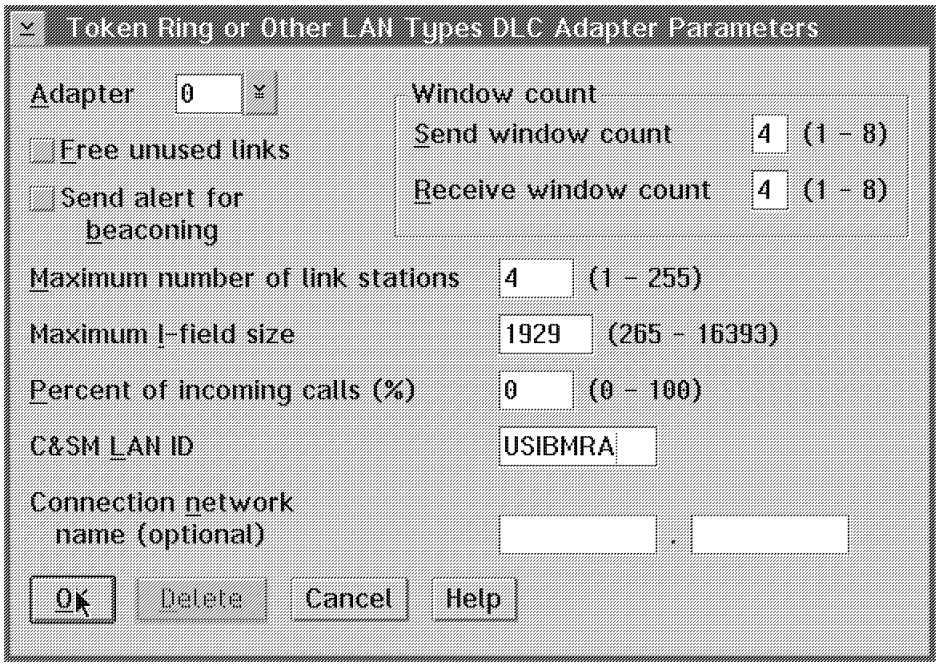


Figure 241. Token Ring or Other LAN Types DLC Adapter Parameters

Click on **OK**.

9. Select **SNA local node characteristics**.

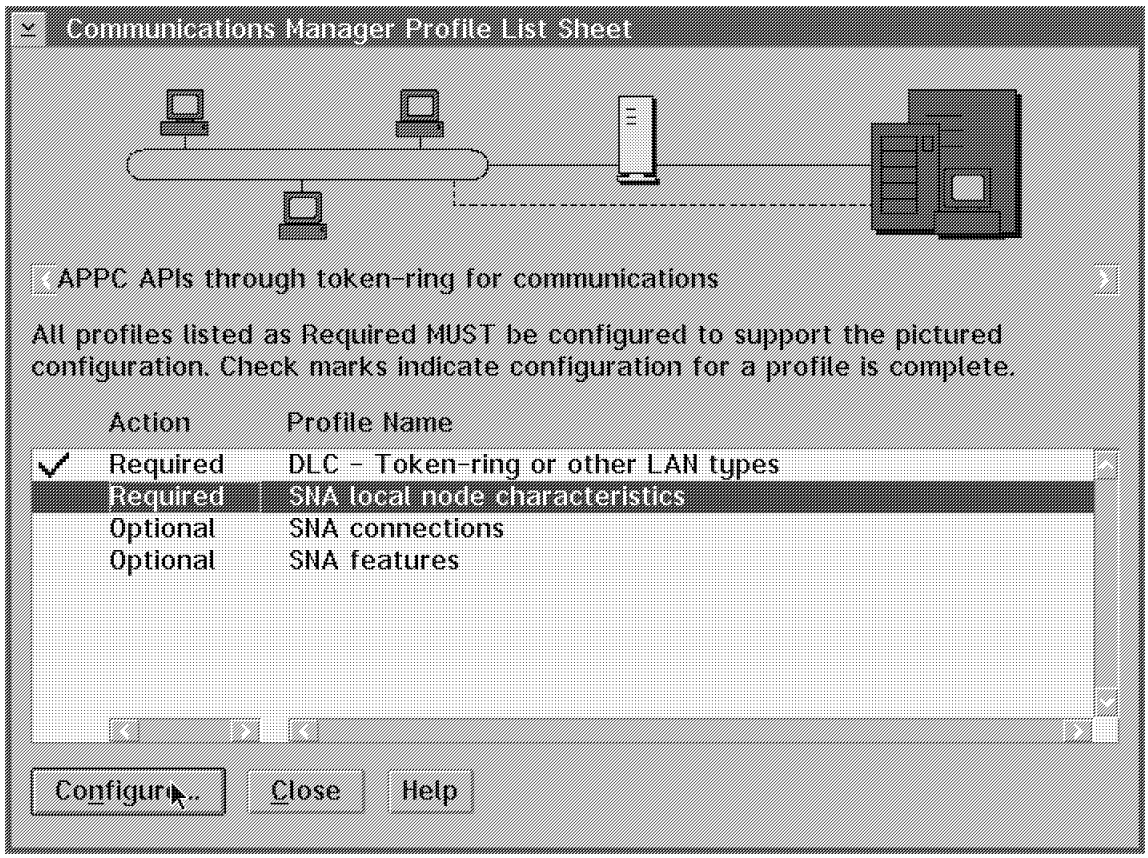


Figure 242. Communications Manager Profile List Sheet

Click on **Configure**.

10. This panel allows you to configure the SNA network values for this workstation:

Parameter	**Our Value**
Network ID	USIBMRA
Local Node Name	WTR32275
Local Node ID	32275

    When we configured machine WTR32240, we used these values:

Parameter	**Our Value**
Network ID	USIBMRA
Local Node Name	WTR32240
Local Node ID	32240

    Ensure that these values comply with your network coordinator's SNA naming convention. Click on **OK**.

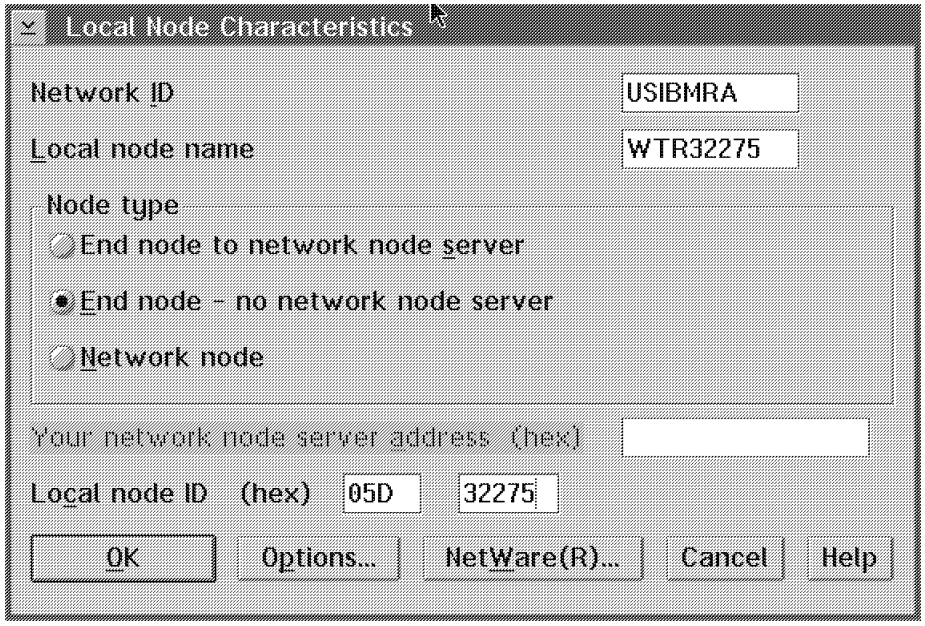


Figure 243. Local Node Characteristics

11. Select **SNA Connections**.

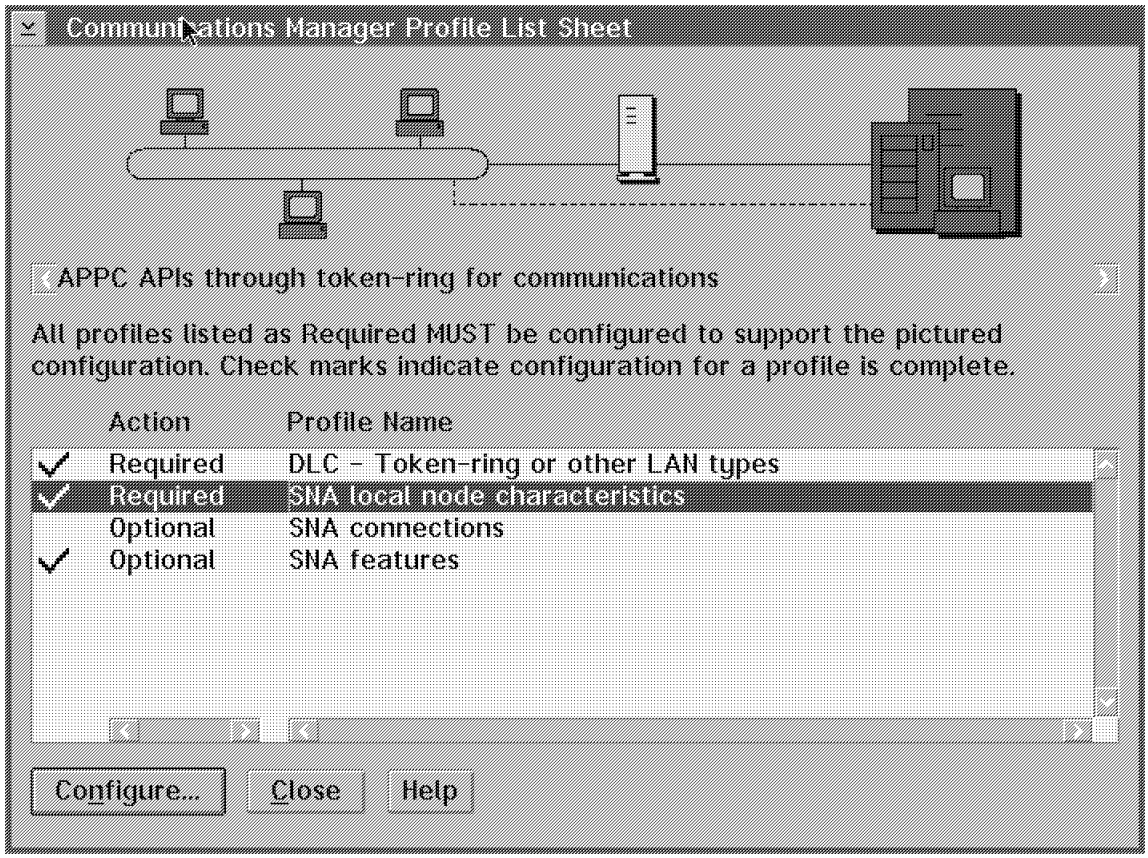


Figure 244. Communications Manager Profile List Sheet

Click on **Configure**.

12. This panel is used to configure SNA connections to other workstations. We configure a connection to machine WTR32240. Select **To Peer Node**.

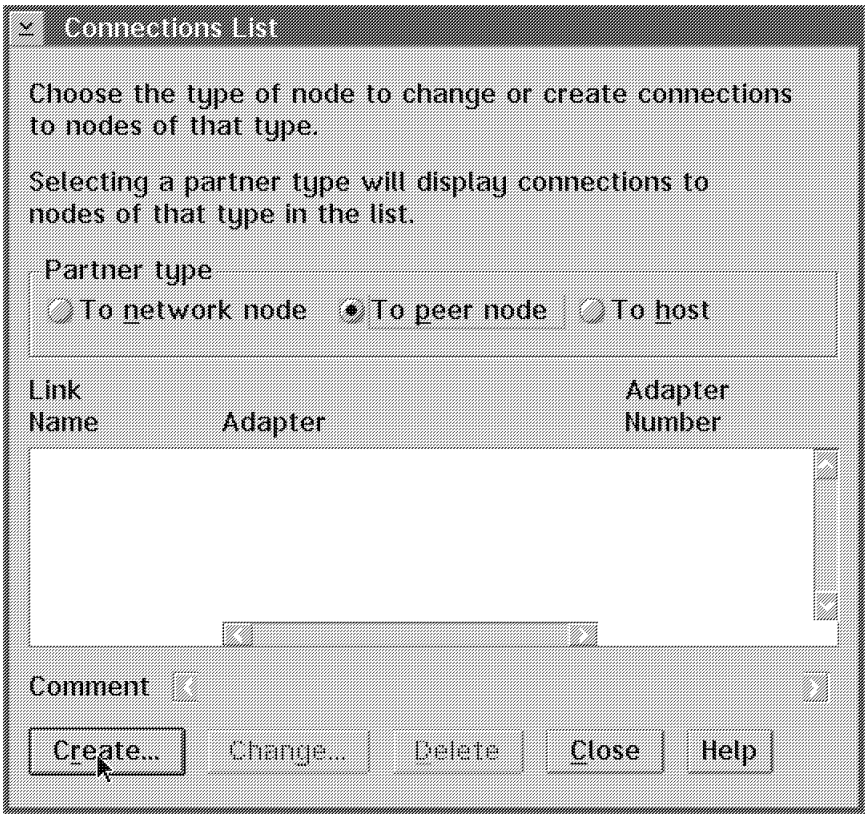


Figure 245. Connections List

Click on **Create**.

13. Select **Token-ring or other LAN types**.

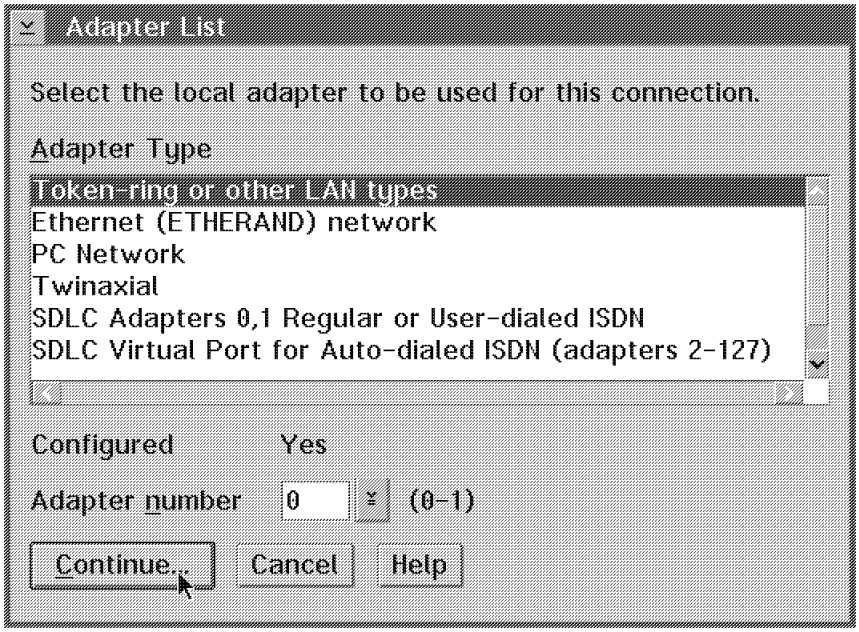


Figure 246. Adapter List

Click on **Continue**.

14. These are the values that we used to configure machine WTR32275:

Parameter	Our Value
LAN destination address	400000032240
Partner Network ID	USIBMRA
Partner Node Name	WTR32240
Optional Comment	Direct SNALINK via TR to Martin's machine

15. We used these values when we configured machine WTR32240:

Parameter	Our Value
LAN destination address	400000032275
Partner Network ID	USIBMRA
Partner Node Name	WTR32275
Optional Comment	Direct SNALINK via TR to Scott's machine

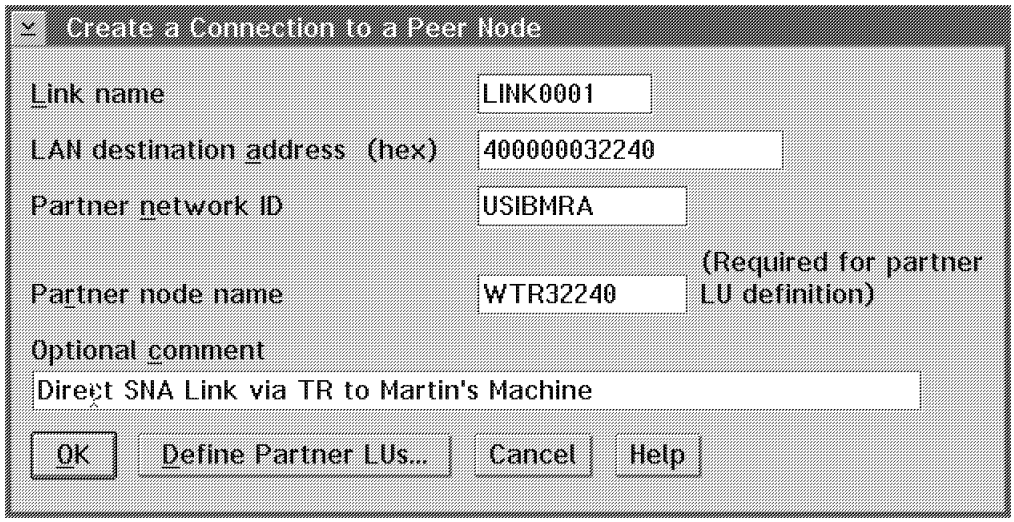


Figure 247. Create a Connection to a Peer Node

Click on **Define Partner LUs...**

16. This panel allows you to configure a Partner LU at the other end of the connection.

    You should ensure that you configure any partner LUs on this panel. Partner LUs can also be configured in the SNA features profile list, but SNALIO needs to know which connection can access the partner LU.

