Softerm TCP/IP NFS and TCP/IP Kernel Manual

Please note that while using a pre-configured TCP/IP connection is relatively straightforward, actually configuring a TCP/IP connection can be very difficult for the inexperienced. We suggest that yor Network Administrator use the Kernel portion of this on-line manual to install and properly configure Softerm TCP/IP Kernel.

The Softerm TCP/IP Kernel is bundled with every copy of Softerm Modular. Softerm TCP/IP NFS is available as a separate product. For more information, contact Softronics, Inc.

Copyright Statement

Changes are made periodically to the information herein; these changes will be incorporated in new editions of this publication. Contents are property of Softronics, Inc..

Softronics, Inc. provides this guide without warranty of any kind, either implied or expressed, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Softronics, Inc. may make improvements or changes in the product(s) and/or the programs(s) described in this manual at any time.

Trademarks

SOFTERM MODULAR, SOFTERM TCP/IP NFS and SOFTERM TCP/IP KERNEL are trademarks of Softronics, Inc.,
UNIX and UNIX System V are trademarks of AT&T Bell Laboratories.
DESQview is a trademark of Quarterdeck Office Systems.
MS, MS-DOS, Windows/286, Windows/386, Windows 3.0 and LAN Manager are trademarks of Microsoft Corporation.
VT-100, VMS, VAX and DECnet are trademarks of Digital Equipment Corporation.
Ethernet is a trademark of Xerox Corporation.
IBM, PC-DOS, OS/2, PC-AT, PS/2 and Micro-channel are trademarks of International Business Machines Corporation.
DR-DOS is trademark of Digital Research Inc.
SUN/OS is a trademark of Sun Micro Systems.
NETWARE is a trademark of Novell, Incorporated.
Etherlink, Etherlink II, Etherlink Plus, Etherlink/MC, Tokenlink, NDIS and BAPI are trademarks of 3COM Corporation.
EtherCard PLUS is a trademark of Western Digital Corporation.
LanCard is a trademark of Tiara Computer systems, Inc.
EtherPocket is a trademark of ACCTON Technology Corporation
NICpc and NICps2 are trademarks of Ungermann-Bass, Inc.

Acknowledgements

TCP/IP is based on the first release of BSD Networking Software under the license from the Regents of the University of California.. Additionally, portions of this product are derived from the PC/IP software developed and copyrighted in 1984 by the Massachusetts Institute of Technology.

We also would like to express our thanks to the following institututions and corporations for permitting us to distribute their network drivers along with our TCP/IP NFS product: Clarkson University, 3Com Corporation and Western Digital Corporation.

Softronics Address and Phone Numbers

Softronics, Inc.
5085 List Drive
Colorado Springs, CO   80919

Sales: (800) 225-8590
Office: (719) 593-9540
FAX   : (719) 548-1878
Tech Support: (900) 884-7638

Windows Sockets

- A Windows Sockets DLL (v1.1 compliant) is now included with the Kernel package. The DLL allows Windows applications conforming to Windows Sockets interface to access Kernel's network services. It is automatically loaded when a Windows application makes Windows Sockets call. Make sure that the PATH environment variable is set to reference the drive and directory where the WINSOCK.DLL file is located. We place it in our BIN directory since our installation program edits the AUTOEXEC.BAT file and modifies the PATH variable to point to the BIN directory. However, you can copy it to wherever you want it to be.

- Add support for DNS (Domain Name Service).

Kernel Changes

- TCP send/receive windows are now configurable in the ranges of 4 and 64 (of 256 bytes unit). A larger window size usually yields better performance, but more network buffer space is required. Typically, the relationship between the window size and buffer space are as follows:

TCP Window Size (in multiple of 256 bytes)	Buffer Space (in Kbytes)
4 - 16	10
17 - 32	20
32 - 48	30
48 - 64	40

  Make sure you have sufficient buffers after increasing the window size(s). Otherwise, your system may experience excessive packet drops due to lack of network buffers, which causes performance degration. One way to determine whether the kernel has enough buffers for network usage is to check whether the kernel has denied any memory requests. This can be done by using the "NETSTAT -M" command.

- The kernel now provides full support for Token-Ring's source routing processing as specified by IEEE 802 standard. It is now able to communicate with other hosts on remote rings via Token-Ring bridges. It also negotiates for maximum receive packet size during the ARP process. The Softerm TCP/IP Kernel can receive TR packets of up to 2088 bytes including the MAC header.

- The UDP send and receive windows are now expanded to satisfy popular demand from several customers. The minimum window size is:

  min_window_size = 4.5 kbytes

  The window is expandable as more network buffer space is available. They are computed according to the following formula:

  window_size = MAX(min_window_size, 20% of avail. buf space)
  window_size = MIN(window_size, 45% of avail. buf space)

  The available buffer space can be configured by varying the field "Network Buffer Space" in the SITE menu of TCPSETUP. The receive and send windows have the same setting.

- Softerm TCP/IP Kernel forwards all packets destined to hosts on other networks to the default gateway which must be specified by using the SITE menu of TCPSETUP.

Driver Interfaces

Using Softerm TCP/IP Packet-driver kernel (TCPIP_PD.EXE) under Windows.

When loading TCPIP_PD.EXE with a packet driver to use under Windows, you do NOT need to specify the **-w** option on the packet driver command line.

Installing TCP_PD.EXE with SLIPPER/CSLIPPER

Add support for shareware SLIPPER (SLIP) and CSLIPPER (compressed SLIP) drivers. The drivers are not included with Softerm TCP/IP Kernel but are available from the author (Peter Tattam) at Internet address:

peter@psychnet.psychol.utas.edu.au

or, you can directly download the drivers from:

ftp.utas.edu.au:/pc/trumpet/slipper/slipr13.zip

To use Softerm TCP/IP Packet driver kernel with SLIPPER or CSLIPPER driver.

1. Use TCPSETUP to set up SITE information. Make sure to set the "Default Gateway's IP Address" field with the IP address of the remote machine that acts as an IP router

2. Use a communications program (such as Softerm Modular) to dial the IP router,

3. Enter the login/password as prompted

4. If successfully logged in, enter the IP router's command that puts the port into SLIP or CLIP mode.

5. Exit from the communications program without hanging up the call

6. Load SLIPPER or CSLIPPER as instructed above

   ```
   C> SLIPPER .....

         or,

   C> CSLIPPER ....
   ```

7. Load Softerm TCP/IP Packet driver kernel

   ```
   C> TCPIP_PD
   ```

8. Try the PING command to verify connectivity

You can skip step 3 to 4 if user authentication is not required by the IP router.

Loading Softerm TCP/IP Kernel With Packet driver Multiplexor

Here is a sample configuration that shows you how Softerm TCP/IP Kernel can be configured to coexist with other protocol stacks using the Packet driver multiplexor. Note that all the commands can be put together in a batch file.

The following installation allows Softerm TCP/IP Kernel, Novell IPX and WinQVT to share the same packet driver (NE2000.COM). The only problem with this is that WinQVT's ftp server does not work for unknown reasons. A Winsock version of WinQVT (currently, in beta) solves this problem.

1. First, you need to run TCPSETUP to configure the SITE information.
   (Also local HOSTS table if necessary).

2. Second, follow the below example:

```
REM Load the packet driver for NE2000 adapter
C:\PKT\NE2000 0x7f 3 0x300

REM Now load PKTMUX (available from all good Simtel20 mirrors)
C:\PKT\PKTMUX 4

REM (create pseudo drivers - 3 are used here)
REM Now instantiate all 4 drivers before loading windows.
C:\PKT\PKTDRV 68
C:\PKT\PKTDRV 69
C:\PKT\PKTDRV 6a

REM Run IPX over packet driver (also from simtel) - it finds 0x68.
C:\NET\IPXPD
REM And NETX (Novell standard)
C:\NET\NETX

REM And PKTINT is required for QVT, but not needed if you have WinQVT/Winsock
C:\QVT\PKTINT

REM Load TCP/IP (packet driver interface) Kernel
C:\TCPOPEN\BIN\TCPIP_PD
```

PKTMUX, PKTDRV, PKTINT and IPXPD are not included with the Softerm TCP/IP Kernel package.

NOTES:

1. Softerm TCP/IP Kernel searches down to find a packet driver, and hits the one at 0x6a.

All normal packet driver applications search UP from 0x60)

2. Since WinQVT/Net requires a specific address, give it 0x69.
3. Note that WINPKT is NOT needed - since PKTMUX does all that.

ODI Driver

If you install Softerm TCP/IP to coexist with Netware Lite or version 3.1x using ODI driver, edit the existing Netware's NET.CFG and insert the following lines under the "Link driver" heading:

```
Link driver 3c503

; insert the following lines to support Softerm TCP/IP Kernel

        FRAME   ETHERNET_II
        PROTOCOL IP 800   ETHERNET_II
        PROTOCOL ARP 806   ETHERNET_II
```

3c503 is given as an example - it should be replaced with the name of the driver of the adapter in-use.

NDIS Driver

TCPIP_ND supports NDIS drivers conforming to NDIS specification version 2.01.   The protocol manager driver supporting NDIS version 2.01 is now included with TCP/IP Kernel release 1.21 or later. Note that the release 2.01 protocol manager supports dynamic binding and unbinding of the protocol drivers (TCP/IP or whatever).   This feature allows the user to dynamically load/unload the protocol driver without having to reboot the system.

There are a number of new utility programs to support the new Protocol Manager, version 2.01.   They are:

UBTCP.EXE
UNBIND.EXE
READPRO.EXE
BINDSTAT.EXE
Example usage

NOTE: THE ABOVE COMMANDS DO NOT WORK WITH THE OLD PROTOCOL MANAGER (version 1.0x).

UBTCP.EXE

This command intructs the Protocol Manager to unbind a protocol module from the protocol binding structure. It must be invoked before the TCP/IP protocol kernel can be unloaded from memory (by the TCPIP_ND -D command).

Syntax: UBTCP [-V] [-N <module-name>]

Where:-V specifies verbose mode
-N <module-name> specifies the name of the module to be unbinded. By default, UBTCP unbinds TCPIP module.

UNBIND.EXE

This command is developed by 3Com.   It unbinds protocol modules in the reverse order they were bound; i.e. last-in, first-out. UNBIND does not allow selective unbinding of a named protocol module.

Comment: does not work with Microsoft's protocol manager v2.1.

READPRO.EXE

This command updates the Protocol Manager's memory image of the PROTOCOL.INI file. The PROTOCOL.INI file is assumed to be in the same location that the Protocol Manager read when it was loaded. The command can only be used when the Protocol Manager is in dynamic loading operation mode (i.e. DYNAMIC=YES). It should be invoked before (re)loading of any dynamic modules.

Comment: does not work with Microsoft's protocol manager v2.1.

BINDSTAT.EXE

This command traverses the protocol binding structure maintained by the Protocol Manager and prints binding information at the console. It is only useful when there are several protocol modules and/or NDIS drivers loaded in the system.

Example Usage

The following command sequences will load and then unload the TCP/IP kernel from the MAC NDIS driver, and then later reload it.

- Load the TCP/IP kernel, initialize and bind it to the MAC NDIS driver the first time:

```
C> TCPIP_ND
C> IFCONFIG ND0
C> NETBIND
```

- Unbind the TCP/IP kernel from the MAC driver and then unload the TCP/IP kernel (remove its memory image):

```
C> UBTCP
C> TCPIP_ND -D
```

- Reload the TCP/IP kernel:

```
C> READPRO(does not work if Microsoft's protocol manager is used)
C> TCPIP_ND
C> IFCONFIG ND0
C> NETBIND
```

Updates to the Manual

The following documents the new TCP/IP Kernel changes since the previous releases and the new commands/features that are not documented in the Softerm TCP/IP Kernel user's guide.

Using TCPIP_ND.EXE with Microsoft's LAN Manager (version 1.2 or later)

When using Lanera NDIS kernel in coexistence with Microsoft's LAN Manager, follow these guidelines:

1. Install LAN Manager first

2. Modify LAN Manager's PROTOCOL.INI file to include:

```
[TCPIP]
        DriverName=TCPIP$
        BINDINGS="ELNK3_NIF"

[ELNK3_NIF]
        ::::::::
        ::::::::
```

   (ELNK3_NIF is given as an example, replace it with appropriate module name for the adapter in-use.)

3. Modify TCPOpen's NETINFO file to contain the following entry:

```
MODNAME=ELNK3_NIF
```

   ELNK3_NIF is the module name for 3Com Etherlink III adapter (exactly as specified in the protocol.ini).

   NOTE: If you have the following error message when loading the TCPIP_ND kernel, TCPIP_ND0304: Get PMINFO failed (0x....) this step will correct this problem.

4. Always load and start NETBEUI first. If NETBEUI is loaded after TCPIP_ND, you can not unload TCPIP_ND at will.

5. Load and start TCPIP_ND kernel.

```
C> NETBIND  (to enable the dynamic loading/unloading of TCPIP_ND)
C> TCPIP_ND
C> IFCONFIG ND0
C> NETBIND
```

   Later, if you want to unload TCPIP_ND, do:

```
C> UBTCP
C> TCPIP_ND -D
```

   You can then reload the TCPIP_ND kernel again if desired.

NOTE: The first NETBIND must be executed before any protocol stack is loaded in order for the protocol stacks to be dynamiccally loaded and unloaded at a later time.

SLIP Kernel

- TCPIP_SL.EXE (SLIP) now supports COM1 to COM4.

- If you experience network buffer shortage, you can either increase the "Network Buffer Space" or reduce the "TCP Send Window Size" parameter using the TCPSETUP program. This problem occurs because there are too many messages waiting, at the interface send queue, to be transmitted. The pending messages consume a lot of network buffers.

- Reducing the TCP Window Size helps to reduce the number of messages waiting for transmission on the wire, if the application is TCP-based. If the application is UDP-based, it must control the amount of data sent to the interface in order to avoid long interface send queue.

  Use "NETSTAT -Q" to monitor the queues' length.
  Use "NETSTAT -M" to monitor network buffer usage.

- SLCONFIG had been modified to support only three commands instead of six commands as before. The ATTACH, DETACH, COMMAND commands are dropped. SLCONFIG now can only be used to SET a port configuration (baurate, etc ...), DIAL a remote and HANGUP or disconnect a call.

- A Windows-based SLCONFIG is now available (WSCM.EXE).

- Support for auto-login/dynamic IP address assignment

  Support for AUTO-LOGIN for IP routers is now available. To support this feature, SLCONFIG performs a series of batch-mode commands as described below:

  The WAITFOR/SEND combination can be repeated

  The batch commands are stored in the {INSTALL}\SETUP\PHONES database. The command(s) should be added RIGHT AFTER the name-phone number mapping entry of the remote destination. Valid syntax for these commands are:

  SEND <string>
  Sends a string to remote. This command deals with the situation where the remote requires some triggering characters before sending the login prompt to the local station. For example, some IP routers expect to receive a CR ('\r') before prompting for user login.

  WAITFOR <string1> SEND <string2>
  The WAITFOR keyword instructs SLCONFIG to scan the modem input stream for the string <string1> before the SEND can take place. If such a string is not detected in the input stream, SLCONFIG waits forever until interrupted by the [Ctrl]+[Break] key.

  SEND instructs SLCONFIG to send <string2> to remote immediately upon receiving <string1>. Note that SEND does not append any character to <string2> when sending, so if the remote expects to see CR ('\r') or LF ('\n') as the string terminator, you must explicitly specify that character ([cr] for CR, [lf] for LF) in <string2>.as many times as necessary. Apparently, the WAITFOR/SEND mechanism is suitable to deal with user authentication procedure (i.e. login/password prompts).

WAITFOR_IP <string>
WAITFOR_IP requests SLCONFIG to look for a dot-notation IP address immediately after the leading <string> in the modem input stream.   Once a valid IP address is detected, SLCONFIG will automatically set the local IP address (of the loaded SLIP kernel) to that of picked-up from the modem input stream.

WAITFOR_NETMASK <string>
WAITFOR_NETMASK requests SLCONFIG to look for a dot-notation IP address immediately after the leading <string> in the modem input stream.   Once a valid IP address is detected, SLCONFIG will automatically set the local IP address (of the loaded SLIP kernel) to that of picked-up from the modem input stream.

WAITFOR_GATEWAY <string>
WAITFOR_GATEWAY requests SLCONFIG to look for a dot-notation IP address immediately after the leading <string> in the modem input stream. Once a valid IP address is detected, SLCONFIG will automatically set the default gateway's IP address (of the loaded SLIP kernel) to that of picked-up from the modem input stream.

SLIPMODE
Puts the TCP/IP kernel in SLIP mode (after connected or done with other batch command execution) and terminates SLCONFIG. SLIPMODE is the last batch command to execute. By default, SLCONFIG goes into INTERACTIVE mode after connected or finishing batch command execution, if any.

****
Note that <stringX> consists of printable characters and [cr] for '\r' and [lf] for '\n
Note that the above address settings effect only the kernel setting, not the settings in the NETINFO file.   This mechanism is provided to handle dynamic IP address assigment implemented by many IP routers.

To verify whether the assigned IP address, Netmask, and default gateway's IP address are in effect, use the IFCONFIG command as follows:

```
C> IFCONFIG SL0
```

which displays the kernel's IP address settings.

'. NO SPACE OR TAB IS ALLOWED IN THE LEADING STRING.
****

Sample {INSTALL}\SETUP\PHONES database file:

```
.....
remote1   234-5678            # destination dial-up IP router
SEND [cr][cr]
WAITFOR login: SEND steve[cr]
WAITFOR password: SEND steve_secret[cr]
```

```
     remote2 345-6789              # remote2 will assign IP address
     WAITFOR login: SEND steve[cr]
     WAITFOR password: SEND steve_secret[cr]
     WAITFOR_IP Address
     .....

NOTES: *****************************************************************
 ______________________________________________________________________
|
|    1. If you specify the phone number (234-5678) instead of name (remote1)
|       on the SLCONFIG command line when dialing, the batch commands are
|       ignored. Once the phone line is connected, you interact with the
|       IP router's command session. The interactive mode terminates under
|       two conditions:
|
|       a. IP router goes into SLIP mode, which eventually forces the
|          SLIP kernel to enter SLIP mode (when receiving a FRAME_END
|          (0xC0) character from remote).
|
|       b. or the [Ctrl]+[Break] key is pressed, which causes SLCONFIG to
|          prompt you for actions.
|
|    2. SLCONFIG works only on the SLIP kernel (TCPIP_SL.EXE) but not
|       with the Packet, NDIS or ODI kernel.
|
|    3. A new line that does not begin with one of the above keyword signals
|       the end of batch command mode. SLCONFIG will put you into
|       interactive mode after executing all the batch commands unless a
|       SLIPMODE command was specified.
|
|    4. The SLIP kernel is in RAW mode after the phone connection completes,
|       and then goes into the SLIP mode only when one of the following
|       conditions occurs:
|
|       a. A SLIPMODE command was found while in batch mode
|       b. Put it into SLIP mode by responding to SLCONFIG's prompt
|
|       Once the SLIP kernel is in SLIP mode, SLCONFIG ignores all the
|       remaining WAITFOR/SEND batch commands, if any.
|
|    5. While in batch or interactive mode, the user can break out of
|       SLCONFIG any time by hitting the [Ctrl]+[Break] key. When the
|       key combination is pressed, in either mode, you will be prompted
|       for whether you want to hangup the call or to put the kernel in
|       SLIP mode. If you choose to hangup, SLCONFIG hangs up the call
|       and terminates. Otherwise, it puts the SLIP kernel into SLIP mode
|       and also terminates.
|_________________________________________________________________-
```

- When using SLIP under Windows, make sure to exclude the COM port used by the SLIP kernel from Windows configuration by adding the following lines in Windows' SYSTEM.INI file:

  ```
  comXirq = -1
  comXbase = 0
  ```

  where X is the COM port ID (i.e. 1, 2, ...) for the port used by SLIP.

  If you fail to do this, Windows will take over the COM port on start-up and the SLIP kernel is no longer working.

Driver Deinstallation

- Packet Driver Deinstallation:

  The Packet Driver can be deinstalled by using the TERMIN command. You have to invoke TERMIN twice to actually remove the packet driver from memory. The first invocation is to unlink the Packet Driver from higher-layer protocol driver (TCP/IP kernel). The second invocation actually remove it from memory. For example:

  ```
  C> TERMIN -S <packet_int_no>
  C> TERMIN   <packet_int_no>
  ```

  If the TCP/IP kernel was already removed prior to the Packet Driver removal, you do not need to issue "TERMIN -S..." since the TCP/IP kernel already broke the linkage when it was removed.

- ODI Driver Deinstallation:

  ODI driver is deinstalled by specifying the -U option on the driver command line. For example:

  ```
  C> 3C503 -U
  ```

  will remove a previously-loaded 3C503 ODI driver.

  When deinstalling the ODI driver and TCP/IP kernel, remember to unload the drivers in the reverse-order of loading. For example, to load the ODI driver for 3Com's 3C503 adapter and TCP/IP kernel:

  ```
  C> LSL
  C> 3C503
  C> TCPIP_OD
  C> IFCONFIG OD0            ; to trigger driver binding process
  ```

  and to unload the above loaded drivers:

  ```
  C> TCPIP_OD -D
  C> 3C503 -U
  C> LSL
  ```

- NDIS Driver Deinstallation:

  NDIS driver can not be deinstalled since it is a DOS driver (unless you have a special utility that can deinstall a DOS driver)

  However, TCPSETUP now generates a batch file: RMTCP.BAT, which contains the necessary commands to remove the kernel. Try this batch file before you try the manual method as outlined above.

Monitor Network I/O Traffic

Included in this release is a new network statistics displaying program (namely, NETMON.EXE). The program is a full-screen menu-driven implementation of the traditional NETSTAT.EXE program. A nice thing about this program is that it periodically updates the screen with new statistics. Therefore, you can use it to monitor network I/O activities in the TCP/IP protocol stack.

Give it a try. You may find it useful in pin-pointing problems. It can be invoked at a DOS prompt or in Windows 3.X.

IP Address Assigment Using BOOTP Client

A BOOTP client (BOOTPC.EXE) is now included with the kernel package. This program should be ran RIGHT AFTER THE KERNEL IS LOADED. Its function is to contact a BOOTP server to request for an IP address. Once a BOOTP server replies with an IP address, BOOTPC will set the local IP address to the assigned IP address.

You could append BOOTPC statement into the LOADTCP.BAT file to execute it right after the kernel is loaded.

```
WARNING: ---------------------------------------------------------------------------
|
|   1. You should set the initial local IP address to a network address with
|       the network part to reflect your network address and the host part of
|       all 0's for BOOTPC to work properly.
|
|   2. Subnet and default gateway's IP address information must be locally
|       configured.
|__________________________________________________________________________
```

The BOOTPC command syntax is:

```
BOOTPC [-T <time-out>] [-R]
```

where:

```
    -T <time-out>  specifies the wait-for-reply time-out. Def: 2 secs.
    -R             specifies Token-Ring hardware type. Def: Ethernet.
```

PC Eudora: Windows-based E-mail package (SMTP/POP)

If you are running PC-Eudora package, remember to set the field "TCP Send Window Size" in the SITE menu if TCPSETUP to at least 16 (4 Kbytes). Otherwise, large mails can not be sent.

X-terminal emulators

If you are running X-terminal emulator or any applications that open several connections and all these connections may be active at the same time, it is a good idea to increase the "Network Buffer Space" field in the SITE menu of TCPSETUP to at least 20 Kbytes. This setting avoids the kernel from running out of network buffers when the traffic is heavy.

Oracle Database

When installing Oracle database (Windows Sockets version), you will be prompted to select TCP/IP interface from a list of vendors.   If Softerm TCP/IP is not yet in the list of vendors, you can select "NetManage" entry for instead.   This selection instructs Oracle to use NetManage Windows Sockets interface which Softerm TCP/IP also supports.

Known Problems

- Executing the WINSTART.BAT file during MS Windows 3.1 startup could cause TCP/IP kernel to not properly functioning after exiting from Windows. This problem occurs even if WINSTART.BAT is empty (no thing to execute).

- Running SLCONFIG in Windows DOS box causes receive data to be dropped.

NOTES / WARNINGS

- Running the same serialized copy of TCP/IP kernel on more than one PC station results the following message on the screen:

  TCPIP_XX: SAME SERIAL NUMBER with station: xx:xx:xx:xx:xx:xx.

  xx:xx:xx:xx:xx:xx is hardware address of the station on the network that runs the same serialized copy of the TCP/IP kernel.

  The kernel is disabled after printing the above message.

- The TCP/IP kernel can be loaded high provided there is enough space to load it. Typically, it requires about 90 Kbytes of free continuous space to be loadable in high-memory (even though it uses less than that amount of memory after loaded). The kernels have been successfully tested with QEMM and DOS 6.0 MemMaker.

Softerm TCP/IP NFS Manual

TABLE OF CONTENTS

Preface

Chapter 1. - INTRODUCTION

Chapter 2. - INSTALLATION

Chapter 3. - ADMINISTRATION

ERROR MESSAGES

TROUBLE SHOOTING

Requirements

Softerm TCP/IP Kernel is bundled with Softerm Modular for Windows and is also available separately as an optional package. Softerm TCP/IP NFS requires the Softerm TCP/IP Kernel product to be installed to provide the low-level networking services.

You will also need TCP/IP and NFS server software on the server machine(s).

Related Manuals

This document supplements the Softerm TCP/IP Kernel Installation Guide. The TCP/IP Kernel Guide describes how to install and setup Softerm TCP/IP Kernel to provide support for TCP/IP NFS operations.

Conventions Used in This Guide

The following conventions are used throughout the guide.

Keys: The table below shows the symbols that represent the keys on your keyboard.

Symbol	Key Represented
[Esc]	Escape
[Alt]	Alternate
[Ctrl]	Control
[Backspace]	Backspace
[Space]	Space bar
[F1] - [Fx]	Function keys F1 through Fx
[Return] or **[Enter]**	Return or Enter key

These symbols are printed in boldface whenever they are mentioned in the text.

Key Combinations: If two or more keys are to be pressed simultaneously, the key symbols are linked with a "+" sign. For example, the following key combination resets a workstation:

[Ctrl]+[Alt]+[Del]

Commonly-Used Terminology

Remote refers to a computer attached to the network via appropriate cabling system.

Local refers to your own computer system.

Host refers to a computer system, may either be remote or local.

File System refers to a hierarchical (tree structure) organization of files within a computer system.

Mounting the mapping of a remote file system to a local drive so that files on the remote file system can be accessed as if they were on the local drive.

Getting Started

Softerm TCP/IP NFS is a PC/DOS implementation of SUN Microsystems NFS (Network File System) specification for transparent access of remote file systems. Using TCP/IP NFS in conjunction with Softerm TCP/IP Kernel, a user can share files on one or more NFS-based network file server with other users on the network.

For additional information, select:

- Requirements
- Related Manuals
- Conventions Used in his Guide
- Commonly-Used Terminology

Overview

Softerm TCP/IP NFS is an implementation of SUN Microsystems NFS specification for the PC/DOS environment. It allows PC users to transparently share files in a heterogeneous TCP/IP network environment.   Remote file systems can be mounted to local drives. All well-behaved DOS applications and multi-tasking programs that use DOS file access services such as DESQview or Windows 3.X can access the remote files as if they were on a local drive.   Softerm TCP/IP NFS can coexist with other network operating systems such as Novell Netware, Microsoft LAN Manager, etc. ... This capability makes TCP/IP NFS an ideal tool for merging and managing dissimilar file systems in a diversified network environment.   Since the NFS protocol is available on most UNIX implementations (as well as in other operating environments), the TCP/IP NFS product primarily deals with the issues of file sharing between DOS and UNIX machines.   Furthermore, TCP/IP NFS also works with other operating environments where an NFS server and TCP/IP protocol suite are available.