    These are the values that we used for machine WTR32275:

    Parameter Our Value

    LU name W3224000

    Alias snalm

    These are the values that we used for machine WTR32240:

    Parameter Our Value

    LU name W3227500

    Alias snals

Click on **Add**. You should now see the partner LU in the list on the right-hand side of the panel:

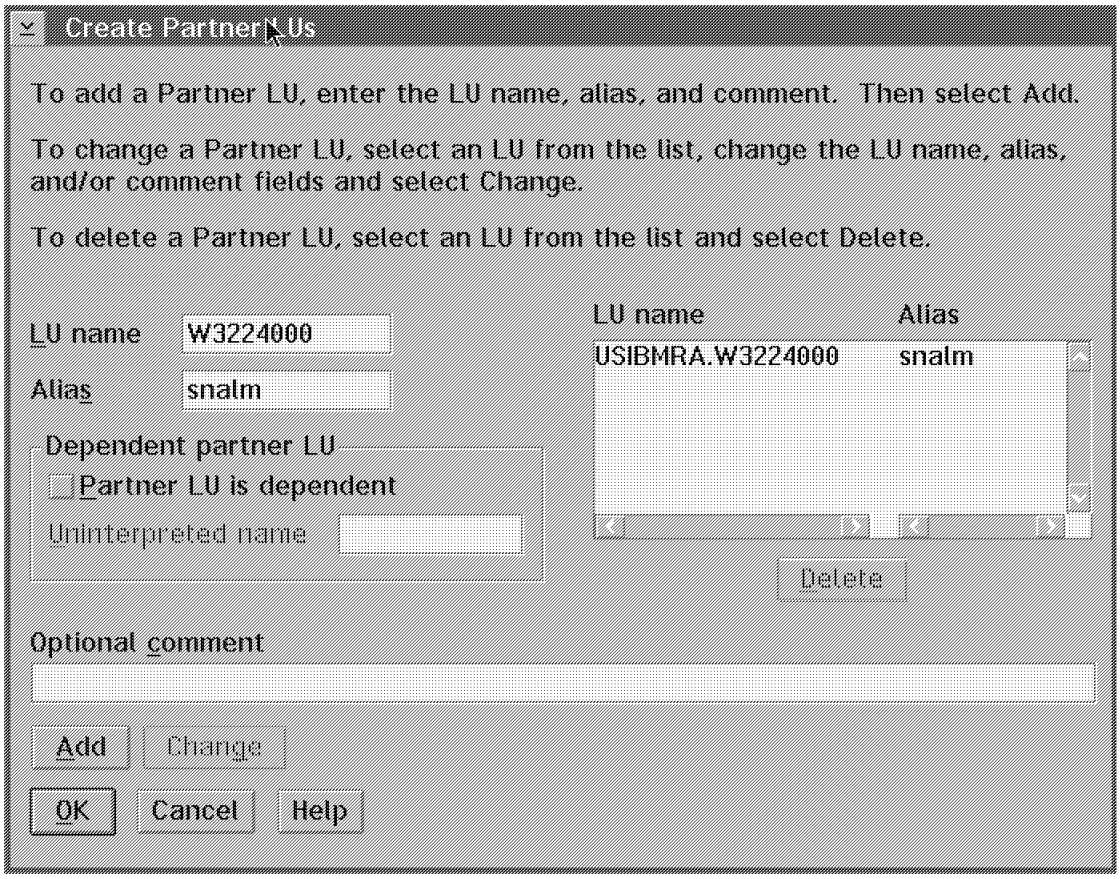


Figure 248. Create Partner LUs

Click on **OK**, and then click on **OK** again from the Create a Connection to a Peer Node panel.

Click on **Close**.

17. Select **SNA Features**.

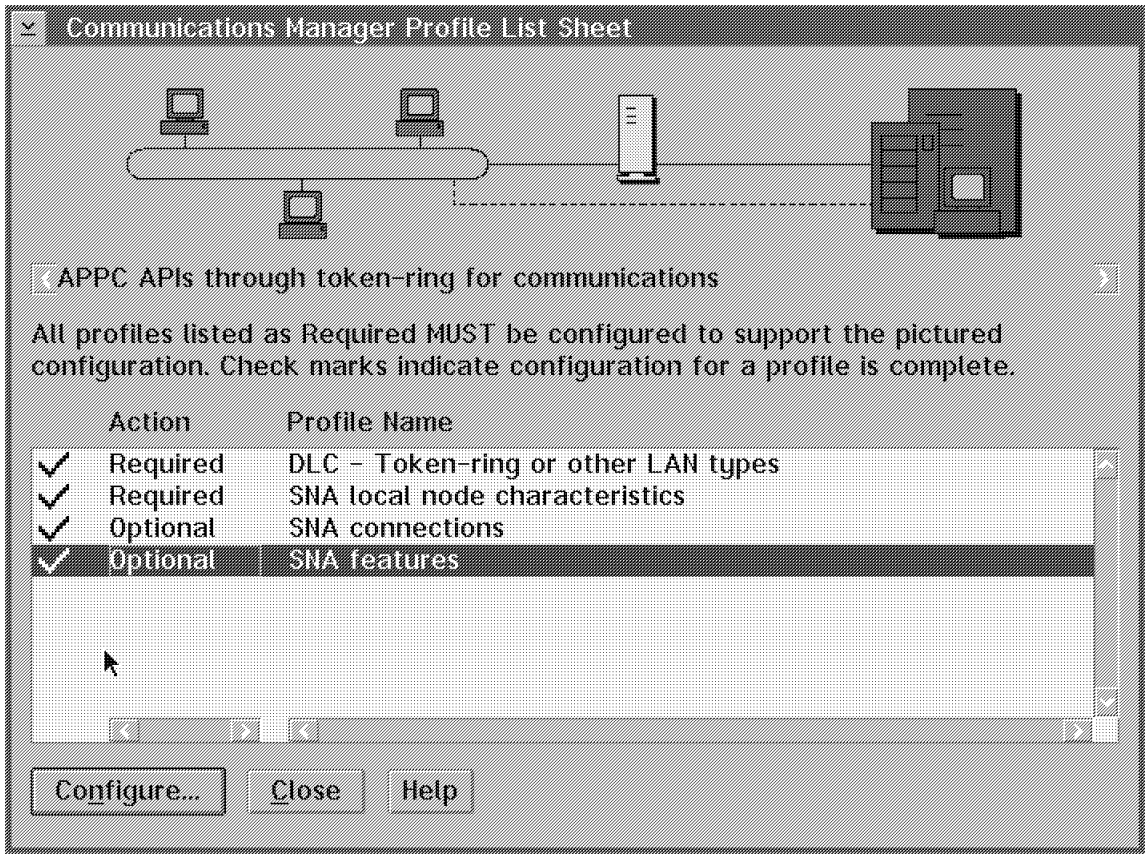


Figure 249. Communications Manager Profile List Sheet

Click on **Configure**.

18. Select Local LUs.

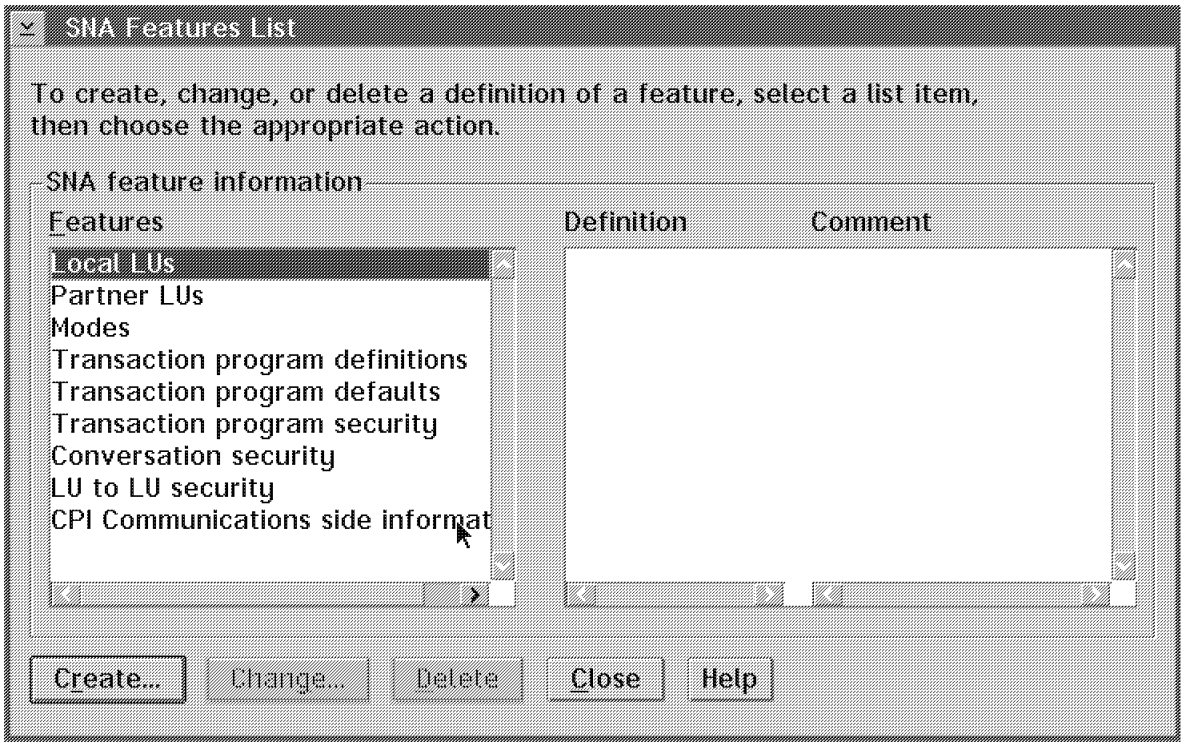


Figure 250. SNA Features List (Local LUs)

Click on **Create**.

19. This panel allows you to create a Local LU.

    We used these values on machine WTR32275:

    Parameter Our Value

    LU name W3227500

    Alias snals

    We used these values on machine WTR32240:

    Parameter Our Value

    LU name W3224000

    Alias snalm

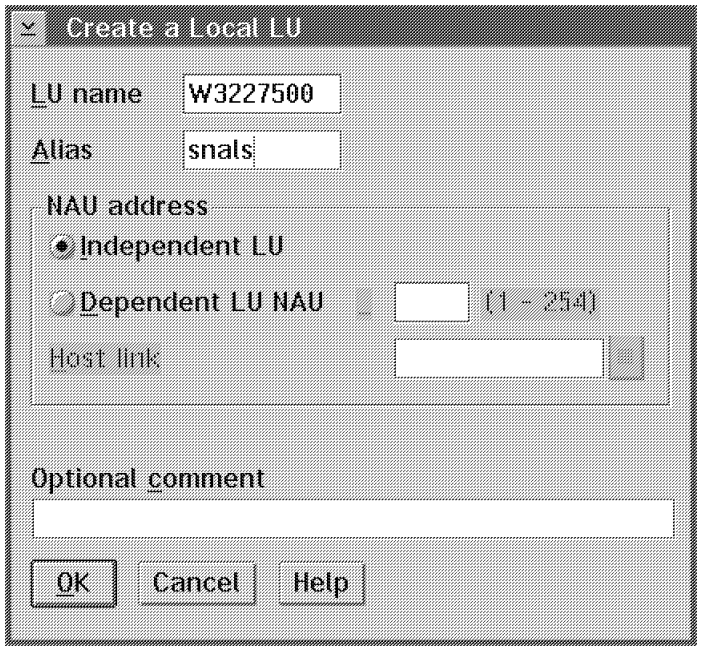


Figure 251. Create a Local LU

Click on **OK**.

20. Select **Modes**.

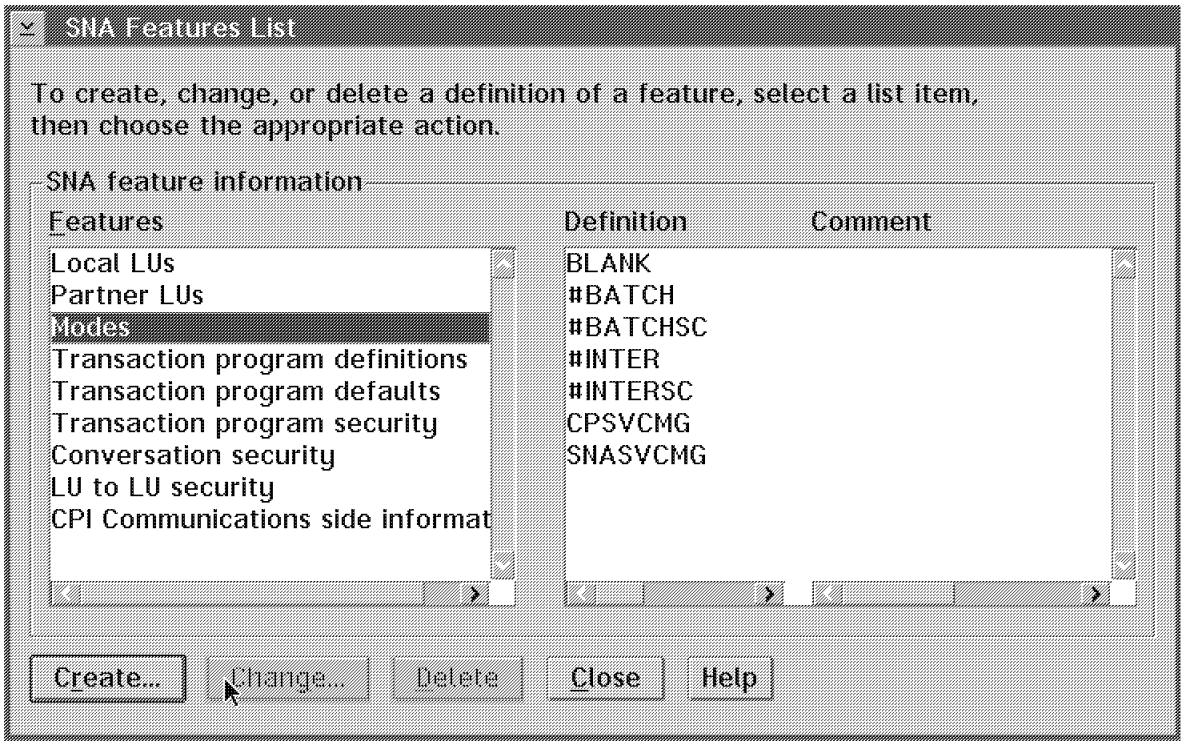


Figure 252. SNA Features List (Modes)

Click on **Create**.

21. This panel allows you to create a mode definition. We define our own mode definition, which is also used later for our connection through VTAM. We create a mode definition **DSIL6MOD** with default values.

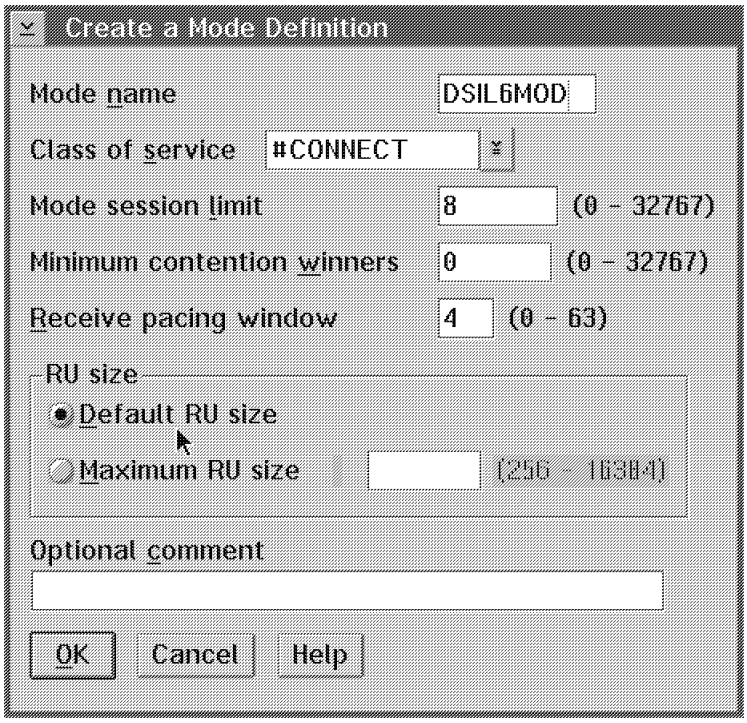


Figure 253. Create a Mode Definition

Click on **OK**.

22. Select **Transaction program definitions**.

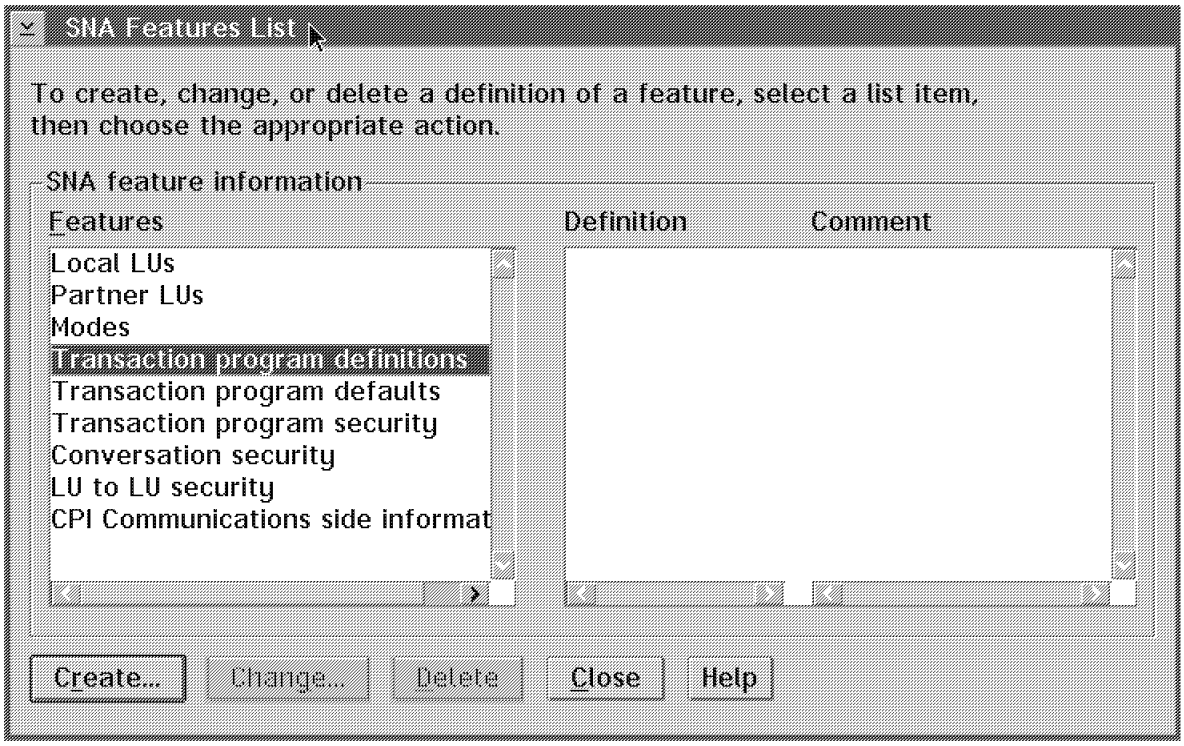


Figure 254. SNA Features List (Transaction program definitions)

Click on **Create**.