Currently, TCP/IP NFS implements the client portion of the SUN Microsystems NFS protocol which allows a PC to behave as an NFS client machine accessing remote files on NFS file servers. The advantage of using NFS to share network files is that it allows the mapping of remote file systems to local drives.   This capability enables local applications to access a remote file system as if it were a local file system. Therefore, file access is transparent to the users.   This means minimal training is required to use TCP/IP NFS for accessing and managing remote files, as opposed to other file transfer programs such as FTP.

While NFS has proven to be a very useful tool in a high-speed network environment, the excessive handshaking procedures between the client and the server in most file access operations make it impractical to operate on a slow link such as serial line.

Components

The Softerm TCP/IP NFS package consists of the following software components:

NFS Administrative Tools
NFS Kernel
NFS Utilities

NFS Administration Tools

A full-screen menu-driven application that allows the user to:

- setup the Softerm TCP/IP NFS environment,
- administer remote file systems,
- display NFS Kernel and file access statistics

A command-line version (NFS command) which offers similar functionality is also included for users who want to start up TCP/IP NFS using a DOS batch file.

NFS Kernel

The component that stays resident in DOS memory and is responsible for mapping all local file operations on a mapped remote file system to corresponding NFS-conformed requests to the associated NFS server. The NFS Kernel supports the mapping of up to 15 remote file systems on several different remote hosts simultaneously. A transaction user ID can be dynamically configured for added security. Read/Write transaction block sizes and NFS Kernel buffer space can be optimized for the best performance and memory space usage.

NFS Utilities

A set of utility programs that allows users to convert text file formats from DOS to UNIX and vice versa. The utilities also allow users to deal with UNIX-specific file attributes such as file access permissions.

Installation

Softerm TCP/IP Kernel is available with the Softerm Modular package or as an optional package. Softerm TCP/IP NFS is also available as an optional package. Softerm TCP/IP NFS is distributed on the diskette labeled "NFS". Before you install this diskette, **make sure you have already installed Softerm TCP/IP Kernel diskette**, since TCP/IP NFS depends on the TCP/IP Kernel to provide network services.

To install Softerm TCP/IP NFS, insert the diskette labeled "NFS" into the floppy drive, let's say A:, then enter:

```
C> A:INSTALL [Enter]
```

The INSTALL program then guides you through the installation procedure.

Installed Files

The following files are installed on your system after a successful installation of the diskette. Assuming your selected target drive and directory is C:\TCPOPEN, you should have the following files:

C:\TCPOPEN\BIN\NFS.EXE	Command-line NFS administration command
C:\TCPOPEN\BIN\NFSADMIN.EXE	Menu-driven NFS administration command
C:\TCPOPEN\BIN\NFSCLT.EXE	NFS Kernel
C:\TCPOPEN\BIN\NCHMOD.EXE	File-attribute manipulation program
C:\TCPOPEN\BIN\NLS.EXE	UNIX-style file listing command
C:\TCPOPEN\BIN\D2U.EXE	Convert DOS text file format to that of UNIX
C:\TCPOPEN\BIN\U2D.EXE	Convert UNIX text file format to that of DOS
C:\TCPOPEN\BIN\LOADNFS.BAT	Batch file to load and setup NFS Kernel
C:\TCPOPEN\BIN\RMNFS.BAT	Batch file to remove NFS Kernel from memory
C:\TCPOPEN\BIN\NFSADMIN.ICO	NFSADMIN icon for Windows 3.X
C:\TCPOPEN\ETC\PASSWD	Password database used for mapping a user ID to a user name and user authentication

Related Database Files

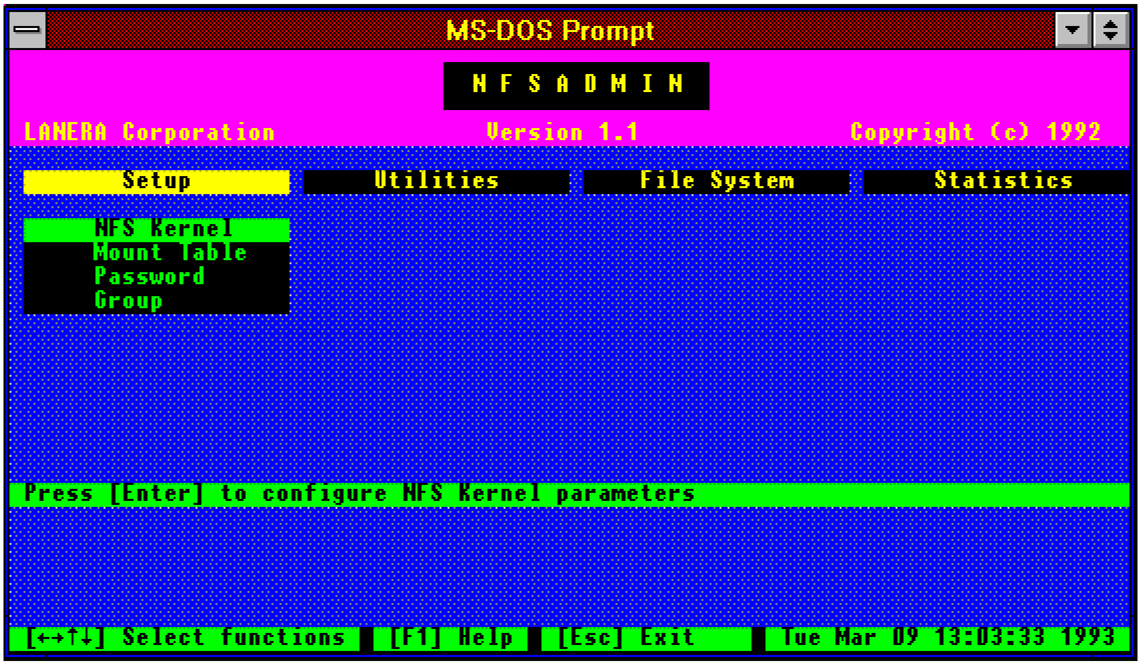
In addition to the above executable files, NFSADMIN will later create and maintain the following database files:

C:\TCPOPEN\ETC\GROUP	Group-ID-to-Group-Name mapping table
C:\TCPOPEN\ETC\MNTTAB	Local drive/Remote file system mapping table

Overview

Before proceeding with setting up Softerm TCP/IP NFS, make sure you have already set up TCP/IP Kernel by using TCPSETUP program and verify that it is functioning properly. Refer to the TCP/IP NFS User's Guide for information on how to set up TCP/IP NFS using the TCPSETUP program.

TCP/IP NFS can be set up by manually editing the NFS database files or by using NFSADMIN program. NFSADMIN is a screen-oriented menu-driven application that allows users to set up TCP/IP NFS operational parameters, to manage the remote file systems, to display NFS Kernel statistics, and to perform other administrative functions. NFSADMIN is a menu-driven program. It is a convenient way to quickly view NFS configuration data and NFS Kernel status.



N F S A D M I N

LANERA Corporation Version 1.1 Copyright (c) 1992

Press [Enter] to configure NFS Kernel parameters

Tue Mar 09 13:03:33 1993

Figure 1. Sample NFSADMIN screen

To invoke NFSADMIN, you can either click on the NFSADMIN icon if you are in Windows 3.X or type the following command:

```
C> NFSADMIN [Enter]
```

at a DOS prompt. The NFSADMIN menu screen as illustrated in figure 1 is then displayed on your monitor. As you can see, NFSADMIN provides four menus as described below:

SETUP

This menu sets up the following information:

- NFS Kernel parameters
- MNTTAB database
- PASSWD database
- GROUP database

UTILITIES

This menu provides some useful utilities such as verifying connectivity with a remote NFS server, deinstalling the NFS Kernel, setting the NFS Kernel's default UMASK on the server, and logging into an NFS server.

FILE SYSTEM

This menu displays the currently-mounted file system(s), and manages the mapping of remote file systems.

STATISTICS

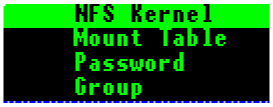
This menu displays NFS Kernel statistics. Technical users can use this statistical information to diagnose problems if they occur.

Setup

The following items need to be set up before Softerm TCP/IP NFS can be started:

- NFS Kernel parameters
- MNTTAB mount table
- PASSWD database
- GROUP database

To set up the above information, you select the **SETUP** menu on the NFSADMIN screen by high-lighting the SETUP menu.   You will see the following sub-menus:



NFS Kernel Parameters

The NFS Kernel parameters are maintained in the {INSTALL}\SETUP\NETINFO file. The NFS-specific parameters are always prefixed with "NFS_". Some NFS commands use these parameters to determine how to communicate with the NFS Kernel. The following are the tunable kernel parameters:

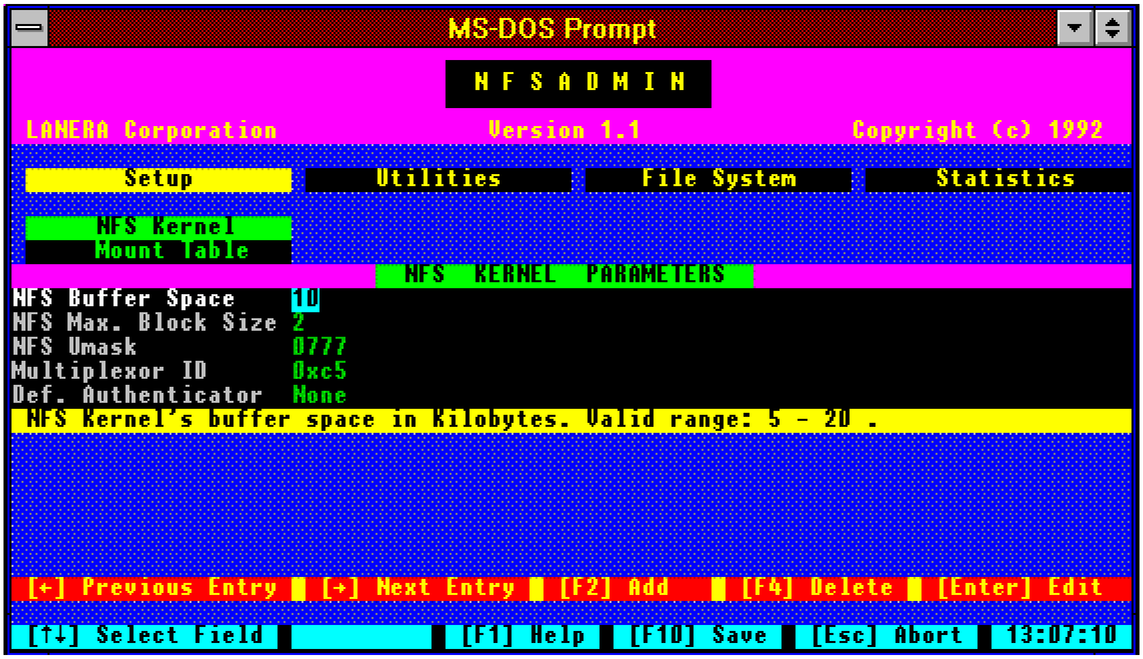
Buffer Space
Block Size
Default UMASK
Multiplexor ID
Default Authenticator

Configure Kernel Parameters

To configure NFS Kernel parameters, select the **Kernel** sub-menu by high-lighting it and then press the [Enter] key. The window, as illustrated in figure 2 below, then appears on the screen. It displays the current setting of NFS Kernel parameters. These parameters must be properly set before loading the kernel. You can select to edit any of these parameters by high-lighting the field to be edited and then press the [Enter] key to start editing.

Save Kernel Parameter Settings

After setting the NFS Kernel parameters to the desired values, press [F10] to save these parameters in the NETINFO file. At this point, NFSADMIN also generates new LOADNFS.BAT and RMNFS.BAT files based on the newly-configured parameters. These batch files reside under {INSTALL}\BIN directory and are used later to load and to unload the NFS Kernel, respectively.



```
NFS Buffer Space     10
NFS Max. Block Size  2
NFS Umask            0777
Multiplexor ID       0xc5
Def. Authenticator   None
 NFS Kernel's buffer space in Kilobytes. Valid range: 5 - 20 .
```

Figure 2. Sample NFS Kernel Parameters data-entry screen

NFS Buffer Space

Specifies the amount of memory to be used for NFS Kernel buffers.   The default NFS buffer space is 10 KBytes.   The valid range for this parameter is from 10 to 40, inclusive. This parameter is saved as NFS_BUFFER in the NETINFO file.

NFS Max. Block Size

Specifies the maximum NFS read/write block size in KBytes.   The bigger the block size, the better the performance is.   This is simply because it takes less read/write operations to complete a file access transaction.   However, the NFS Kernel may require more buffer space to accommodate larger block sizes. The default value for this parameter is 2 KBytes.   If it is set to 4, you should increase NFS Buffer Space parameter to 15.   This parameter ranges between 1 and 8, inclusive and is saved as NFS_BLOCKSIZE in the NETINFO file.

UMASK

Is the default file attribute to be set for a network file when it is created. UMASK is maintained by the NFS Kernel and can be changed by either the command-line NFS command or NFSADMIN. The coding for this parameter follows the same convention as that of on UNIX system. It is specified in *octal* with the bit assignments as follows:

	Owner	**Group**	**Other**
	r w x	r w x	r w x
	1 1 1	1 1 1	1 1 1
Example: 0640 →	1 1 0	1 0 0	0 0 0

r → read permission
w → write permission
x → execute permission

Bits set to 1 means to allow the permission and 0 means to disallow. For example, if UMASK is set to 0640, the created file is then readable and writable by the owner; however, it is only readable by users who are members of the group assigned to the file. All other users have no access to the file at all. The default UMASK is 0644. This parameter is saved as NFS_UMASK in the NETINFO file.

See the "User Authentication" section for more explaination on file attribute.

Multiplexor ID

Specifies the INT 2F multiplex ID that the NFS Kernel uses as its ID for responding to the installation check command. The default value for this parameter is 0xC5. You may have to assign another ID in the range of 0xC0 and 0xCF if there is another program running on your system using the same multiplexor ID as that specified by this parameter.   This parameter is saved as NFS_MULTIPLEXOR in the NETINFO file.

Default Authenticator

Specifies the hostname or IP address of the server that, by default, is contacted for authentication when the local user executes the login. The authentication server must have *pcnfsd* running. This parameter is saved as NFS_AUTH in the NETINFO file.

Mount Table (MNTTAB)

The mount table contains the local drive/remote file system mapping entries. The entries are specified in the format:

Local-Drive: Remote-Host Remote-Pathname Options

Local-Drive is a one letter standard DOS drive ID followed by a colon (:). DOS can recognize any drive from A to Z or to the limit specified by the LASTDRIVE configuration parameter in the CONFIG.SYS file.

Remote-Host is the remote NFS server hostname or IP address in dot notation. Hostnames can be up to 60 characters in length and consist of printable characters.

Remote-Pathname is the pathname of the directory on the NFS server system to be mounted (mapped) to the Local-Drive. The pathname specification must be specified exactly as can be understood by the remote system. TCP/IP NFS does not attempt to do any conversion on Remote-Pathname. The maximum length for this parameter is 60 characters.

Options specify a number of options that can be set up for these entries. The following are available options:

AM (Auto-Mount) means the associated entry will be selected for mounting whenever the "NFS MOUNTAM" command is invoked.

RO (Read-Only) means the associated file system is for read-only access. Any attempt to write or modify the file system will fail.

LK (Locking Enable) means Softerm TCP/IP Kernel will support file and record locking by cooperating with the network lock manager (on UNIX machines, this program is *lockd*) on the remote NFS server. File locking provides protection for network files against multiple concurrent accesses that may jeopadize data integrity.

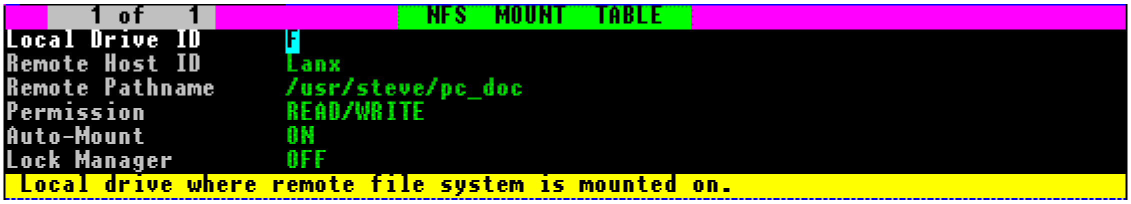
Note that the fields must be separated by space(s) or tab(s), while the options are separated from each other by comma (,). Lines begin with the pound sign (#) are comment lines and ignored by NFSADMIN. ***Invalid format lines are simply ignored without any warning.*** A sample mount table entry is as follows:

```
E: lanx:/usr/steve/pc_files RO,AM
```

In the above example, *E:* is the local PC drive, *lanx* is the remote host name, */usr/steve/pc_files* is a valid directory on *lanx* and RO,AM indicate the remote file system is for read-only and auto-mount, respectively.

Set up the Mount Table

To set up the mount table, {INSTALL}\ETC\MNTTAB, you first high-light the **Mount Table** sub-menu in the SETUP menu and then press [Enter]. The following window then appears on the screen:



```
 1 of   1            NFS  MOUNT  TABLE
Local Drive ID        F
Remote Host ID        Lanx
Remote Pathname       /usr/steve/pc_doc
Permission            READ/WRITE
Auto-Mount            ON
Lock Manager          OFF
 Local drive where remote file system is mounted on.
```

Figure 3. Sample mount table data-entry screen

The window in figure 3 displays one entry of the mount table at a time. You can use the [¬] or the [→] key to go back to the previous entry or advance to the next entry, respectively. The numbers in the upper left-hand corner indicate the order of the current entry and the total number of *valid* entries in the table.

Edit the Mount Table Entry

The window in figure 3 presents the fields of a mount table entry and allows you to edit or select your desired configuration. To edit a field, you simply high-light the field by using the [] and [↓] keys and then press the [Enter] key to begin editing. The last three (3) fields are toggle fields, you can use the [Enter] key to select their settings.

Save the Edited Mount Entry

Once you have done with editing the current entry, you need to save it by pressing the [F10] key. The mount table is then updated with the new entry. The old mount table is backed up to the file {INSTALL}\ETC\MNTTAB.BAK. You need to do this step on every entry you edited since NFSADMIN does not buffer these entries. Advancing to another entry without saving (by pressing [F10] key) will result in losing the edited information.

Add a New Entry

When you first install Softerm TCP/IP NFS, the mount table is empty, you will have to add entries into it. To do this, press the [F2] key. NFSADMIN will present you with a blank entry with some default settings. You can then edit the entry to your desired settings and save it as described above.

Repeat this process if you want to add more entries to the mount table. You can invoke this function any time a new entry is needed.

Delete an Existing Entry

You can delete an existing entry by using the [←] or the [→] key to position to the entry to be deleted, and then press the [F4] key to delete the entry.

User Information Databases

The **SETUP** menu also allows you to create and maintain user and group information databases, namely PASSWD and GROUP. The PASSWD database contains information about the local user's login accounts on various remote hosts. The GROUP file contains the user's group information. The NLS command uses these databases to map user-ID and group-ID to names for displaying UNIX file information. These databases reside under {INSTALL}\ETC directory. See "User Authentication" later in this chapter for further discussion on how user ID and group ID are used.

For additional information, choose:

PASSWD
GROUP
Setting up the Databases

User Login Database (PASSWD)

The user login database contains login entries that allows the NLS command to map a user-ID on a UNIX file to a user name. The user login database's entry has the following format:

Login-ID : Password : User-ID : Comment

Login-ID is the user login ID as assigned on one or more remote NFS server. For example, if your login name (ID) on a remote UNIX server is *steve,* then you would use the same name (ID) here. *Login-ID* can be up to 40 characters. An entry with a *Login-ID* length that exceeds the maximum allowance will be discarded by NFSADMIN.

Password is the encrypted password for the above login ID. The password must be the same as that of on the remote server since it is used to authenticate the user on the server's side. Password can be up to 15 characters. An entry with a Password length that exceeds the maximum allowance will be discarded by NFSADMIN.

User-ID is the same user ID associated with your login account on the remote NFS server.

Comment is a string of up to sixty (60) characters.

User Group Database (GROUP)

The group database contains a user's group entries that allows the NLS command to map a group-ID on a UNIX file to a group name.   This file can be constructed by simply copying the */etc/group* file from the UNIX server and modifying it to conform to the following entry format:

Group-Name : Group-ID : Comment

Group-Name is the unique name of a group.

Group-ID is the group ID associated with the above group name.

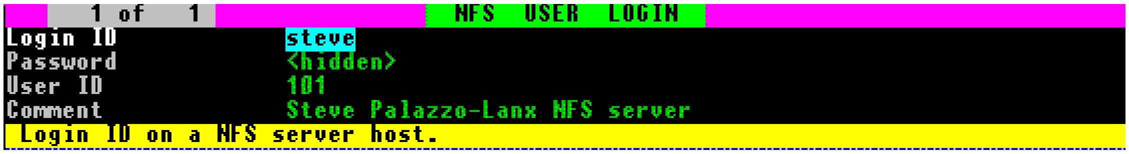
The group ID and group name association must be the same as that of on the NFS file server so that the NLS command can display correct information.

Comment is a string of up to sixty (60) characters.

Setup The PASSWD and GROUP Databases

The PASSWD and the GROUP databases are set up very much the same way. To set up for the PASSWD database, you first high-light the **Password** sub-menu under the **SETUP** menu and then press the **[Enter]** key. A data-entry window as illustrated in figure 4 then appears on the screen.

The user login database is displayed one entry of a time. The numbers in the upper left-hand corner indicate the order of the current entry and the total number of ***valid*** entries in the database. Use the [←] and the [→] keys to move from one entry to another in the database.



```
    1 of    1                NFS  USER  LOGIN
Login ID              steve
Password              <hidden>
User ID               101
Comment               Steve Palazzo-Lanx NFS server
 Login ID on a NFS server host.
```

Figure 4. Sample User Login ID data-entry screen

Edit User Login Information

To edit an entry, you must first position to that entry by using the [←] and [→] keys as mentioned above, then use the [] and [↓] keys to position to the field to be edited, and then press the [Enter] key to begin editing. Pressing the [Enter] key again will allow you exit from the editing mode.

The *Password* field is particularly worth noting here since your input will not be echoed to the screen while you are entering the password for the entry. When displaying an entry, the *Password* field displays the word *<hidden>* if there is a password specified. Otherwise, the field is blanked out. The password is encrypted when written to the PASSWD database.

WARNING

If a password is specified, NFS LOGIN function will use it for user authentication. While this is convenient since you do not have to enter the password every time logging in an NFS server, it creates a security problem that allows other user to log in the server (from this PC) without having to enter the password. It is up to your judgement to decide whether you should enter a password here.

Note that Login ID field must be unique for every entry in the PASSWD database (or the Group Name field for the GROUP database). An attempt to add an entry of which Login ID is already used by another entry will result in an error message from NFSADMIN.

Save User Login Entry

Once you have finished editing the current entry, you need to save it by pressing the [F10] key. If a password already exists for the entry, you will be prompted to enter the password before the changes can take place. The PASSWD database is then updated with the new data. The old PASSWD database is backed up to the PASSWD.BAK. You need to do this step on every entry you edit since NFSADMIN does not buffer these entries. Advancing to another entry without saving (pressing [F10]) first will result in losing the edited information.

Add a New Entry

When you first install Softerm TCP/IP NFS, the user login database is empty, you will have to add entries into it.   To do this, press the [F2] key.   NFSADMIN will present you with a blank entry.   You can then edit the entry and then save it by pressing the [F10] key.

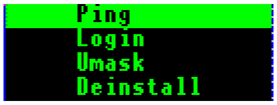
Repeating the above process if you want to add more entries to the user login database.   You can invoke the add function any time a new entry is needed.

Delete an Existing Entry

You can delete an existing entry by using the [←] or the [→] key to position to the entry to be deleted, and then press the [F4] key to delete the entry.

Utilities

NFSADMIN has some utilities that allow the user to change the kernel settings without the need to reload the Kernel such as changing the Kernel's default UMASK or changing the kernel's user information. In addition, a simple connectivity verification tool is also provided and can be used to conveniently verify the communication path between the NFS client (Kernel) and a specified NFS server. To use these utilities, you select the UTILITIES menu by high-lighting it, you immediately see the following sub-menu under UTILITIES menu heading:



Connectivity Verification (PING)

You can verify the communication path between your NFS client and an NFS server by using the **Ping** function. Select this function by high-lighting it and then press the [Enter] key. The following prompt appears on the bottom line of the screen:

Server Host ID: _

You response to the prompt with a hostname or IP address (dot notation) of the server that you want to verify (followed by the [Enter] key). If the communication path is operational, NFSADMIN will eventually respond with the following message:

lanx is alive

In this example, **lanx** is the name of the server host that you want to query.

User Authentication (LOGIN)

To protect individual's files in a multi-user environment, the UNIX operating system defines several levels of file ownership and access permission to every file created on the system.

For more information, choose:

- NFS File Ownership
- NFS Access Permissions
- NFS Authentication

NFS File Ownership

UNIX uses the user-ID and the group-ID to identify the owner of a file. By default, every file is assigned the user-ID and group-ID of the user who creates the file. The ownership can be changed later using the *chown* command on the UNIX system. User-ID and group-ID are usually 16-bit integer quantities and are assigned when a user is added to the system. This information, along with user's login ID and user's password, is maintained in the */etc/passwd* file on the UNIX system.

NFS Access Permission

To increase the flexibility in file sharing and, at the same time, to provide sufficient security protection for every user in a multi-user environment, UNIX defines three classes of users: owner, group and other. *Owner* is the user who creates the file, *group* is the group of users that the owner usually belongs to, and *other* is defined as any other user who does not belong to the file's group. For every user class, UNIX defines three types of access permissions: read, write and execute; which mean the file is readable, writable and executable respectively. File access permission is a part of the file attribute which will be fully explained in chapter 5. You can find this information in a directory listing command's output such as in the following example:

```
% ls -l file1
     -rwxrw-r--     steve       pcusers           50134     July 24   18:11   file1
%
```

In the above example, *steve* is the owner of the file, *pcusers* is the user group of the file, and the file attribute for *file1* is:

-rwxrw-r--

which is explained as follows:

- the first letter is used to indicate the type of a file which may be one of the following:

  - regular file (-),
  - directory (d),
  - character device file (c),
  - block special device (b),
  - named-pipe (or FIFO) file (p),
  - socket file (s),
  - or symbolic link file (l).

- the next three letters after the file type are used to code the owner's file access permissions in the order of: read, write, and execute. A dash in any position means the corresponding access permission is disabled.

- the next three letters after the owner's file access permissions are used to code the group's file access permissions in the order of: read, write, and execute. A dash in any position means the corresponding access permission is disabled.

- the last three letters are used to code the other's file access permissions in the order of: read, write, and execute. A dash in any position means the corresponding access permission is disabled.

When a file is accessed, the system will classify the accessing user into one of the above three user classes, based on the given user-ID, and then check for access permission against those set for that particular class in order to qualify an access permission.

When a file is created, its access permissions are set to that of the UMASK environment variable. File access permissions can be changed by the file's owner or the "superuser" by using the *chmod* command on the UNIX system. Softerm TCP/IP NFS provides an equivalent command, NCHMOD, which can be invoked from the PC side.

NFS Authentication

The UNIX operating system associates every file with a unique user-ID and a group-ID. Based on this association, it implements a file access qualification scheme to prevent users from accessing or possibly damaging one another's files. NFS client accesses remote files on a UNIX system by pretending to be a UNIX user and communicates the user information (user-ID/group-ID) with the NFS server on every file access request. The server uses the user information that it receives from an NFS client in a file access request to qualify file access privilege at the system level. By default, Softerm TCP/IP NFS uses the well-known *nobody* user-ID on file access requests. File access permissions for this user- ID on the server system are usually limited to the public file areas. However, if a PC user has a login account on the server system and wishes to access files permitted to his/her user-ID on that particular server, the user can request NFS Kernel to use his/her user-ID for all future NFS requests and thus, enables s/he to access remote files permitted to his/her user-ID.