23. This panel creates a definition of the APPC transaction program (SNALIO,EXE). You should use these values:

Parameter	Value
Transaction Program (TP) Name	IPXPORT
OS/2 program and path and file name	C:TCPIPBINSNALIO.EXE
Icon path and file name	C:TCPIPBINSNALIO.ICO

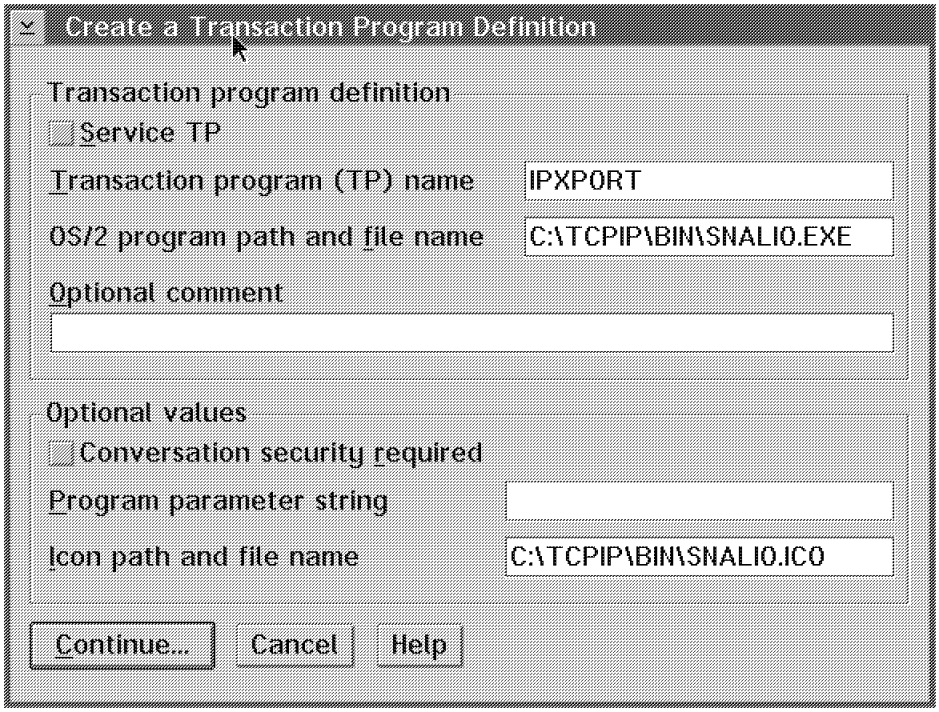


Figure 255. Create a Transaction Program Definition

Click on **Continue**.

24. You will start SNALIO from a command prompt, hence you should select **Background** and **Queued, operator started**.

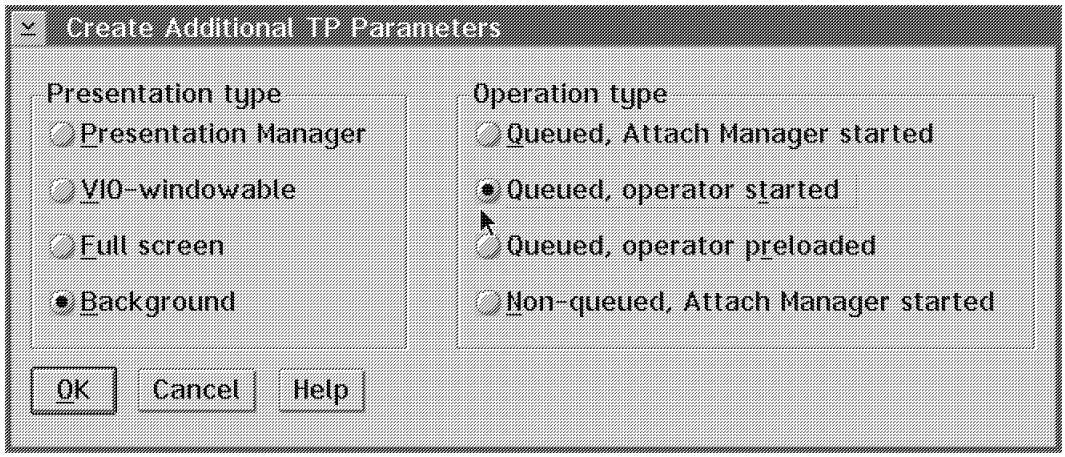


Figure 256. Create Additional TP Parameters

Click on **OK**.

25. Close each panel for Communications Manager Setup.

16.2.1.2 OS/2 Workstation to OS/2 Workstation through VTAM

In this scenario we set up an SNALINK between two OS/2 workstations via a VTAM host. We still use token-ring for all physical connections. However, if required you could connect each workstation to VTAM using any of the following media:

- LAN
- SDLC
- X.25

This type of connection through VTAM is not normally adopted in these circumstances. It is easier and more efficient to connect the workstations directly across the token-ring as described in the previous section. VTAM is normally used to form an SNALINK between two workstations under the following circumstances:

- They are physically located in different geographic locations.
- They use different mediums to connect to the network. For example one of the workstations is connected via token-ring and the other is connected via SDLC to the SNA network.

We have set up this scenario so that you can easily compare the setup required to a direct peer connection.

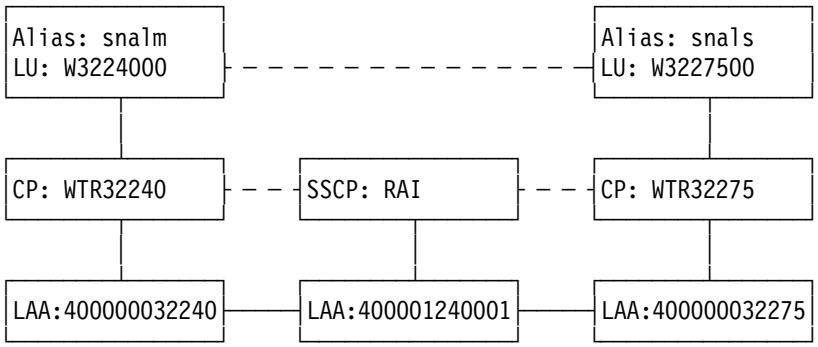


Figure 257. OS/2 to OS/2 Workstation via VTAM SNALINK Configuration

1. Follow steps 1 to 6 in 16.2.1.1, “OS/2 Workstation to OS/2 Workstation” on page 367.
2. Select SNA connections.

   Click on **Configure**.
3. Our previous configuration used a direct connection between two OS/2 workstations for the LU6.2 conversations used by SNALIO. We need to delete this connection. Select **To peer node** and then **LINK0001**.

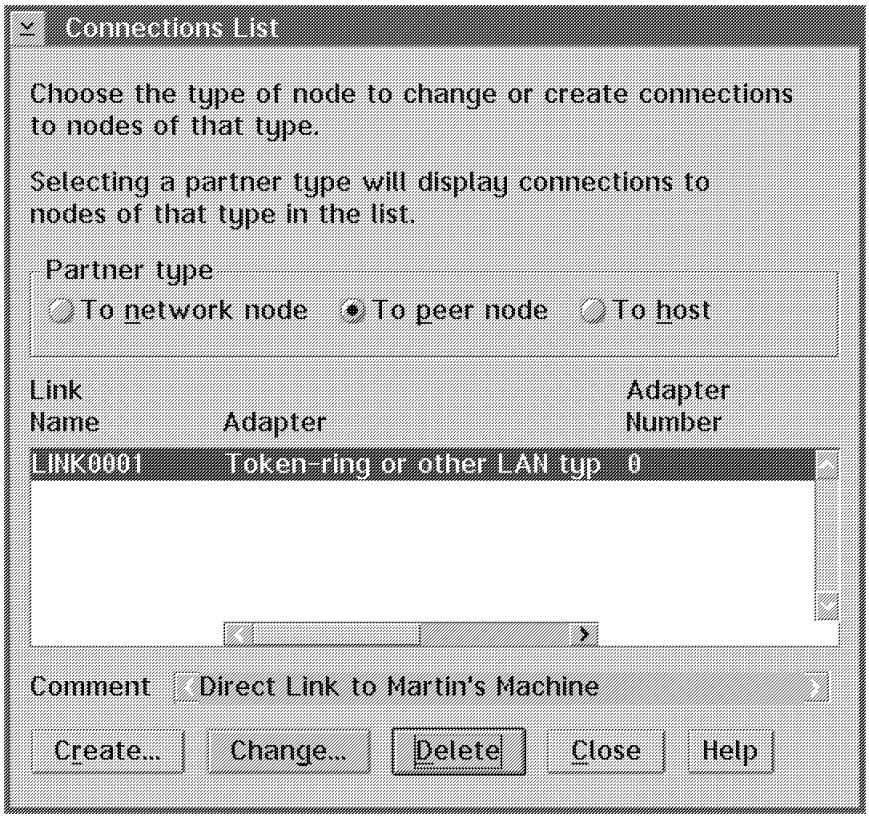


Figure 258. Connections List

Click on **Delete**.

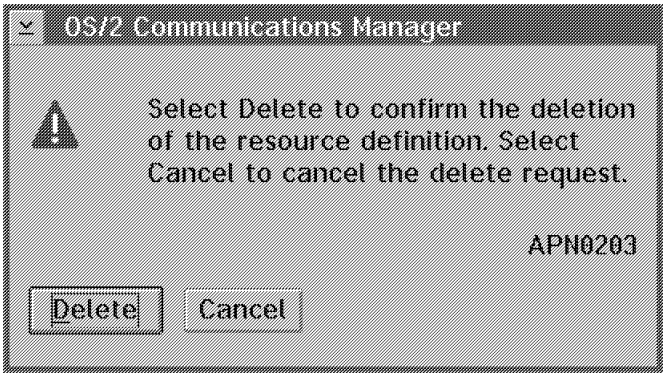


Figure 259. Delete a Connection Confirmation

Click on **Delete**.

4. We need to define a connection via the host system, so we select **To host**, and click on **Create**.
5. Select **Token-ring or other LAN types**, and click on **Continue**.
6. These are the values that we used to configure machines WTR32275 and WTR32240:

Parameter	**Our Value**
Link name	HOST0001
LAN destination address	400001240001

Partner Network ID	USIBMRA
Partner Node Name	RAI
Optional Comment	SNALINK via VTAM to Martin's machine

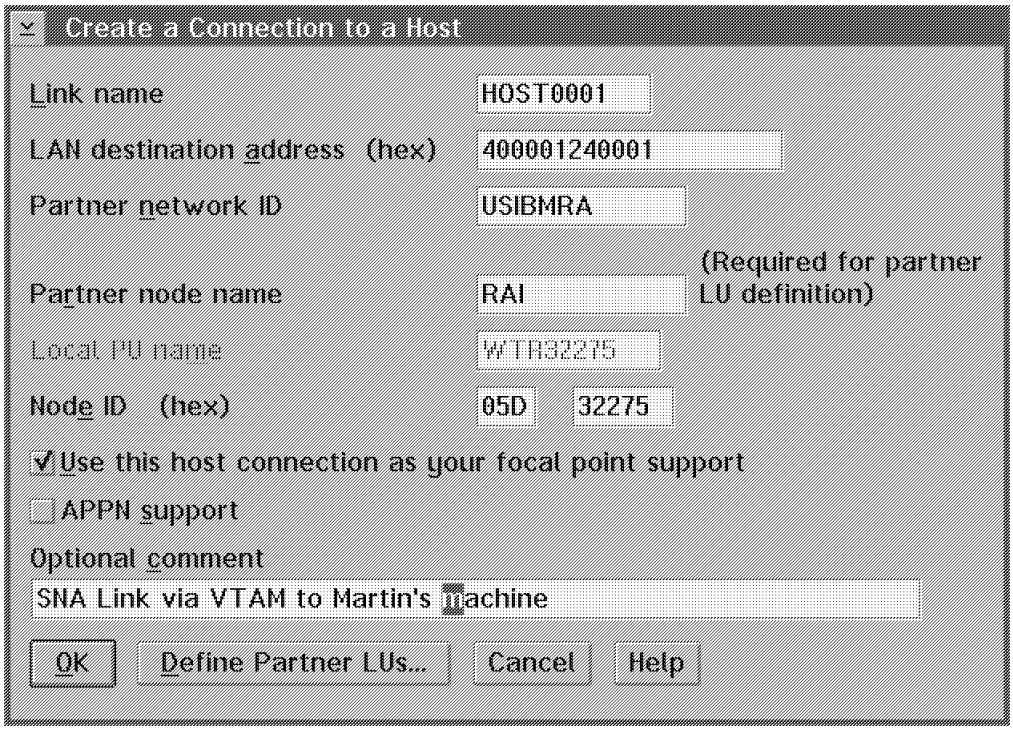


Figure 260. Create a Connection to a Host

Click on **Define Partner LUs...**

7. This panel allows you to configure a partner LU at the other end of the connection.

   You should ensure that you configure any partner LUs on this panel. Partner LUs can also be configured in the SNA features profile list, but SNALIO needs to know which connection can access the partner LU.

   These are the values that we used for machine WTR32275:

   Parameter Our Value

   LU name W3224000

   Alias snalm

   These are the values that we used for machine WTR32240:

   Parameter Our Value

   LU name W3227500

   Alias snals

   Click on **Add**. You should now see the partner LU in the list on the right-hand side of the panel:

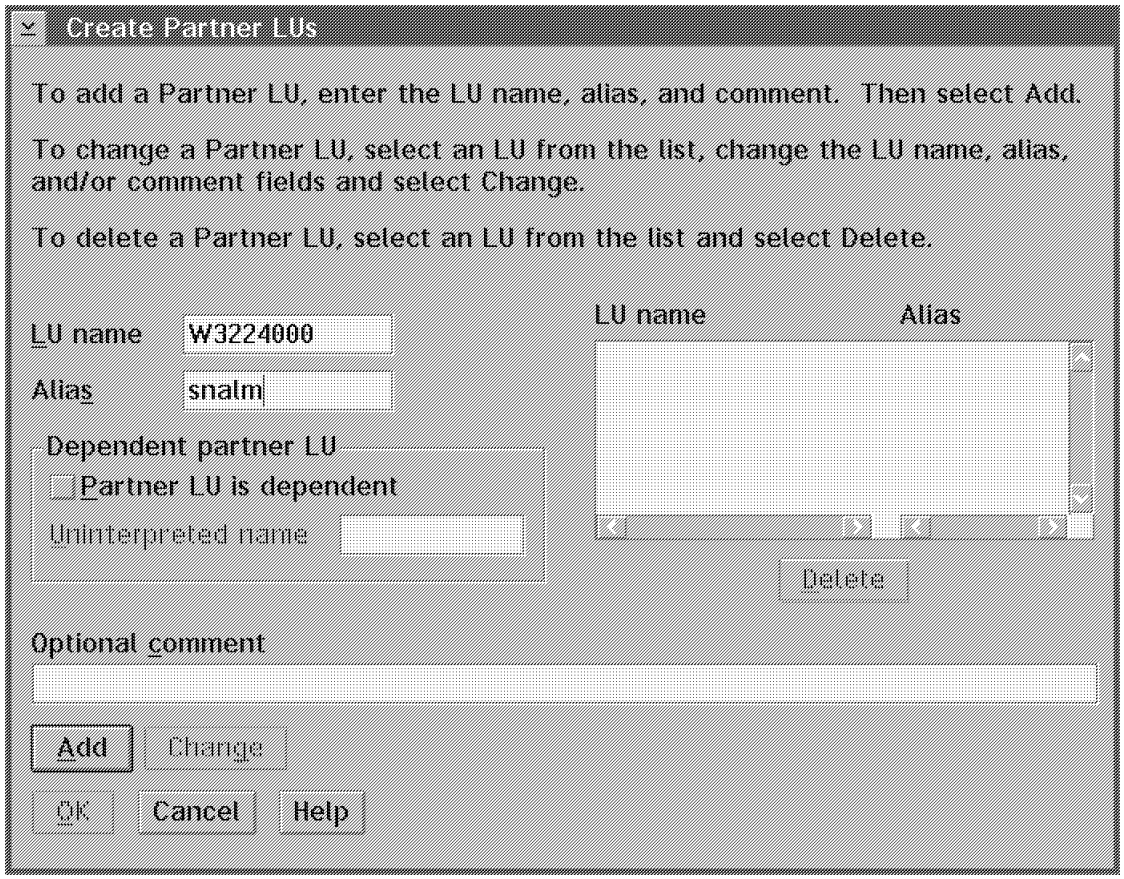


Figure 261. Create Partner LUs

Click on **OK**, and then click on **OK** again from the Create a Connection to a Host panel.

Click on **Close**.

8. Close each panel for Communications Manager Setup.

16.2.1.3 OS/2 Workstation to MVS

When you connect OS/2 to MVS using the SNALINK feature of the Extended Networking kit, you should first install the MVS PTF for MVS APAR PN44647.

If you do not apply this APAR, the following may occur:

- When an OS/2 user tries to activate the SNA LU 6.2 sessions from the Communications Manager SNA subsystems management, the sessions remain inactive.
- No TCP/IP traffic can use the SNALINK.

This occurs because the transaction program (TP) names on OS/2 and MVS are different.

This APAR has no effect on SNALINK connections from one OS/2 system to another OS/2 system. This information is also documented in this file shipped with the product:

```
TCPIPDOCREADME.XTN
```

16.2.2 Configuring TCP/IP

1. Start the TCP/IP Configuration panel from the TCP/IP desktop folder.
2. Click on the **SNALINK** tab.

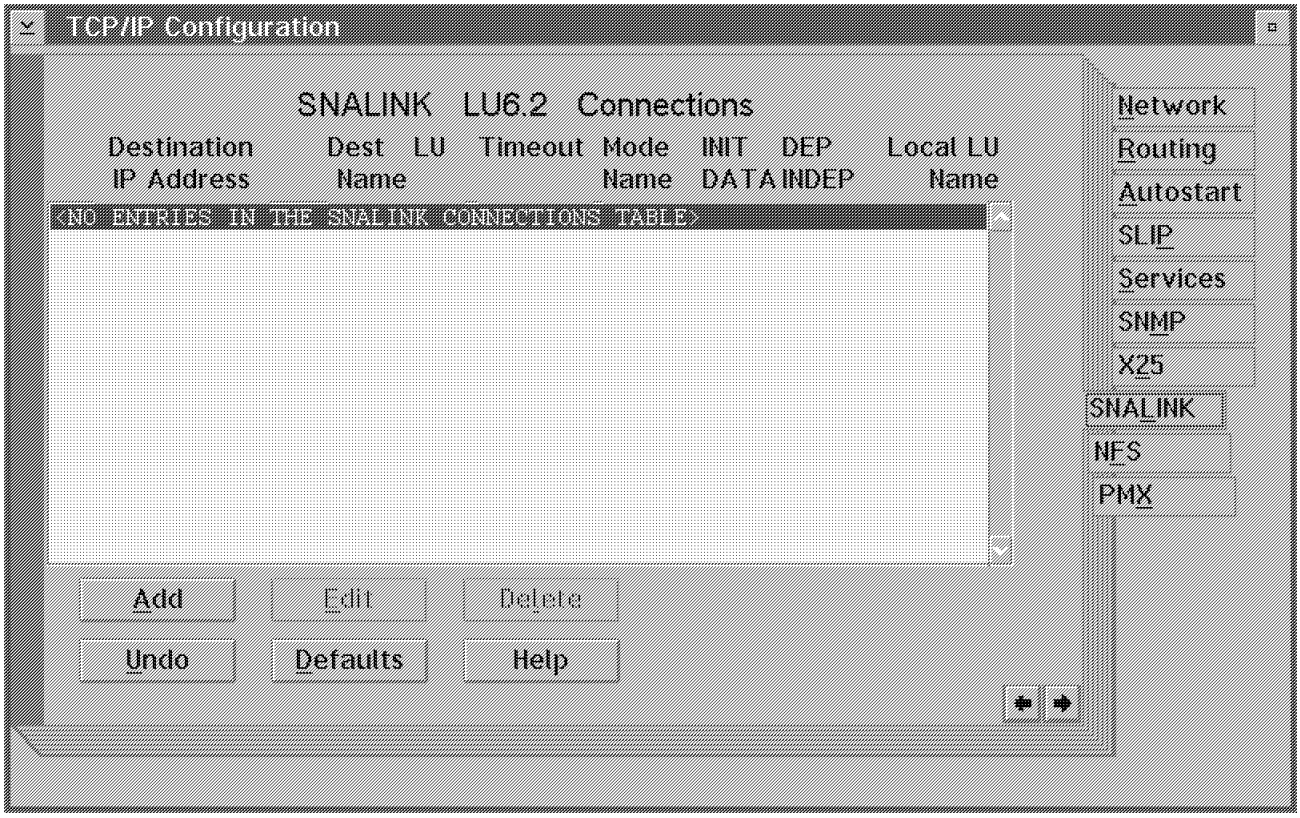


Figure 262. SNALINK LU6.2 Connections

Click on **Add**.

3. Type in:

Parameter	**Value**
IP Address	9.67.32.1
Destination LU Name	snalm
Timeout	0
Mode Name	DSIL6MOD
Define as a Dependant LU	Do not tick
Local LU Name	snals
Start Session at Initialization	Off

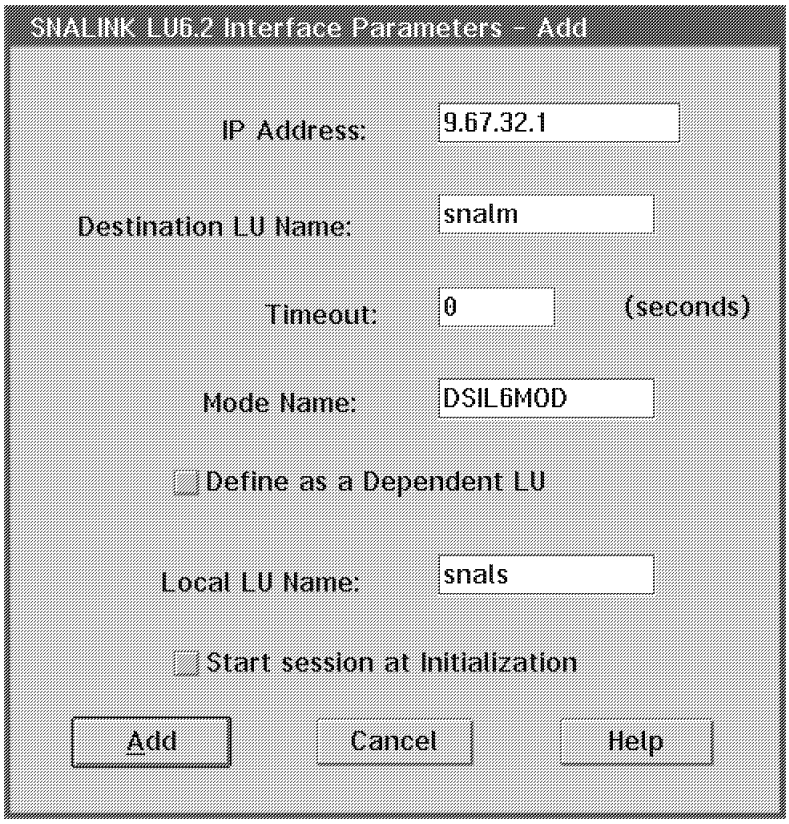


Figure 263. SNALINK LU6.2 Interface Parameters - Add (1 of 2)

Click on **Add**.

4. You should now see the link listed in the LU6.2 Connections:

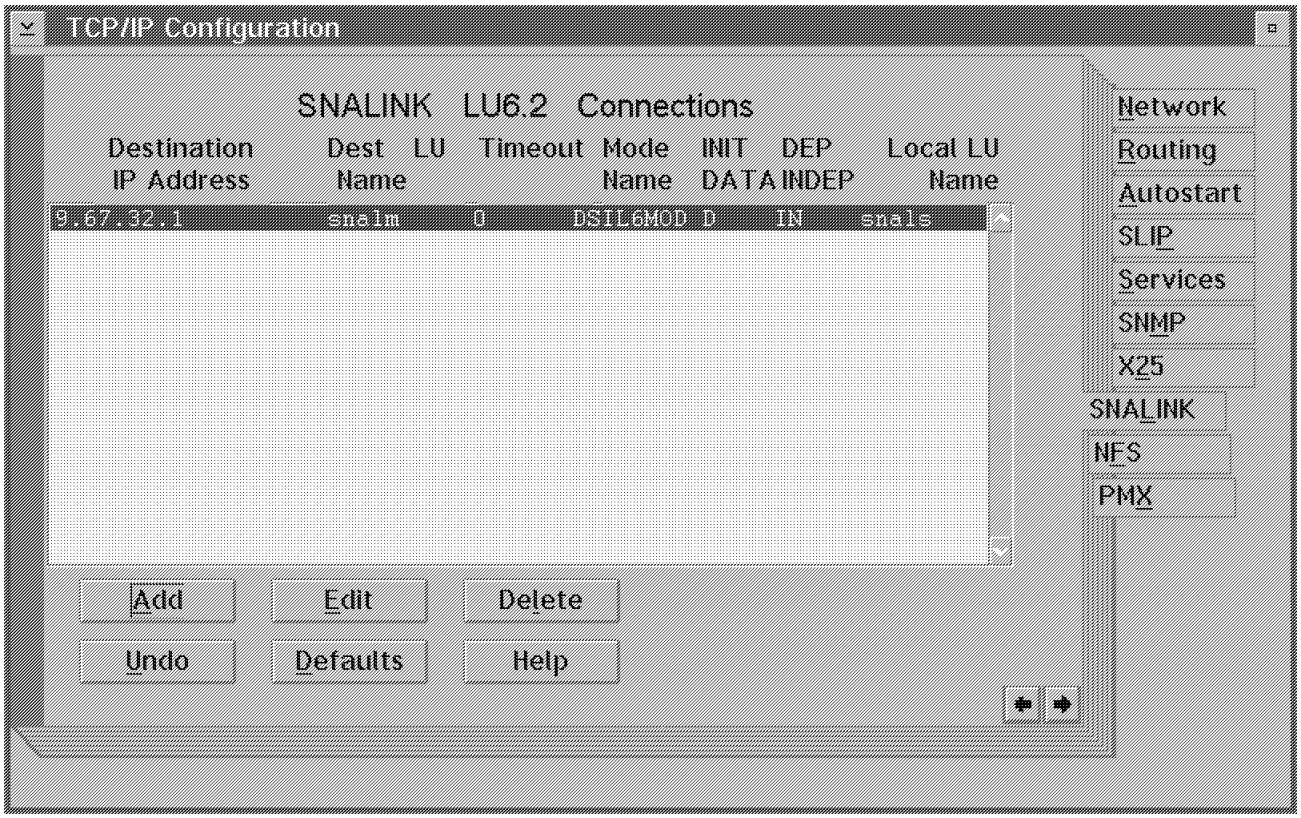


Figure 264. SNALINK LU6.2 Interface Parameters - Add (2 of 2)

5. Double-click on the top left-hand corner of the panel. This will close the Configuration panel. Click on **Save** to ensure that your changes are made.

6. You can manually view the file that we have just configured by typing this command from an OS/2 command prompt:

```
[C:tcpipetc]type snalip.cfg
* Destination  Destination  Timeout  Mode  INIT  DEP   Local
* IP Address   LU Name                   Name  DATA  INDEP LU Name
*=============================================================
9.67.32.1      snalm        0        DSIL6MOD INIT INDEP snals
```

You should ensure that this file contains the configuration information. If it does not then should create one manually, which looks similar to the configuration created in this example.

16.2.3 Starting

1. Start SNALIO.EXE from an OS/2 command prompt:

```
[C:tcpipbin]snalio
SNA2009I: Parsing the configuration file "\tcpip\etc\snalip.cfg".
SNA2010I: Successfully parsed the configuration file.

  SNALIO: Available using INET interface unit #2.  SNALIO version 2.0

SNA2000I: SNALINK interface now available.
SNA2004I: Waiting for an incoming ALLOCATE request.
SNA2006I: Session allocated successfully to remote LU "snalm   ".
```

2. In a separate OS/2 session, run the following commands:

   Command Purpose

   SNALWAIT This will ensure that SNALIO has started before you run the IFCONFIG command.

   IFCONFIG This will configure an IP address to use the SNALINK.

   NETSTAT This will show the status of the IP address just configured with IFCONFIG.

   Your OS/2 session will look similar to the following:

```
[C:tcpipbin]snalwait

[C:\tcpip\bin]ifconfig snal 9.67.32.2 netmask 255.255.255.192

[C:\tcpip\bin]netstat -a
addr        9.67.38.106 interface 0 mask ffffffc0 broadcast        9.67.38.127
addr          9.67.32.2 interface 12 mask ffffffc0 broadcast            0.0.0.0
```

16.2.4 Verifying the Connection

Once the connection has been started. You can check if it works by executing a PING:

```
ping 9.67.32.1 10 5
```

If your response from a PING is successful, then you may use the address at the other side of the SNALINK like any other TCP/IP address on your network. We established a Telnet session using this command:

```
telnet 9.67.32.2
```

This is the result of the Telnet session establishment:

```
OS/2 Version 2.1 (martin)

login:martin
password:
0

[<martin>-C:\]
```

Chapter 17. Application Programming Interfaces

TCP/IP V3.x for OS/2 Warp contains the following application programming interfaces (APIs) that can be used to write network applications in the C programming language:

- BSD Sockets
- Sun Remote Procedure Call (RPC)
- FTP
- SNMP DPI
- X Window System Client (X11)

The X11 APIs are provided by the X Window System Client kit; all other APIs are provided by the TCP/IP for OS/2 Warp V3.0 Programmer's Toolkit. See the Chapter 18, "Problem Determination" on page 405 for more information about the X Window System Client APIs. These kits provide the following files to support developing applications:

- Header and include files
- Link time libraries
- Dynamic link libraries
- Sample programs

TCP/IP V3.x for OS/2 Warp also contains the REXX FTP API package and the REXX Socket Support package. Your REXX applications can access the OS/2 TCP/IP FTP API and the OS/2 TCP/IP socket API by using these two packages.

This chapter refers to the application programming interfaces provided with TCP/IP V3.x for OS/2 Warp and the Programmer's Toolkit.

17.1 System Requirements and Installation

In order to develop applications using the APIs provided with TCP/IP V3.x for OS/2 Warp, you must have one of the following items installed on your system:

- IBM OS/2 Warp Version 3.0 or higher with the Internet Connection for OS/2 (from the BonusPak).
- IBM OS/2 Warp Connect Version 3.0 or higher with TCP/IP V3.X for OS/2 Warp. (TCP/IP V3.0 for OS/2 Warp is a part of Warp Connect.)

In addition, you must have the following installed on your system:

- Any IBM 32-bit compiler for OS/2, including the following:
  - VisualAge C++
  - C Set++

To install the TCP/IP for OS/2 Warp V3.0 Programmer's Toolkit, insert the Toolkit diskette into your diskette drive A: and enter the following command from an OS/2 command prompt:

```
A:INSTALL
```

Select the **Install** push button and follow the installation instructions.

The installation program will put the necessary files in the following directories:

Table 30. Directories for TCP/IP for OS/2 Warp V3.0 Programmer's Toolkit Files

API Files	Directory
Header and Include Files	TCPIPINCLUDE
Link Time Libraries	TCPIPLIB
Dynamic Link Libraries	TCPIPDLL
Sample Program Code	TCPIPSAMPLES

After you have installed the Toolkit, you may want to set your environment variables to find the following:

- Header files
- Link libraries
- Executable programs

You can set your environment variables interactively or you can include them in your CONFIG.SYS file. Detailed information on how to compile, link and run the sample programs is provided in the online *IBM TCP/IP for OS/2 Warp Programmer's Reference*.

17.2 Multi Thread and DLL Support

TCP/IP Version 3.0 for OS/2 Warp supports multi-threaded and dynamic link libraries (DLL) for all of the APIs. This allows writing TCP/IP applications that execute multiple threads. Each thread runs as independent code, but shares the same resources. A Presentation Manager application that issues communication requests should do this in a separate thread, because a short turnaround time for the request is not guaranteed. If the network is very slow or congested, it takes a long time for the request to complete. During this time, the whole Presentation Manager is blocked while it is waiting for the network request to complete.

A real server that uses the socket APIs is another example of the multitasking need. It should be able to communicate with many clients at the same time. This requires multiple threads in an OS/2 environment.

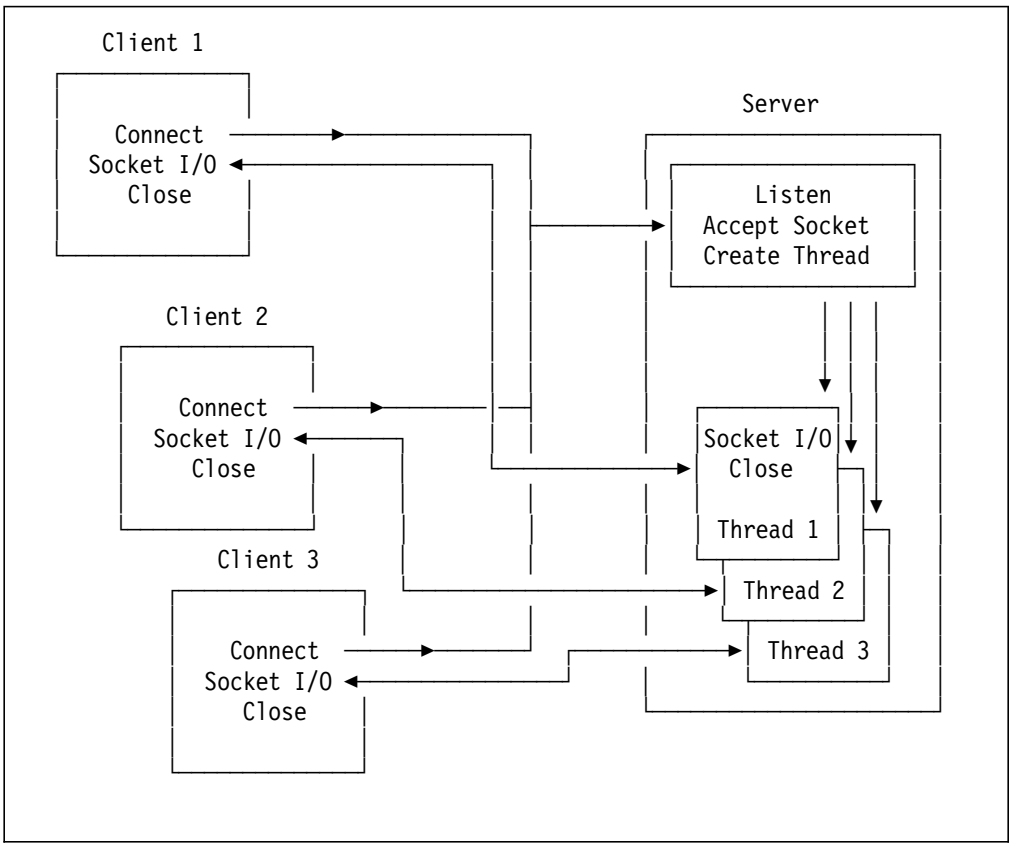


Figure 265. Multitasking Server

In the figure above, the server consists of a control thread that waits for incoming requests from any client and creates a new thread that actually handles the I/O and the processing. This allows communication with several clients simultaneously.

The DLL libraries offer additional advantages. The functions in the DLL libraries are used only at run time. They are linked dynamically when a program runs, instead of at link time. Applications that use DLLs need less run-time memory and also need less user disk space. If a DLL is needed by multiple applications, it is loaded only once and shared by all applications simultaneously.

For more information about DLLs, please refer to the *OS/2 Warp Technical Library, Control Programming Guide, G25H-7101*.

17.3 Socket API

IBM′s Network Services for TCP/IP Version 3.0 for OS/2 Warp provides a solution to interconnect applications across networks. Network services provides a 32-bit sockets API for the Internet (TCP/IP), local interprocess communication (local IPC), and NetBIOS communication domains. Network services sockets is based on the Berkeley Software Distribution (BSD) Version 4.3 sockets implementation.

The sockets API allows you to write distributed or client/server applications in supported communication domains to allow applications to communicate across networks. In addition, the interface allows interprocess communication within the same workstation. Applications can have full network access by just using the sockets API. You can run an existing sockets application in another

communications domain by modifying the communications domain selection and the networking addressing parameters used by the application. You must then recompile and relink the application.

A socket is a communication endpoint (uniquely identified by the host IP address and a port number) that a TCP/IP application uses to communicate with another TCP/IP application (on the same host or on a different host). You can, for example, make use of the sockets interface when you write a client application that may communicate with a server application running on the same or another workstation. Sockets are duplex, which means that data can be transmitted and received simultaneously.

The network services support four socket types:

Datagram Datagram sockets are connectionless. Data is sent in both directions without any guarantee of delivery using UDP. Data may be lost or duplicated and datagrams may arrive out of order. NFS is built on datagram sockets.