NFS Kernel obtains this user-ID information via an authentication process normally known as *login*. NFS authentication is done by having the NFS client (Kernel) to contact an authentication process on the NFS server, which, in turn, looks up a centralized user information database, usually the user password file, on the remote system for the requested user. If the look-up is successful and the given password matched that stored in the database, the server passes the user information back to the NFS client. Once the NFS Kernel (client) has the new user information, it uses this information for all future NFS requests that require user information to be specified. A well-known authentication process is *pcnfsd* which is available on many UNIX systems where NFS is available such as System V Release 4 or SUN OS. Softerm TCP/IP NFS works with *pcnfsd* to provide authentication capability. The authentication procedure is illustrated in figure 4. Assuming the server is a UNIX system.

```
NFS Client                                        NFS Server

User name
& Password --------------------------------> Lookup /etc/passwd database

                         <-------------------------------- If successful, respond with
                                                              user-id and group-id.
                 file open request, etc, ...
User-id          -------------------------------->
    & Group-id
```

Figure 4. User authentication scenario

If you are communicating with only one NFS server host or if you have the same user ID and group ID setup on every NFS servers in your network. You will never need to use this facility to change your user ID information. However, if you have different user IDs on different NFS server hosts, you may have to use this utility to set the Kernel's user ID information to your user ID on the server you are about to access files from. Otherwise, your file access will be denied since the old user ID information is not valid on the later server even though it is absolutely valid on the previous server.

To change the Kernel's user ID information, you have to go through a login process. You can initiate the login process from NFSADMIN by selecting the **Login** sub-menu under the **UTILITIES** menu and then press the **[Enter]** key. NFSADMIN will then prompt you for the login ID and the password. Note that you can also login using the NFS command (see NFS Command Reference).

Setting The NFS Kernel's UMASK

NFS Kernel maintains a default UMASK parameter which is used to set the default file access permission to all files created on the remote file systems by local PC applications (see the Command Reference for discussion of UMASK).

From time to time, you may want to change the NFS Kernel default's UMASK value.   To do this, select the UMASK function in UTILITIES menu by high-lighting it and press the [Enter] key.   You are then prompted to enter the new UMASK value as follows:

```
Enter new UMASK:   _
```

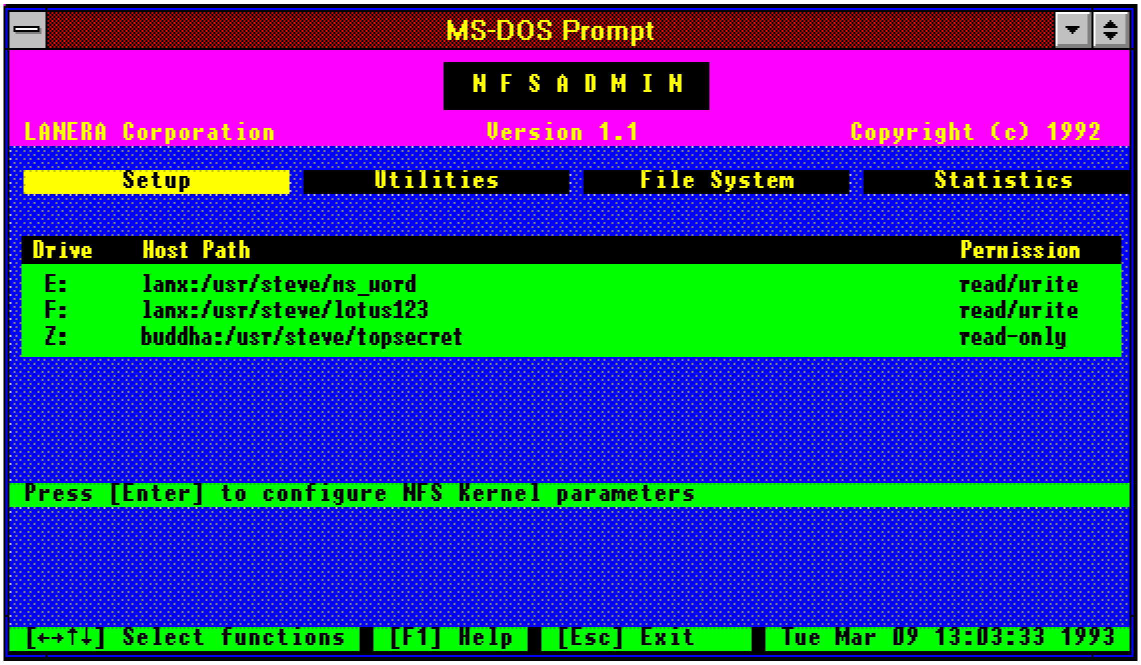
You then simply enter the new value for UMASK at the prompt followed by the [Enter] key.   You can verify if the update is successful by checking the value display next to the UMASK function bar when you high-light it.

Deinstalling The NFS Kernel

This function allows you to deinstall the NFS Kernel from NFSADMIN.   However, the NFS Kernel can not be reinstalled from NFSADMIN.   You must exit from NFSADMIN and invoke the LOADNFS batch file from a DOS prompt in order to reinstall the NFS Kernel.

File System

This menu displays the currently-mounted remote file systems and allows you to manage your system's file system resources by mounting or unmounting remote file systems.   To select this function, use the [←] or the [→] key to high-light the **File System** menu.   If there are some drives already mounted, these drives are immediately displayed on the screen as illustrated in figure 5.   The mounted drives are displayed with the remote hostname, the pathname and the access permission of the remote file systems. The bottom line displays a number of functions you can select to do while in this menu.



```
                          N F S A D M I N
LANERA Corporation           Version 1.1            Copyright (c) 1992

Drive    Host Path                                        Permission
 E:      lanx:/usr/steve/ms_word                          read/write
 F:      lanx:/usr/steve/lotus123                         read/write
 Z:      buddha:/usr/steve/topsecret                      read-only

Press [Enter] to configure NFS Kernel parameters

[←→↑↓] Select functions  [F1] Help  [Esc] Exit     Tue Mar 09 13:03:33 1993
```

For more information, choose:

- Mount a File System
- Mount all Auto-Mount File Systems
- Unmount an existing NFS Drive
- Unmount all Existing NFS Drives

Mount A File System

To mount a remote file system, press the [F9] key. NFSADMIN then brings up the mount table entries. You can select one of these entries for mounting by positioning to the to-be-mounted entry and then press the [F10] key to mount it, or you can specify an entry which is not in the mount table by pressing the [F2] key. Once this key is pressed, NFSADMIN displays a blank mount table entry that you can use to specify the file system to be mounted. After you have entered all the necessary information, press the [F10] key to mount the file system you have just specified. If the operation is successful, the display is updated with the new file system added.

Mount All Auto-mount File Systems

You can select to mount all the file system entries in the mount table which are marked AUTO. This function avoids you the tedious job of mounting several file systems one by one. To invoke this function, press the [F10] key. NFSADMIN will then look up the mount table, find the ones marked with AUTO and then execute the mount requests.

Unmount An Existing NFS Drive

To unmount a current NFS drive, use the [] or the [↓] key to position to the file system to be unmounted by high-lighting it, then press the [F7] key to invoke the unmount.   If the operation is successful, the unmounted entry is removed from the display.   Otherwise, an error message is displayed at the bottom line.

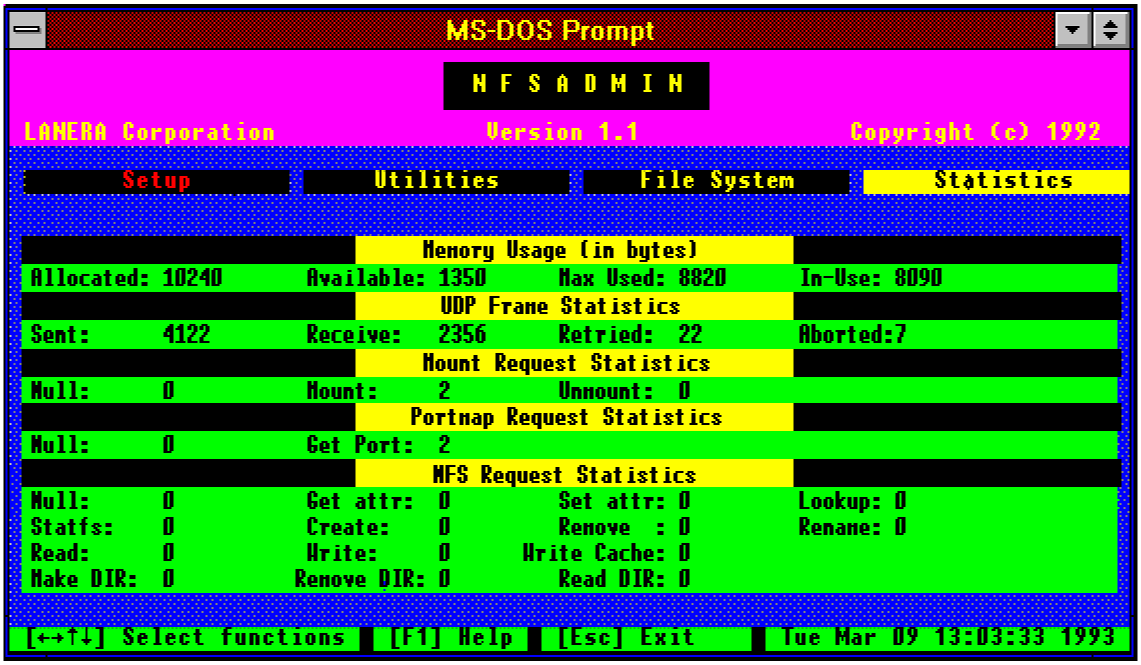
Unmount All Existing NFS Drives

To unmount all current NFS drives from your local system, press the [F8] key. NFSADMIN will prompt you to confirm whether you want to proceed. If your response to the prompt is Y (for yes), NFSADMIN will go ahead to unmount all the existing remote file systems from your local drives. When the operation completes, the screen displays only the heading with all the file system entries removed.

Statistics

The STATISTICS menu displays Softerm TCP/IP NFS Kernel's statistics. Using this utility, the users can learn about the kernel memory usage, UDP frame send/receive and other DOS system call information. This menu displays the following statistics as shown in figure 6:

- Kernel's memory usage statistics
- UDP frame statistics
- Mount request statistics
- Portmap request statistics
- NFS request statistics
- DOS system call statistics



```
                              N F S A D M I N
LANERA Corporation              Version 1.1              Copyright (c) 1992

      Setup            Utilities          File System          Statistics

                          Memory Usage (in bytes)
Allocated: 10240     Available: 1350     Max Used: 8820     In-Use: 8090
                           UDP Frame Statistics
Sent:       4122     Receive:    2356    Retried:   22      Aborted:7
                          Mount Request Statistics
Null:       0        Mount:      2       Unmount:   0
                         Portmap Request Statistics
Null:       0        Get Port:   2
                           NFS Request Statistics
Null:       0        Get attr:   0       Set attr: 0        Lookup: 0
Statfs:     0        Create:     0       Remove   : 0       Rename: 0
Read:       0        Write:      0    Write Cache: 0
Make DIR:   0       Remove DIR:  0       Read DIR: 0

[←→↑↓] Select functions  [F1] Help  [Esc] Exit    Tue Mar 09 13:03:33 1993
```

Figure 6 - Statistics display

Most of the above information is only useful to TCP/IP NFS developers who use the information for debugging purposes. End-users are most likely interested in the first two categories.

They are:

Memory Usage Statistics
UDP Frame Statistics

Kernel's Memory Usage Statistics

This statistical information allows the users to monitor the NFS Kernel memory usage. The NFS Kernel may behave unpredictably when it lacks memory to complete an operation. This is the kind of information that should be checked first when you experience difficulty in accessing a mounted file system. If the memory usage level is high, it is possible that your particular application requires a large amount of the NFS Kernel's memory. In this case, you can use the SETUP menu to reconfigure the Kernel buffer space parameter.

UDP Frame Statistics

This information records the number of UDP frames to be sent or received by the NFS Kernel.   The statistics also record the number of retried or aborted frames which usually indicate problems in the communication path between the NFS Kernel and an NFS server.

Using Softerm TCP/IP NFS

Using Softerm TCP/IP NFS is simple as it is intended to be. Assuming that you have already configured and successfully loaded TCP/IP Kernel, the following list outlines a step-by-step procedure to get TCP/IP NFS loaded and running:

- Configure NFS Kernel parameters and generate LOADTCP.BAT and RMNFS.BAT files
- Load the NFS Kernel
- Mount file system(s)
- Login an NFS server if neccessary
- Access files on a mounted file system
- Unload the NFS Kernel

CONFIGURE NFS KERNEL PARAMETERS

Before loading the NFS Kernel, you must ensure it is properly configured. Configuring the NFS Kernel is straight forward and simple by using the NFSADMIN program. You can follow the intructions in SETUP section of Chapter 3 in order to configure the NFS Kernel parameters and generate new LOADNFS.BAT and RMNFS.BAT files.

For additional information, choose:

Loading NFS Kernel
Accessing NFS Files
Unloading NFS Kernel

Load NFS Kernel

Once the setup is done, you can start loading the NFS Kernel. The NFS Kernel requires the network services from the TCP/IP Kernel that must be loaded and properly functioning before the NFS Kernel can be loaded. The NFS Kernel is loaded from a DOS prompt as follows:

```
C> LOADNFS [Enter]
```

LOADNFS is a batch file generated by NFSADMIN when you saved the configured NFS Kernel parameters. It contains NFS commands to do the following:

- Load the NFS Kernel in memory
- Mount all auto-mount file system entries in the mount table
- Attempt to login as the user specified by the USER parameter in the NETINFO file with the default authentication server.

If its execution is successful, you should be able to access the mounted file system(s) thereafter without having to do anything else.

Access NFS Files

After the NFS Kernel is loaded and all the remote file systems mounted, files can be copied to or from a mounted remote file system as if they were on a local drive. DOS and Windows 3.X applications can create, access, or delete any files on a mounted file system provided the local user is permitted to do so.

Unload NFS Kernel

The NFS Kernel can be dynamically loaded and unloaded. This means when you have no need for NFS services, you can remove it from memory. Unloading the NFS Kernel is done by invoking the RMNFS batch file at a DOS prompt as follows:

```
C> RMNFS[Enter]
```

RMNFS unmounts all the mounted file systems and returns the memory occuppied by the NFS Kernel back to the DOS operating system.

Filename Conversion

Every operating system implements their own filename and file attribute conventions depending on the system design goals, constraints and other factors, DOS implements its own rules for filenames and file attributes which is much simpler than other multi-user operating systems such as UNIX or VMS. This is primarily due to the fact that DOS does not have to concern itself with file protection among users who share the same system. In a diversified network environment, where dissimilar file systems are accessible from any system on the network, some differences need to be resolved when files are accessed from a foreign operating system. Softerm TCP/IP NFS deals primarily with the conversion between DOS and UNIX files.

For additional information, choose:

- DOS File and Directory Names
- UNIX File and Directory Names
- DOS/UNIX Filename Conversion
- Pathname Conversion
- Filename Case

DOS File and Directory Names

DOS file and directory names have the following general syntax:

FILENAME[.EXT] or DIRNAME[.EXT]

A file or directory name could be followed by a dot and an extension. The name and the extension must follow the following rules:

- file or directory names must be from 1 to 8 characters
- extensions must be from 1 to 3 characters
- extensions always begins with a dot
- names and extensions are case-insensitive
- filenames and extensions are composed of:
    + uppercase letters (A-Z)
    + numerical letters (0-9)
    + and the following special characters:
      $ % ' @ - { } ~ ` ! # ( ) _ ^ &

All other characters are invalid and should not be used to name a DOS file or directory.

UNIX File and Directory Names

On UNIX, the file and directory names are composed of the full ASCII character set and can be 256 character long. Some UNIX implementation limit the length of the file name and directory name to 14 characters. UNIX file and directory names are case-sensitive, i.e. files whose names are Abc, abc, aBc, abC are all different files.

DOS/UNIX Filename Conversion

Obviously, because of the differences in filename conventions, filenames must be validated and properly converted when files are transferred (or copied) from DOS to UNIX and vice versa. Softerm TCP/IP NFS implements an algorithm for automatically converting filenames on the fly without user intervention. To do this, TCP/IP NFS maintain a cache of the last 50 accessed file or directory names which are subject to conversion. Valid DOS filenames are not converted and, therefore, not stored in the conversion cache. Invalid DOS file name are converted and are presented to the user in the following general format:

FFFFFF~L[.XXX]

where:

FFFFFF are the 6 first character of the UNIX filename converted to uppercase character. If one or more converted character is invalid DOS filename's character, the character is replaced by a tilda (~).

~ this special character at the 7th character position is a marker used by Softerm TCP/IP NFS to indicate that the filename is a converted filename.

L is a uppercase letter (A-Z). It is used to enforce filename uniqueness in case more than one converted filename has the same resulting name. It is automatically assigned by Softerm TCP/IP NFS while converting a filename.

XXX are the first 3 characters of the UNIX file name extension. If one or more converted character is invalid for DOS, the character is replaced by a tilda (~).

Softerm TCP/IP NFS only keeps track of the last 50 conversions. It does this by storing the last 50 converted filenames in a cache buffer. Recently accessed files found in the cache are placed at the top of the cache. When a filename needs to be converted at the time the cache is full, the last entry in the cache is reused. This may cause some problems for some programs that read directories and retain the filenames in their local memory and then reuse them later. This happens because some of the entries in the cache have been wiped out to make room for new entries. Xtree by Executive Systems Inc. is one such program.

The following table gives some examples of how UNIX filenames are converted to DOS file name when the user accesses UNIX files using TCP/IP NFS.

UNIX filenames	®	**DOS filenames**	**Comments**
DOS.EXE		DOS.EXE	No conversion
DOS		DOS	No conversion
DOS.EXEC		DOS~~~~X.EXE	Extension too long
DOSFILENAME.EXE		DOSFIL~X.EXE	Filename too long
DOSFILENAME.EXEC		DOSFIL~X.EXE	Filename & extension too long
DOS+FI.EXE		DOS~FI~X.EXE	Invalid character in filename
DOS.B.EXE		DOS~B~~X.EXE	Invalid character in filename
DOS.E+E		DOS~~~~X.E~E	Invalid character in filename
dos.exe		DOS~~~~X.EXE	Lower case to uppercase

dosfilename	DOSFIL~X	Lower case to uppercase & Filename too long

Pathname Convention

For file manipulation programs such as DOS COPY, XCOPY, etc..., the filename separator to be used is \ (backslash). For file system operations such as "NFS MOUNT ...", the remote pathname separators must follow the remote file system standard, since Softerm TCP/IP NFS does not attempt to do any translation of the remote pathname.

Filename Case

When DOS creates a file or directory on a mounted file system, the filename or directory name is always upper case.   This convention allows the directory listing command to show filenames without conversion.

Text File Conversion

DOS and UNIX text file formats are slightly different on the coding of the end-of-line (EOL) and end-of-file (EOF) markers. UNIX EOL marker is NL (0x0A), and DOS EOL marker consists of two (2) characters, CR (0x0D) and NL (0x0A), in that order.   UNIX files do not have any EOF marker while DOS uses CNTL-Z (0x1A) as the EOF marker.

Softerm TCP/IP NFS does not convert text files when they are read or written. Therefore, when accessing a file on a remote UNIX operating system, DOS commands may misinterpret the file contents since the EOL and the EOF characters do not have the same meaning on the remote system.

For example, let's assume we have a UNIX file system mounted on drive L: and FILE is a file on that (UNIX) file system.   The original content of the file is as shown below on the UNIX system:

```
$ cat FILE
This is a UNIX file which does not have any carriage returns
(CR), but has a CNTL-Z (^Z) which in fact
does not indicate the End Of File.
$
```

However, on the DOS side where the same file is seen through the mounted file system, the file looks like:

```
L:>TYPE FILE
This is a UNIX file which does not have any carriage returns

(CR), but
has a CNTL-Z (

L:>
```

As you can see, DOS TYPE command treats UNIX end-of-line character just likes a linefeed character, without moving the cursor back to the first row, and the ^Z character as a DOS end-of-file and, therefore, not output to the console.

There are several ways to deal with such situation such as those suggested below:

- Use a file transfer program (FTP, TFTP, or RCP) which performs the EOL and EOF conversion for you automatically.

- Use the command U2D (UNIX to DOS file conversion) to do the conversion manually, file by file.   In addition, D2U (DOS to UNIX file conversion) can also be used to convert DOS format file to UNIX file when you want to make a DOS file to be accessed by UNIX applications.   See Command Reference section in this manual for information on these commands.

- Use a UNIX utility for DOS command such as CAT by MKS to display the file.

UNIX-to-DOS Mapping

When copying files from UNIX to DOS, UNIX file attributes are mapped as followed:

- UNIX *Owner-read* (only) is mapped to DOS Read-Only
- UNIX *Owner-write* is mapped to DOS Read-Write
- UNIX *Owner no read* and *no write* is mapped to DOS *Hidden*
- UNIX *Owner-Execute* is ignored
- UNIX *Group* and *Other* permission settings are ignored

File Attribute Mappings

Another area where DOS and UNIX file systems have some differences is file attributes.
DOS uses the following attributes for its files:

- *Read-only* which indicates that the file can not be written.
- *Hidden* which indicates that the file can not be displayed on the output of DIR command.
- *System* which indicates that the file is a system file.
- *Volume* which indicates that the file is a Volume serial for a disk.
- *Directory* which indicate the file is a directory.
- *Archive* which indicate the file has been modified since the last backup.

On the other hand, UNIX uses the following attributes for its files:

- *Owner-read* which indicates that the file is readable by the owner.
- *Owner-write* which indicates that the file is writable by the owner.
- *Owner-execute* which indicates that the file is executable by the owner.
- *Group-read* which indicates that the file is readable by members of the group which the file belongs to.
- *Group-write* which indicates that the file is writable by members of the group which the file belongs to.
- *Group-execute* which indicates that the file is executable by members of the group which the file belongs to.
- *Other-read* which indicates that the file is readable by everyone else.
- *Other-write* which indicates that the file is writable by everyone else.
- *Other-execute* which indicates that the file is executable by everyone else.
- *Regular* which indicates the file is a regular file.
- *Directory* which indicates the file is a directory.
- *Character* which indicates the file is a character device file.
- *Block* which indicates the file is a block device file.
- *Symbolic link* which indicates the file is a symbolic link.
- *Pipe (or FIFO)* which indicates the file is a named-pipe.
- *Socket* which indicates the file is a bound socket.

DOS has no sense of file ownership because it has been designed for a single user environment. In a network environment, this may create some security problems.
However, Softerm TCP/IP NFS cooperates with the remote NFS server to provide some degree of protection against non-permitted file accesses on the remote server. It does this by accompanying every file access request a user-ID. The NFS server uses the user-ID to assign a owner to the created files, if the operation is file creation, or to qualify file access permission, if the operation is open, read or write. This user-ID can be dynamically controlled by the local user through an authentication process called *login*. When the local user logs in, a new user-ID associated with the login user will be set and used for all future file access requests made by the Softerm TCP/IP NFS Kernel. When there is no login, a special default user-ID (called *nobody*) is used for all file access requests initiated at the local PC. This user-ID allows manipulation on files owned by *nobody*, and files of which the UNIX file permissions are set for *Other* users.

For additional information, choose:

UNIX-to-DOS Mapping
DOS-to-UNIX Mapping
File Attribute Manipulation

DOS-to-UNIX Mapping

When copying files from DOS to UNIX, DOS file attributes are mapped as follow:

- DOS *Read-only* is mapped to UNIX *Owner-read* and *Owner-Execute*.
- DOS *Read-Write* is mapped to UNIX *Owner-read, Owner-write* and *Owner-Execute*.
- DOS *Hidden* is ignored because UNIX does not have equivalent attribute.
- DOS *System* is ignored because UNIX does not have equivalent attribute.
- DOS *Volume* is not supported except for read operation. In such case, the returned filename is NFS.
- DOS *Directory* is converted to UNIX *directory*.
- DOS *Archive* is ignored because UNIX does not have equivalent attribute. The file attributes return to DOS application (by TCP/IP NFS Kernel) always have the Archive bit on. This allows backup programs to save network files.

File Attribute Manipulation

Since the DOS-to-UNIX file attribute mappings do not manipulate the permission bits for *group* and *other* users, the UMASK setting is used to manipulate these bit settings if so desired. In addition, TCP/IP NFS also provides a command to allow you to manipulate the file attribute of files on a mounted (UNIX) file system. This command's function is similar to UNIX's CHMOD command or DOS' ATTRIB command. From time to time, you may want to use this command, namely NCHMOD, to set a file or a group of files to a desired mode setting.

UMASK Setting

UMASK is a parameter maintained by the TCP/IP NFS Kernel. Its setting reflects the default file acess permissions for all files created on a mounted file system by local applications. You can dictate the DOS-to-UNIX file access permission mappings by setting the UMASK to a value other than 0, since the mapped file access permission is always "or-ed" with the UMASK to yield the final access permission to be set for the remote file. By default, UMASK is set to the value specified by the NFS_UMASK parameter in the NETINFO file, NFSADMIN Kernel Parameters function edits this parameter. UMASK can be change after the TCP/IP NFS Kernel is loaded. For information on how to change the UMASK setting, see Chapter 2 (Administration).

NCHMOD Command

Reference the NCHMOD command reference for information on how to use this command.

LIMITATIONS

The following are some limitations that you may encounter from time to time when using Softerm TCP/IP NFS.   These limitations are not imposed by TCP/IP NFS but   by the DOS operation system:

For additional information, choose:

- Pathname Length
- Pathname Depth
- File Size

Pathname Length

Since DOS pathname can not be longer than 80 characters. A mounted file system's pathname exceeds this limitation will not show up in a directory listing command's output. Any other local applications when tries to specify a pathname longer than permitted will fail with an error message equivalent to "Invalid pathname". One way to keep the pathname short is to name the directories with fewer characters.

Pathname Depth

DOS limitation on directory depth level is 15. An attempt to access a remote directory or file at this level or deeper will fail. For example, if you have a pathname like the following:

```
\1\2\3\4\5\6\7\8\9\10\11\12\13\14\15\16\17\18
```

on a mounted file system. When you do a DIR command, the command lists only up to the directory 15 and then quits. It is because DOS has reached its limitation and gave up.

File Size

DOS has a file size limit of up to 4 Gigabytes. Since most PC hard disks have much smaller size, this figure is actually no limit at all. On a UNIX system, the file size is limited to a system configuration parameter, namely ULIMIT, which is typically set to 16 Mbytes. If you encounter error message saying "File too big" when transferring a DOS file to a mounted UNIX file system, it is likely that you have a file size limit error. In this case, you should contact the system administrator to increase the ULIMIT for your user-ID on the system where your mounted file system is.

Command Reference

This section provides a quick reference to the commands provided in Softerm TCP/IP NFS package except for the NFSADMIN command. They are prepared in UNIX-style command reference format, which generally describes the command syntax and options followed by a detailed description of what the command does for every option selected. Any related programs or files are also mentioned.

The commands are listed in alphabetic order for easy reference.

D2U
NCHMOD
NFS
NLS
U2D

Error Messages

This section documents error messages generated by Softerm TCP/IP NFS commands when they encounter an error condition.   Error messages are always assigned with a unique error code which is in the following format:

<PRG-NAME>   <ERROR-CODE>:   <ERROR MESSAGE>

<PRG-NAME> is the name of the program that generates the error message.   For example, if you are using the NFS command, the <PRG-NAME> is then NFS.

<ERROR-CODE> is a four-digit decimal number intended to be unique within every program.   <ERROR-CODE> of which number begins with 1, such as **1**020, indicates the error is common which may occur in various Softerm TCP/IP commands.   Common error messages are documented in the standard Softerm TCP/IP Kernel Manual.