Raw Raw sockets interface to the ICMP and IP protocols. PING uses raw sockets.

Sequenced packet Sequenced packet sockets define a reliable connection-oriented service. Data is sent without error or duplication and is received in the same order as it was sent. Flow control is built in order to avoid data overruns. Every sequenced packet packet is sent and received as a complete record.

Stream Stream sockets transmit data reliably in both directions between two applications by using the TCP protocol. FTP is implemented with stream sockets.

You should consider the following factors in choosing a socket type for new applications:

- **Reliability**: Stream and sequenced packet sockets provide the most reliable connection. Connectionless datagram and raw sockets are unreliable because packets can be discarded, duplicated, or received out of order. This may be acceptable if the application does not require reliability, or if the application implements the reliability on top of the sockets API. The trade-off is the increased performance available compared to stream and sequenced packet sockets.
- **Performance**: The overhead associated with reliability, flow control, packet reassembly, and connection maintenance degrades the performance of stream and sequenced packet sockets so that these types do not perform as well as datagram sockets acting in a connectionless mode.
- **Amount of data to be transferred**: Datagram and sequenced packet sockets impose a limit on the amount of data transferred. As the amount of data in a single transaction increases, it is preferable to use stream or sequenced packet sockets.

Each communication domain supports certain socket types. The following table shows the communication domains supported:

Table 31. Communication Domains Supported			
Communication Domain	**Protocol Family**	**Supported Protocols**	**Supported Socket Types**
Internet	PF_INET	ICMP, IP, TCP, UDP	Datagram, Raw, Stream
Local IPC	PF_OS2 or PF_UNIX	Local IPC	Datagram, Stream
NetBIOS	PF_NETBIOS or PF_NB	NetBIOS	Datagram, Sequenced Packet

The IBM OS/2 socket implementation differs from the Berkeley socket implementation as follows:

- Sockets are not OS/2 files or devices. Socket numbers have no relationship to OS/2 file handles. Therefore, read(), write(), and close() do not work for sockets. Using read(), write(), or close() gives incorrect results. Use the recv(), send(), and soclose() functions instead.
- Socket calls require that you call the sock_init() routine before you call them. Therefore, always call sock_init() at the beginning of programs using the socket interface.
- Error codes set by the OS/2 TCP/IP sockets implementation are not made available via the global errno variable. Instead, error codes are accessed by using the sock_errno() API described in sock_error(). Use psock_errno(), instead of perror(), to write a short error message to the standard error device describing the last error encountered during a call to a socket library function. It is not possible for an application to assign new values to error codes.
- The select() call has a different interface. Unlike the Berkeley select() call, you cannot use the OS/2 select() call to wait for activity on devices other than sockets.
- The ioctl() implementation might differ from the current Berkeley ioctl() implementation.

For more information about porting a socket application, please refer to the online *TCP/IP for OS/2 Warp Programmer's Reference*.

17.4 Remote Procedure Call APIs (RPC)

The remote procedure call (RPC) API is a higher-level and more powerful interface than the socket API. It is built on top of sockets and is used to develop distributed or cooperative processing client/server applications. RPC extends the procedure mechanism in a program to allow procedures to be distributed in the network. The idea is that a remote procedure call should look the same to the programmer as the usual local procedure call. The programmer should not be concerned with network details like sockets and addressing and therefore can be more productive.

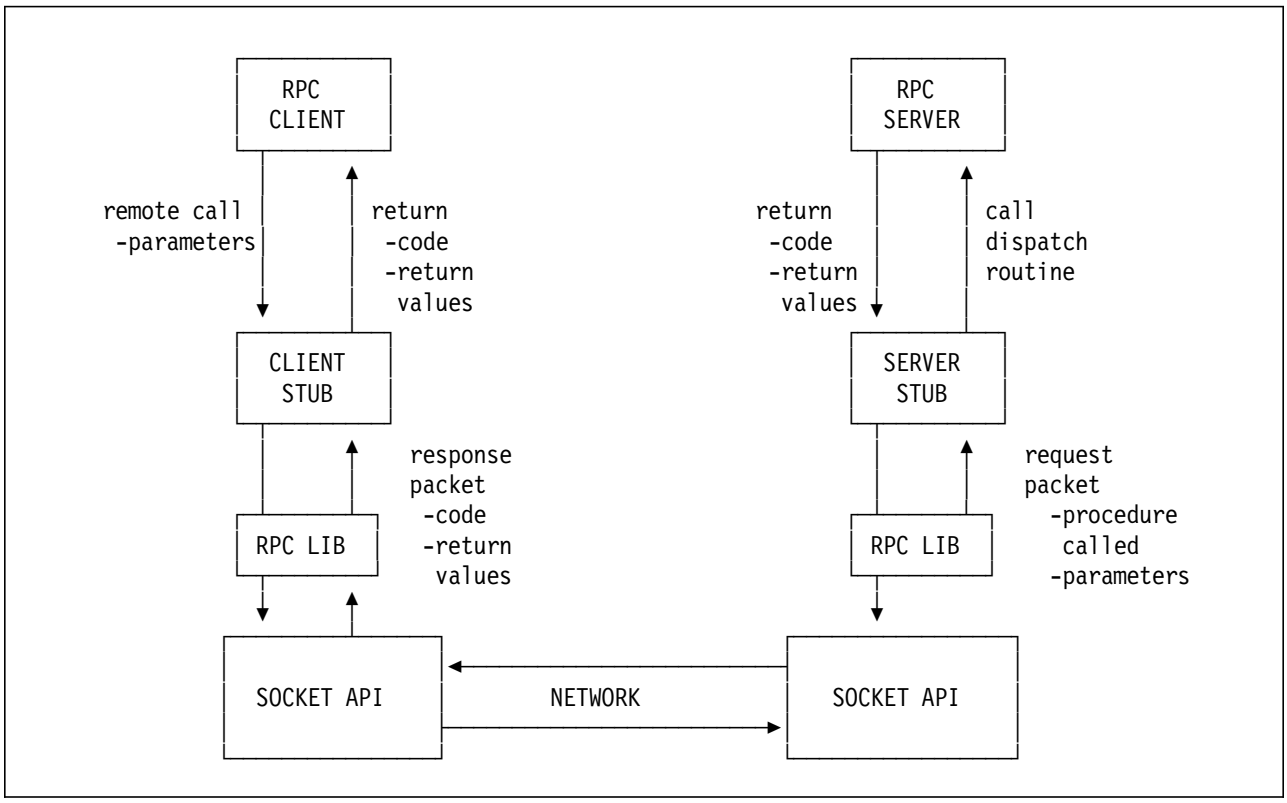


Figure 266. The RPC Mechanism

When The RPC client program issues a remote procedure call, it actually calls a local stub procedure (see Figure 266).

The client stub packages the parameters in a standard format understood by the server stub, builds request packets and transfers them over the network. The server stub receives the packets, calls a dispatch routine that services the call, builds a response packet and transfers it back to the client stub. The client stub converts any return values to the RPC client′s native format and returns them to the RPC client.

TCP/IP V3.x for OS/2 provides an RPC interface based on Sun Microsystem′s RPC Library (Sun RPC). Currently the Sun RPC is also implemented in IBM′s DOS, AIX, VM and MVS systems.

Data Representation: Sun RPC defines an intermediate data representation protocol called external data representation (XDR) to account for architectural differences in data representation between a server and a client. XDR is always used, even if the server and client use the same computer architecture.

Interface Language: The Sun RPC defines the RPC language (RPCL) to describe the remote procedure characteristics. RPCL allows remote procedures to have only one parameter. When more than one parameter is needed, they must be packaged in a structure and XDR procedures must be written to marshal and unmarshal the parameters. The RPCL definition is compiled with the RPCGEN tool, which generates C source code for the client and server stubs.

Transport Protocol: Sun RPC uses either TCP or UDP. If UDP is used, there is a current limit of 2 KB that can be sent at one time and there is no guarantee that the remote procedure will be executed only once.

Binding and Port Location: The location of an RPC server must be known by the client program or supplied by the user to the program. When an RPC server starts, it registers itself (by supplying its port number) with a local program called Portmapper. A remote client program can obtain the server's port number by contacting Portmapper on the server's host via a well-known port.

For more information about the RPC and XDR protocols, see Sun Microsystems publication, *Networking on the Sun Workstation: Remote Procedure Call Programming Guide*, RFC 1057 and RFC 1014.

The following components make up the Sun RPC Support provided by TCP/IP V3.x for OS/2:

- RPC Library
- RPCGEN
- PORTMAPPER
- RPCINFO

The IBM OS/2 RPC implementation differs from the Sun Microsystems RPC implementation as follows:

- The global variables svc_socks() and noregistered are used in place of the svc_fds global variable.
- Functions that rely on file descriptor structures are not supported.
- The svc_getreq() call supports the socks and noavail global variables. In the Sun Microsystems implementation, the svc_getreq() call supports the rdfds global variable.
- TYPES.H for RPC has been renamed to RPCTYPES.H.

To use the RPCs described in this section, you must have the following header files in your TCPIPINCLUDE directory:

PC Header File	What It Contains
RPCAUTH.H	Authentication interface
RPCAUTH_UNI.H	Protocol for UNIX-style authentication parameters for RPC
RPCCLNT.H	Client-side remote procedure call interface
RPCPMAP_CLN.H	Supplies C routines to get to PORTMAP services
RPCPMAP_PRO.H	Protocol for the local binder service, or pmap
RPCRPC.H	Includes the RPC header files necessary to do remote procedure calling
RPCRPC_MSG.H	Message definitions
RPCRPCNETDB.H	Data definitions for network utility calls
RPCRPCTYPES.H	RPC additions to <TYPES.H>
RPCSVC.H	Server-side remote procedure call interface
RPCSVC_AUTH.H	Service side of RPC authentication
RPCXDR.H	External data representation serialization routines

The RPC routines are in the RPC32DLL.LIB file in the LIB directory. You must also have the SO32DLL.LIB and TCP32DLL.LIB files in your LIB directory. Put the following statement at the beginning of any file using RPC code:

```
#include <rpcrpc.h>
```

You must define the OS2 variable by doing one of the following:

- Place #define OS2 at the top of each file that includes TCP/IP header files.
- Use the /DOS2 option when compiling the source for your application.

17.5 File Transfer Protocol API

The file transfer protocol (FTP) 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. It allows up to 256 simultaneous connections and enables third-party proxy transfers between pairs of FTP servers. Consecutive third-party transfers are allowed between any sequence of pairs of FTP servers. An example of such an application is FTPPM.

The FTP API tracks the servers to which an application is currently connected. When a new request for FTP service is requested, the API checks whether a connection to the server exists and establishes one if it does not exist. If the server has dropped the connection since last use, the API re-establishes it.

Note: The FTP API is not re-entrant. If you are using a multithreaded program, you must serialize the access to the APIs. For example, without serialization, the program may fail if it has two threads running concurrently and each thread has its own connection to a server.

The following are the API calls supported through the FTP:

Table 32. FTP API Calls

ftpappend()	ftpcd()	ftpdelete()	ftpdir()
ftpget()	ftplogoff()	ftpls()	ftpmkd()
ftpping()	ftpproxy()	ftpput()	ftpputunique()
ftppwd()	ftpquote()	ftprename()	ftprmd()
ftpsite()	ftpsys()	ftptrcoff()	ftptrcon()
ftpver()	ping()		

To use the FTP API described in this section, you must have the <FTPAPI.H> header file in your TCPIPINCLUDE directory. The FTP API routines are in the FTPAPI.LIB file in the LIB directory. You must also have SO32DLL.LIB and TCP32DLL.LIB files in your LIB directory. Put the following statement at the top of any file using FTP API code:

```
#include <ftpapi.h>
```

Define the OS2 variable to the compiler by doing one of the following:

- Place #define OS2 at the top of each file that includes TCP/IP header files.
- Use the /DOS2 option when compiling the source for your application.

17.6 SNMP Agent Distributed Protocol Interface (DPI)

The SNMP DPI API provides the necessary files for writing your own SNMP subagents. An SNMP subagent can dynamically support its own network management variables and generate its own alerts to a SNMP monitor. It can add, delete, or replace MIB values, and generate SNMP TRAPs. Supported are DPI V2.0 and RFC 1592 interfaces.

For the SNMP DPI V2.0 API, some functions are implemented as macros, because the older DPI V1.x had the same function names with different parameters. The new implementation has new function names, which are not always the most intuitive. By defining the macros with the more natural names for the functions, the non-intuitive names are hidden. This was done because the macros have the same names as the functions were named in DPI V1. It is thus

possible to provide either the DPI V1.x or the the DPI V2.x API by properly defining the macros.

You can keep your existing DPI V1.1 subagent and communicate with a DPI-capable agent that supports DPI V1.1 in addition to DPI V2.0. For example, the OS2 agent for TCP/IP provides support for multiple versions of DPI, namely DPI V1.0, DPI V1.1 and DPI V2.0. Normally you would compile your DPI V1.1 subagent with the DPI V1.1 <dpisnmp_dpi.h> include file and link-edit it with the provided DPI V1.1 level DPI32DLL.LIB. At run time, you then need access to the DPI32DLL.DLL. You can continue to do this until you are ready to migrate to DPI V2.0. For more information about migrating your SNMP DPI subagent to DPI V2.0, please refer to the online *TCP/IP for OS/2 Warp Programmer's Reference*.

To use the DPI library routines provided with TCP/IP for OS/2, you must have the <snmp_dpi.h> header file in your TCPIPINCLUDE directory. The DPI20DLL.LIB file in the LIB directory contains the DPI library routines. You must also have the SO32DLL.LIB and TCP32DLL.LIB files in your LIB directory.

You must define the OS2 variable to the compiler by doing one of the following:

- Place #define OS2 at the top of each file that includes TCP/IP header files.
- Use the /DOS2 option when compiling the source for your application.

17.7 REXX FTP API and REXX Socket Support

The REXX FTP API package provides access to the OS/2 TCP/IP FTP APIs from your REXX program. It is contained in the file rxftp.dll. This dynamic link library needs to be placed in a directory listed in your LIBPATH statement in your CONFIG.SYS file.

We wrote a sample program to show the usage of the REXX FTP API. It connects to the FTP server named rs6ktw3 then gets the specified files from server. This sample program is listed as follows:

```
/* RXFTPSMP.CMD - Sample program of REXX FTP APIs */

 rc = RxFuncAdd("FtpLoadFuncs","rxFtp","FtpLoadFuncs")
 say "RxFuncAdd:(FtpLoadFuncs), return:" rc
 FtpLoadFuncs()
 say "FtpLoadFuncs"

 hostname = "rs6ktw3"
 say "Login to host:" hostname
 say "  Enter your user id:"
 parse pull userid
 say "  Enter" userid"'s password"
 parse pull password
 rc = FtpSetUser(hostname,userid,password)
 say "FtpSetUser, return:" rc

 rc = FtpSetBinary("Ascii")
 say "FtpSetBinary:(ASCII), return:" rc

 rc = FtpLs("*.c","files.")
 say "FtpLs:(*.c), return:" rc
 if files.0 > 0
 then do
```

```
  say files.0 "files"
  do i = 1 to files.0
    say "Get remote file:" files.i
    rc = FtpGet(files.i, files.i)
    say "  FtpGet, return:" rc
  end
end

rc = FtpLogoff()
say "FtpLogoff, return:" rc

FtpDropFuncs()
say "FtpDropFuncs"
/* End of RXFTPSMP.CMD */
```

The REXX Socket Support package provides access to the OS/2 TCP/IP socket APIs from your REXX program. It is contained in the file rxsocket.dll. This dynamic link library needs to be placed in a directory listed in your LIBPATH statement in your CONFIG.SYS file.

We wrote a pair of client/server sample programs to show the usage of the REXX socket support. The server listens for the client on a specified port. After the connection is established the client send a short message to server, then the server re-sends this received message back to client. The server program is listed as follows:

```
/* RXTCPS.CMD - Sample program of REXX Socket Support */

rc = RxFuncAdd("SockLoadFuncs","rxSock","SockLoadFuncs")
say "RxFuncAdd:(SockLoadFuncs), return:" rc
SockLoadFuncs()
say "SockLoadFuncs"

say "Please enter the port number:"
parse pull server.port

rc = SockInit()
say "SockInit, return:" rc

s = SockSocket("AF_INET", "SOCK_STREAM", "0")
say "SockSocket, socket is" s
if s < 0 then exit

server.family = "AF_INET";
server.addr = "INADDR_ANY";

rc = SockBind(s, "server.")
say "SockBind, return:" rc
if rc < 0 then exit

rc = SockListen(s, 1)
say "SockLitsen, return:" rc
if rc <> 0 then exit

ns = SockAccept(s, "client.")
say "SockAccept, socket is" ns
if ns = -1 then exit

rc = SockRecv(ns, buf, 12)
```

```
 say "SockRecv, received" rc "bytes"
 if rc = -1 then exit
 say "  the received message is ("buf")"

 rc = SockSend(ns, buf)
 say "SockSend, sent" rc "bytes"
 if rc < 0 then exit

 rc = SockSoClose(ns);
 say "SockSoClose:(" ns "), return:" rc
 rc = SockSoClose(s);
 say "SockSoClose:(" s "), return:" rc

 SockDropFuncs()
 say "SockDropFuncs"
/* End of RXTCPS.CMD */
```

The following is the list of the client program:

```
/* RXTCPC.CMD - Sample program of Rexx Socket Support */

 rc = RxFuncAdd("SockLoadFuncs","rxSock","SockLoadFuncs")
 say "RxFuncAdd:(SockLoadFuncs), return:" rc
 SockLoadFuncs()
 say "SockLoadFuncs"

 rc = SockInit()
 say "SockInit, return:" rc

 say "Please enter the server's name:"
 parse pull host.name
 say "Please enter the port number:"
 parse pull server.port
 rc = SockGetHostByName(host.name, "host.")
 say "SockGetHostByName:(" host.name "), return:" rc
 if rc <> 1 then exit
 say "  server's address is" host.addr
 server.family = "AF_INET";
 server.addr = host.addr;

 s = SockSocket("AF_INET", "SOCK_STREAM", "0")
 say "SockSocket, socket is" s
 if s < 0 then exit

 rc = SockConnect(s, "server.")
 say "SockConnect, return:" rc
 if rc < 0 then exit

 rc = SockSend(s, "the message")
 say "SockSend, sent" rc "bytes"
 if rc < 0 then exit

 rc = SockRecv(s, buf, 12)
 say "SockRecv, received" rc "bytes"
 if rc = -1 then exit
 say "  the received message is ("buf")"

 rc = SockSoClose(s);
 say "SockSoClose:(" s "), return:" rc
```

```
 SockDropFuncs()
 say "SockDropFuncs"
/* End of RXTCPC.CMD */
```

Chapter 18. Problem Determination

This section lists the utilities provided with TCP/IP for OS/2 that help you find out what is going wrong in case of communication problems. Please note that this is only a brief overview of what you can do, since thorough problem analysis and tracing would go far beyond the scope of this document. For a more detailed discussion of problem determination in a TCP/IP environment, please refer to the *TCP/IP for MVS, VM, OS/2 and DOS Troubleshooting Guide, GG24-3852*.