This the following programs' error messages are documented in this section:

D2U
NCHMOD
NFS
NFSADMIN
NLS
U2D

Trouble Shooting

Although we have tried to minimize errors during the installation and setup processes by automating many steps, however, it is extremely difficult to cover all the variables introduced by so many adapter cards and programs available for the PC environment. Therefore, when you first install and setup Softerm TCP/IP KERNEL, you may run into difficulty in "talking" with the other hosts on the network.

Below is a list the four most common problems that you may experience when using NFS and shows you how to diagnose and recover from these problems.

Problem #1: Communication Path Problem
Problem #2: System Hangs when you invoke a TCP/IP application
Problem #3: Read Fault
Problem #4: Write Fault

D2U

DOS-to-UNIX text file conversion command

synopsys

```
d2U [ infile | - ] [ outfile ]
```

DESCRIPTION

D2U command copies the file infile or STDIN to the file outfile or STDOUT. While copying the file, input file's DOS end of lines (\r\n) characters are converted to UNIX end of lines (\n) characters. If both infile and outfile are not specified on the command line, the command copies STDIN and output to STDOUT. The dash (-) is used to indicate STDIN as input file when an output file is also specified on the command line.

SEE ALSO

U2D

NCHMOD

UNIX-style file attribute manipulation program

synopsis

nchmod octal fileS ...
nchmod [ugO]{+-}{rwx} fileS ...

Description

NCHMOD allows the users to manipulate the file permission settings of the files on a mounted NFS file system. OCTAL has the same syntax and meaning as UMASK (see NFS Command Reference). When OCTAL is specified, NCHMOD sets all the files specified by FILES with the file permission setting indicated by it. For example, the following sets DOC1's and DOC2's permission bits to 0655, i.e. readable and writable by owner, readable and executable by group and other users:

```
C> NCHMOD 0655 DOC1 DOC2
```

Optionally, the user can also use a more user-friendly syntax as follows:

```
nchmod [ugO]{+-}{rwx} fileS ...
```

where:

U	specify the followed setting will effect *owner* attribute
G	specify the followed setting will effect *group* attribute
O	specify the followed intended will effect *other* attribute
+	means *to set*
-	means *to reset*
R,W,X	read, write, execute permission bits respectively

When none of U, O or G is specified, the setting will effect all user classes. For example, to set the owner bits of DOC1 to readable and writable:

```
C> NCHMOD U+RW DOC1
```

or to disable writing by group and other users for the same file:

```
C> NCHMOD GO-W DOC1
```

or to make MY_PROG.EXE to be executable by everyone:

```
C> NCHMOD +X MY_PROG.EXE
```

Note that NCHMOD does not take wildcard character (*) in filenames.

SEE ALSO

NLS

NFS

Command-line NFS administration utility program

SYNOPSYS

```
nfs mount drIVE: host:/path [ ro,lk ]
nfs mount { drive: | host:/path }
nfs mountam
nfs umount drive:
nfs umountall
nfs showmount [ drive: ]
nfs ping host
nfs umask [ value ]
nfs login [ userid [ passwd ] ]
nfs logout
nfs stats
nfs stop { server | client }
```

DESCRIPTION

This command implements all the administrative tasks to manage the NFS operating environment provided by Softerm TCP/IP NFS package. The command provides many sub-commands that allow the user to:

- mount or unmount one or more file systems to local drive(s)
- display currently-mounted file systems and NFS Kernel statistics
- verify connectivity with NFS servers
- modify default remote file attribute settings and remote file ownership

The followings describe how the NFS command can be used:

nfs mount drive: host:/path [ro,lk]
Mount a file system specified by HOST:/PATH to a drive specified by DRIVE:. RO and LK are optional. When specified, they have the following meanings:

- RO: indicates the file system is for READ-ONLY. No writing is allowed.
- LK: enables file/record locking. A lock manager (*lockd* on UNIX machines) must be available for this function to work properly.

nfs mount { drive: | host:/path }
Mount a file system as defined in the MNTTAB database file. Either the DRIVE: or the HOST:/PATH argument needs to be specified. NFS will look up the MNTTAB database for the other information.

NFS MOUNTAM
Mount all file system mapping entries in the MNTTAB database that are marked with AM option. This command provides a quick way to mount several file systems after loading the NFS Kernel.

nfs umount drIVE:
Unmount a mounted file system specified by the given drive ID, DRIVE:

nfs showmount [drive:]
Display information of a mounted file system, if a drive ID is specified, or all the mounted file systems, if no drive ID is specified.

nfs ping host

Verify connectivity with a remote NFS server by sending ICMP echo messages.

nfs umask [NEW-value]
Display the current NFS Kernel's UMASK setting, or set it to a new value if NEW-VALUE is given. UMASK specifies the default file permission attribute when a new file is created on a mounted file system. It has the same permission bit settings as UNIX's UMASK and must be specified in octal, i.e. preceded by 0. For example, 0755 is a valid UMASK.

The bit pattern defined for UMASK is as follows:

	Owner	Group	Other
	r w x	r w x	r w x
Example: 0755	1 1 1	1 0 1	1 0 1

r: read permission
w: write permission
x: execute permission

Thus, 0755 means all files created on a mounted file system will be readable, writable and executable by the owner of the file. The users belong to the same group as that of the owner and all other users have only the read and execute permissions.

nfs login [[AUTH:]userid [passwd]]
Display the current login user ID used by the NFS Kernel as the default owner for all files created on a mounted file system, if no argument is specified; or changes the default login user ID in the NFS kernel if USERID is specified. Before the default login user ID can be changed, the new login ID (USERID) must be authenticated with a default authentication host or with the host AUTH, if specified. The new login ID takes effect only if the authentication succeeds. PASSWD is the password associated with USERID. If USERID is specified, the password must be given. If it is not specified in the command line, NFS will prompt for the password.

nfs logout
Clear the current NFS Kernel's user ID setting. The user ID is then reset to the default user ID of *nobody*, which is a limited-access user ID.

nfs stats
Display NFS Kernel and system call statistics.

nfs stop client
Stop NFS Kernel and deinstall it from DOS memory.

FILES

{INSTALL}\ETC\MNTTAB
{INSTALL}\ETC\PASSWD
{INSTALL}\ETC\GROUP

NLS

UNIX file/direcotry listing command

synopsis

nls [-RadCxmnlogLrtucpFbqisfS] [FILES | DIRECTORIES]

Description

NLS is used to list UNIX files or directories on a mounted file system.   For each directory argument, NLS lists the contents of the directory; for each file argument, NLS repeats its name and any other information requested. The output is sorted alphabetically by default. When no argument is given, the current directory is listed.   A null ('') parameter will be interpreted as '.' and the current directory is listed. Passing N null parameters lists the current working directory N times. When several arguments are given, the arguments are first sorted appropriately, but file arguments appear before directories and their contents.

There are three major listing formats. The default format is to list one entry per line, the -C and -x options enable multi-column formats, and the -m option enables stream output format. To determine output formats for the -C , -x , and -m options, NLS assumes the screen having 80 columns so that it can internally determine the number of character positions available on one output line.

NLS supports the following options:

-R	Recursively lists sub-directories encountered.
-a	Lists all entries including entries whose names begin with a period (.)
which are not normally listed.
-d	If an argument is a directory, only lists its name (not its contents). Often
used with -l to get the status of a directory.
-L	If an argument is a symbolic link, list the file or directory the link
	references rather than the link itself.
-C	Multi-column output with entries sorted down the columns.
-x	Multi-column output with entries sorted across rather than down the page.
-m	Stream output format.
-l	Lists in long format, giving mode, number of links, owner, group,
	size in bytes, and time of last modification for each file (see below).
	If the file is a special file, the size field contains the major
	and minor device numbers rather than a size. If the file is a
	symbolic link, the file name is printed followed by " à " and the
	path name to the referenced file.
-n	The same as -l , except that the owner's UID and group's GID numbers
	are printed, rather than the associated character strings.
-o	The same as -l , except that the group is not printed.
-g	The same as -l , except that the owner is not printed.
-r	Reverses the sort order to get reverse alphabetic or oldest first as
	appropriate.
-t	Sorts by time modified (latest first) instead of by name. (See -u and -c.)
-u	Uses time of last access instead of last modification for sorting (with
	the -t option) or printing (with the -l option).
-c	Uses time of last modification of the i-node (file created, mode changed,
	and so forth) for sorting (-t) or printing (-l).
-p	Puts a slash (/) after each filename if that file is a directory.
-F	Puts a slash (/) after each filename if that file is a directory
	and puts an asterisk (*) after each filename if that file is
	executable, and an ampersand (@) if the file is a symbolic link.
-b	Forces printing of non-graphic characters to be in the octal ddd

notation.
-q Forces printing of non-graphic characters in file names as the question mark character (?).
-i For each file, prints the i-number in the first column of the report.
-s Gives size in 4Kbyte blocks, including indirect blocks, for each entry.
-f Forces each argument to be interpreted as a directory and lists the name found in each slot. This option turns off. -l, -t, -s, and -r, and turns on -a.
The order is the order in which entries appear in the directory.

Note that the above options are *case-sensitive*, i.e. specifying option L yields a quite different result from specifying option l.

The mode printed under the -l option consists of 10 characters that are interpreted as follows:

The first character is one of the following:

d If the entry is a directory
l if the entry is a symbolic link;
b If the entry is a block special file
c If the entry is a character special file
p If the entry is a FIFO (a named pipe) special file
s If the entry is a socket
- If the entry is an ordinary file.

The next 9 characters are interpreted as three sets of three bits each. The first set refers to the owner's permissions; the next set refers to permissions of others in the user-group of the file; and the last to all others in the user group. Within each set, the three characters show permission to read, to write, and to execute the file as a program, respectively. For a directory, execute permission is permission to search the directory for a specified file.

The NLS -l (long list) command prints its output as follows:

```
-rwxrwxrwx   1 smith   dev      10876   May 16 9:42 part2
```

This horizontal configuration provides a good deal of information.

Reading from right to left, you see that the current directory holds one file, named "part2." Next, the last time that file's contents were modified was 9:42 A.M. on May 16. The file is moderately sized and contains 10,876 characters, or bytes. The file owner, or the user, belongs to the group *dev* (perhaps "development"), and the user's login name is *smith*. The number, in this case 1, indicates the number of links to the file named *part2*. Finally, the row of dash and letters tell you that user, group, and others have permissions to read, write, execute the file named *part2*.

The execute (x) symbol here occupies the third position of the three-character sequence. A - (dash) in the third position would have indicated a denial of execution permissions. The permissions are indicated as follows:

r If the file is readable
w If the file is writable
x If the file is executable
- If the indicated permission is not granted
l Mandatory locking occurs during access (the set-group-ID bit is on and the group execution bit is off).

For user and group permissions, the third position is sometimes occupied by a character other than x or -.   s also may occupy this position, referring to the state of the setID bit, whether it be the user's or the group's. The ability to assume the same ID as the user during execution is, for example, used during login when you begin as root but need to assume the identity of the user stated at login.

In the case of the sequence of group permissions, l may occupy the third position. l refers to mandatory file and record locking. This permission describes the capability of a file to allow other files to lock its reading or writing permissions during access.

For others permissions, the third position may be occupied by t or T.   These refer to the state of the sticky bit and execution permissions.

Examples: The first set of examples refers to permissions:

```
-rwxr--r--
```

This describes a file that is readable, writable, and executable by the user and readable by the group and others.

```
-rwsr-xr-x
```

The second example describes a file that is readable, writable, and executable by the user; readable and executable by the group and others; and allows its userID to be assumed, during execution, by the user presently executing it.

```
-rw-rwl---
```

This example describes a file that is readable and writable only by the user and the group and can be locked during access.

```
NLS -a
```

This command prints the names of all files in the current directory, including those that begin with a dot (.), which normally do not print.

```
NLS -aisn
```

This command provides you with quite a bit of information including all files, including non-printing ones (a), the i-number(the memory address of the i-node associated with the file (printed in the left-hand column (i); the size (in blocks) of the files, printed in the column to the right of the i-numbers (s); finally, the report is displayed in the numeric version of the long list, printing the UID (instead of user name) and GID (instead of group name) numbers associated with the files.

The group-execute permission character is given as s if the file has set-group-ID mode; likewise, the user-execute permission character is given as s if the file has set-user-ID mode. The last character of the mode (normally x or -) is t if the 1000 (octal) bit of the mode is on; see NCHMOD for the meaning of this mode. The indications of set-ID and 1000 bits of the mode are capitalized   S and T respectively) if the corresponding execute permission is not set.

When the sizes of the files in a directory are listed, a total count of blocks, including indirect blocks, is printed.

SEE ALSO

NLS

U2D

UNIX-to-DOS text file conversion command

SYNOPSYS

u2d [infile | -] [outfile]

DESCRIPTION

U2D command copies the file infile or STDIN to the file outfile or STDOUT. While copying the file, input file's UNIX end of lines (\n) characters are converted to DOS end of lines (\r\n). If both infile and outfile are not specified on the command line, the command copies STDIN and output to STDOUT. The dash (-) is used to specify STDIN as input file when an output file is also specified on the command line.

SEE ALSO

D2U

D2U

D2U0001: Internal error flag=value

This message indicates an D2U's internal error.

> **Action:** Retry the command again. If the problem perists, notify Lanera technical support personnel for assistance.

D2U0005: Invalid argument

This message indicates a problem on the command line. The usage of the command is printed after this message.

> **Action:** Reference the Command Reference for the correct command syntax and reenter the command again.

NCHMOD

NCHMOD0001: Invalid argument count

An unexpected number of arguments was specified on the command line.

Action: The command should have at least 2 arguments. Reference the Command Reference and the retype the command again with valid number of arguments.

NCHMOD0002: Wildcard characters not supported

The wildcard character '*' is not supported by this command.

Action: Do not specify wildcard characters in a file or directory name.

NCHMOD0003: Cannot chmod file <filename> error(x): reason

Failed to change the permission mode of the file <filename>.

Action: If the target file exists, verify the file ownership to ensure you have the permissionto change the file permission mode. Only the owner of the file is allowed to change the file permission mode.

NCHMOD0004: Invalid 'mode'

The 'mode' specification is invalid.

Action: Reference the NFS Command Reference on the discussion of UMASK and retry the command with a valid file mode.

NCHMOD0027: Invalid interrupt vector <vector>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the NFS Kernel interrupt vector. This value should be between 0x60 and 0x7F.

Action: Use an editor or TCPSETUP to verify the validity of the NETVEC parameter. If it is not valid, change it to a value in the range mentioned above.

NCHMOD0028: Invalid multiplexor <multiplexor>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the NFS Kernel multiplexor as indicated by <multiplexor>. This value should be between 0xC0 and 0xCF.

Action: Use an editor or TCPSETUP to verify the validity of the NFS_MULTIPLEXOR parameter. If it is not valid, change it to a value in the valid range mentioned above.

NCHMOD0029: Invalid userid <userid>

The USER parameter in the {INSTALL}\SETUP\NETINFO file contains an invalid length user ID.

Action: Use an editor to verify the length of the USER parameter. The length of a user ID should not longer than 40 characters.

NCHMOD0030: Invalid umask <umask>

The NFS_UMASK parameter in the {INSTALL}\SETUP\NETINFO file contains an invalid value for the NFS Kernel default UMASK. This value should be between 0 and 0777.

Action: Use an editor or TCPSETUP to verify the validity of the NFS_UMASK parameter. If it is not valid, change it to a value in the range mentioned above.

NFS ERROR MESSAGES

NFS0001: Invalid argument count.

This message indicates a syntax error on the NFS command line. The number of arguments is less or more than the number of the required arguments. A usage message is printed after this message.

Action: Reference the Command Reference for correct command syntax and then retype the command with a valid number of arguments.

NFS0002: Invalid function

This message indicates an invalid function was specified on the command line. A function is the second keyword on the NFS command line. A brief usage message is printed after this message.

Action: Reference the Command Reference for a list of valid command functions and then retype the command with a valid fucntion.

NFS0003: Invalid driveid

An invalid drive ID argument is detected either from a command line or from a mount table entry. The usage of the command is printed after this message.

Action: A DOS drive ID consists of a letter A to Z followed by a colon (:). If the specified drive ID did not conform to the syntax. Retype the command with a correct drive ID specification, if it was entered from the keyboard, or using NFSADMIN or a text editor to correct the drive ID in the mount table.

If the <drive-id> specification is perfectly valid, it is possible that <drive-id> exceeds the LASTDRIVE specification in the CONFIG.SYS. In this case, you need to extend LASTDRIVE specification to include <drive-id> and then reboot the system.

NFS0004: Invalid host:/path

This message indicates a syntax error in the <host:/path> argument specification either from the command line or from a mount table entry. A usage message is printed after this message.

Action: Check the hostname and the pathname specification. A hostname is either a string of up to 60 printable characters or an IP address in dot notation (a.b.c.d). A pathname is also up to 60 characters long. Hostname and pathname must be separated by a colon.

NFS0005: Invalid option

This message indicates a syntax error either from the command line or from a mount table entry. The argument <option> which is supposed to be one or more of the following keywords: ro, lk, and auto separated by commas. The usage of the command is printed after this message.

Action: Retype the command with the correct option specification. For example, the following command mounts a file system which is read-only (RO) and file-locking enable (LK).

```
NFS MOUNT F: lanx:/usr/steve/pc_files RO,LK
```

Note that if the remote host treats the hostname and pathname as case-sensitive strings, you must ensure to specify these names in a way that is acceptable to the remote host. Softerm

TCP/IP NFS does not attempt to do any conversion in this regard since it has no idea what type of system the remote is.

NFS0008: Invalid sub-function

This message indicates an error on the command line. The second argument (sub-function) is invalid. The usage of the command is printed after this message.

Action: Reference the Command Reference for correct command syntax and then retype the command with valid arguments.

NFS0009: Invalid host

This message indicates an error on the command line. The host ID argument, which supposes to be a remote hostname or IP address (in DOT notation), is invalid. The usage of the command is printed after this message.

Action: Reference the Command Reference for correct command syntax and then retype the command with valid arguments.

NFS0010: Invalid umask value

This message indicates an error on the command line. The UMASK specification, which supposes to be an octal value between 0 and 0777, is invalid. The usage of the command is printed after this message.

Action: Reference the Command Reference for correct UMASK syntax and then retype the command with a valid UMASK. Remember to always prefix the UMASK with a 0, since it is an octal value.

NFS0011: Invalid userid

The specified USERID in NFS LOGIN command does not exist in the {INSTALL}\ETC\PASSWD database.

Action: Use NFSADMIN's SETUP function to add a new entry for USERID if necessary.

NFS0012: Invalid password

The password specification for a USERID in NFS LOGIN command is not valid.

Action: Repeat the command and then reenter the password.

NFS0013: Cannot mount: <reason>

The NFS MOUNT command failed to mount a file system. <reason> indicates the reason for failure.

Action: If the <reason> is "Read fault" or "Write fault", either the NFS server is not up yet or the communication path is broken. Use the NFS PING command to verify connectivity between the local NFS client and the remote NFS server.

If <reason> is either "Not an NFS drive" or "Invalid drive", it is likely that the specified drive is not supported by DOS. By default, DOS supports up to drive E: unless specified by the LASTDRIVE parameter in the CONFIG.SYS file. If you want to mount a remote file system to a local drive after the drive E:, you need to add the LASTDRIVE parameter in the CONFIG.SYS file which specifies at least up to the drive you want to mount the remote file system on.

NFS0014: Cannot umount: <reason>

The NFS UMOUNT command failed to unmount a file system.

Action: If the <reason> is "Read fault" or "Write fault", either the NFS server is not up yet or the communication path is broken. Use the NFS PING command to verify connectivity between the local NFS client and the remote NFS server.

If <reason> is either "Not an NFS drive" or "Invalid drive", the drive you try to unmount is not an NFS drive or invalid.

NFS0015: Cannot showmount: <reason>

The NFS showmount command failed.

Action: If the <reason> is "Read fault" or "Write fault", either the NFS server is not up yet or the communication path is broken. Use the NFS PING command to verify connectivity between the local NFS client and the remote NFS server.

If <reason> is either "Not an NFS drive" or "Invalid drive", the drive you specified is not an NFS drive or invalid.

NFS0016: Cannot ping: <reason>

The NFS PING command failed.

Action: The communication path between the local station and the remote NFS server is broken. It may be due to problems at the remote server, problems at the local station or problems at an intermediate node. To determine whether it is a local failure, try to ping some other NFS servers on your local network which you know are up for sure.

If the problem is local, deinstall the NFS Kernel and TCP/IP driver and then reload them again.

NFS0017: Cannot set umask: <reason>

The NFS UMASK command failed to set NFS Kernel's default UMASK.

Action: Normally, this command should succeed. An error may indicate an unstable NFS Kernel. Deinstall and reload the NFS Kernel if necessary.

NFS0018: Cannot get umask: <reason>

The NFS UMASK command failed to retrieve NFS Kernel's default UMASK.

Action: Normally, this command should succeed. An error may indicate an unstable NFS Kernel. Deinstall and reload the NFS Kernel if necessary.

NFS0019: Cannot retrieve userid: <reason>

The NFS USERID command failed to retrieve NFS Kernel's user ID setting.

Action: Normally, this command should succeed. An error may indicate an unstable NFS Kernel. Deinstall and reload the NFS Kernel if necessary.

NFS0020: login failed: <reason>

The authentication fails for a given user ID and password. <reason> gives a brief error message indicating the cause of failure.

Action: It is possible that you have entered a wrong password. Retype the NFS LOGIN command and slowly re-enter the correct password at NFS prompt followed by the [Enter] key. If the problem persists, you may have forgotten the correct password or you may use the wrong user ID on the authentication host. Try to login the authentication host from a terminal to verify if the user ID/password you entered is valid.

NFS0021: Cannot login: <reason>

Failed to set the new user ID.

Action: Normally, this command should succeed. An error may indicate an unstable NFS Kernel. Deinstall and reload the NFS Kernel if necessary.

NFS0022: Cannot get statistics: <reason>

The NFS command failed to obtain NFS Kernel statistics.

Action: Normally, this command should succeed. An error may indicate an unstable NFS Kernel. Deinstall and reload the NFS Kernel if necessary.

NFS0024: Cannot stop client: <reason>

The NFS STOP command failed.

Action: If the <reason> indicates it is unsafe to deinstall, it is possible that another TSR program has overlaid a NFS Kernel interrupt vector. In this case, you have to deinstall the overlaid TSR program before you can deinstall NFS Kernel. Any other error messages may indicate an unstable system environment. In this case, reboot the system and report the problem to Lanera technical support personnel.

NFS0025: invalid <userid, password>

The authentication fails for a given user ID and password.

Action: It is possible that you have entered a wrong password. Retype the NFS LOGIN command and slowly re-enter the correct password at NFS prompt followed by the [Enter] key. If the problem persists, you may have forgotten the correct password or you may use the wrong user ID on the authentication host. Try to login the authentication host from a terminal to verify if the user ID/password you entered are valid.

NFS0026: no authentication host specified

NFS LOGIN requires an authentication host to be specified. However, none is specified.

Action: Usually NFS LOGIN uses a default authenticator specified by the NFS_AUTH parameter in the NETINFO file. The parameter specifies the name of a host to be contacted for login authentication. You can use a text editor to add this parameter to the NETINFO file or

NFSADMIN to configure this parameter.

Alternately, you can directly specify an authenticator on the NFS LOGIN command line.

NFS0027: no authentication port

NFS LOGIN fails to locate the authentication process port. It is due to the process on the authentication host, normally PCNFS, that responds to the port query request is not active.

Action: Check if PCNFS (or equivalent process) is running on the authentication host. If not, start it up; or sometimes a restart may also solve the problem.

NFS0028: Invalid multiplexor <mux-id>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the NFS Kernel multiplexor. This value should be between 0xC0 and 0xCF. <mux-id> is the invalid value detected.

NFS0029: Invalid userid <user-ID>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the USER parameter. <user-ID> is the invalid value detected.

NFS0030: Invalid umask <mask>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the NFS Kernel default umask. This value should be between 0 and 0777. <mask> is the invalid umask.

NFS0033: Invalid entry in mnttab: <entry>

<entry> in the mnttab file has an invalid syntax. Use NFSADMIN's SETUP function to restore the entry or manually correct it by editing the MNTTAB file.

Action: Use NFSADMIN or a text editor to correct the invalid entry in the mount table.

NFS0034: Too many entries in mnttab

The maximum entries in the MNTTAB database is 20.

Action: Use NFSADMIN or a text editor to delete some unused entries in the mount table to make room for new ones.

NFS0035: Invalid entry (HOST) in mnttab file <entry>

<entry> in the mnttab file has a too long hostname. Hostname can not be longer than 60 characters.

Action: If the hostname was wrongly specified, use NFSADMIN or a text editor to correct the hostname. If the hostname was a valid hostname in the HOSTS dattabase, use a text editor to shorten it in both HOSTS and MNTTAB databases.

NFS0036: Invalid entry (PATH) in mnttab file <entry>

<entry> in the mnttab file has an invalid remote path specification. The maximum pathname's length is 60 characters.

Action: If the pathname was wrongly specified, use NFSADMIN or a text editor to correct the pathname. Otherwise, the remote pathname must be shortened by assigning shorter names for directories, etc... before it can be mounted.

NFSADMIN

NFSADMIN0010: invalid drive ID: <drive>

User has entered an invalid drive ID. A valid drive ID is from A to Z. <drive> is the invalid drive letter entered.

Action: Reenter valid drive ID.

NFSADMIN0011: drive <drive:> is a local drive

User has entered a drive ID which is already be assigned for a local drive. <drive> is the invalid drive letter entered.

Action: Reenter a drive ID not yet being used by your local system.

NFSADMIN0015: drive <drive:> entry already exists

User has attempted to add a new entry or modify an existing entry to use a drive ID that is already assigned to another entry in the mount table.

Action: Use another drive ID for the new or modified entry.

NFSADMIN0026: mount failed: <reason>

An attempt to mount a remote file system failed. <reason> indicates the reason for failure.

Action: See TROUBLE SHOOTING for NFS Kernel error messages.

NFSADMIN0028: unmount failed: <reason>

An attempt to unmount a mounted file system failed. <reason> indicates the reason for failure.

Action: See TROUBLE SHOOTING for NFS Kernel error messages.

NFSADMIN0029: display mount failed: <reason>

An attempt to display currently-mounted file systems failed. <reason> indicates the reason for failure.

Action: See TROUBLE SHOOTING for NFS Kernel error messages.

NFSADMIN0040: Name already in-use: <name>

An attempt to specify a login user ID or group name failed because the specified name has been used for another entry in the PASSWD or GROUP database.

Action: Specify another name or user ID instead.

NLS Error Messages

NLS0001: Invalid argument

An unexpected number of arguments was specified on the command line.

Action: Reference the Command Reference and the retype the command again.

NLS0002: Wildcard characters not supported

The wildcard character '*' is not supported by this command.

Action: Do not specify wildcard characters in a file or directory name.

NLS0003: Cannot open directory <directory>

An attempt to open the directory for reading failed.

Action: Make sure the directory pathname is valid or <directory> is actually a directory.

NLS0027: Invalid interrupt vector <vector>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the TCP/IP Kernel interrupt vector. This value should be between 0x60 and 0x7F.

Action: This problem should not occur. If it does, either NETINFO file is damaged or your system is unstable. Check the NETINFO file's content using an editor or reboot your system.

NLS0028: Invalid multiplexor <mux-id>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the NFS Kernel multiplexor. This value should be between 0xC0 and 0xCF.

Action: Use an editor or TCPSETUP to verify the validity of the NFS_MULTIPLEXOR parameter. If it is not valid, change it to a value in the range mentioned above.

NLS0029: Invalid userid <user-id>

The {INSTALL}\SETUP\NETINFO file contains an invalid default login userid.

Action: Use an editor or TCPSETUP to verify the validity of the USER parameter. If it is not valid, change it to a value in the range mentioned above.

NLS0030: Invalid umask <UMASK>

The {INSTALL}\SETUP\NETINFO file contains an invalid value for the NFS Kernel default UMASK. This value should be between 0 and 0777.

Action: Use an editor or TCPSETUP to verify the validity of the NFS_UMASK parameter. If it is not valid, change it to a value in the range mentioned above.

U2D Error Messages

U2D0001: Internal error flag=value

This message indicates an U2D's internal error.

Action: Reinvoke the command again. If the problem perists, notify Lanera technical support personnel for assistance.

U2D0005: Invalid argument

An unexpected number of arguments has been specified on the command line. The usage of the command is printed after this message.

Action: Reference the Command Reference for correct command syntax and then retype the command with valid arguments.

Problem #1: Communication Path Problem

The most common problem is TCP/IP application fails to connect to a server host. This may be due to one of the following reasons:

a. the cable is loose
b. TCP/IP Kernel is not functioning
c. TCP/IP driver on the server is not functioning

Diagnostics:

First try the PING command to ping either the server host or another host on the same network that is known to be up and running or both to narrow down the cause. If none of them responds to PING command, it is possible that:

a. the cable is loose

Check the cable connection. If you find a loose end, tighten it and use the PING command to check for connectivity again.

b. the network adapter driver failed to talk to the adapter card

Use the PING command to send ICMP echo messages to a host known to be up and running; and then use the NETSTAT command to monitor I/O traffic and error statistics. If output packets are observed but there is no indication of input packets, either the cable is loosen or the network adapter driver fails to communicate with the adapter card.

Go to "a" if you suspect the cable connection.

Another possibility is that the IRQ that was configured for the adapter card conflict with another adapter in your system. You can try to reconfigure the IRQ to another interrupt level.

c. It is possible that the server networking sub-system is no longer functioning. Try to "ping" the server from another PC or host to verify that. If this is the case, notify the server's system administrator for assistance.

Problem #2: System hangs when you invoke a TCP/IP application

This problem is likely due to the memory address conflict between the memory-map on the adapter card and one of the memory management program on your system such as EMM or QEMM. If you have selected to use memory-map for your network adapter card, make to exclude the memory region used by the adapter from EMM or QEMM.

See these programs' manual for instruction how to do this. Alternately, if your network adapter card has an option to disable the memory-map, you can choose to disable it. There is no significant performance penalty from the application level if the memory-map is disabled.

Problem #3: Read Fault

This error message indicates the NFS Kernel has timed out on a read from server operation. A Kernel read operation may follow a read request or an NFS command sent to the server that the Kernel expects a response for it. A read time-out may occur in the one of the following situations:

Cause #1: Communication path problem

First try the PING command to ping either the server host or another host on the same network that is known to be up and running or both to narrow down the cause. If none of them responds to PING command, it is possible that you have a communication path problem.

Diagnostics:

Follow the diagnostic hints in Problem #1 to diagnose and to correct the problem.

Cause #2: NFS server problem

The NFS server machine has experienced some problems and failed to receive previous command and/or to respond to NFS Kernel read request.

Diagnostics:

Check the server machine to see if it is still functioning by using the NFS PING command. If this command fails, it is likely that the server has crashed or hung up.

Alternately, if you have another PC having a mounted file system on the same server, try a directory listing command on the mounted file system to see if the server responds properly. If it does not respond, notify the server's system administrator about the situation.

Problem #4: Write fault

This problem rarely occurs. However, when it occurs, it indicates either the Kernel fails to send a command or data packet or the NFS server encounters a file writing error.

Cause #1: Communication path problem

First try the PING command to ping either the server host or another host on the same network that is known to be up and running or both to narrow down the cause. If none of them responds to the PING command, it is possible that you have a communication path problem.

Diagnostics:

Follow the diagnostic hints in Problem #1 to diagnose and to correct the problem.

Cause #2: Lack of TCP/IP Kernel buffers

TCP/IP Kernel may fail a send request if it runs out of network buffers (MBUFs). To verify if this is the case, use "NETSTAT -m" command to display network buffer usage statistics. If the statistics indicate that the TCP/IP Kernel has denied some memory requests, you may need to reload the TCP/IP Kernel again with more buffer space configured for the Kernel.

Cause #3: NFS server has encountered a write error

It is difficult to tell whether the server has ran into a write error. However, if that is the case, the NFS Kernel is notified by the server. In future release, NFS Kernel will be upgraded to maintain a set of statistics for server's error notifications.

Softerm TCP/IP Kernel Manual

TABLE OF CONTENTS

Part 3: User's References

Part 4: Error Messages

Appendices

Before You Begin

This Softerm TCP/IP /Kernel User 's Guide describes how to install and configure Softerm TCP/IP NFS protocol drivers to support TCP/IP applications in a DOS environment.

Before using this guide, you should feel comfortable using DOS commands and be able to work with files and directories. If you are not familiar with DOS, refer to your DOS manual.

System Requirements

Softerm TCP/IP Kernel requires certain hardware and software. The required hardware and software provide you with a minimum equipment configuration; the recommended components are optional, but make it easier to use TCP/IP Kernel and to improve performance.

Hardware:

You should have the following hardware:

- an 80286- or 80386-based PC-AT
- 512KB of RAM minimum; 640KB is recommended
- 2MB of disk space available
- an Ethernet adapter card with either a NDIS-conformed MAC driver or a Packet driver.

Software:

- DOS 3.1 or later version
- MAC driver for the adapter card in-use

Network:

Softerm TCP/IP Kernel interoperates with all TCP/IP-based implementations on various hardware and software platforms.

Conventions Used in this Guide

The following conventions are used throughout this guide. These conventions are identical for all both the Softerm TCP/IP NFS and Softerm TCP/IP Kernel Installation Manuals.

Keys

Key Combinations

Install Directory

TCP/IP Commands

File Names

Keys

The table below shows the symbols that represent the keys on your keyboard.

Symbol	Key Represented
[Esc]	Escape
[Alt]	Alternate
[Ctrl]	Control
[Backspace]	Backspace
[Space]	Space bar
[F1] - [Fx]	Function keys F1 through Fx
[Return] or **[Enter]**	Return or Enter key

These symbols are printed in boldface whenever they are mentioned in the text.

Key Combinations

If two or more keys are to be pressed simultaneously, the key symbols are linked with a + sign. For example, the following key combination resets a workstation:

[Ctrl]+[Alt]+[Del]

Install Directory

{INSTALL} denotes the directory where Softerm TCP/IP is installed on your system.

For example:

{C:\SOFTERM\TCP_IP}

Commands

Commands have the following general syntax:

Command **Options** **Arguments**

Command
Command is the name of the command that can be entered at a DOS prompt and is case insensitive. If properly setup, a command can also be invoked from a multi-tasking window program such as DESQview or Window 3.0.

Options
Options are the command options that allow a user to control a command action and dictate its behavior. Options come in two forms:

- one or more alphabetic letters followed by a dash (-). The dash is the option switch that signals the command parser that an option follows. An option may have an argument immediately following the letter after a space. The following is an example:
  -A : an option
  -A <opt_argument> : an option with its own argument

- sub-command or keyword that tells the command what to do such as: ADD, DELETE, UP, DOWN.

Options in a command are case insensitive when they are listed in single case (either upper or lower case). When they are listed as a mix of upper case and lower case, they must be entered exactly as they appear in the usage message or in this manual.

Arguments
Arguments are the information necessary for the command to operate on a host name, file name, time-out value, and so on. In this manual and in the command usage messages, the greater and less than signs denote a command argument. However, these signs should not be entered in the command line.

<argument**>** : argument

Arguments may be a string of characters, an integer, an IP address or in any format expected by a command. Some arguments may be optional. The square brackets denote an optional argument as in the following:

[<argument>**]** : optional argument

When an argument is optional, the command may have the default value in case it is not given. Refer to the individual command reference in Chapter 2 of Part 4 for the default values.

The following arguments have special meaning:

Argument	Description
<drive:>	Refers to a disk drive name. You need to specify a drive name only if you are using a file that is not on the default drive.
<dir-name>	Refers to a directory name which may or may not include a <drive> or directory

path. When no drive is specified, the current drive is assumed. When no path is specified, the current directory is assumed.

```
[drive]:[\directory][\directory . . .]\directory
```

<file-name> Refers to a file, and includes any file name or extension. The filename does not refer to a **<local-file>** device or drive name. A filename may or may not include a directory path. When no drive is specified, the current drive is assumed. When no path is specified, the current directory is assumed.

```
[drive]:[\directory][\directory . . .]\filename
```

<host> Refers to a remote host name or IP address. The host name is a string of printable characters, the IP address is expressed in the dot notation format as follows:

```
a.b.c.d
```

a, b, c, and d are decimal digits between 0 and 255 inclusive.

<user> Refers to the remote user login ID. Usually, the specification of the USER entry in the {INSTALL}\SETUP\NETINFO file is taken as the default remote user login ID.

Other
The manual uses the following conventions for command
Conventions options and arguments:

<u>Convention</u> <u>Usage</u>

<item> You must supply the text for any of the variable items shown in enclosed <>. For example, when <file-name> appears, you should type the name of your file. Do not type the <> themselves.

[brackets] Items in brackets are optional. To include optional information, type only the information within the brackets. Do not type the brackets themselves.

... (ellipsis) An ellipsis (...) means that you can repeat an item as many times as necessary.

plurals A plural (word suffixed with -s or -es) means that you can repeat an item as many times as necessary.

separators Unless otherwise specified, you must use spaces to separate commands from their options; for example,

```
rcp -v   file1   file2   remote1:/tmp
```

In this manual, spaces separate commands from their options and arguments.

hexadecimal numbers In this manual, hexadecimal numbers are presented with either a '0x' prefixed or an 'H' suffixed.

File Names

Local system file names in the text are always in all capital letters, such as {INSTALL}\SETUP\NETINFO.

Remote system file names in the text are in lower-case italics, such as /etc/hosts. File names in example command lines are in lower-case as you would use them in a command.

The actual names of local files as you use them on your system can be in upper or lower case. Remote file names must follow the conventions of the remote system.

TCP/IP Networking Concepts

TCP/IP is the popular name for a networking protocol family developed by the US Department of Defense to run on the ARPANET, a packet switching wide area network. ARPANET is now a part of wide area network known as the DoD (Department of Defense) Internet network.

In the early days, TCP/IP was used by the DoD and universities around the country for connecting computers used for research purposes. Today, TCP/IP has become the de-facto standard for connecting heterogeneous network because of its availability on many different types of machines from many different computer manufacturers.

Softerm TCP/IP is an implementation of the TCP/IP protocol suite on the PC/DOS environment. It allows PC-AT or compatible machines running the DOS operating system to connect to host computers on a TCP/IP-based network. Softerm TCP/IP applications can concurrently operate in a multi-tasking environment such as DESQview or Windows 3.X. Softerm TCP/IP can run on Ethernet, Token ring or serial line network, and requires at least a network adapter or a COM port to provide the physical connection with the physical network.

This chapter introduces some basic networking concepts to new network users. Experienced users can skip this section.

Client/Server Model

TCP/IP networking bases on the client and server model. In this model, a server application running on a machine waits for services request from corresponding client application running on any machine on the network. Once the service request is accepted, the server and client applications begins to exchange data and control messages. Many types of services have been defined and implemented in a TCP/IP network. These include file transfer service, virtual terminal
service, electronic mail service, and so on.

Most of Softerm TCP/IP application programs are client-type applications which used by a PC user to obtain the network services from server machines on the network. The server machines are typically multi-user systems, such as UNIX or VMS systems.

Addressing

Every computer on a TCP/IP network is assigned an address called Internet Address or IP address. The Internet address is a way for addressing a particular host on a network.

If your machine is connected to the Internet network, you must acquire a unique Internet address from the DDN Network Information Center (NIC).

DDN Network Information Center
333 Ravenswood Avenue
Room EJ291
Menlo Park, CA 94025
(800) 235-3155
(415) 859-3695

You should contact your network administrator before contacting NIC since she/he may already have some information relating to acquiring and assigning the Internet address.

If your site is not connected to the Internet network, the Internet addresses are locally created and maintained by your network administrator. You do not need to connect to the Internet network if you are using TCP/IP network software to form a Local Area Network (LAN) connecting your in-house machines for local resource sharing.

Note that DDN NIC is also the agency to obtain the RFCs (Recommend For Changes) that defines the TCP/IP protocol suite.

Internet Address

The Internet address format typically consists of four parts of Format decimal digits separated by dots, e.g. 123.0.1.5. For this reason, the IP address is often referred to as the dot-notation IP address. Each part of the Internet address is internally represented by a byte or octet. Therefore, they range from 0 to 255.

Although the Internet address is conventionally written as four parts. It is actually used to contain two types of information: the network ID and the host ID. The network ID part is used to identify a particular network within an internetwork such as the Internet network. The host ID part is used for identifying a particular local host within a local network.

Internet Address Classes

There are a number of ways that a network ID and a host ID can share the four octets that make up the Internet address. This further leads to the classification of Internet address into various classes as follows:

Class A Internet Address
Class B Internet Address
Class C Internet Address
Class D Internet Address
Class E Internet Address

Class A Internet Address

In a class A address, the highest-order bit of the left-most byte is set to 0. The remaining bits of the left-most byte (7 bits) are used to represent the network ID. The rest are used to represent the host ID. This address class allows up to 127 network addresses and over 16 million host addresses. This class of address is suitable for large network such as the Internet.

```
0          8             16             24             32
0nnnnnnn      hhhhhhhh       hhhhhhhh       hhhhhhhh
```

nnn... : network ID
hhh... : host ID

Class B Internet Address

Class B address has the highest-order bits set to 10. The rest of the left-most two bytes are used for the network ID. The remaining two bytes are then used to contain the host ID. This class allows up to about 16,000 network IDs and about 64,000 host IDs. This class of address is suitable for networks of large organization with approximately 64,000 hosts or less per network.

```
0         8             16             24             32
10nnnnnn      nnnnnnnn      hhhhhhhh      hhhhhhhh
```

nnn... : network ID
hhh... : host ID

Class C Internet Address

Class C has the highest-order bits set to 110, the remaining bits of the left-most three bytes are used for the network ID and the right-most byte is used for the host ID. This class of address is suitable for a large network of small organizations of 255 hosts or less.

```
0          8              16              24              32
110nnnnn      nnnnnnnn      nnnnnnnn      hhhhhhhh
```

nnn... : network ID
hhh... : host ID

Class D Internet Address

Class D address has the highest-order bits set to 1110, the remaining 28 bits are used to represent the multicast address, which is used to address a group of hosts with the same Internet address.

```
0         8              16              24              32
1110xxxx        xxxxxxxx        xxxxxxxx        xxxxxxxx
```

xxx... : multicast address

Class E Internet Address

This class of address are reserved by the NIC for future assignment. It has the highest-order bits set to 1111.

```
0            8                 16                        24                        32
1111RRRR     RRRRRRRR     RRRRRRRR     RRRRRRRR
```

RRR... : reserved

Subnet Mask

A network may be further divided into subnetworks for administrative purposes such as organizing the network based on the departments, groups, buildings, or different types of transmission medium such as Ethernet, or Token ring, or serial line. These subnetworks are then connected together by means of IP gateways (or routers).

The subnet ID in an Internet address is taken from the unused highest-order bits of the host ID part of the Internet address. The number of bits used for subnet ID can be figured out based on the maximum allowable number of hosts per subnet, which decides how many bits in the host ID part is sufficient for host ID. The unused highest-order bits of the host ID part then can be used to contain the subnet ID.

As an example, consider a network of class B address of which network address is 129.127 and assume 4 bit of the host ID is used for subnet mask. The network 129.127 can have up to 16 subnetworks (4 bits), each of which can address up to 4096 hosts.

```
0          8                16             24                32
10000001       011111111       01010000       00000001

                                Host ID = 1
                                Subnet ID = 5
```

The above example shows the address of a host in subnet 5 of the network 129.127. The host ID is 1. The host's Internet address is 129.127.80.1. In this case, the subnet mask is 240 (11110000).

IP Broadcast

By default, the protocol driver uses the IP broadcast address with the host ID part of all 1's and the network ID part (plus the subnet part, if any) containing the local network ID.

Host Name

Host name is a user-friendly way to reference a host on the network. It is basically a mapping of the Internet address to a string of characters which can be arranged into a name so that a host address can be easily remembered. The Internet address-to-host name mappings are contained in the {INSTALL}\ETC\HOSTS file, also known as the host table. It is in the form of a string of characters of up to 64 characters long. TCPOpen applications automatically detects whether the user specifies a host name or an Internet address and performs the appropriate mapping if the specified address is a host name. The conversion of a host name to an   Internet address is called host name resolution.

Host Name Resolution

Normally, host name resolution is done locally by using the local host table. However, when a domain name resolver is configured Ä by specifying a name server (NSIPADDRESS parameter) in the file {INSTALL}\SETUP\NETINFO Ä a TCPOpen application will first contact the name server to resolve a host name. Only when this attempt fails, the application then uses the local host table for resolving the host name.

Domain Name

The concept of Domain Name Service arises due to the need to manage host names in a large network, such as the Internet. The basic idea is to hierarchically organize the network into domains and sub-domains. A name server is designed to have a complete knowledge of the domains it serves and interacts with other name servers to resolve names outside of the its domains. When name servers are used to resolve host names, a host name must be always appended with a complete domain name. For example, a host A which is known to belong to domain B, will have a complete host name as : A.B. The dot, in this case, is used to signal a domain name followed. If the domain B is, in turn, a sub-domain of another domain (e.g. C), the host name for A is then A.B.C.

When using a name server for name resolution, your host's domain name must be included with the host name. There are two ways you can specify the local domain name in your host name. You can either:

- specify your host's domain name in the DOMAIN entry in the {INSTALL}\SETUP\NETINFO file, or

- specify your host's domain name in your host name, which is assigned to the HOST entry in the {INSTALL}\SETUP\NETINFO file.

If both specified, the domain name specified in the DOMAIN entry will be used.

References

Users who are interested in better understanding of TCP/IP technology can find more information in the following books:

- Internet Protocol Handbook, Volume 1-4. These books are available from the DDN NIC.
- Internetworking with TCP/IP, Volume I and II, by Douglas E. Comer and David L. Stevens.

TCP/IP Network Components

Softerm TCP/IP consists of three major components:

- the Protocol driver (also referred to as "kernel"),
- the MAC driver,
- and TCP/IP application programs

Protocol Driver

The protocol driver is an implementation of the TCP/IP protocol stack as shown in figure 1.2.1. It communicates with application programs through a Berkeley-compatible socket interface. The protocol driver is hardware independent. It supports interface to either NDIS, ODI or Packet drivers that directly interact with the network adapters for the purpose of transmission and reception of data packets to and from the physical network.

One of the major responsibility of the protocol driver is to establish logical connections with remote server applications on the request of local application programs and then buffers incoming and outgoing data exchange between the two processes. Other responsibilities include the ability to route information to the correct destination.

Note that the protocol driver comes with three different versions, each with a different type of data link interfaces as shown in figure 1.2.1. Only one of these interface types can be used at a time. Each version of the protocol driver supports one interface to a corresponding network adapter driver type.

FTP VNT DRTAR Application

SOCKET INTERFACE
TCP UDP ICMP RAW Transport Network
ARP IP ROUTE
ND PD OD SLIP PPP* Data Link
MAC Driver
Hardware Devices Physical

* not yet available in release 2.1

Figure 1.2.1. TCP/IP Protocol Driver (Kernel)

Network Layer

The network layer has two components. The main component is the Internet Protocol (IP). The Internet protocol performs fragmentation of transmitted packets and reassembly of received packets as necessary. IP depends on a default gateway for routing packets intended for stations outside of the local network. The second component is the ARP (Address Resolution Protocol) which provides address translation service for the IP when a physical (hardware) address of a destination host is unknown.

Note that the IP protocol is not capable to redirect incoming packets intended for another host. Therefore, TCPOpen protocol driver is incapable to function as a router. This functionality is available in an optional package (TCPOpen/Router) which allows a PC to function as an IP router. Incoming packets with routing-related options (loose source routing, strict source routing and record routing) will be responded with unreachable destination ICMP messages.

Transport Layer

The transport layer implements two important protocols that provides connection-oriented (TCP) and connectionless (UDP) data transport services to applications.

TCP is an implementation of RFC-793. It supports multiple end-to-end connections and implements the MAXSEGSIZE option which allows the local host to negotiate the maximum packet size with the remote host at the time of connection establishment. TCP's send and receive window size can be configured using TCP_SENDWIN and TCP_RECVWIN entries in the {INSTALL}\SETUP\NETINFO file. By default, they are set to 2.5 Kbytes.

UDP is an implementation of RFC-768. It permits the application to send datagram to an addressed remote host without having to first make an end-to-end connection. Due to buffer space limitation, the maximum datagram can be queued at a socket is 8 Kbytes and the maximum transmit datagram's size is 4 Kbytes.

ICMP and **RAW** are not transport protocols. However, they can be accessed directly from the socket layer and use IP services. ICMP stands for the Internet Control Message Protocol (RFC-792). It recognizes and responds to all the message types specified in the protocol. RAW module provides an application interface to send raw IP datagrams. Raw IP datagrams have no transport-layer header.

The protocol driver can be dynamically loaded and unloaded on demand since it is a TSR program. This means a user can load it when s/he needs to use TCP/IP applications and then unload it, without rebooting the system, when s/he needs to reclaim the memory occupied by the protocol driver for executing another application program. Appendices A to C of this manual shows how to load, configure and unload the protocol driver.

Data Link Layer TCPOpen protocol driver recognizes packets encoded according to RFC 894 for the DIX Ethernet network, RFC 1042 for the Token Ring network and the SLIP protocol for the serial line network.

		Packet Sizes		
	Layers	IP		MAC
Mediums	Min.	Max.	Min.	Max.
DIX Ethernet	60	1500	60	1536
Token Ring (802.5)	28	2044	28	2088
SLIP	1	1024	N/A	N/A

Table 1.2.1 - Limitation on packet lengths

Table 1.2.1 shows the minimum and maximum packet sizes in relation to the transmission medium used and at different protocol levels.

MAC Driver

The MAC driver may either be a NDIS driver or a Packet driver and is not a part of the TCP/IP protocol driver. Most network adapter vendors supply these drivers along with their adapter packages. However, for your convenience, we include in our distribution package the NDIS drivers for the most popular network adapters from the leading network adapter vendors (3Com and Western Digital). In addition, we also include a diskette of some of the Packet drivers available from Clarkson University.

Again, we distribute these drivers for your convenience, in case you could not find one. Neither we assume the responsibility or liability resulting from using these drivers nor we will provide support for them. However, we will keep track of the vendors' development concerning the drivers and try to ship the latest version of the drivers at the time of future TCPOpen releases.

Application Programs

Softerm TCP/IP comes with a set of applications that allows a user to obtain many services provided by a server machine on the network. These services include file transfer, remote command execution, virtual terminal, remote backup/restore, remote printing, etc . . .

These commands are documented in Softerm TCP/IP Application User's Guide.

Softerm TCP/IP Kernel Installation

This chapter explains how to install SOFTERM TCPOpen/Kernel and provides you with some information about the operating environment.

INSTALLATION PROCEDURES

You will install SOFTERM TCPOpen/Kernel from a floppy drive. The Install program copies the Softerm TCP/IP Kernel files from the distribution diskette to your PC workstation. The files will be decompressed and verified for data integrity before being transferred to the destination drive.

To install Softerm TCP/IP Kernel, follow these steps:

- **Step 1:** Insert the SOFTERM TCPOpen/Kernel Install Disk #1 into drive A:

- **Step 2:** At the DOS prompt, type:

  A:INSTALL **[Enter]**

  The following window displays:

  ```
  TCPOpen Kernel version 2.1 Installation


          LANERA CORPORATION

          San Jose, CA (USA)
          Copyright 1990-1991
          All rights reserved.

          Serial Number: TCPO-520-1234567
  ```

- **Step 3:** Press any key to continue. The following window appears:

  ```
      This program installs TCPOpen /KernelVersion 2.1 onto your computer system and verifies
      the integrity of the distribution files. You may press the [Esc] key at any time to abort the
      installation.



          Press [Esc] to quit, any key to continue ...
  ```

- **Step 4:** The next window asks for the destination disk drive:

  ```
  Please specify the disk drive where you wish to install TCPOpen/Kernel:
          Drive A:
          Drive B:
          Drive C:
          Drive D:

      Use the [Up Arrow], [Dn Arrow], [Pg Up], & [Pg Dn] keys to select the destination disk drive on
      which you wish to install TCPOpen /Kerneland then press the [Enter] key.
  ```

- **Step 5:** The next window asks for the destination directory:

  ```
  Please specify the destination directory where TCPOpen /Kernel will be installed:

                  Which Directory?
                  \TCPOPEN
  ```

  The default \TCPOPEN directory is displayed on the screen. Press [Enter] to proceed with the installation or press [Esc] and then type in another directory of your choice.

After completing Step 5, the following window will appear:

```
Transferring Files      Please Wait . . .

        Reading:        Library File > ARP.EXE
        Verifying:      ARP.EXE
                          Reading:              Library File > IFCONFIG.EXE
        Verifying:      IFCONFIG.EXE
        Making: TCPOPEN
        Making: BIN
        Writing: C:\TCPOPEN\BIN\ARP.EXE
        Decompressing:          ARP.EXE
        Writing: C:\TCPOPEN\BIN\IFCONFIG.EXE
        Decompressing:          IFCONFIG.EXE
        Reading:        Library File > NETSTAT.EXE
        Verifying:      NETSTAT.EXE
                :
                :
                :
```

After all files have been transferred to the destination directory, the Install program will automatically modify the AUTOEXEC.BAT file on your computer. For example, if you are installing TCPOpen/Kernel   into C:\TCPOPEN directory, you will see the following display:

```
The node "C:\TCPOPEN\BIN" has been added to the existing
path command.
The entire line "SET TCPOPEN=C:\TCPOPEN" has been added.

The AUTOEXEC.BAT file on drive C: has been
created/modified. The original AUTOEXEC.BAT
file has been renamed to AUTOEXEC.BAK.


 Press any key to continue.
```

The Install program will then display the following screen:

```
Please press any key to complete the TCPOpen /Kernel
installation procedure. You should then perform a system
reboot and run TCPSETUP to configure TCPOpen/Kernel on
```

```
your PC. Refer   to Chapter 2 of Part 2 of the manual for more
information on how to use the TCPSETUP program.


Press any key to continue ...
```

- **Step 6:** Reboot the PC.
- **Step 7:** Execute the TCPSETUP program to configure the TCPOpen/Kernel on your PC.

Installed Files

After completing the installation, you should have the following files installed on your system. Assume that the drive and directory where the files are installed is represented by {INSTALL}.