18.1 Overview

The following list describes the utilities that you need most to find the reason for a communications problem:

Program — **Description**

PING — Sends an ICMP message to a destination IP address or IP network and reports the response time if that destination can be reached. By default, PING sends a 56-byte packet continuously until you terminate it with Ctrl-C. You can also specify the number of packages to be sent in order to prevent too much network traffic or keeping routers too busy. Use this command to determine whether you can reach a destination on an IP network.

NETSTAT — NETSTAT helps you to obtain information about your own IP interfaces. In most cases you will need the following to:

- List IP addresses in use at your workstation type

  ```
  netstat -a
  ```

- List TCP/IP routing tables in use at your workstation type

  ```
  netstat -r
  ```

- List connections of TCP/IP clients and servers at your workstation type

  ```
  netstat -s
  ```

- Show characteristics of IP interfaces at your workstations type

  ```
  netstat -n
  ```

- List the ARP table in use at your workstation type

  ```
  netstat -p
  ```

- Show the status of TCP at your workstation type

  ```
  netstat -t
  ```

- Show the status of UDP at your workstation type

  ```
  netstat -u
  ```

- Show the status of IP at your workstation type

  ```
  netstat -i
  ```

- Show the status of ICMP at your workstation type

  ```
  netstat -c
  ```

- Show the memory buffer usage type

  ```
  netstat -m
  ```

In case a connection cannot be established, use NETSTAT to find the cause of the problem.

IFCONFIG Initializes an IP interface at your workstation and allows you to query its status. This is helpful if you have to find out if an IP interface is active (UP).

ARP Displays the IP-to-hardware address mapping table at your workstation. You can manually add a hardware address to your ARP table, for instance if you lose connection to a router.

Note: Be careful when updating the ARP table. Entering a wrong IP-to-hardware address pair will result in communication errors.

18.2 Problem Case

The steps provided in this section should be used as a guide. We recommend that you follow these steps when you are unsuccessful in your attempts to use one of the services provided in TCP/IP for OS/2.

1. Determine if your interface to the network is up by running this command:

   ```
   ifconfig interface
   ```

   Where interface is the name of the interface that you are trying to use to access the network. It must be one of the following:

Interface	Description
lo	Local loopback
lan0 to lan7	LAN interfaces (token-ring, Ethernet, PC Network, 3174 Peer LAN over Coax, FDDI, Frame Relay, 3172 Offload)
sl0	SLIP interface
x25	X.25 interface
snal	SNALINK interface
sna0	Sockets over SNA interface provided by the Anynet/2 program.

   This is the result of this command when used to check the status of the interface of a token-ring adapter on a machine that was working with no problems.

```
[C:tcpipbin]ifconfig lan0
lan0: flags=3063<UP,BROADCAST,NOTRAILERS,RUNNING,BRIDGE,SNAP>
        inet 9.24.104.77 netmask ffffff00x broadcast 9.24.104.255
```

   You will notice that the link is up, and it shows all information associated with that interface.

   This is the result of this command when used it to check the status of the interface LAN1, which is not physically installed on the machine.

```
[C:tcpipbin]ifconfig lan1
lan1: flags=3032<BROADCAST,POINTOPOINT,NOTRAILERS,BRIDGE,SNAP>
ioctl (SIOCGIFADDR): Can't assign requested address
        inet 0.0.0.0 ioctl (SIOCGIFNETMASK): invalid argument
ioctl (SIOCGIFDSTADDR): invalid argument
--> 0.0.0.0 netmask 0x ioctl (SIOCGIFADDR): invalid argument
```

You will notice that the command fails and the error message says that the interface specified is not valid and therefore not operational.

If your interface is not functioning correctly then you should make sure that you have run C:MPTNBINSETUP.CMD and configured your interface information correctly in the TCP/IP Configuration Notebook.

2. Determine if your physical connection to the network is functioning by using this command:

   ```
   PING IP-address
   ```

   Where IP-address is the IP address of an active host.

   Notes:

   a. If you cannot PING your own address that does not necessarily mean that your network connection is down since your PING to yourself might not go out to the network. If it is unsuccessful, the reason might be that the TCP/IP protocol stack has not been initialized at your workstation.

   b. You cannot PING yourself on a SLIP connection (loop-back not supported).

   c. You can only PING yourself on an X.25 or SNALINK connection if both sides of the link are up.

   This is the result of a successful attempt to PING a destination address:

   ```
   [C:tcpipbin]ping 9.24.104.1 10 5
   PING 9.24.104.1: 10 data bytes
   18 bytes from 9.24.104.1: icmp_seq=0. time=0. ms
   18 bytes from 9.24.104.1: icmp_seq=1. time=0. ms
   18 bytes from 9.24.104.1: icmp_seq=2. time=0. ms
   18 bytes from 9.24.104.1: icmp_seq=3. time=0. ms
   18 bytes from 9.24.104.1: icmp_seq=4. time=0. ms

   ----9.24.104.1 PING Statistics----
   5 packets transmitted, 5 packets received, 0% packet loss
   round-trip (ms)  min/avg/max = 0/0/0
   ```

   This is the result of an unsuccessful attempt to PING the destination address:

   ```
   [C:tcpipbin]ping 9.24.104.2 10 5
   PING 9.24.104.2: 10 data bytes

   ----9.24.104.2 PING Statistics----
   5 packets transmitted, 0 packets received, 100% packet loss
   ```

   If you cannot PING the destination IP address, then you should ensure that:

   a. The machine with the destination address is functioning correctly.

   b. You are on the same IP network or subnet, or your router can access the destination IP address.

   c. There are no physical network problems.

3. Determine if you can reach a router for the desired destination using PING. If that is unsuccessful and you are sure that the router itself is working, use:

   ```
   [C:]ARP -a
             ARP table contents:
   interface     hardware address         IP address   minutes since
                                                          last use
       0           10005aa8d769           9.24.104.108     4
       0           10005ac95035        192.200.200.200     5
   ```

```
0          10005ab1c42c          9.24.104.209     0
0          10005ab187f8           9.24.104.30     1
0          10005ac92ceb           9.24.104.60     0
0          10005ab1afe9          9.24.104.109     0
0          10005ab1d731          9.24.104.215     0
0          400052005011            9.24.104.1     0
0          10005ab1b0fe           9.24.104.12     0
0          10005ab1ac7d           9.24.104.15     0
0          10005a4f58ce           9.24.104.26     0
```

In this way, you can find out whether the router's hardware address is in your ARP table. If not, you can manually add it using the following command:

```
arp -s hostname hardware_address
```

4. Find out if a server process (ftpd, telnetd, snmpd, etc.) is running at your workstation if someone else cannot communicate with it, but PING, ARP, and IFCONFIG do not indicate an error condition. Enter the following command to see TCP and UDP activity at your workstation:

```
[c:]netstat -s
-------------------------------------------------------------------------------
                        AF_INET Address Family :

SOCK     TYPE        FOREIGN          LOCAL        FOREIGN     STATE
                        PORT           PORT           HOST
====  =========  =============   =============   =============  =============
1425    STREAM              0    printer..515         0.0.0.0    LISTEN
1424    STREAM              0         ftp..21         0.0.0.0    LISTEN
1423    STREAM              0      telnet..23         0.0.0.0    LISTEN
1402    STREAM           1080            1474      9.14.1.101    ESTABLISHED
1399    STREAM           1080            1473      9.14.1.101    ESTABLISHED
1396    STREAM           1080            1472      9.14.1.101    ESTABLISHED
560     STREAM     telnet..23            1095      9.164.34.3    ESTABLISHED
529     STREAM        ftp..21            1084      9.164.34.3    ESTABLISHED
22      STREAM              0        smtp..25         0.0.0.0    LISTEN
-------------------------------------------------------------------------------
                        AF_OS2  Address Family :
```

This will show you the sockets in use, the type of connection, the foreign and local ports, the addresses of the foreign hosts that are connected, and the state of the connection. A state of LISTEN refers to a server at your workstation. In the above example, the lpd server, the ftp, the sendmail and the telnet daemons are active and listening for incoming requests. There is also an active Telnet and FTP session to host 9.164.34.3 . Another socket application is communicating through three active ports with host 9.14.1.101 .

The port assignments can be found in the MPTNETCSERVICES file.

18.3 Trace Utilities

The following topics describe the facilities provided with TCP/IP for OS/2 to do tracing.

18.3.1 TRACERTE

The tracerte command is intended for use in network testing, measurement, and management. It should be used primarily for manual fault isolation. Because of the load it imposes on the network, the tracerte command should not be used during normal operations or from automated scripts.

The tracerte command attempts to trace the route an IP packet follows to an Internet host by launching UDP probe packets with a small maximum time-to-live (Max_ttl) variable, then listening for an ICMP TIME_EXCEEDED response from gateways along the way. Probes are started with a Max_ttl value of one hop, which is increased one hop at a time until an ICMP PORT_UNREACHABLE message is returned. The ICMP PORT_UNREACHABLE message indicates either that the host has been located or the command has reached the maximum number of hops allowed for the trace.

The tracerte command sends three probes at each Max_ttl setting to record the following:

- max_ttl value
- Address of the gateway
- Round-trip time of each successful probe

The number of probes sent can be increased by using the -q flag. If the probe answers come from different gateways, the command prints the address of each responding system. If there is no response from a probe within a 3-second time-out interval, an * (asterisk) is printed for that probe.

The tracerte command prints an ! (exclamation mark) after the round-trip time if the Max_ttl value is one hop or less. A maximum time-to-live value of one hop or less generally indicates an incompatibility in the way ICMP replies are handled by different network software. The incompatibility can usually be resolved by doubling the last Max_ttl value used and trying again.

Other possible annotations after the round-trip notation are:

- !H Host unreachable
- !N Network unreachable
- !P Protocol unreachable
- !S Source route failed
- !F Fragmentation needed

If the majority of probes result in an error, the tracerte command exits.

Syntax:

```
tracerte [ -m Max_ttl ] [ -n ] [ -p       Port ] [ -q Nqueries ]
 [ -r ] [ -s SRC_Addr ] [ -t TypeOfService ] [ -v ]
 [ -w WaitTime ] Host [ PacketSize ]
```

The only mandatory parameter for the tracerte command is the destination hostname or IP number. The default probe length is 38 bytes, but may be increased by specifying the packet size (in bytes) after the destination hostname. The UDP probe packets are set to an unlikely value so as to prevent processing by the destination host.

Flags

-m Max_ttl: Sets the maximum time-to-live (maximum number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).

-n: Prints hop addresses numerically rather than symbolically and numerically. This flag saves a name-server address-to-name lookup for each gateway found on the path.

-p Port: Sets the base UDP port number used in probes. The default is 33434. The tracerte command depends on an open UDP port range of base to (base + nhops - 1) at the destination host. If a UDP port is not available, this option can be used to pick an unused port range.

-q Nqueries: Specifies the number of probes the tracerte command sends at each Max_ttl setting. The default is three probes.

-r: Bypasses the normal routing tables and sends the probe packet directly to a host on an attached network. If the specified host is not on a directly attached network, an error is returned. This option can be used to issue a ping command to a local host through an interface that is not registered in the routed daemon's routing table.

-s SRC_Addr: Uses the next IP address in numerical form as the source address in outgoing probe packets. On hosts with more than one IP address, the -s flag can be used to force the source address to be something other than the IP address of the interface on which the probe packet is sent. If the next IP address is not one of the machine's interface addresses, an error is returned and nothing is sent.

-t TypeOfService: Sets the TypeOfService variable in the probe packets to a decimal integer in the range of 0 to 255. The default is 0. This flag can be used to investigate whether different service types result in different paths. Useful values are -t 16 (low delay) and -t 8 (high throughput).

-v: Receives packets other than TIME_EXCEEDED and PORT_UNREACHABLE (verbose output).

-w WaitTime: Sets the time (in seconds) to wait for a response to a probe. The default is 3 seconds.

Host: Specifies the destination host, either by hostname or IP number. This parameter is required.

PacketSize: Specifies the probe datagram length. The default is 38 bytes. This number can be increased by specifying the packet size, in bytes, after the destination hostname.

18.3.1.1 Examples

This is the result of a successful attempt to trace the route to a remote destination address:

```
[C:tcpipbin]tracerte 9.164.34.3
 0  6611ral (9.24.104.1)  0 ms  0 ms  0 ms
 1  6611ral (9.24.104.1)  0 ms  0 ms  969 ms
 2  9.24.96.1 (9.24.96.1)  0 ms  0 ms  0 ms
 3  6611slk.sl.dfw.ibm.com (9.24.1.1)  156 ms  188 ms  187 ms
 4  * 9.142.1.1 (9.142.1.1)  188 ms *
 5  9.32.108.1 (9.32.108.1)  219 ms  250 ms  250 ms
 6  9.32.1.13 (9.32.1.13)  344 ms  250 ms  188 ms
 7  * * *
 8  9.31.42.2 (9.31.42.2)  250 ms  344 ms  344 ms
 9  141.94.158.4 (141.94.158.4)  344 ms  282 ms  406 ms
10  141.95.240.10 (141.95.240.10)  438 ms  437 ms  500 ms
11  141.94.126.3 (141.94.126.3)  562 ms  469 ms  563 ms
12  141.95.125.38 (141.95.125.38)  531 ms  562 ms  438 ms
13  141.94.102.194 (141.94.102.194)  531 ms  531 ms  500 ms
14  141.95.125.210 (141.95.125.210)  406 ms  469 ms  438 ms
15  141.94.95.165 (141.94.95.165)  437 ms  532 ms  500 ms
16  * * *
17  9.164.34.3 (9.164.34.3)  781 ms  500 ms  500 ms
```

This is the result of an unsuccessful attempt to trace the route to a remote destination address:

```
[C:tcpipbin]tracerte 9.164.34.5
 0  6611ral (9.24.104.1)  0 ms  0 ms  0 ms
 1  6611ral (9.24.104.1)  0 ms  31 ms  0 ms
 2  6611ral.superlab.css.ibm.com (9.24.96.1)  0 ms  0 ms  31 ms
 3  6611slk.sl.dfw.ibm.com (9.24.1.1)  63 ms  62 ms  32 ms
 4  * * *
 5  9.32.108.1 (9.32.108.1)  94 ms  63 ms  62 ms
 6  9.32.1.13 (9.32.1.13)  94 ms  125 ms  94 ms
 7  * * *
 8  9.31.42.2 (9.31.42.2)  218 ms  250 ms  282 ms
 9  141.94.158.4 (141.94.158.4)  218 ms  219 ms  250 ms
10  141.95.240.10 (141.95.240.10)  344 ms  313 ms  375 ms
11  141.94.126.3 (141.94.126.3)  344 ms  313 ms  281 ms
12  141.95.125.38 (141.95.125.38)  375 ms  375 ms  406 ms
13  141.94.102.194 (141.94.102.194)  375 ms  375 ms  406 ms
14  141.95.125.210 (141.95.125.210)  312 ms  312 ms  344 ms
15  141.94.95.165 (141.94.95.165)  344 ms  312 ms  344 ms
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

TRACERTE will constantly increase the time-to-live value of the packages sent, and it will stop after 30 attempts.

Note: A route to a destination may still exist (you can PING the remote address) though TRACERTE cannot track it. The reason may be that the response time of some routers is longer than it takes TRACERTE to finish.

18.3.2 IPTRACE

The utility IPTRACE traces all packets received from and sent to an interface.

The syntax of the IPTRACE command is:

```
iptrace [-i ] [interface]
```

-i Specifies that only IP packets are to be traced. The default is to include all information (such as hardware type). Certain interfaces (for example, X.25 and SNALINK) require this parameter.

interface Specifies an interface to be traced (for example, lan0 or sl0). If not specified, all interfaces are traced.

Note: The following additional characteristics are useful to be aware of when using IPTRACE:

1. IPTRACE writes data to IPTRACE.DMP in the directory from which you initiated the IPTRACE command. IPTRACE records all traffic sent and received, but does not check for sufficient disk space to record that information. As a result, running IPTRACE can impact your workstation′s performance as the IPTRACE.DMP file continues to grow larger.
2. IPTRACE is not a network monitor. It can trace only data received by and sent from the specified interfaces.
3. IPTRACE provides a time stamp, recording when the packet was sent or received.
4. To stop IPTRACE, press Enter, Ctrl-Break, or Ctrl-C.
5. Use IPFORMAT to convert the IPTRACE.DMP file into a user-readable format.

During tracing, every packet causes an entry in the trace window. The complete packet data is stored in the IPTRACE.DMP file.

```
[C:\]iptrace
lan0: tracing enabled
 lan0:[  0.000]: process_pkt: len=58, type=9
 lan0:[  0.875]: process_pkt: len=56, type=9
 lan0:[  0.000]: process_pkt: len=56, type=9
 lan0:[  0.000]: process_pkt: len=56, type=9
 lan0:[  0.125]: process_pkt: len=58, type=9
 lan0:[  1.000]: process_pkt: len=58, type=9
 lan0:[  0.562]: process_pkt: len=80, type=9
 lan0:[  0.032]: process_pkt: len=58, type=9
 lan0:[  0.218]: process_pkt: len=68, type=9
 lan0:[  0.250]: process_pkt: len=220, type=9
 lan0:[  0.032]: process_pkt: len=220, type=9
 lan0:[  0.093]: process_pkt: len=202, type=9

lan0: tracing disabled
the ip trace taken
```

18.3.3 IPFORMAT

The IPFORMAT utility converts the data in the IPTRACE.DMP file and the SLIPTRC.DMP file (if the file was created without VJ header compression on) to either human-readable format, which is displayed to the screen, or to a file that can be used as input to a network analyzer. If you choose to convert the data into human-readable format, you can redirect the output to a text file.

IPFORMAT reads the header information in the trace to determine the type of packet received; for example, token-ring (TRC) or ethernet (ENC). It then separates the data by the IP, TCP, UDP, and ICMP layers. The rest of the packet is displayed as hexadecimal output.

The syntax of the IPFORMAT command is:

```
ipformat [-a ] [ -d ] [-h ] [ -f filename ]
 [-n ] [ -s hwaddress ] [-x ] > filename
```

-a Do not format ARP or RARP packets.

-d Do not display the data portion of a packet.

-f filename Specifies the input file name. The default is IPTRACE.DMP.

-h Display the raw data packet after the formatted information.

-s hwaddress Format data only for the specified hardware address. hwaddress is the 12-digit hexadecimal address for the Ethernet or token-ring adapter. You can use the NETSTAT -N command to display this address.

-n Do not display hexadecimal data for unknown data type.

-x Converts IPTRACE data to a format that can be read by a Network General Sniffer.

18.3.3.1 Examples

The following trace packet shows some key entries of a formatted output of the IPFORMAT utility. It has been taken out of an LPR session between a VM - Host (9.165.68.3) sending print data to an OS/2 LPD server (9.24.104.77).