File/Directory	Comment
{INSTALL}	This directory contains the entire Softerm TCP/IP Kernel product files
KERNEL.UDP	A text file which contains all the latest product information that may not be found in the documentation or release notes.
{INSTALL}\ETC	This directory contains networks databases.
HOSTS	Host name-to-Internet address database
NETWORKS	Network name-to-network ID database
PROTOCOL	Internet Protocols' ID
SERVICES	Network services database
{INSTALL}\SETUP	This directory contains setup files.
MODEMS	Modem setup file for use with SLIP
NDIS.INF	NDIS driver configuration database
NETINFO	TCPOpen configuration database
PACKET.INF	Packet driver configuration database
PHONES	Name-to-phone number mapping database
SLIP.INF	SLIP driver configuration database
PROTOCOL.INI	NDIS Protocol Manager's configuration file
{INSTALL}\BYU	This directory contains BYU packet drivers.
PDSHELL.OBJ	BYU LAN driver for Novell Netware shell generation
PDSHELL.LAN	BYU LAN driver for Novell Netware shell generation
APDLAN.OBJ	BYU LAN driver for Novell Netware 2.1 bridges
BPDLAN.OBJ	"
CPDLAN.OBJ	"
DPDLAN.OBJ	"
PDLAN.LAN	"
READ.ME	BYU packet driver installation instructions
{INSTALL}\NDIS	This directory contains NDIS drivers.
ELNK.DOS	NDIS driver for 3COM Etherlink adapter
ELNKII.DOS	NDIS driver for 3COM EtherlinkII adapter
ELNKPL.DOS	NDIS driver for 3COM Etherlink Plus adapter
ELNK16.DOS	NDIS driver for 3COM Etherlink/16 adapter
ELNKMC.DOS	NDIS driver for 3COM Etherlink/MC adapter
MACWD.DOS	NDIS driver for Western Digital W8003E series adapter s
PRO.MSG	3COM Protocol Manager's message file
PROH.MSG	3COM Protocol Manager's help message file
PROTMAN.DOS	3COM Protocol Manager driver
TLNK.DOS	NDIS driver for 3COM Tokenlink adapter
BINDSTATUS.EXE	
UNBIND.EXE	
{INSTALL}\BIN	This directory contains TCPOpen /Kernel executable files.
ARP.EXE	ARP table maintenance program

ARP.PIF	Windows 3.x PIF file for ARP.EXE
ARP.ICO	Windows 3.x icon file for ARP.EXE
IFCONFIG.EXE	Network interface configuration program
IFCONFIG.PIF	Windows 3.x PIF file for IFCONFIG.EXE
IFCONFIG.ICO	Windows 3.x icon file for IFCONFIG.EXE
NETSTAT.EXE	Network statistics program
NETSTAT.PIF	Windows 3.x PIF file for NETSTAT.EXE
NETSTAT.ICO	Windows 3.x icon file for NETSTAT.EXE
NETMON.EXE	Menu-driven network statistics program
NETMON.PIF	Windows 3.x PIF file for NETMON.EXE
NETMON.ICO	Windows 3.x icon file for NETMON.EXE
PING.EXE	Network fault-isolation tool
PING.PIF	Windows 3.x PIF file for PING.EXE
PING.ICO	Windows 3.x icon file for PING.EXE
SLCONFIG.EXE	Serial-line interface configuration program
SLCONFIG.PIF	Windows 3.x PIF file for SLCONFIG.EXE
SLCONFIG.ICO	Windows 3.x icon file for SLCONFIG.EXE
TCPIP_ND.EXE	TCP/IP driver with NDIS interface
TCPIP_OD.EXE	TCP/IP driver with ODI interface
TCPIP_PD.EXE	TCP/IP driver with Packet driver interface
TCPIP_SL.EXE	TCP/IP driver with serial line interface
TCPSETUP.EXE	TCPOpen setup program
TCPSETUP.PIF	Windows 3.x PIF file for TCPSETUP.EXE
TCPSETUP.ICO	Windows 3.x icon file for TCPSETUP.EXE
ECONFIG.EXE	Configuration utility for Novell's Netware shells & drivers
NETBIND.EXE	3COM Protocol binding program
LOADTCP.BAT	A batch file created by TCPSETUP used to load the protocol driver (kernel)
RMTCP.BAT	A batch file created by TCPSETUP used to unload the protocol driver (kernel)
READPRO.EXE	
UBTCP.EXE	

{INSTALL}\PDS This directory contains Packet drivers.

3C501.COM	Packet driver for 3COM Etherlink adapter
3C503.COM	Packet driver for 3COM EtherlinkII adapter
3C505.COM	Packet driver for 3COM 3C505 adapter
3C507.COM	Packet driver for 3COM 3C507 adapter
3C523.COM	Packet driver for 3COM 3C523 adapter
IPXPKT.COM	IPX tunnel packet driver
NB.COM	NETBIOS tunnel packet driver
TERMIN.COM	Program used to unload the Packet driver from memory
TIARA.COM	Packet driver for Tiara LanCard adapter
UBNICPC.COM	Packet driver for Ungermann-Bass NICpc adapter
UBNICPS2.COM	Packet driver for Ungermann-Bass NICps2 adapter
WD8003E.COM	Packet driver for Western Digital W8003E series adapters (or SMC 8003E)

{INSTALL}\ODI This directory contains ODI drivers.

NW1000.COM	ODI driver for 3COM NW1000 adapter
SMCPLUS.COM	ODI driver for SMC 8003E series adapters

Network Administration Commands

The Softerm TCP/IP Kernel package provides a number of network administration commands. These commands allow the user to display network-related information and track down problems that may occur on a local system and on the network. The following is a summary of these commands:

- ifconfig configure the network interfaces.
- slconfig configure the serial line (SLIP) interfaces.
- netstat display statistics collected by various protocol layers.
- route maintain the protocol driver routing table.
- arp maintain the protocol driver ARP (Address Resolution Protocol) table.
- pingverify connectivity with remote hosts or gateways.

Ifconfig - Interface Configuration

The ifconfig command configures a network interface. The interface may be either an Ethernet controller, a serial line port, or any other supported device. It can be used to bring up, bring down, or configure a network interface.

Softerm TCP/IP kernel (or protocol driver) supports three popular standard network interfaces for Ethernet and Token Ring adapters: Packet Driver interface, NDIS interface and ODI interface. These interfaces allow the TCPOpen   kernel to operate on a variety of network adapters from many hardware vendors.

INVOCATION

The ifconfig command has the following syntax:

```
IFCONFIG <interface>
        [ [<address>] [<dest-addr>] [UP] [DOWN] [NETMASK <mask>] ]
        [ TYPE <adapter-type>]
        [ TRAILERS   | -TRAILERS]
        [ ARP | -ARP ]
        [ MTU <mtu> ]
        [METRIC <n>]
```

Each interface is associated with a name. An interface name is constructed with:

- two letters to identify the type of interface. The valid types are:

pd	Packet driver interface
nd	NDIS driver interface
od	ODI driver interface
sl	SLIP driver interface

- a decimal digit unit number that identifies a sub-device of a specific interface type.

The interface type and unit number combination identifies a particular network interface device.

For example, a name for a Packet driver Interface would be pdí where pd is the type of interface, a Packet driver, and í is a decimal digit specifying a particular sub-device within the Packet driver interface type. A complete description of the ifconfig command's parameters can be found in the Command Reference.

Ifconfig commands:

Display the Status of an Interface
Network IP Mask
Subnet Mask
Internet (IP) Broadcast
Configure a Network Internet Interface

Display the Status of an Interface

From time to time, the user may need to check the status of an interface if communications between a local system and a remote host appear to have a problem. Assume you have just loaded the tcp/ip kernel and have not yet configured the interface. Furthermore, assume your hardware interface is an Ethernet controller and is controlled by a Packet driver.

To check the status after downloading the tcp/ip kernel, type:

```
C> ifconfig pd0
Interface (pd0) Status:   <BROADCAST, NO TRAILER>
Adapter Type:   DIX
Internet Address:         0.0.0.0
Network ID Mask:          0x0
IP Broadcast Address:     0.0.0.0

C>
```

Interface Status This line reflects the status of an interface. The following keywords reveal the states of an interface:

UP indicates that the interface is ready to transmit and receive packets to and from the network.

DOWN indicates that the interface is down and not communicating with the network. All transmit or receive packets are dropped by the interface driver when the interface is marked down.

BROADCAST indicates that this is an interface to a broadcast network such as an Ethernet network.

POINT-TO-POINT indicates that this is an interface to a point-to-point network such as a serial line network.

TRAILER see Enable/Disable Trailer Encapsulation

Internet Address This line displays the configured Internet address of this interface. If the interface (pd0) has never been configured, this field shows all zeros. Otherwise, it shows the Internet address in dot notation.

Network ID Mask

The protocol driver uses the network ID mask to determine the network number from an interface IP address. A network (IP) address can be one of three classes as shown in figure 2.3.1. Depending on the class of the interface IP address, the network ID mask must be set to correctly reflect the width of the network number field. Figure 2.3.1 shows the network masks for various network address classes.

CLASS A Network Address

0xxxxxxx

NETW HOST HOST HOST

Network ID mask should be: 0xff000000.

CLASS B Network Address

10xxxxxx

NETW NETW HOST HOST

Network ID mask should be: 0xffff0000.

CLASS C Network Address

110xxxxx

NETW NETW NETW HOST

Network ID mask should be: 0xffffff00.

Figure 2.3.1. Network masks for various network address classes

Internet address classes are coded using the left-most few bits of the Internet address. The following table summarizes the Internet address class coding scheme:

CLASS	Leftmost Byte
A	0XXXXXXXX
B	10XXXXXXX
C	110XXXXXX

Table 2.3.1. Internet Address Classes

Based on the encoding scheme, the protocol driver automatically figures out and sets the network mask for an interface when it is assigned an Internet address.

Subnet Mask

If your PC is attached to a network of which address has a subnet ID. You need to explicitly specify the network mask for both the network ID part and the subnet ID part in the ifconfig command line. For example, if your network address is 64 (class A), and the next four bits are used as the subnet ID, then your network mask becomes 0xFFF00000 instead of 0xFF000000 as it would normally.

Internet (IP) Broadcast

TCP/IP uses the Internet broadcast address to broadcast a message on the network identified by the network number Address of the attached interface. All hosts residing on the same network should receive broadcast messages destined for that network.

According to RFC-919 (see References), the Internet broadcast address is the address with the host field as all 1's. However, the Berkeley 4.2 BSD TCP/IP implementation uses the host field as all 0's. You can use the ifconfig command to modify the broadcast address if necessary.

Configure a Network Interface

Before an interface can communicate with other hosts on the Internet network, it has to be configured properly. The following information is associated with an interface and must be configured.

Set Interface Address

This is the address that identifies the interface to the other Internet hosts on the network.

Interface Internet Broadcast Address

This is the address used when transmitting an IP broadcast message on the Internet network.

Network Mask

This mask reflects the network number field in an Internet address. Figure 2.3.2 shows a sample configuration. remote1, remote2, and yourÄPC are host names that correspond to the IP addresses shown. The association of a host name with an IP address is done by the HOSTS database (see Part 4 Chapter 1) on every machine attached to the network or by one or more hosts designated as name servers. On the DOS system using the TCPOpen TCP/IP package, the HOSTS database is under the {INSTALL}\ETC directory.

Furthermore, the network number shows that the network address of the above network is class A since the leftmost bit is 0. The default netmask for a class A network address is 0xff000000.

Decimal	Hexadecimal	Binary
90 f	0x60 f	01100000

The following command assigns an Internet address to your interface:

```
C> ifconfig   pd0   your_pc
```

Alternately, you can specify the dot-notation IP address in place of the host name:

```
C> ifconfig   pd0   90.0.0.5
```

After the interface is configured, the interface status display becomes:

```
C> ifconfig pd0
Interface (pd0) Status:   <BROADCAST, NO TRAILERS>
Internet Address:         90.0.0.5
Network ID Mask:          0xff000000
IP Broadcast Address:     90.255.255.255

C>
```

Set the Network Mask

By default, the protocol driver automatically sets the network ID mask based on the class of the IP address assigned to an interface. The driver does this by checking the most significant bits of the left most byte of the assigned IP address. Then, it sets the interface network mask according to the scheme shown in figure 2.3.1.

However, using the ifconfig command, you can revise the network ID mask for an interface. For example, to change the network ID mask to 0xffff0000 use the command:

```
C> ifconfig   pd0   your_pc netmask   0xffff0000
```

WARNING: Always set the netmask and the IP address on the same ifconfig command. Setting it separately using another ifconfig command will not work.

Set the Internet Broadcast Address

Similarly, the IP broadcast address is automatically configured by the protocol driver by setting the host part of your interface IP address to all 1's. However, the ifconfig command allows you to change the address if necessary.

For example, to change the broadcast address to 90.0.0.0 which is used by 4.2 BSD's TCP/IP implementation, use the command:

```
C> ifconfig   pd0   broadcast   90.0.0.0
```

Bring Up the Network Interface

The next step is to bring up the interface. Bringing up an interface usually causes the interface driver to invoke the following actions:

- start the network interface hardware.
- mark the interface as up. This enables higher protocols to use the interface for transmitting and receiving data packets.

The following command brings up an already configured interface:

```
C> ifconfig   pd0   up
Interface (pd0) Status: <up,broadcast,notrailers,running>
Internet Address:        90.0.0.5
Network ID Mask:         0xff000000
IP Broadcast Address:    90.255.255.255

C>
```

Bring Down a Network Interface

An active network interface can be brought down with the command:

```
C> ifconfig   pd0   down
```

After the interface is brought down, all network activities of the interface are halted, and all transmit and receive packets are dropped and discarded.

Enable/Disable Address Resolution Protocol (ARP)

The ifconfig command can disable or enable the ARP protocol, which maps the link-level (Ethernet) address to a network-level (Internet) address. By default, the ARP protocol is enabled.   Note that the ARP works only on broadcast-based networks   such as Ethernet or Token Ring.

The following command enables ARP mapping:

```
C> ifconfig   pd0 arp
```

The following command disables ARP mapping:

```
C> ifconfig   pd0   -arp
```

Enable/Disable Trailer Encapsulation

Trailer encapsulation is a link-level technique that minimizes the number of memory-to-memory copy operations by the receiver. Since other host link-level drivers may not provide support for this encapsulation technique, it should be used with caution to avoid incompatibility. When encapsulation is enabled, the link-level driver encapsulates each out-going packet before sending it to the network. Encapsulation is disabled by default.

The following command enables trailer encapsulation:

```
C> ifconfig   pd0   trailer
```

The following command disables encapsulation:

```
C> ifconfig   pd0   -trailer
```

Set the Adapter Type

The adapter-type setting determines the type of packet header used by the link-level driver to encapsulate and deencapsulate transmit and receive packets. This parameter can only be set for the Packet and NDIS interface drivers. By default, this setting is obtained from the MAC driver. However, the user can change this default setting by specifying the type keyword in the ifconfig command as follows:

```
C> ifconfig   pd0   type 802.5
```

This causes the pd0 interface to use the Token Ring packet header. See ifconfig in the Command Reference, Part 4 of this manual, for a list of valid adapter types.

Set the MTU

The ifconfig command can be used to reduce the default MTU (Maximum Transmission Unit), which reflects the maximum IP packet size an interface driver can send or receive. Note that MTU is not the same as the maximum network packet size supported by the physical device. This setting allows the interface driver to allocate intermediate transmit and receive buffers to accommodate the largest possible packet size. Otherwise, packets of which size is larger than the MTU are dropped. The MTU can be set to another value by specifying   the -M option when loading the protocol driver. The ifconfig command can only change the MTU to a smaller value than the one set at load time.

SLCONFIG - Slip Configuration

The slconfig command configures the serial interface that supports TCP/IP over serial ports. This command only works on the protocol driver that includes support for the serial interface. The netstat command with option -i shows whether the installed protocol driver has a serial interface. The name of this interface is slX, where X is the interface unit number.

With the SLIP interface, a user can connect to a remote host over a telephone line by using a modem attached to a serial port. The advantage of this type of dial-up connection over KERMIT or any other communication protocol is that users can use familiar TCP/IP applications. However, the SLIP protocol is considerably slower than other communication protocols because of the high overhead associated with the TCP acknowledgment algorithm.

INVOCATION

slconfig has the following general syntax:

```
SLCONFIG     <interface>
        [ SET   [<baud>],[<data>],[<stop>],[<parity>] ]
        [ ATTACH <modem-entry> ]
        [ DETACH ]
        [ COMMAND <modem-command>[time-out] ]
        [ DIAL <phone>   [<time-out>]]
        [ HANGUP ]
        [PORT <port-id>]
```

<interface>
is the name of the serial line interface: sl0, sl1 . . .

SET, ATTACH, DETACH, COMMAND, DIAL, HANGUP,PORT
are slconfig's sub-commands that tell it what to do.

<baudrate>
is one of the following speeds: 300, 1200, 2400, 4800, 9600, 19200, and 38400.

<data>
is the number of data bits, which may be one of the following values: 5, 6, 7, or 8.

<stop>
is the number of stop bits, which may be one of the following values:

1 Ä	for 1 stop bit
2 Ä	for 2 stop bits

<parity>
is the number for parity check, which is one of the following values:

N	Ä	for no parity
O	Ä	for odd parity
E	Ä	for even parity
M	Ä	for mark parity
S	Ä	for space parity

<modem-entry>
is the name of a modem entry in the modem setup file, {INSTALL}\SETUP\MODEM.SU.

<modem-command>

is Hayes-compatible modem commands.

<time-out>
is modem command time out in seconds. The default is 5 seconds for the COMMAND command and 40 seconds for the DIAL command.

<phone>
is either a phone number prefixed by the at sign (@) or a phone entry in the {INSTALL}\SETUP\PHONES file.

<port-id>
is serial port number, and it must be 1, 2, 3, or 4.

SAMPLE USAGE

The following paragraphs explain how this command configures the serial port with the TCPOpen protocol driver.

Assign COM port To Serial Line
By default, every serial line interface is associated with a COM port as follows:

Interface		
sl0	-	COM1
sl1	-	COM2

However, the user can reassign a COM port to any serial line interface as necessary. This is done using the slconfig PORT sub-command as follow:

```
slconfig <interface> port <port-id>
```

For example,

```
C> slconfig sl0 port 2
```

The above command associates COM port 2 with the serial line interface sl0, which interfaces with COM port 1 by default. Once port 2 is associated with sl0, any future slconfig command issued on sl0 is directed to COM port 2.

Note that a serial line interface must be brought down before being switched to another port, and then brought up after the new port is assigned to it. This is to ensure that all previous COM port activities are properly terminated.

Serial Line Baudrate and Data Format Setup

By default, the serial line is set to operate:

- at the speed of 9600 baud.
- with the data format as: 8 data bits, 1 stop bit, and no parity.

The COM port can be reconfigured to operate at a different baud rate and data format if necessary. For example, the following command sets the serial interface sl0 to a baud rate of 2400, 8 data bits, 1 stop bit, and no parity:

```
C> slconfig sl0   set   2400,8,1,N
```

This command can also alter the configuration of the serial port after the TCP/IP driver is installed. The command arguments can be omitted; however, the commas must be reserved. If an argument is omitted, its value will not be affected by the execution of the slconfig command.

Note that changing the port configuration when the interface is up and running may damage on-going packet transmission or reception.

Read a Port Setting

The following command displays the current port settings of a serial port, COM1, associated with a serial interface, sl0.

```
C> slconfig   sl0

 sl0 Settings:

Port            :        COM1
Baud rate       :        9600
Date            :        8
Stop            :        1
Parity          :        None
C>
```

Attach and Configure a Modem

To set up a modem for TCP/IP usage, enter the following command:

```
slconfig <interface> attach <modem-name>
```

attach - is the command for downloading and executing the modem configuration specified by the <modem-name> entry.

modem-name is the name of the modem configuration entry in the {INSTALL}\SETUP\MODEMS file (see Chapter 1 of Part 3).

For example, the following command attaches a Smart modem to the sl0 interface, configures the modem according to the setup parameters, and issues the modem commands in the MODEMS file:

```
C> slconfig sl0 attach Smartmodem
```

Smartmodem is the entry name in the MODEMS file for the Smart modem.

Detach a Modem from SLI

To detach the modem, use the following command:

```
slconfig <interface> detach
```

For example, the following command detaches the modem from the serial line interface:

```
C> slconfig sl0   detach
```

Dial a Remote Host

After connecting a modem to the serial line interface, the user can dial any remote system by using the following command:

```
slconfig <interface> dial <phone> [<time-out>]
```

<phone> may either be a phone number prefixed by the at sign (@) or a phone entry in the PHONES database. For example,

```
C> slconfig sl0   dial   @1234567
```

or,

```
C> slconfig sl0   dial slremote
```

The first example dials the phone number 1234567 using the Smartmodem modem. Note that the at sign (@) signals the slconfig program that a phone number follows. The second example shows the use of a name for dialing. slremote is   the name of a phone entry in the PHONES database file.

The <time-out> argument is optional. When specified, it indicates the maximum time slconfig waits for the modem response. If the time-out occurs during a call, the call is terminated and slconfig reports a time-out condition. The default time-out is 40 seconds.

Hanging Up

To hang up a phone in the middle of a call or when the remote host does not answer the call, use the following command:

```
slconfig <interface> hangup
```

For example, the following command hangs up the call in progress.

```
C> slconfig sl0 hangup
```

WARNING

A sudden loss of power while the modem is connected
requires a reset of the interface driver by bringing it
down and bringing it back up again. This action resynchronizes the states of the driver and the modem.

A phone connection loss requires the HANGUP
command to be issued before redialing.

Sending a Command to Modem

The user can issue a Hayes-compatible command to the modem by using the slconfig's COMMAND sub-command.   The following command issues a modem command to the attached modem:

```
slconfig <interface> command <modem-command> [<time-out>]
```

For example,

```
C> slconfig sl0 command ATS0=2
```

The above command sets the modem to auto-answer mode. Modem commands can only be issued one at a time. There is a default time-out of 5 seconds for every command issued. However, when a command is expected to take longer to execute (such as dialing), the user can select a longer time-out by specifying the argument <time-out>.

NETSTAT - Network Statistics

The netstat command displays the connection table, the routing table, and various statistics collected by the protocol driver. The command can also display the network interface hardware configuration and collected statistics. This command can monitor packet transmission and reception through various protocol layers within the protocol driver to track down failures in the protocol stack.

INVOCATION

NETSTAT   [-aimnqr] [-pdo <interface>] [<interval>] [-s[<protocol>]]

A summary description of this command can be found in the Command Reference of Part 3. Note that netstat command option letters are case-sensitive. That means they must be entered as shown.

Show Active Connections

To display active connections, invoke:

```
                C> netstat -a

        Active Internet connections (including servers)
        Proto    Recv-Q Send-Q Local Address    Foreign Address         (state)
        tcp      0        0        dave.1028        remote1.ftp       ESTABLISHED
        tcp      0        0        dave.1027        remote2.telnet   ESTABLISHED tcp        0          0
        dave.1026        remote3.telnet   ESTABLISHED tcp        0          0          dave.1023
        remote3.login    ESTABLISHED

                C>
```

Proto The protocol used in the connection. In our example, we have four TCP-based connections: two for TELNET connections, one for RLOGIN, and one for FTP.

Recv-Q The receive queue. The number of received characters waiting to be processed.

Send-Q The send queue. The number of characters waiting to be sent.

Local Address It consists of the symbolic host name and port number for the local connection. The first field is the symbolic name of the local machine, and the second is the TCP port number. Thus, dave is the local host name.

Foreign Address It consists of the symbolic host name and port number for the remote connection. In this example, remote1, remote2, and remote3 are remote hosts.

State The current state of the connection.

For additional information, choose:

Show Interface Statistics
Show Routing Tables
Protocol Statistics Display
Show Network Interface Parameter
Transmit Statistics
Receive Statistics
Show Network Interface Statistics

Show Interface Statistics

The command netstat -i displays activities on all of the local machine's interfaces to the net. The cumulative statistics are displayed in the form of a table as follows:

```
            C> netstat -i

      Name   Mtu    Type   Network        Address          Ipkts   Ierrs   Opkts   Oerrs
      nd0    1500   DIX    subnet5lanera  797     0        872     0
```

Name Network interface name. For example, nd0 is the name of the NDIS interface of type DIX (Ethernet). When the interface is down, an asterisk (*) is shown next to the interface name. For example, nd0*.

Mtu Maximum transmission unit in bytes. This is the largest size permitted for any single packet sent through this interface.

Type The interface type used by the link-level driver to determine the type of packet header used.

Network The name of the network address of the interface as given in {INSTALL}\ETC\NETWORKS or in a name server's database.

Address The name of the machine address of the interface as given in {INSTALL}\ETC\HOSTS or in a name server's datebase.

Ipkts Input packets. The number of packets received by this interface.

Ierrs Input errors. The number of errors detected in packets of data received by this interface.

Opkts Output packets. The number of packets transmitted by this interface.

Oerrs Output errors. The number of errors detected in packets of data transmitted by this interface. For NDIS, ODI or Packet driver interfaces, this error usually indicates the hardware settings may have a conflict with another adapter in the local system.

Show Routing Tables

Below is a sample display of the netstat -r command:

```
C> netstat -r

Routing tables:

Destination     Gateway         Flags   Refcnt  Use     Interface
remote1         lanera   UGH    0       0       ec0
remote2         gateway1        UGH     0       0       ec0
remote3         gateway2        UGH     0       0       ec0
remote4         gateway2        UGH     0       0       ec0
```

Destination The network or remote host for this route.

Gateway The name of the gateway configured for this route. If directly connected, a local machine name is displayed. Otherwise, the name of the machine that packets are routed to is displayed.

Flags The state of the route. Valid states are:

U	Up
G	A route to a gateway
N	A route to a network
H	A route to a host

Refcnt The current number of active connections using the route. Connection-oriented protocols, such as TELNET, RLOGIN, and FTP, normally maintain a route for the duration of the connection. On the other hand, connectionless protocols obtain a route as needed and then discard it after use.

Use The current number of packets sent using this route.

Interface The name of the physical network interface on the local end of the route.

Protocol Statistics Display

The option -s displays protocol-specific errors. Errors are grouped for each protocol. Below is a sample output of this option.

```
C> netstat -s

IP Statistics:
        7613 total packets received
        0 bad header checksum
        0 with size smaller than minimum
        0 with data size < data length
        0 with header length < data size
        0 with data length < header length
        0 fragment received
        0 fragment dropped (dup or out of space)
        0 fragment dropped after timeout
        0 packet forwarded
        0 packet not forwardable
        0 redirect sent
        0 maximum input queue length

ICMP Statistics:
        0 call to icmp_error
        0 error not generated because old message was icmp
        Output histogram:
                echo reply: 4
        0 message with bad code fields
        0 message < minimum length
        0 bad checksum
        0 message with bad length
        Input histogram:
                echo: 4
        4 response messages generated

TCP Statistics:
        5 connections attempted
        0 connection accepted
        5 connections established
        0 connection dropped
        0 connection time-out

TCP Send Statistics:
        10991 total packets sent
        5512 total data packets sent
        6887 total data bytes sent
        0 total data packet retransmitted
        0 total data byte retransmitted

TCP Receive Statistics:
        7409 total packets
        7121 total packets in sequence
        0 bad checksum
        0 bad header offset field
        0 too short
        11 window-probe packets

UDP Statistics:
        200 incomplete headers
        0 bad data length field
        0 bad checksum

Routing Statistics:
        0 bad routing redirect
        0 dynamically created route
        0 new gateway due to redirect
        0 destination found unreachable
        0 use of a wildcard route
```

Show Network Interface Parameter

The option -p displays a specific interface parameter. To invoke, enter:

```
C> netstat -p <name>
```

where <name> is the name of an interface such as nd0 or pd0.

For a Packet Interface

To display the Packet driver parameters, invoke:

```
C> netstat -p pd0

PD0 Parameters:

Interface Class: EtherNet
Interface Type:  Western Digital WD8003
Interface Number:        0x0
Local Net Address:       00:00:C0:B7:70:1B
Packet Driver Name:      WD8003E
Packet Driver Version:   9
Software Interrupt No.:  0x7E
1st PD Handle:  0x140
2nd PD Handle:  0x158
```

Interface Class This is the class of interface. The class identifies the network media that the hardware interface supports. Currently, the TCP/IP driver supports only DIX or Bluebook Ethernet, and Token Ring interfaces.

Interface Type This is the type of interface. This specifies a particular instance of an interface supporting a class of network medium. Normally, hardware vendors use this field for their adapter model ID such as: 3Com 3C503 or 3C523, Western Digital WD8003, etc.

Interface Number This number distinguishes different classes and types if a machine is equipped with more than one interface. Currently, it is not used and set to 0.

Local Net Address This identifies the current Etherent address.

Packet Driver Name This identifies the name of the Packet Driver.

Packet Driver Version This identifies the version of the Packet Driver.

Software Interrupt No. This is the software interrupt number that the protocol (TCP/IP) driver uses to communicate with the Packet driver.

1st and 2nd PD Handles This is an arbitrary integer value associated with each MAC-level demultiplexing type that has been established between the TCP/IP driver and Packet driver.

For an NDIS Interface

To display the NDIS interface parameters, invoke:

```
C> netstat -p nd0

ND0 Parameters:

Permanent Station address       00:00:c0:b7:70:1b
Current Station address 00:00:c0:b7:70:1b

Adapter Type:   DIX+802.3
Adapter Description:      Western Digital EtherCard Plus
Link Speed:     2000000         bits/sec Service Flags:   0xbb
Service Status: 0x1f            Recv Filter:      0x3
Xceivers Frame Size:    1514    bytes    Size of Xmit Bufs:         0 bytes
Size of Recv Bufs:      3120    bytes    IRQ:     64

Transmit Statistics
Total Frames:   1         Total Bytes:    60
Multicast Frames:         0       Broadcast Frames:       1
Multicast Bytes: 0        Broadcast Bytes:        60
Frames Ä Timeout:         0       Frames Ä H/W Error:     0

Receive Statistics
Total Frames:   19        Total Bytes:    2400
Multicast Frames:         0       Broadcast Frames:       19
Multicast Bytes: 0        Broadcast Bytes:        2400
Frames Oversize:          0       Frames Undersize:       0
Frames Ä Error: 0         Frames Ä CRC Error:    0
Frames Ä No Buff:         0       Frames Ä H/W Error:     0
```

Permanent Station address This is the network interface address.

Current Station address This is the assigned network interface address.

Adapter Type This is the type name of the MAC. For example DIX+802.3. Others are ARCNET and FDDI.

Adapter Description This is the name of the network adapter.

Link Speed The transfer speed in bits per second of the network adapter. Some adapters do not maintain such records and the indicated value could be wrong.

Service Flags

This field provides a description of the MAC services in bit coding as follows:

Bit	Service (when 1, 0 indicates otherwise)
0	Broadcast supported
1	Multicast supported
2	Functional/group addressing supported
3	Promiscuous mode supported
4	Software settable station address
5	Statistics are always current in the service specific status table
6	Initiate Diagnostics supported
7	Loopback supported
8	Type of service
	0 -- MAC does primarily Receive Look ahead indications
	1 -- MAC does primarily Receive Chain indications

9	IBM source routing supported
10	Reset MAC supported
11	Open/Close Adapter supported
12	Interrupt Request supported
13	Source Routing Bridge supported
14	Global Descriptor Table addresses supported.

Service Status

The service status of the MAC in bit coding as follows:

Bit	Meaning		
0-2			
	0	Ä	Hardware not installed
	1	Ä	Hardware failed startup diagnostics
	2	Ä	Hardware failed because of a configuration problem
	3	Ä	Hardware not operational because of a hardware fault
	4	Ä	Hardware operating marginally because of a software fault
	5	Ä	Reserved
	6	Ä	Reserved
	7	Ä	Hardware fully operational
3	If set, MAC is bound, else no bound		
4	If set, MAC is open, else not open		
5-7	Reserved, zero.		

Recv Filter

A packet filter bit mask that specifies the type of packets the MAC is directed to accept and forward to the protocol driver.

Bit	Meaning
0	directed and multicast or group and functional
1	broadcast
2	promiscuous
3	all source routing

Xceivers Frame Size The maximum frame size in bytes that may be both sent and received by the MAC.

Size of Xmit Bufs The size of an allocated transmit buffer in bytes.

Size of Recv Bufs The size of an allocated receive buffer in bytes.

IRQ The interrupt request level used by the network adapter to interrupt the PC's CPU.

Transmit Statistics

Total Frames The total number of frames transferred.

Total Bytes The total number of bytes transferred.

Multicast Frames The total number of Multicast frames transferred.

Broadcast Frames The total number of Broadcast frames transferred.

Multicast Bytes The total number of bytes contained in all Multicast frames transferred.

Broadcast Bytes The total number of bytes contain in all Broadcast frames transferred.

Frames - Timeout The total number of frames not transferred because of time-out error.

Frames -H/W Error The total number of frames not transferred because of hardware error.

Receive Statistics

Total Frames The total number of frames received.

Total Bytes The total number of bytes received.

Multicast Frames The total number of Multicast frames received.

Broadcast Frames The total number of Broadcast frames received.

Multicast Bytes The total number of bytes contained in all Multicast frames received.

Broadcast Bytes The total number of bytes contained in all Broadcast frames received.

Frames Oversize The total number of received frames discarded because of frames exceeding maximum size.

Frames Undersize The total number of received frames discarded because of frames smaller than minimum size.

Frames w/ Error The total number of received frames discarded because of error.

Frames w/ CRC Error The total number of received frames discarded because of frames with CRC errors.

Frames w/ No Buff The total number of received frames discarded because of shortage of buffer space.

Frames w/ H/W Error The total number of received frames discarded because of hardware errors.

Show Network Interface Statistics

The option -d displays the network interface statistics. To invoke,

```
C> netstat -d <name>
```

where <name> is the network interface such as nd0 or pd0. For example,

```
C> netstat -d nd0

Transmit errors:

Timeout:          0        Re-Xmit Timeout:        0        Re-Xmit:          0
Bad packet len: 0

Receive errors:

Packet Drops:   0        Null Packets:     0
Bad Packet Type:         0          Sanity:  0
Out of Mbufs:    0        Queue Full:       0
```

Transmit Errors

Timeout he total number of frames dropped because of transmission timeout.

Re-Xmit Timeout The total number of frames dropped because of re-transmission timeout.

Re-Xmit The total number of frames re-transmitted.

Bad packet length The total number of frames with incorrect data length.

Receive Errors

Packet Drops The total number of received frames dropped.

Null Packets The total number of invalid frames dropped.

Bad Packet Types The total number of frames dropped because of incorrect frame type.

Sanity The total number of frames dropped because frames contain an invalid field.

Out of Mbufs The total number of frames dropped because of a shortage of receive buffers.

Queue Full The total number of frames dropped because a data queue between the application layer and driver layer is full.

ROUTE - Route Table Management

The route command maintains the routing table. The routing table contains gateway information used by the IP protocol to forward data packets to the addressed hosts or networks.

Using the route command, the users can view the content of the routing table, and manipulate the content of this table by adding or deleting routing entries. Note that this command comes with TCPOpen/Router product only.

INVOCATION

The general syntax of this command is:

```
ROUTE [-N]      [LIST] [FLUSH]
                [ADD <dest> <gateway> <metric>]
                [DELETE <dest> <gateway>]
```

LIST, FLUSH, ADD, DELETE
are route sub-commands.

-N displays routing entry addresses in dot-notation.

<dest> is the host to which you wish to connect.

<gateway> is the gateway that is responsible for routing your packets to the <dest>. Both <dest> and <gateway> can be specified by the host name or dot-notation IP address.

<metric> specifies the number of intermediate hosts your packets may have to travel through in order to reach the destination. By default, <metric> is 0.

SAMPLE USAGE

The following are examples of the route command. To list the content of the routing table:

```
C> route list
Routing tables:
Destination: 139.313.20.1           Gateway: gateway1
Destination: 139.313.50.4           Gateway: gateway2
Destination: 139.313.20.5           Gateway: gateway2
```

To add a remote host remote4 through gateway1 with two (2) hops, invoke:

```
C>   route add remote4 gateway1 2
ADD   host :remote4     gateway: gateway1
```

To delete a route entry for remote3, enter:

```
C>   route delete remote3 gateway2
DELETE   host:   remote3     gateway: gateway2
```

ARP- ARP Table Management

The ARP command displays or modifies the Internet-to-Ethernet address translation table, also called the ARP table. This table is maintained by the Address Resolution Protocol (ARP) for Ethernet-based networks only.

TCP/IP addresses with its peers using logical network level Internet addresses. However, the Ethernet interfaces attached to the same local network (LAN) exchange data packets with each other using the physical Ethernet addresses. Therefore, hosts on an Ethernet LAN need to map the Internet address of their data packets to the corresponding Ethernet address before sending the packets to a locally-attached host through an Ethernet interface.

The ARP protocol is for broadcast-based networks such as Ethernet or Token Ring. It does not work for point-to-point networks such as serial line network.

The ARP table contains the entries of the locally-attached hosts only. Hosts on a distant network do not have an entry in this table although gateways attached to the local networks do. By default, the table can hold up to 10 entries. An older entry is deleted to make room for a new entry when the table is full.

INVOCATION

The arp command has the following syntax:

```
ARP     [ LIST] [ LOAD <file-name>] [ DELETE <host>]
        [ ADD <host> <ether-addr> [TEMP] [PUB] [TRAIL] ]
```

LIST lists the entries in the ARP table.

DELETE <host> deletes an existing entry specified by <host> in the ARP table. <host> may be specified as either a host name or a dot-notation IP address.

ADD <host> <ether-addr> [TEMP] [PUB] [TRAIL] adds a new entry in the ARP table. <host> is the Internet address in either host name or dot-notation representation. <ether-addr> is the Ethernet address associated with <host>. The Ethernet address is given as six colon-separated, two-digit hexadecimal numbers. For example: 02:05:F0:E9:DC:01.

If **TEMP** is specified, the entry is marked as temporary, and is automatically removed if not used within 1 minute. Otherwise, the newly-added entry is assumed to be permanent, which stays in the table until explicitly deleted by the arp command.

If **PUB** is specified, the entry is marked as "published". The local system ARP protocol responds to requests for the <host> even though <host> is not the local host.

If **TRAIL** is specified, <host> is using link level trailer encapsulation technique as mentioned in the inconfig command section.

LOAD <file-name>

instructs arp to add multiple entries in the ARP table from the file <file-name>. The entries in the file must be in the following format:

```
<host> <ether-addr> [TEMP] [PUB] [TRAIL]
```

Entries are separated by a new line.

ADD AN ENTRY INTO THE ARP TABLE

A new entry is added to the ARP table either by the ARP protocol, when it successfully resolves an IP address or as a result of receiving a broadcast packet from a station on the network, or by the arp command.

By ARP Protocol When the user invokes a TCP/IP application that sends a packet to a host on the network, the ARP protocol looks into its own table to map the Internet address of the destined host to a physical address, e.g. Ethernet. If the mapping fails, the ARP protocol automatically sends a broadcast packet on the LAN to query all the local hosts for an Ethernet address associated with the given Internet address. If the host associated with the given Internet address is up, its ARP protocol responds to the query with an <Internet, Ethernet> address pair that associates a given Internet address with an Ethernet address. The local ARP protocol then updates its table with a new entry containing the address pair.

By ARP Command The address translation process usually happens automatically whenever the user invokes a TCP/IP application. However, if some host on the local network does not support ARP, the user may have to manually create the ARP entries in the local ARP table for those hosts. The arp command allows you to modify the ARP table at will.

DELETE AN EXISTING ENTRY FROM THE ARP TABLE

Similarly, an entry in the ARP table can be deleted automatically by the ARP protocol or manually by users.

Every entry in the ARP table is marked either temporary or permanent. All temporary entries are automatically removed from the ARP table if they are not used within one minute. Permanent entries can only be deleted by the arp command.

SAMPLE USAGE

The following example shows some of the arp command usage.

```
C> ping remote1
ping:    remote1 is alive!

C> ping remote2
ping:    remote2 is alive!

C> arp list
remote1 (139.313.5.1) at 8:0:20:0:45:7b
remote2 (139.313.5.2) at 9:0:30:0:56:8b

C> arp delete remote1
remote1 (139.313.5.1) deleted

C> arp list
remote2 (139.313.5.2) at 9:0:30:0:56:8b

C>
```

The ping command is used to for the ARP protocol in the protocol driver to automatically and quietly updates the ARP table with entries for remote1 and remote2. The list command then shows the updated ARP table. Finally, the delete command deletes the entry for remote1 from the ARP table.

PING - Fault-Isolation Utility

The ping command checks the connectivity between the local DOS machine and a remote host or gateway. It is particularly useful in isolating network failures.

Ping uses the ICMP ECHOÄREQUEST datagram to trigger an IMCP ECHO - RESPONSE from a remote machine. Note that some machines may not provide support for ICMP's ECHO - RESPONSE. ping cannot verify the connectivity of such machines.

INVOCATION

The ping command has the following syntax:

```
        PING    [-R] [-V] [-T <time-out>] [-D <data-size>]
                        [-N <npackets>] <host1> <host2> . . .
```

-R instructs ping to bypass the routing table and send the packets to a host directly attached to the local network. If the host is not on the directly-attached network, an error is returned.

-V turns on verbose mode. ping displays a summary on the sent datagrams, their sequence number, and the round trip time.

-T <time-out> specifies how long ping should wait for ICMP's ECHO_REQUEST response. By default, ping waits for 3 seconds.

-D <data-size> specifies the size of the data included in the ICMP datagrams being sent. This size does not include the 8 byte ICMP header. By default, the size is 56 bytes.

-N <npackets> specifies the number of datagrams to be sent to the remote machines. By default, npackets is 5.

<host1> <host2>. . . are the target hosts that ping sequentially probes for connectivity.

WARNING

You cannot ping the local host running Softerm's TCP/IP since Softerm's TCP/IP driver does not support a local host interface. However, you can ping the local host from any other host using rsh.

Avoid pinging a slow or congested link, such as a serial line. Ping may overload the link. If you have to, change the number of packets sent with the -n option.

SAMPLE USAGE

Ping operates in two modes: sequential and continuous.

In a sequential mode, ping sends a sequence of datagrams one datagram at a time and then waits for a response from the remote host. ping resumes the transmission when it either receives the response from the remote host for the transmitted packet or it times out (the default is 3 seconds).

This mode also implies verbose mode, which causes ping to generate in-progress messages to the monitor as shown in the following examples. Assuming remote1 is up and network is active, remote2 is down.

To invoke ping in sequential mode, use the -v option.

```
          C> ping -v remote1

          PING remote1 (89.0.0.1): with 5 packets of 56 data bytes.

          Transmit     64 bytes     to     remote1:     ICMP Seq# 0.
          Receive      64 bytes     from   remote1:     ICMP Seq# 0. Roundtrip Time =
59ms
          Transmit     64 bytes     to     remote1:     ICMP Seq# 1.
          Receive      64 bytes     from   remote1:     ICMP Seq# 1. Roundtrip Time =
50ms
          Transmit     64 bytes     to     remote1:     ICMP Seq# 2.
          Receive      64 bytes     from   remote1:     ICMP Seq# 2. Roundtrip Time =
59ms
          Transmit     64 bytes     to     remote1:     ICMP Seq# 3.
          Receive      64 bytes     from   remote1:     ICMP Seq# 3. Roundtrip Time =
50ms
          Transmit     64 bytes     to     remote1:     ICMP Seq# 4.
          Receive      64 bytes     from   remote1:     ICMP Seq# 4. Roundtrip Time =
59ms

          ----remote1 PING Statistics----
          5 packets transmitted, 5 packets received, 0% packet loss.
          Minimum round-trip time:      50 ms
          Average round-trip time:55 ms
          Maximum round-trip time:      59 ms

          remote1 is alive!

          C> ping -v remote2

          PING remote2 (89.0.0.2): with 5 packets of 56 data bytes.

          Transmit     64 bytes     to     remote2:     ICMP Seq# 0.   No response!
          Transmit     64 bytes     to     remote2:     ICMP Seq# 1.   No response!
          Transmit     64 bytes     to     remote2:     ICMP Seq# 2.   No response!
          Transmit     64 bytes     to     remote2:     ICMP Seq# 3.   No response!
          Transmit     64 bytes     to     remote2:     ICMP Seq# 4.   No response!

          ----remote2 PING Statistics----
          5 packets transmitted, 0 packets received, 100% packet loss.

          remote2 does not respond!

          C>
```

In continuous mode, ping sends five datagrams to the remote host before waiting for responses. If the five responses come back before the time-out period (the default is 3 seconds) expires, ping prints a message at the monitor as follows:

```
          C> ping remote1
          remote1 is definitely alive!
```

Otherwise, ping eventually times out and terminates with a failure message as shown in the following example:

```
C> ping remote2
remote2 does not respond!
```

This mode is useful for probing the connectivity of several hosts or gateways in a single command .

USING PING TO ISOLATE FAULTS

The following example demonstrates the use of ping to isolate a failure on the path between your computer and a remote host.

Assume you have the following configuration:

Assume that you fail to connect from your_pc to remote3. The problem is hard to determine since the failure may be at any intermediate host or gateway through which your packets travel before reaching the destination. The failure may be at the destination itself.

The following sequence of actions will isolate the failure:

- ping a locally-attached host that you know is functioning to make sure that your DOS station is functioning. Alternately, you can ping your own DOS machine from another machine on the same local network using rsh.

- ping gateway1 to see if the path between YOURÄPC and gateway1 is connected.

- If you get a positive response from gateway1, then ping gateway2 to see if the connection between gateway1 and gateway2 is healthy.

- If the responses are good so far, the problem is likely with remote3. However, you should also ping remote3.

Using the ping command, you can probe for the connectivity of gateway1, gateway2 and remote3 in a single command as follows:

```
C> ping gateway1 gateway2 remote3
ping: gateway1 is alive !
ping: gateway2 does not respond !
ping: remote3 does not respond !

C>
```

ping sequentially sends datagrams to each of the specified hosts and prints the result of every checked point to the PC monitor. In the above example, gateway2 is the point where the problem begins.

Configuration and Setup Files

This chapter describes the configuration and setup files required by various TCPOpen commands. Many of these files are automatically created during the installation process. The user can edit them with any text editor.   Some of them are updated by TCPSETUP program.

The following table identifies the files, their users (applications) and whether they are required or optional.

Files	**Applications**	**Status**
HOSTS	all	required
MODEMS	slconfig	required
NETINFO	all	required
NETWORKS	all	required
PHONES	slconfig	optional
PROTOCOLS	all	required
SERVICES	all	required

HOSTS

NAME HOSTS - host name data base

SYNOPSIS {INSTALL}\ETC\HOSTS

DESCRIPTION The HOSTS file contains information regarding the known hosts on the network. For each host a single line should be present with the following information:

<Internet-address>   <official-host-name>   <aliases>

Items are separated by any number of blanks or tab characters. A '#' indicates the beginning of a comment; characters up to the end of the line are not interpreted by parsing routines. A sample of this file is created during TCPOpen installation. However, if your PC is connected to the Internet network, this file is normally created from the official host data base maintained at the Network Information Control Center (NICC), though local changes may be required to bring it up to date regarding unofficial aliases or unknown hosts.

<Internet address> is specified in the conventional dot notation. <official-host-name> may contain any printable character other than a field delimiter, new line, or comment character. <aliases> is one or more names by which the host is also known. <official-host-name> and <aliases> are case-sensitive.

EXAMPLE
Here is a typical line from the HOSTS file:

```
89.0.0.3          dave              # dave's DOS machine.
```

CAVEATS This file is accessed by Softerm TCP/IP's application programs to resolve a host name to an Internet address. However, if a name server is available, it is always contacted first to resolve the hostname. If this attempt fails, the application program then looks into the HOSTS file to resolve the name.

It is updated by TCPSETUP program.

MODEMS

NAME MODEMS - Modem setup database

SYNOPSIS

{INSTALL}\ETC\MODEMS

DESCRIPTION

The MODEMS file is a text file and contains modem setup entries for various types of modems used by the serial line interface. Entries can be added or deleted using any text editor. Every entry begins with a keyword NAME and a name entry used by the user to specify a particular modem setup. The entry name must consist of printable characters, is case-insensitive, and can be up to twenty (20) characters long. Comment lines begin with a pound sign (#).

Following the entry name is a custom setup for the serial port that the modem is attached to:

- Serial port data format

  This entry specifies the serial port data format parameters to be used with a modem and must be specified in the following format:

  ```
  SET [<baudrate>],[<data>],[<stop>],[<parity>]
  ```

  The arguments must be specified in the above order, and separated by a comma or space. Leading or trailing spaces or tabulations are discarded. The arguments may be omitted but the commas must be preserved. If omitted, the port parameter associated with the omitted argument will not change. See slconfig in Chapter 2 of this part for possible values for these parameters.

- Modem start-up command sequence

  This command sequence consists of one or more modem commands that are issued by slconfig to initialize the modem when it is attached to the serial port. The modem commands may be on the single COMMAND line or multiple COMMAND lines. They must begin with keyword COMMAND and be specified in the format that is recognizable by the modem. slconfig passes them to the modem without any processing.

The following is a sample of such entries:

```
        # setup entry for Smart modem
        #
NAME smartmodem
SET 2400,8,1,N
COMMAND ATE0 L3 S7=60 S11=55 V0 X1 S0-2

# setup entry for VIVA modem
#
NAME VIVA
SET 2400,8,1,N
COMMAND ATE0
COMMAND ATL3 S7=60
```

```
COMMAND ATS11=55 V0
COMMAND ATX1 S0=2
```

Name-to-Phone Number Mapping Database, PHONE.BK

This database also is a text file that contains name-to-phone number mapping entries. It allows a user to specify a dial-up number by name instead of by numbers. The entry must have the following format:

```
<name>          #<phone_number>
```

For example,

```
        DOWJONES     #1-234-567-8900                                    PENTAGON
#00987654321
```

<name> consists of alpha-numeric characters and is case insensitive. It can be up to twenty characters long. <phone-number> must be prefixed by a pound sign (#) and in the format recognizable by the attached modem. Consult the modem Manual for this information.

NETINFO

NAME

NETINFO - TCPOpen configurable parameters.

SYNOPSIS

{INSTALL}\SETUP\NETINFO

DESCRIPTION

The NETINFO file contains various TCPOpen configurable parameters. TCPOpen applications and the protocol driver use the parameters. Most of the parameters are optional and already set to their default values and can be set up either by using tcpsetup program or by using any text editor. A sample of the NETINFO file is given in the {INSTALL}\SETUP directory.

TCP/IP Application Parameters

TCP/IP application programs use these parameters. They give information required by certain TCPOpen applications.

HOST (required) is your PC's host name as known by other hosts in your network. The name is retrieved by the application whenever the application needs the local host's name.

Note that this entry must be the same as the host name for your PC in the {INSTALL}\ETC\HOSTS database file. If there is a dot (.) in a host name, the string follows the dot is assumed to be a domain name.

This parameter consists of any printable characters and the maximum length is 64 characters.

USER (required) is necessary whenever an application attempts to auto-login to a remote host, or to execute a remote command. In these situations, the application must pass the remote user's login ID to the remote host. This parameter is taken as the default for the remote user login ID on the remote host. However, applications usually allow the user to override this setting with an optional argument.

The parameter consists of any printable characters and the maximum length is 40 characters. The remote server applications may also impose length limitation on this parameters which may be less than 40 characters.

DOMAIN (optional) is the name of the domain that the local PC belongs. This parameter is specified only when a name server is available in your domain and when the NSIPADDRESS parameter is also specified.

TERM (optional) specifies the terminal type to be negotiated with the remote server when a TELNET or RLOGIN connection is attempted by either vnt or tapi. vnt uses the default of VT-100 when TERM is not specified, tapi sets the default to unknown.

LPSERVER (optional) specifies the host name of the remote host acting as a print server. The lpr command takes its default printer server name from this parameter.

This parameter consists of any printable characters and the maximum length is 64 characters.

LPPRINTER (optional) specifies the name of the remote printer on the remote host specified by

LPSERVER. This parameter is used by the remote printing applications: lpr and lpq. An application may optionally override this parameter.

This parameter consists of any printable characters and the maximum length is 30 characters.

RMTPATH (optional) specifies the path name of the remote archive server program on the remote host acting as the remote archive server. drtar uses this parameter to know which program to invoke to serve its archive requests. If not specified, RMTPATH is set to /etc/rmt.

The maximum length of this parameter is 80 characters.

ATTRIBUTE (optional) specifies the types of files that are matched during a directory search performed by TCPOpen applications (rcp, ftp, ftpd, etc...) in a file name expansion (globbing) attempt. DOS supports seven file types: NORMAL (read/write), READ-ONLY, HIDDEN, SYSTEM, VOLID (volume ID), SUB-DIRECTORY and ARCHIVE. By default, TCPOpen applications match files with the following file types: NORMAL, READ-ONLY, SUB-DIRECTORY and ARCHIVE files.

ATTRIBUTE is set with the above keywords separated by a space, a tab or a comma. For example, the following setting is equivalent to the default ATTRIBUTE setting:

```
ATTRIBUTE=NORMAL, READ-ONLY, SUB-DIRECTORY, ARCHIVE
```

Unrecognized keywords are simply ignored.

NETVECN (required) specifies the software interrupt vector that TCPOpen applications use to interface with the protocol (TCP/IP) driver for network system services. By default, it is set to 61H. It ranges from 60H to 7FH, inclusive.

TCP/IP Driver Parameters

The TCP/IP driver uses the following parameters:

BUFSPACE (optional) specifies the amount of memory (in Kbytes) to be used for creating network buffers, mbufs. Its value ranges from 10 to 40. By default, it is set to 10 Kbytes.

IPADDRESS (required) specifies the local PC's IP address in dot notation. This is also the network interface's IP address.

NSIPADDRESS (optional) specifies the name server's IP address in dot notation. When this parameter is specified, either the DOMAIN parameter is also specified or the HOST parameter implies a domain name; i.e. having a dot followed by a domain name.

IPBROADCAST (required) specifies the IP address in dot notation to be used for IP broadcast messages. By default, it has the network number part to be that of the local network and the host part of all 1s.

IPGATEWAY (optional) specifies the IP address in dot notation of the default IP gateway/router. The TCP/IP driver routes all the packets of which route is unknown to the default gateway.

IPMTU (required) specifies the maximum IP packet's size can be transmitted on the network interface. By default, it is set to 576 bytes. The maximum allowable MTU varies depending on the type of the physical interface in use. For example, the maximum MTU for a DIX Ethernet interface is 1500. See Table 1.2.1 for other interface types.

SUBNETBITS (optional) specifies the number of bits used as subnet ID. These bits are assumed to immediately follow the network number part of the network address. See Chapter 1 of Part 1 for discussion on network address.

TCPÄSENDWIN (optional) specifies TCP's maximum send window size. This parameter defines how many bytes per connection the protocol driver can send before it has to wait for acknowledgement from the remote host. The actual send window size is usually negotiated between the driver's TCP protocol and that of a remote host during the connection establishment phase.

This parameter is set to a multiple of 256 bytes and must be between 4 and 48, inclusive. By default, it is set to 10 (or 2.5 Kbytes).

TCPÄRECVWIN (optional) specifies the TCP maximum receive window size. This parameter defines how many bytes the TCP/IP driver can receive (or buffer) for a connection before it has to stop receiving. Additional data arriving when the receive window is full is discarded. The receiving process resumes as soon as the application reads or flushes the buffered data.

Changing this parameter effects the protocol driver's performance and memory usage. Generally, a bigger window results in better performance, but it also requires more buffers to accommodate the performance gain.

This parameter is set to a multiple of 256 bytes and must be between 4 and 48, inclusive. By default, it is set to 10 (or 2.5 Kbytes).

ARP_ENT (optional) specifies the maximum number of entries in the address resolution table (See Chapter 6 of Part 3). By default, it is set to 10. The maximum number of entries in this table is 256 entries.

CAVEATS

In multi-tasking environment such as Windows 3.X, the window sizes should be set to 10 or smaller in order to avoid too much data being queued in the Kernel, which potentially consumes all available network buffers.

UDP send and receive window sizes are set to support datagrams of up to 4 Kbytes.

NETWORKS

NAME

NETWORKS - names and numbers for the network.

SYNOPSIS

{INSTALL}\ETC\NETWORKS

DESCRIPTION

The NETWORKS file lists networks. Each line describes a single network and consists of the following blank separated fields:

```
<name>   <number>   <aliases>
```

<name> is the official name of the network. All hosts on the network should use the same official name for a given network.

<number> is the network number. All hosts on the network must use the same number for a given network.

<aliases> is a blank-separated list of local aliases for the network.

The routines that search this file ignore comments (portions of lines that begin with a #) and blank lines.