```
------------------------------- #:16 -------------------------------
  Delta Time:  0.000   Packet Length: 78 bytes (4E hex)
 802.5:  Dest: 08:00:5A:94:30:B8   Source: C0:00:52:00:50:11
 802.5:  Dest: 009.024.104.077.    Source: 009.165.068.003.
---------------------- IP HEADER -----------------------
 IP:  Version: 4 Correct    Header Length: 20 bytes
 IP:  Type Of Service: 00
 IP:     000. ....  Routine
 IP:     ...0 ....  Normal Delay
 IP:     .... 0...  Normal Throughput
 IP:     .... .0..  Normal Reliability
 IP:  Total Len: 50 (x32) bytes          Id: 7BC4
 IP:  Flags: 0
 IP:     .0..       May Fragment
 IP:     ..0.       Last Fragment
 IP:  Fragment Offset: 000
 IP:  Time To Live: 46 sec    Protocol: 06 (TCP)
 IP:  Header Checksum: 51F5
 IP:  No Options
---------------------- TCP HEADER ----------------------
 TCP:  Source Port: 721        Dest Port: 515  (Printer)
 TCP:  Sequence #: 515366977
 TCP:  Ack #: 133824002
 TCP:  Offset: 20 bytes
 TCP:  Flags: 18
 TCP:     ..0. ....  Urgent bit Off
 TCP:     ...1 ....  Ack bit On
 TCP:     .... 1...  Push bit On
 TCP:     .... .0..  Reset bit Off
 TCP:     .... ..0.  Synchronize bit Off
 TCP:     .... ...0  Finish bit Off
 TCP:  Window: 8192      Checksum: 99B6
 TCP:  No Options
-------------------------------- DATA --------------------------------
0000 02 57 54 52 50 52 54 30    32 0A                     .WTRPRT02.
```

The following are the key fields:

Packet Header It contains information about source and destination IP addresses as well as a time stamp relative to the previous packet. This might be useful information in the case of timing dependent problems.

IP Header Shows usage of TCP or UDP.

TCP Header Shows the used port numbers and in the case of a well-known port the type of application (in this case a printer session).

User Data In the case when the packet contains user data, it will be provided at the end of this packet (in this case the printer queue name, where the data should be printed to).

18.3.4 Other trace facilities

The following TCP/IP servers and clients that come with TCP/IP for OS/2 also provide some tracing or logging capability:

- DDNS
- DHCP Server
- DHCP Cient
- BOOTPD
- NSLOOKUP
- PMX
- ROUTED
- SNMP
- SNMPD
- TELNETD
- Telnet client

In most cases debugging information can be obtained by starting the servers or clients with one or more additional -d parameters. Please refer to the provided online documentation for how to start or invoke tracing and what additional information can be obtained.

Chapter 19. Network Management

This chapter describes the capabilities and tools provided with TCP/IP for OS/2 to manage resources in a TCP/IP environment. The following programs are provided for those purposes:

- SNMP
- SNMPGRP
- PMTRAPS
- PING
- PMPING
- NETSTAT
- ARP
- RPCINFO

19.1 SNMP Concepts

Simple Network Management Protocol (SNMP) defines an architecture that consists of network management stations (SNMP monitors or clients) and network elements (SNMP agents or servers). Refer to Figure 267 on page 418 as an example of an implementation for the following description of the SNMP concepts.

SNMP monitor collects and analyzes data sent by SNMP agents residing on nodes in the network. This data can specifically be requested by an SNMP monitor by querying an SNMP agent for desired information (sending a GET or GETNEXT API call). The agent responds with the requested data (GET_RESPONSE).

The collection of all the data that can be obtained from an SNMP agent is called the management information base (MIB). The MIB defines objects that are relevant to a TCP/IP environment, such as packet counts and routing tables, and divides them into the following groups:

- System
- System Interfaces
- Address Translation
- IP
- ICMP
- TCP
- UDP
- EGP.
- Transmission
- SNMP

An SNMP agent can also send an unsolicited message (called a trap) containing information about, for example, the failure of a network resource.

GET, GET_NEXT, GET_RESPONSE and TRAP are called protocol data units (PDUs). SNMP also defines a fifth PDU called SET, that is issued from an SNMP monitor to update data at an SNMP agent.

SNMP also has a concept of subagents. A subagent is a separate program that communicates with an agent via a Distributed Protocol Interface (DPI) over a

TCP connection. A subagent can register its own MIB objects and groups with an agent. The agent will then get the information from the subagent, when an SNMP monitor requests it from the agent. A subagent can also generate traps, which the agent will forward to the SNMP monitor. This makes it possible to generate your own traps by developing your own subagents with the SNMP DPI.

19.2 Overview of the SNMP Implementation in OS/2

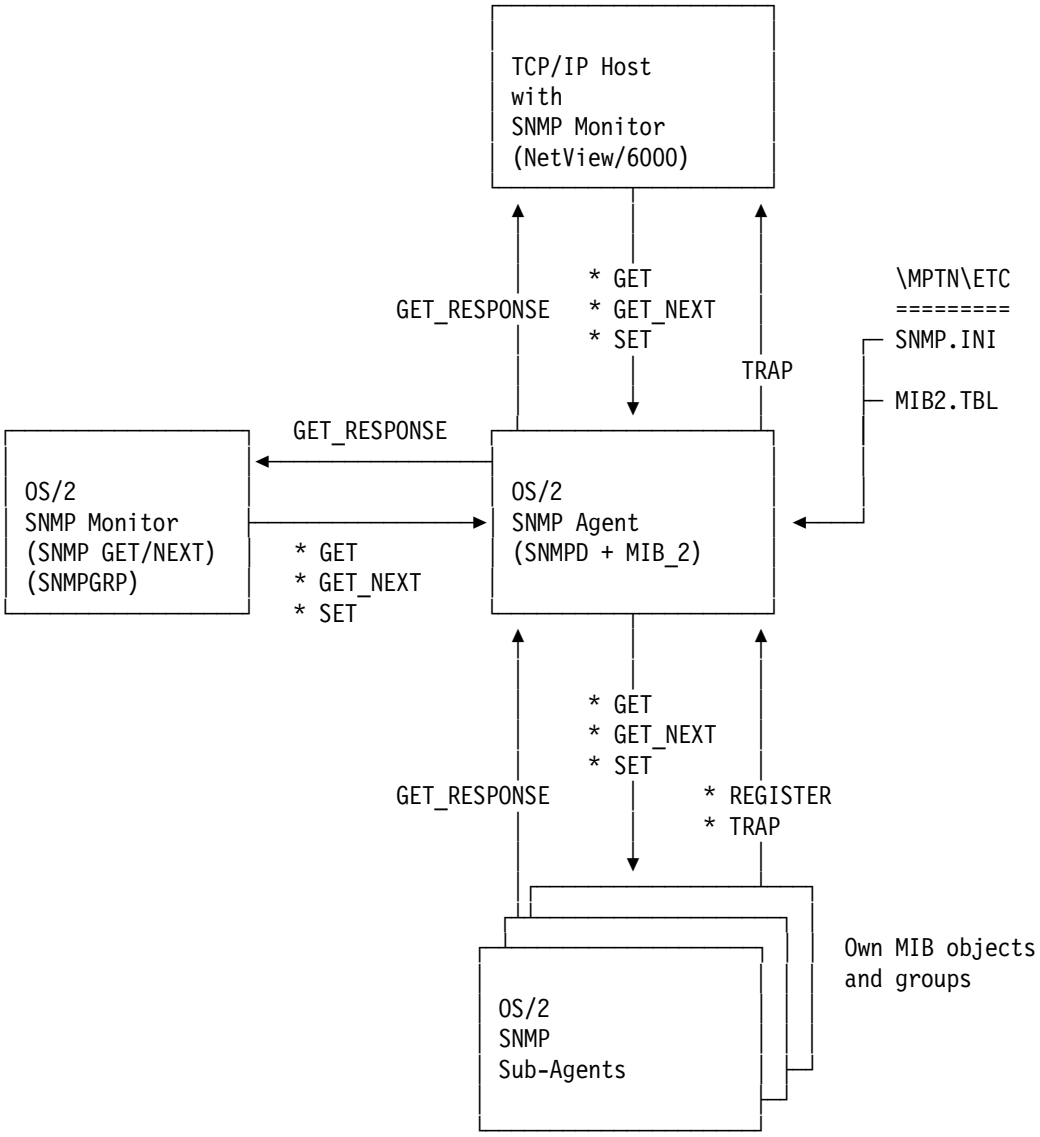


Figure 267. SNMP Implementation in OS/2

The implementation of SNMP in OS/2 is shown in Figure 267. TCP/IP for OS/2 implements the following SNMP functions:

- An SNMP agent (SNMPD + MIB_2) that supports the following:
  - All currently defined MIB groups *except* the Exterior Gateway Protocol (EGP)
  - Two TRAPs:

COLD_START (agent has reconfigured itself)

AUTHENTICATION_FAILURE (a monitor request was not properly authenticated)

Its own TRAPs can be implemented by developing subagents.

– The protocol data units (PDUs) according to Figure 267 on page 418.

- The possibility of developing its own subagents with the SNMP DPI and connecting them via TCP or shared memory to SNMPD. A subagent can run on the same OS/2 machine as SNMPD or on a separate workstation.
- SNMP monitor functions SNMP GET/NEXT, SET and SNMPGRP make it possible to retrieve and alter MIB information from SNMPD running in a remote TCP/IP host. See Figure 267 on page 418. SNMPGRP can be regarded as a ″remote″ NETSTAT command.

19.3 Setting Up the SNMP Agent (Server) in OS/2

Use the TCP/IP Configuration Notebook to set up the SNMP agent on your OS/2 system. The following screens describe the configuration pages:

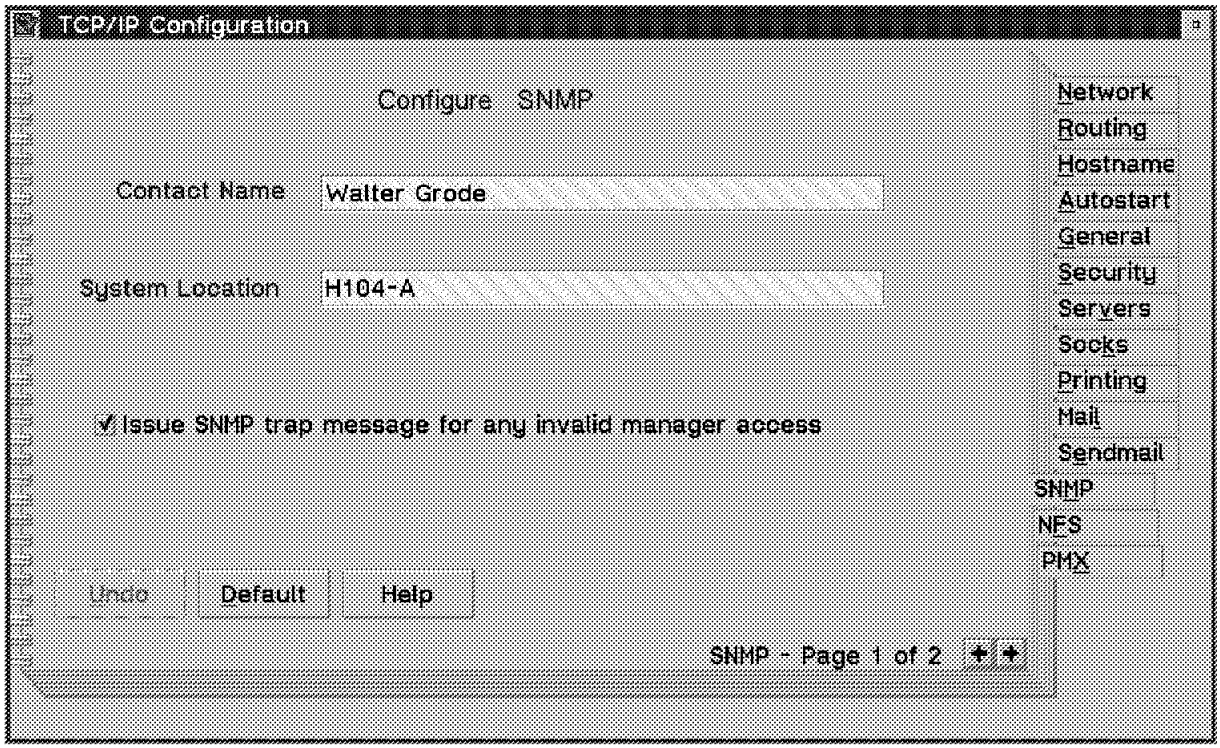


Figure 268. TCP/IP Configuration Notebook for SNMP, Page 1

Setting	Meaning
Contact Name	Name of the person responsible for that OS/2 workstation
System Location	Location of that OS/2 workstation

With the check box you can enable this SNMP agent to send a trap message to a given destination in case of an unauthorized access attempt (in case of an invalid community name).

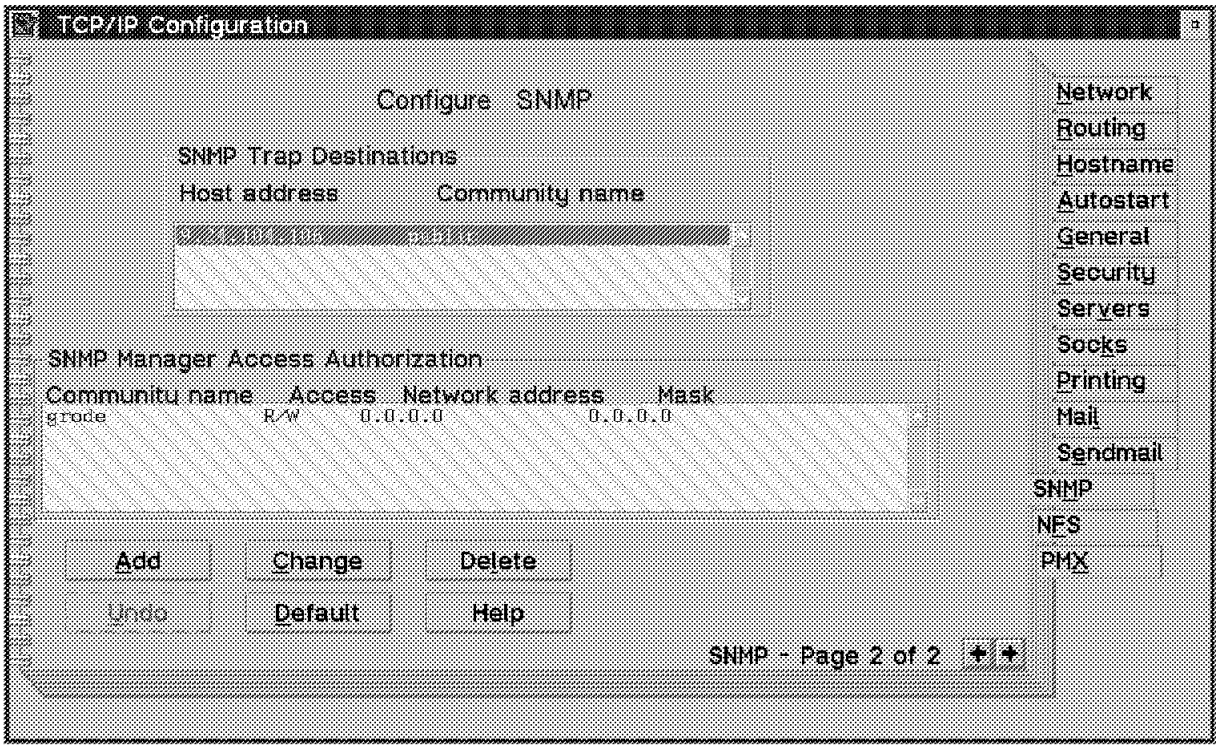


Figure 269. TCP/IP Configuration Notebook for SNMP, Page 2

Setting	Meaning
SNMP Trap Destinations	Specify hosts where your SNMP agent will send TRAPs.
Access Authorization	Specify the SNMP communities and networks that you want to share MIB information with.
	Community Name Name for an SNMP community
	Access Allow read/write of MIB values
	Network Address IP network base address
	Mask Subnet mask for that IP network

These values allow an SNMP monitor to contact your SNMP agent. The IP address of an incoming request to SNMPD is logically ANDed with the SNMP mask and compared with the desired network. If they match, the password (or community name) is compared to the password in the incoming request. If that also matches, the request is accepted.

In the above example, access to SNMPD is restricted to SNMP monitors using the password grode. They can be located in any network.

Note: The community names are case sensitive.

The configuration will make changes to the following file:

- MPTNETCSNMP.INI

There are no longer environment variables in your CONFIG.SYS file being changed as was the case with prior TCP/IP releases.

After leaving the Configuration Notebook, the community names are automatically scrambled into the SNMP.INI file. There is no longer a need to execute the MAKE_PW program.

To start the OS/2 SNMP agent and to enable the SNMP monitor, type:

```
start SNMPD -dpi shm
start MIB_2
```

from an OS/2 command prompt.

19.4 Network Management Utilities

TCP/IP for OS/2 contains a few basic utilities to monitor the TCP/IP network and provide information about local activity and remote hosts.

19.4.1 SNMP and SNMPGRP

With these commands you can get information from the SNMPD agent running in an IP host. They search an internal table first to find the requested information. This table defines the mapping between an object′s ASN.1 notation and its textual notation, and lists the objects syntax. For example, when you issue an `snmp get` command and specify the textual description (for example sysDescr.0) the OS/2 SNMP client looks for that object in the internal table and uses the corresponding ASN.1 notation in the SNMP request to an SNMP agent. If the mib_name does not exist in the internal table, SNMP searches the MIB2.TBL for the information.

The internal table contains all the textual names defined in the following RFCs:

- 1155
- 1213
- 1231
- 1385
- 1315
- 1398

During installation, the MIB2.TBL file is placed in the directory defined by the ETC environment variable in your CONFIG.SYS file. Modify this file as needed for vendor-specific MIB objects, and save the new file in that ETC subdirectory. For more information about the layout of this file refer to the provided online documentation.

Following are some examples showing snmpgrp commands to host walter requesting MIB data:

```
OS/2     Ctrl+Esc = Task List                                Type HELP = help

[F:]snmp -h walter -c grode get syslocation.0
sysLocation.0 : H104-A

[F:]snmpgrp -h walter -c grode sys

SYSTEM group -------------------------

     Descr:  OS/2 SNMP Agent version 1.3 with DPI version 2.0 (October 25, 1994)

  ObjectId:  1.3.6.1.4.1.2.6.46
    UpTime:  6666800 - 18 hours, 31 minutes, 8.0 seconds
   Contact:  Walter Grode
      Name:  walter
  Location:  H104-A
  Services:  76

End of group sys -----------------


[F:]snmpgrp -h walter -c grode tcp

TCP group-----------------------------

    RtoAlgorithm:  4
          RtoMin:  2000
          RtoMax:  128000
         MaxConn:  255
     ActiveOpens:  462
    PassiveOpens:  413
    AttemptFails:  5
     EstabResets:  808
       CurrEstab:  14
          InSegs:  321271
         OutSegs:  272487
     RetransSegs:  1269
           InErr:  0
         OutRsts:  0
                :  10966

End of group tcp -----------------

[F:\]
```

19.4.2 SNMPTRAP

SNMPTRAP is a Presentation Manager application that receives, interprets and displays SNMP traps. The trap in the following sample was produced by restarting the SNMP agent on system walter:

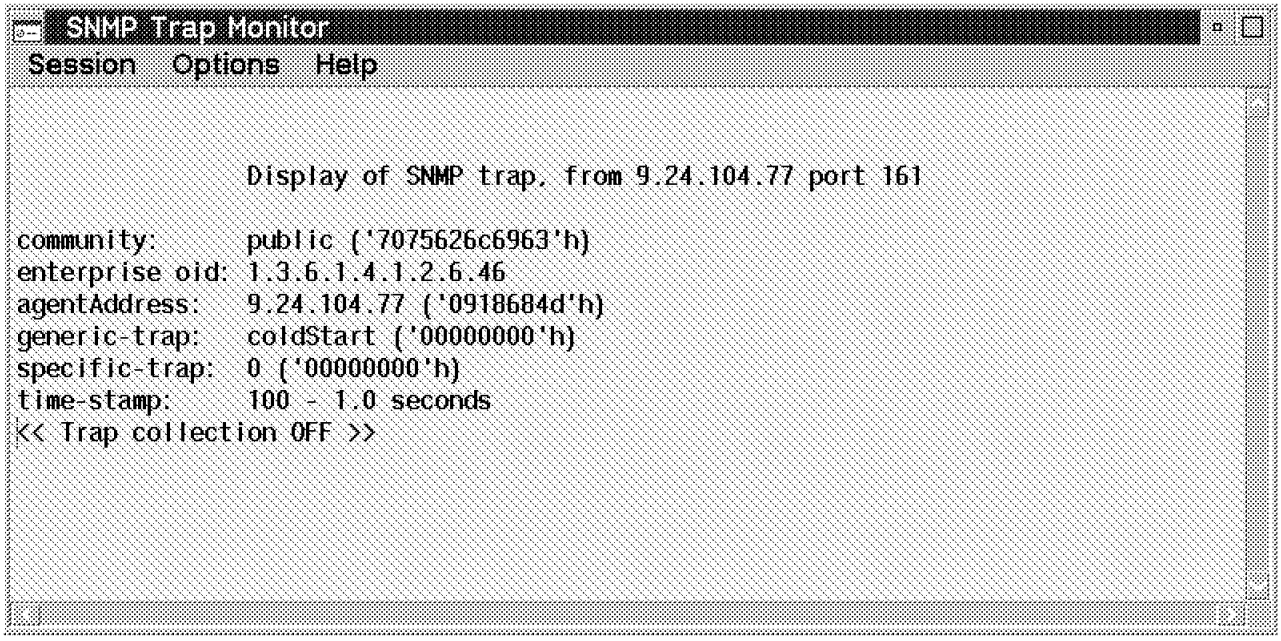


Figure 270. SNMPTRAP Showing a Trap

19.4.3 PMPING

PMPING periodically PINGs a number of hosts defined in a file called PINGHOST.LST. You can create that file manually, or use the TCP/IP Configuration Notebook (page 2 of the Services tag). The turnaround time, or a highlighted error indicator for each defined host is displayed.

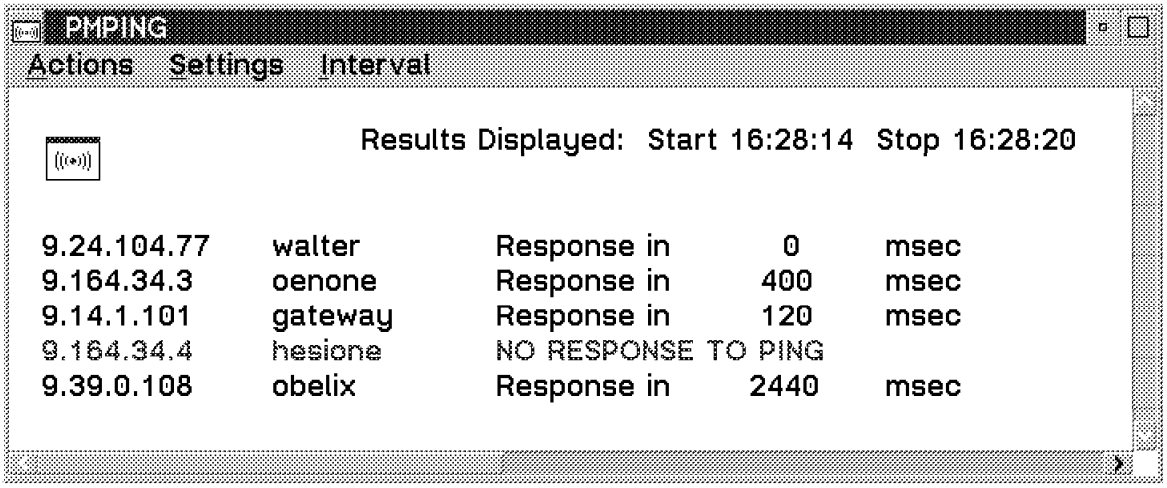


Figure 271. PMPING Showing Turnaround Times

The PMPING application can be used to show the status of all important application servers or network routers.

19.4.4 NETSTAT

NETSTAT displays all available information about the local TCP/IP system and in the case of communication problems it is a helpful tool for finding the cause. The routing and ARP tables especially should be verified in case of problems.

```
[C:tcpiptmp]netstat -a
addr        9.24.104.77 interface 0 mask ffffff00 broadcast  9.24.104.255

[C:\tcpip\tmp]netstat -s
SOCK      TYPE         FOREIGN           LOCAL         FOREIGN      STATE
                          PORT            PORT            HOST
==== ==========      ==========      ==========      ========== ==========
1056     DGRAM              0               0         0.0.0.0   UDP
1309    STREAM           6000            1028     9.24.104.77   CLOSE_WAIT
81      STREAM              0            1024         0.0.0.0   LISTEN
80       DGRAM              0             161         0.0.0.0   UDP
79       DGRAM              0            1024         0.0.0.0   UDP
78       DGRAM              0               0         0.0.0.0   UDP
10      STREAM              0             139         0.0.0.0   LISTEN
9        DGRAM              0             138         0.0.0.0   UDP
8        DGRAM              0             137         0.0.0.0   UDP

[C:\tcpip\tmp]netstat -n
Interface 0: 802.5
physical address    08005a817174      MTU 1500

speed 4000000 bits/sec
unicast packets received 11294
broadcast packets received 3537
total bytes received 2270412
unicast packets sent 10676
broadcast packets sent 11
total bytes sent 1252732
packets discarded on transmission 0
packets discarded on reception 0
received packets in error 0
errors trying to send 0
packets received in unsupported protocols 0

[C:\tcpip\tmp]
```

The example shows how to use the netstat command to do the following:

- Obtain information about locally used IP addresses (-a option)
- Obtain information about local application activity showing active sockets, ports, local and remote addresses, and status (-s option)
- Obtain information about the local hardware interfaces assigned to IP (-n option)

19.4.5 ARP

The ARP table is used to map IP addresses and physical addresses on a LAN. It is dynamically built by the Address Resolution Protocol (ARP) provided by TCP/IP. To reduce network overhead, the entries are kept in memory. If the entry is not used during a certain time period, it will be deleted. Sometimes it is desirable to delete an entry manually, or to refresh the whole ARP table. That can be done with the ARP command:

```
[C:]arp -a
          ARP table contents:

interface     hardware address        IP address   minutes since
                                                     last use
   0          10005aa8d769           9.24.104.108     1
   0          10005ac95035           9.24.104.241     1
   0          10005ab1c42c           9.24.104.209     1
   0          10005ab187f8            9.24.104.30     0
   0          10005ac92ceb            9.24.104.60     0
   0          10005ab1afe9           9.24.104.109     0
   0          08005ac60015           9.24.104.185     1
   0          10005ab1d731           9.24.104.215     0
   0          10005ab155da            9.24.104.25     0
   0          10005aa87023            9.24.104.28     0

[C:\]
```

The example shows all entries in the ARP table on system 9.24.104.106 . The command arp -f deletes all entries. If you then PING any system, it will cause an ARP request first. This can be useful for tracing and debugging purposes.

19.4.6 RPCINFO

This command reports the status and services registered to RPC (Remote Procedure Call) on remote systems. This is especially useful if you want to use an RPC-based application such as NFS, and you want to ensure that communication with a remote host will not result in an error.

```
[C:]rpcinfo -p rs600014
   program vers proto   port
    100000    2   tcp    111
    100000    2   udp    111
    100001    1   udp   1029
    100001    2   udp   1029
    100001    3   udp   1029
    100002    1   udp   1031
    100002    2   udp   1031
    100008    1   udp   1033
    100012    1   udp   1035
    150001    1   udp   1037
    150001    2   udp   1037
    100083    1   tcp   1026
    100068    2   udp   1039
    100068    3   udp   1039
    100068    4   udp   1039
    100068    5   udp   1039
    100024    1   udp    895
    100024    1   tcp    897

[C:\]
```

The example shows RPC status information obtained from an AIX TCP/IP host.

19.5 Managing OS/2 TCP/IP Hosts Using NetView/6000

NetView/6000 is a network management application based on SNMP. It fully implements an SNMP monitor with the capability to send SNMP GET, GET_NEXT and SET commands to SNMP agents on a TCP/IP network.

The following figure shows an OS/2 TCP/IP host running NetView/6000 Entry from an AIX system. It uses PMX, the OS/2 X Window System Server, since NetView/6000 is implemented as an X client application. The figure shows the MIB browser utility to view or change values of a system's MIB.

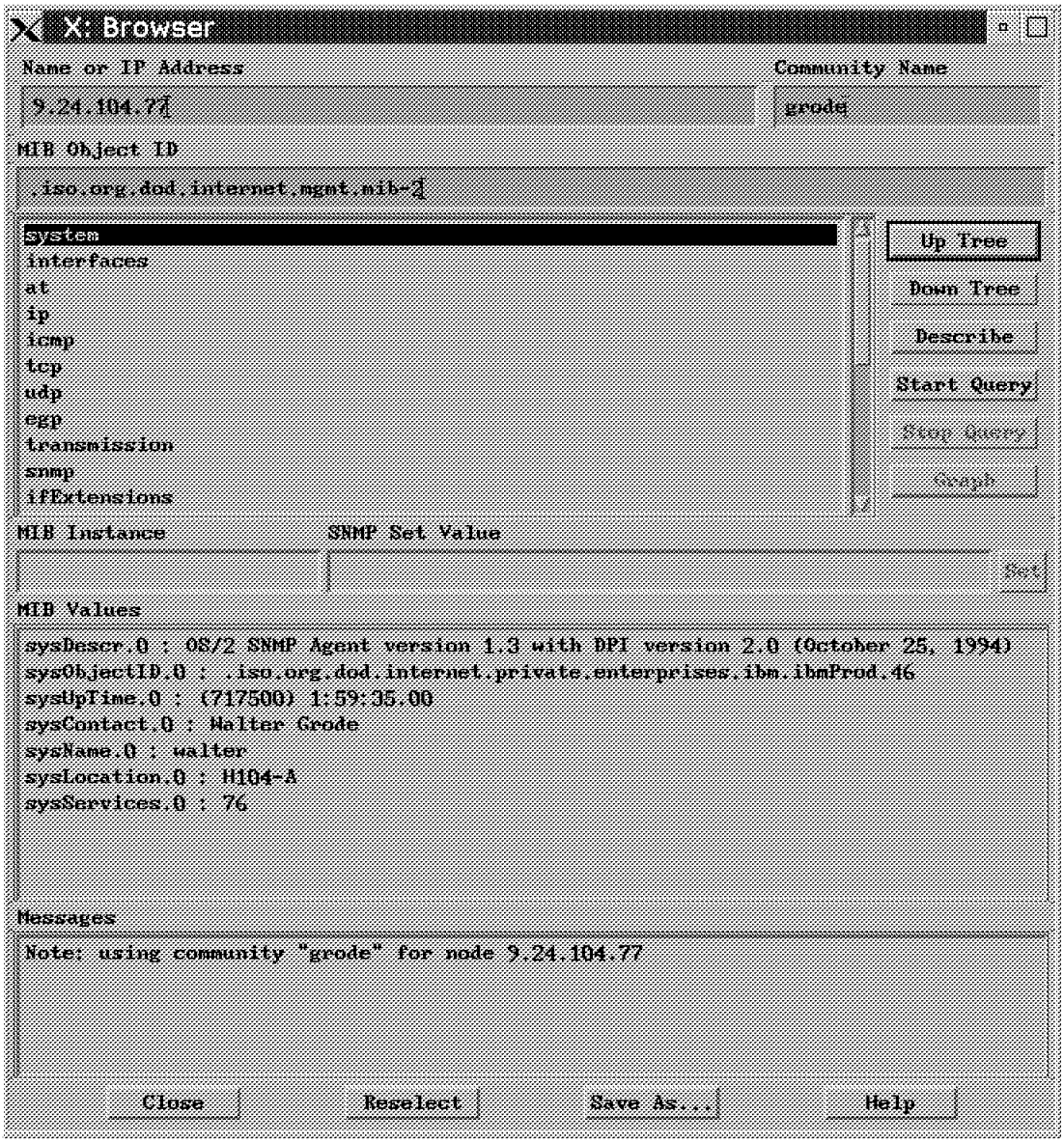


Figure 272. Managing an OS/2 MIB with NetView/6000 from an OS/2 TCP/IP System

19.6 Introduction to SNMP Distributed Protocol Interface

This section gives you a short description of the SNMP DPI interface provided by TCP/IP for OS/2. This interface gives you the ability to implement your own subagents into the existing environment. You may also want to obtain the following documentation:

- RFC1592 SNMP DPI 2.0 RFC
- RFC1440 - RFC1452
- Documentation for IBM TCP/IP for OS/2 Programmers Toolkit V3

19.6.1 SNMP Agents and Subagents

SNMP agents are responsible for answering SNMP requests from network management stations. Examples of management requests are GET, GETNEXT, and SET, performed on the MIB objects.

A subagent extends the set of MIB objects provided by the SNMP agent. With the subagent, you define MIB variables useful in your own environment and register them with the SNMP agent.

When the agent receives a request for an MIB variable, it passes the request to the subagent. The subagent then returns a response to the agent. The agent creates an SNMP response packet and sends the response to the remote network management station that initiated the request. The existence of the subagent is transparent to the network management station.

To allow the subagents to perform these functions, the agent provides for two types of subagent connections:

- TCP connection
- Connection via shared memory

For the TCP connections, the agent binds to an arbitrarily chosen TCP port and listens for connection requests. A well-known port is not used. Every invocation of the SNMP agent could potentially use a different TCP port.

A subagent of the SNMP agent determines the port number by sending a GET request for a MIB variable, which represents the value of the TCP port. The subagent is not required to create and parse SNMP packets because the DPI API has a library routine query_DPI_port(). After the subagent obtains the value of the DPI TCP port, it should make a TCP connection to the appropriate port. After a successful connect(), the subagent registers the set of variables it supports with the SNMP agent. When all variable classes are registered, the subagent waits for requests from the SNMP agent.

The query_DPI_port() function is implicitly executed by the DPIconnect_to_agent_TCP() function. The DPI subagent programmer would normally use the DPIconnect_to_agent_TCP() function to connect to the agent, so it does not need to obtain the value of the DPI TCP port.

For a SHM connection, the subagent can use the DPIconnect_to_agent_SHM() function.

19.6.2 DPI Agent Requests

The SNMP agent can initiate the following DPI requests:

- GET
- GETNEXT
- GETBULK (SNMP Version 2)
- SET
- COMMIT
- UNDO
- UNREGISTER
- CLOSE

The GET, GETNEXT, GETBULK, and SET requests correspond to the SNMP requests that a network management station can make. The subagent responds to a request with a response packet.

The GETBULK requests are translated into multiple GETNEXT requests by the agent. According to RFC 1592, a subagent may request that the GETBULK be passed to it, but the OS/2 version of DPI does not yet support that request.

The TCP/IP for OS/2 Programmers Toolkit V3 provides in-depth documentation and programming samples showing how to code a subagent.

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

Q

R

S

T

U

V

W

X

ITSO Redbook Evaluation

International Technical Support Organization
TCP/IP Implementation in an OS/2 Warp Environment
April 1996

Publication No. SG24-4730-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) Are you an employee of IBM or its subsidiaries: Yes____ No____

b) Do you work in the USA? Yes____ No____

c) Was this redbook published in time for your needs? Yes____ No____

d) Did this redbook meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this redbook?

_____

_____

What other redbooks would you like to see published?

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

_____ _____
Name Address

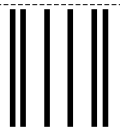
_____ _____
Company or Organization

_____ _____
Phone No.

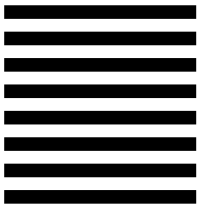
ITSO Redbook Evaluation
SG24-4730-00

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

IBM International Technical Support Organization
Department HZ8, Building 678
P.O. BOX 12195
RESEARCH TRIANGLE PARK NC
USA 27709-2195

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.