EXAMPLE

```
# Building 1 Internet
Engineering      101      #R&D
Administration   17       #Administration, etc.
```

PHONES

NAME

PHONES - Name-to-phone number mapping database

SYNOPSIS

{INSTALL}\SETUP\PHONES

DESCRIPTION

The PHONES file contains name-to-phone number mapping entries. Using this database, the user can specify a remote modem connected to a remote host by name instead of by phone number, which is difficult to memorize. For simplicity, this database is basically a text file and can be edited (to add or delete entries) with any text editor. A sample PHONES file is copied to your system under the {INSTALL}\SETUP directory during the installation procedure. The entry must be in the following format:

```
<name>   @<phone-number> #comment (if any)
```

<name> consists of up to thirty alpha-numberic characters and is case-insensitive. <phone-number> is the phone number of the remote modem and must be prefix with '@'. The pound sign (#) signals the start of a comment line. All characters up to the end of line character are ignored by the parsing routine.

EXAMPLES

The following are some typical entries in this file:

```
batman @1-234-567-8900 # available at night only
joker     @1-900-765-4321 # when you get bored
```

PROTOCOLS

NAME

PROTOCOLS - list of Internet protocols

SYNOPSIS

{INSTALL}\ETC\PROTOCOLS

DESCRIPTION

The PROTOCOLS file lists known DARPA Internet protocols supported by TCPOpen 's protocol driver. Each line describes a single protocol and consists of the following fields separated by blanks:

```
<name>   <number>   <aliases>
```

<name> is the official name of the protocol.

<number> is the protocol number.

<aliases> is a blank-separated list of local aliases for the protocol.

The routines that search this file ignore comments (portions of lines that begin with #) and the blank lines.

Protocol names and numbers are specified by the DDN Network Information Center. Do not change this file.

SERVICES

NAME

SERVICES - list of Internet services

SYNOPSIS

{INSTALL}\ETC\SERVICES

DESCRIPTION

The SERVICES file lists known DARPA Internet services. Each line describes a single service and has the following blank separated fields:

```
<name>   <port#>/<protocol>   <aliases>
```

<name> is the of official name of the service.

<port#> is the service number.

<protocol> is the name of the protocol used by the service.

<aliases> is a blank-separated list of local aliases for the service.

The routines that search this file ignore comments (portions of lines that begin with #) and blank lines.

Service names and numbers are specified by the DDN Network Information Center. Do not change this file unless you are familiar with DARPA Internet internals.

COMMON ERROR MESSAGES

TCPOpen error messages are always prefixed with a special code that helps to identify the source and type of the error. The code has the following format:

PRGxxxx

where PRG is the name of the program that generates the message and xxxx is a four decimal digits that identifies the type of error. Many type of error are classified as common errors which may occur in more than one application, and they are assigned digit code of 1000 or above. The following are the common error messages. Replace PRG with the name of the command that generates the error message.

PRG1000 TCPOPEN environment variable not set

Cause: The TCPOPEN environment variable is not set. All TCP/IP applications use this variable to locate the necessary databases and files.

Action: Use the SET command to set the TCPOPEN environment variable to the drive and directory where TCPOpen is installed. For example, if TCPOpen was installed under the directory named TCPIP in drive C, then the following command should set TCPOPEN:

```
C> SET TCPOPEN=C:\TCPIP
```

PRG1001 <error-message>

Failure to start up the application. <error-message> may be one of the following:

- TCP/IP kernel is not yet installed

Cause: The TCP/IP kernel is not loaded. It must be loaded and configured properly before any TCP/IP application can be invoked.

Action: Follow the instructions in the manual to download and configure the TCP/IP kernel

- TCP/IP kernel is not operational. Needs to be reinstalled

Cause: The TCP/IP kernel is loaded but was disabled. It probably had encountered an error condition.

Action: Use the rmtcp command to remove the non-operational TCP/IP kernel, and reinstall it. A system reboot may be necessary if the TCP/IP kernel can not be removed from memory.

- <NETINFO-ENTRY>: entry is undefined or not found

A required entry in the NETINFO file is not found or is not defined. <NETINFO-ENTRY> is one of the following required entries:

HOST	local host name
USER	local user name
NETVECN	TCP/IP kernel interface vector

Causes: The failure may be due to:

1. The absence of a NETINFO file under {INSTALL}\SETUP directory.

2. The <NETINFO-ENTRY> is required but missing or not defined.

Actions: 1. Make sure that the NETINFO file is under the {INSTALL}\SETUP directory.

2. Make sure the above <NETINFO-ENTRY> exists in the NETINFO file and is properly defined.

PRG1002 <service>/<transport-protocol>: unknown service

A failure of an attempt to resolve a TCP/IP application service to a well-known port number where the remote server is listening for service request.

<service>: TCP/IP application services.

<transport-protocol>: The transport protocol that supports <service>. It is either udp or tcp.

Causes: 1. Either the SERVICES file is not in its intended location, which is {INSTALL}\ETC, or

2. The <service> entry does not exist in the SERVICES file.

Actions: 1. Check if the services file is under {INSTALL}\ETC directory. If it does, check for <service> entry in the file. If it is not there, you can manually add the missing entry and update it. Otherwise, you may need to reinstall TCPOpen from the installation diskette. A copy of the SERVICES file template is created under the {INSTALL}\ETC directory during the installation process. This file contains all the necessary entries for all TCPOpen applications. Therefore, if it had not been modified, you should encounter this type of error.

2. Please read Part 3, Chapter 1 for the SERVICES file's entry format.

PRG1003 <protocol>: unknown protocol

Failure to resolve a protocol name to its associated ID number.

Causes and Actions:
The same as error message PRG1002 except that in this case, the database is {INSTALL}\ETC\PROTOCOLS.

PRG1004 <net-name>: unknown network

Failure to resolve a network name to a network address.

Causes and Actions:
The same as error message 1002 except that in this case, the database file is {INSTALL}\ETC\NETWORKS.

PRG1005 <port-no>: bad port number

The port number specified in the command line is not a valid port number. A valid port number must be positive and no larger than 32,768 (a signed 16-bit integer).

PRG1006 COMSPEC environment variable is not found

The application attempted to execute a DOS command or to exit to the DOS command shell. The DOS command shell cannot be found if COMSPEC is not specified.

Action: Add the following line to the AUTOEXEC.BAT file:

```
SET COMSPEC=<drive>:<path>\COMMAND.COM
```

Where <drive> and <path> identify the drive and the path name of the COMMAND.COM file location on your system. For example, if your COMMAND.COM is in the root directory of drive C, the above statement should be:

```
SET COMSPEC=C:\COMMAND.COM
```

PRG1007 cannot execute <command>: <error-message>

The application failed to execute the command indicated in <command>.

Cause: <error-message> indicates the cause of this error.

Action: Depending upon the type of error, perform the appropriate action to correct the problem. Please contact Softronics's technical support group for help in determining the exact cause.

PRG1008 unkown host name <host-name>: <error-message>

Failure to resolve the host name to an IP address.
<error-message> is one of the following:

- Unknown host

Causes: 1. The host database is not in the expected location, or

2. The <host-name> is not set up in either the local host name, {INSTALL}\ETC\HOSTS, database or a name server's host name database.

Actions: 1. If you rely on the local HOSTS database located under the directory {INSTALL}\ETC for resolving names, make sure the database is in its intended location.

If you are using a name server to resolve host names, consult your network administrator for help.

2. Add <host-name> and its IP address to either one of the above databases, where applicable.

See HOSTS in Chapter 1 of Part 3 if you are not familiar with setting up a host name database.

- Host name lookup failure

Cause: The name server failed to resolve a host name to an IP address.

Action: Contact your network administrator or the person who maintains the name server database to find out the reason for failure.

- Unknown server error

Cause: The name server responded with an unrecognizable error code.

Action: Contact your network administrator or Softronics's technical support group for help.

- No address associated with name

Cause: Failure to resolve the <host-name> to an IP address.

Action: Notify your network administrator.

PRG1010 cannot set drive to: <drive-id>

Cause: Failure to change from the current drive to the drive identified by <drive-id>.

Action: Check to see if the <drive-id> is a valid drive ID, and if it is available on your system.

PRG1012 cannot create <directory-name>

Cause: Fail to create the directory specified by the <directory-name>.

Action: Check to see if the <directory-name> is a valid path name on the current drive. If it is, your drive probably exceeds the maximum number of directories allowed by DOS, or the path name is too long (the DOS maximum path name is 144 characters).

PRG1013 too long an argument list

Failure to execute a DOS command invoked from PRG.

Cause: The command argument list is longer than permitted: 128 characters.

Action: Try a shorter command.

PRG1014 all ports in use

Cause: All reserved TCP/IP ports are in-use. The TCP/IP kernel allows up to 1024 reserved ports.

Action: This error message should never happen. If it does, it may indicate an internal error; restart of the application and TCP/IP kernel. Notify the LANERA's technical support group.

PRG1015 cannot allocate <number> bytes
PRG1015 cannot allocate memory

Cause: Failure to allocate memory. <number> indicates the number of bytes requested.

Action: If <number> is unreasonably large or negative, report the problem to the LANERA's technical support group.

Otherwise, the application's data segment is used up. Report the problem to LANERA's technical support group.

PRG1020 socket: <error-message>

A socket operation failed. <error-message> may be one of the following:

- Out of socket

All available sockets are allocated. No socket is available.

Causes: The following are possible causes:

1. Too few sockets are configured in the {INSTALL}\SETUP\NETINFO file.

2. Some pending applications may have too many sockets opened.

3. Some sockets are closed but hung up in a closing state. The user can verify this situation with the netstat -a command.

Actions: The following actions correspond to the above causes:

1. Reconfigure the SOCKETS entry in the {INSTALL}\SETUP\NETINFO file. Up to 32 sockets may be configured. TCP/IP kernel needs to be reinstalled for the new configuration to take effect.

2. Terminate applications that leave sockets open unnecessarily.

3. Deinstall the TCP/IP kernel using the rmtcp command and then reinstall it.

- No buffer space available

TCP/IP ran out of internal buffers for creating additional sockets.

Causes:

1. The configured buffer space is too low. TCP/IP kernel buffer space is configured with the BUFSPACE entry in the {INSTALL}\SETUP\NETINFO file.

2. TCP/IP is losing message buffers, MBUFs.

Actions:

1. Increase the BUFSPACE parameter.

2. Use the netstat -m command to determine if the TCP/IP kernel is losing buffers. If so, report the problem to the LANERA's technical support group and reinstall the TCP/IP kernel.

PRG1021 listen: <error-message>

A passive open operation failed. <error-message> may be one of the following:

- Socket operation on a non-socket
- Operation not supported on a socket

Cause: Either application error or TCP/IP kernel error.

Action: Restart the application. Load the TCP/IP kernel, if necessary. If the problem persists, contact LANERA's technical support group.

PRG1022 accept: <error-message>

A passive open operation failed. <error-message> may be one of the following:

- Socket operation on a non-socket
- Operation not supported on a socket
- Operation would block

Cause: Either application error or TCP/IP kernel error.

Action: Restart the application. Load the TCP/IP kernel, if necessary. If the problem persists, contact LANERA's technical support group.

PRG1023 bind: <error-message>

An address binding operation failed. <error-message> is one of the following:

- Socket operation on a non-socket
- Invalid argument

Cause: Either application error or TCP/IP kernel error.

Action: Restart the application. If the problem persists, contact LANERA's technical support group.

- Can't assign requested address

Cause: The network interface has not yet been brought up.

Action: Use the ifconfig command to bring up the network interface.

- Address already in use

Cause: The address associated with the host name is already in use. The previous connection to the same address is not completely closed.

Action: Wait for a few minutes to make sure the previous connection closes, then retry the command. If the error happens repeatedly, report the problem to LANERA.

PRG1024 connect: <error-message>

A connect operation failed. <error-message> is one of the following:

- Connection refused

Cause: The remote server is not available.

Action: Inform your local network administrator or the person responsible for the server machine about the problem.

- Network is unreachable

Cause: There is no route to the remote host. This is likely due to the interface(s) is (are) down and/or the default IP gateway is not specified.

Action: If the only interface is down, use the ifconfig command to bring it up. If the remote host is on another network, make sure the default IP gateway address is specified (by using the tcpsetup program).

PRG1025 set socket option (<code>): <error-message>

Cause: Application error.

Action: Restart the application and TCP/IP kernel. If the problem persists, contact LANERA's technical support group.

PRG1026 get socket name: <error-message>

Cause: Application error.

Action: Restart the application and TCP/IP kernel. If the problem persists, contact LANERA's technical support group.

PRG1027 read socket: <error-message>

Failure to read the socket. The cause of failure is indicated by <error-message>.

- Socket operation on a non-socket
- Invalid argument
- Socket is not connected

Cause: Either application error or TCP/IP kernel error.

Action: Restart the command and TCP/IP kernel. If the problem persists, contact LANERA's technical support group.

PRG1028 write socket: <error-message>

Failure to write to a socket. The cause is indicated by <error-message>, which is one of the following messages:

- Socket operation on a non-socket
- Invalid argument
- Socket is not connected
- Message too long
- Permission denied
- Destination address required
- Broken pipe
- Operation would block

Cause: Either application error or TCP/IP kernel error.

Action: Except for the message "Broken pipe", which indicates the connection is closed by the remote server, the system may be unstable and need a system reboot or at least a restart of the TCP/IP kernel.

Report to LANERA's technical support group.

- Output queue is full

Cause: The output queue for a network interface is full. This error generally indicates that the interface has stopped sending for some reasons.

Action: Use netstat -q to monitor the interface output queue size.

If the queue size is constant, the interface driver probably encountered an internal error and stopped working. In this case, a restart of the TCP/IP kernel is necessary.

If the queue size is drained to 0 when observed, the error may be due to transient network congestion. In this case, the user can simply restart the application.

PRG1029 select socket: <error-message>

A polled I/O on a socket failed. <error-message> indicates the type of error:

- Socket operation on a non-socket
- Invalid argument

Cause: Either the application error or the TCP/IP kernel error.

Action: Restart the application. If the problem persists, the system may be unstable and need a system reboot or at least a restart of the TCP/IP kernel.

Please report to LANERA's technical support group if these problems persist.

PRG1030 receive: <error-message>

Failure to receive data from a socket. The cause is indicated by <error-message>, which is one of the following:

- Socket operation on a non-socket
- Invalid argument
- Socket is not connected

Cause: Either application error or TCP/IP kernel error.

Action: Restart application. If the problem persists, restart the command and TCP/IP kernel. If the problem persists, contact LANERA's technical support group.

PRG1031 send: <error-message>

Failure to send a message over a socket. The cause is indicated by <error-message>, which is one of the following:

- Socket operation on a non-socket

- Invalid argument
- Socket is not connected
- Message too long
- Operation would block

Cause: Either application error or TCP/IP kernel error.

Action: Restart application. If the problem persists, the system may be unstable and need a system reboot or at least a restart of the TCP/IP kernel.

Please report to LANERA's technical support group if these problems persist.

- Output queue is full

Cause: The output queue for a network interface is full. This generally indicates that the interface has stopped sending.

Action: Use netstat -q to monitor the interface output queue size.

If the queue size is constant, the interface driver probably has encountered an internal error and stopped working. In this case, a restart of the TCP/IP kernel is necessary.

If the queue size is drained when observed, the error may be due to transient network congestion. In this case, the user can simply restart the application.

PRG1035 lost connection

Cause: The connection with the remote host is closed by the remote server. The server application probably encountered an internal error and terminated the connection.

Action: If the problem happens repeatedly, contact the remote host system administrator for assistance.

PRG1060 status <file-name>: <error-message>

Cause: Failure to check a file, <file-name>, status. <error-message> indicates the type of error.

Action: If <error-message> is "No such file or directory", <file-name> probably is an invalid path name or refers to a non-existent file.

PRG1061 <file-name> is not a regular file

Cause: <file-name> is expected to be a regular file. However, its file attributes indicate otherwise.

Action: Check if the file, <file-name>, is a regular file. If it is, report the error to LANERA's technical support group.

PRG1062 open <file-name>: <error-message>

Failure to open <file-name>. <error-message> indicates the type of error, which is one of the following:

- Permission denied

Cause: The named file does not have an appropriate permission bit set. Because DOS file always has read permission bit set, this problem is likely to happen when a file is read-only opened for writing.

Action: If there is no reason for file protection, use the DOS ATTRIB command to change the file mode to allow writing.

- File already exists

Cause: The named file already exists and is read-only when PRG attempts to open with file creation mode.

Action: If there is no reason for file protection, use DOS's ATTRIB command to change the <file-name> mode to allow writing.

- Too many open files

Cause: The number of opened files exceeds the system allowance specified in the CONFIG.SYS file. It may be because:

  - the configured system allowance, FILES, in the CONFIG.SYS is too low.
  - too many processes are pending with open files.
  - PRG did not close the files it opened.

Action: If the configured system allowance in CONFIG.SYS file is too low, increase it.

If the application did not close the files it opened, report the problem to LANERA's technical support group.

- No such file or directory

Cause: A file or path name is not found.

Actions: Check if <file-name> specifies a valid file (path) name. Report the problem to LANERA's technical support group if it is valid path name.

PRG1065 read <file-name>: bad file number.

Cause: Application error.

Action: Report the problem to LANERA's technical support group. <file-name> indicates the name of the file currently written. <error-message> is one of the following:

- Bad file number

Cause: Application error.

Action: Restart the application or reboot the system. Report the problem to LANERA's technical support group if the problem persists.

- No space left on device

Cause: No more disk space is available.

Action: Remove unused files or directories to make room for new files.

- Unexpected EOF

Cause: The application encountered an unexpected EOF while writing to a file.

Action: Check the disk space. If no disk space is available, remove some unused files to make room for the new files. Otherwise, report the problem to the LANERA's technical support group.

PRG1068 access <file-name>: <error-message>

Failure to access <file-name>. <error-message> may be one of the following:

- Access denied

Cause: The file permission setting does not allow the specified access.

Action: Use the ATTRIB command to change the file permission setting so that the file is accessible by PRG.

- File not found

Cause: Either the file or the specified path name does not exist.

Action: Check if <file-name> is a valid file (path) name or whether it exists.

PRG1069 change mode of <file-name>: file not found

Cause: Failure to change the file mode of the file indicated by <file-name> because either the file or the specified path name does not exist.

Action: Check if <file-name> is a valid file (path) name or whether it exists.

PRG1070 cannot create <directory>

Cause: Failure to create the necessary directory to contain received files.

Action: Check if <directory> is a valid path name on the current drive. If so, the drive exceeds the maximum number of directories allowed by DOS or the path name is too long (DOS maximum path name is 144 characters).

PRG1076 set times <file-name>: <error-message>

Cause: PRG fails to set the modification times on a local file specified by <file-name>. The <error-message> is one of the followings:

- Permission denied
  <file-name> is either a read-only file or a directory.

- Invalid argument
  It is likely that PRG encountered on internal error.

- Too many open files
  The file must be opened in order to change its modification time. When there are already many files open, PRG fails to open this file.

- File or path name not found
  The file (or path) name refers to a non-existent file (or path).

Action:

- Invalid argument
  Report the error to LANERA's technical support group.

- Too many open files
  Under normal circumstance, TCPOpen applications never open more than two files at once. If this error occurs, it may be because:

  A. The FILES variable in the CONFIG.SYS file is wrongly set to a small value.

  This configurable parameter defines the maximum number of files that can be opened at one time, and is read by DOS during boot-up.

  Check the CONFIG.SYS file and make sure FILES is set to at least twenty (20). If FILES is not specified, the DOS system's default value for FILES is 20.

  The system must be rebooted for the new configuration to take effect.

  B: One or more pending processes have opened too many files.

  If the user's system has too many pending processes, it is likely that these pending processes have opened the files and kept them open while being put on hold.

  In this case, the user can terminate one or more of these pending processes before retrying PRG. However, this error indicates that the user may need to increase the total number of open files (see above) in the system to avoid this problem.

- File or path name not found
  Normally, the file modification time is set right after the file is created on the local machine. Report this error to the LANERA's technical support group.

ARP ERROR MESSAGES

Error messages of 1000 or over are common error messages. Please refer to the Common Error Message section for causes and action.

ARP0001 bad entry (<entry>) in file: <file>

Cause: The downloaded file <file> has a badly formatted entry. <entry> is the first argument of the entry line.

Action: Review the downloaded file and ensure that all the entries in the download file conform to the following format:

<Host-Name> <Ethernet-Address> [pub] [trail] [temp]

Where:

<Host-Name> host name or IP address in dot-notation

<Ethernet-Address> six-byte dot-notation Ethernet address

pub, trail, temp optional

ARP0002 set ARP entry: <error-message>

arp failed to add a new entry in the Arpanet table.
<error-message> reveals the cause and is one of the following messages:

- Address family not support by protocol family

Cause: TCPOpen supports only Internet address family. All TCPOpen application uses only this address family. Therefore, this problem should not occurs. If it does, it indicates a TCP/IP internal error.

Action: Reload the TCP/IP driver and report the problem to LANERA.

- Network is unreachable

Cause: The network address of the new entry indicates that the host it represents is on an unreachable network.

Action: If the error is a typo error, re-enter the command with the correct address. Otherwise, the user has to:

Make sure to bring the new interface up by using either the ifconfig or the slconfig command, depending on the type of the interface.

If necessary, use route command to add additional routing entries to direct out-going packets to the destination.

- Can't assign requested address

Cause: There is no more room in the Arpanet table for adding the new entry.

Action: Remove an unused entry from the Arpanet table to make room for the new one.

ARP0003 get ARP entry: <error-message>
ARP0006 delete ARP entry: <error-message>

arp fails to retrieve or delete an Arpanet table entry.
<error-message> reveals the cause.

- Address family not support by protocol family

Cause: TCPOpen supports only Internet address family. All TCPOpen application uses only this address family. Therefore, this problem should not occurs. If it does, it indicates a TCP/IP internal error.

Action: Reload the TCP/IP driver and report the problem to LANERA.

- Network is unreacheable

Cause: The network address of the new entry indicates that the host it represents is on an unreacheable network.

Action: If the error is a typo error, re-enter the command with the correct address. Otherwise, the user has to:

Make sure to bring the new interface up by using either the ifconfig or slconfig command, depending on the type of the interface

If necessary, use route command to add additional routing entries to direct out-going packets to the destination.

ARP0004 <host-name> <IP-address>: no such entry

Cause: A delete or get operation failed because the requested entry is not found in the Arpanet table.

Action: Use the arp -list option to list all existing entries in the Arpanet table. Retry the command with an entry that exists in the table.

ARP0007 internal error (ARPTAB_SIZE)

Cause: An application or TCP/IP driver error.

Action: Reload the TCP/IP driver. If the problem persists, notify LANERA technical support staff.

ARP0008 internal error (ARPTAB)

Cause: An application or TCP/IP driver error.

Action: Reload the TCP/IP driver. If the problem persists, notify LANERA technical support staff.

ARP0009 bad Arpanet table size (<size>)

Cause: An application or TCP/IP driver error.

Action: Reload the TCP/IP driver. If the problem persists, notify LANERA technical support group.

ARP0010 invalid Ethernet Address: <ethernet-address>

Cause: arp expects an Ethernet address in the following format:

```
xx:xx:xx:xx:xx:xx
```

where xx is hexadecimal digits.

Action: If you entered an Ethernet address in a format other than that shown above, please re-enter the address in the expected format.

IFCONFIG ERROR MESSAGES

Error messages of 1000 or over are common error messages. Please refer to the Common Error Message section for causes and action.

IFCONFIG0001 <operation>: <error-message>

ifconfig failed to execute an operation. <operation> indicates the ifconfig operation that failed to execute and is one of the following:

- get interface flags
- set interface flags
- get interface metric
- set interface metric
- get network mask
- set network mask
- get IP address
- set IP address
- set broadcast address
- get adapter type
- set adapter type

<error-message> indicates the reason for failure and is one of the following:

- No such interface

Cause: The specified network interface does not exist in TCP/IP. Make sure to specify a valid network interface supported by the installed TCP/IP driver.

Action: Use the netstat -i command to list all valid network interface supported by the installed TCP/IP driver.

- Can't assign requested address

Cause: A user attempted to assign an IP address to an interface. However, either the specified interface does not exist in the TCP/IP driver, or the address is not valid for the specified interface.

Action: Use the netstat -i command to verify whether the desired interface exists in the TCP/IP driver. If it does, the user may have attempted to assign a broadcast address to a non-broadcast interface medium, which is not allowed.

- TCP/IP kernel has not bound to a MAC driver yet

Cause: An operation required a TCP/IP driver to be bound to a MAC driver.

Action: This error typically occurs when using the NDIS interface version of the TCP/IP driver. In this case, the TCP/IP kernel is not bound to the NDIS MAC driver until the NETBIND command is issued.

Issue the NETBIND command. If successful, retry the ifconfig command.

- No buffer space available

Cause: The TCP/IP driver ran out of MBUF buffers.

Action: Use the netstat -m command to monitor the MBUF statistics. Report the problem to LANERA's technical support group if MBUFs are lost.

- Operation not supported on a non socket
- Permission denied
- Invalid argument

Cause: An application or TCP/IP driver error.

Action: Report to LANERA's technical support group.

IFCONFIG0002 <address> is a bad address

Cause: The specified address at the command line is not a valid IP address.

Action: Specify a valid IP address.

IFCONFIG0003 invalid adapter type

Cause: A non-supported adapter type was specified at the command line. Valid adapter types are:
DIX, 802.3, 802.4, 802.5, 802.6, ARCNET, APPLETALK, FDDI.

However, only DIX, 802.3, and 802.5 are currently supported by the TCP/IP kernel.

Action: Re-enter the ifconfig command with valid adapter types.

NETSTAT ERROR MESSAGES

Error messages of 1000 or over are common error messages. Please refer to the Common Error Message section for causes and action.

NETSTAT0001 internal error (<debug-info>)

Cause: netstat encountered an internal error while trying to obtain statistics information from various data structures maintained by the TCP/IP driver. <debug-info> is one of the following:

IFNETP	Interface table address
INPCB	Internet control block
TCP	TCP statistics table
IP	IP statistics table
UDP	UDP statistics table
ICMP	ICMP statistics table
MBUF	Message buffer statistics table
MFREE	Free message buffer statistics
MCLFREE	Free cluster message buffer statistics
IPINTRQ	IP receive queue statistics
RAWINTRQ	RAW receive queue statistics
RTHOSTP	Host routing table
RTNETP	Network routing table

<debug-info> is for debugging only. It indicates the type of data structure that netstat failed to retrieve.

Action: Reload the TCP/IP driver. If the problem persists, notify LANERA's technical support group.

NETSTAT0002 must specify an interface

Cause: The user should specify an interface with the selected option. Some options require an interface name to be specified while others do not.

Action: Specify one of the following interface names:

pd0, pd1,...	for a Packet driver
nd0, nd1,...	for an NDIS driver
sl0, sl1,...	for a serial line driver

Refer to the Command References or the User Guide for more information.

NETSTAT0003 get NDIS information: < error-message>
NETSTAT0004 get NDIS statistics: <error-message>
NETSTAT0005 get interface statistics: <error-message>
NETSTAT0006 get packet driver's parameters: <error-message>

netstat failed to obtain NDIS driver information or statistics. <error-message> indicates the type of error involved and may be one of the following:

- Operation failed

Cause: The requested operation failed.

Action: The NDIS driver does not support such an operation.

- TCP/IP has not bound to a MAC driver

Cause: The TCP/IP driver has not linked up with the MAC driver. This error typically occurs when TCP/IP is expected to bind with an NDIS driver.

Action: Issue the NETBIND command if both the NDIS driver and the TCP/IP driver have already loaded.

- Socket operation on a non-socket
- Permission denied
- Bad length
- Invalid argument
- Operation not supported on socket

Cause: An application error.

Action: Reload the TCP/IP driver and retry the command. If the problem persists, notify the LANERA's technical support group.

NETSTAT0007 <interface>: no such interface

Cause: The user specified a non-existent interface.

Action: Use netstat -i to check the interfaces available in the installed TCP/IP driver.

PING ERROR MESSAGES

Error messages of 1000 or over are common error messages. Please refer to the Common Error Message section for causes and action.

PING0001 invalid option: <option>

Cause: An invalid ping option was specified in the command line.

Action: Re-enter the command without the invalid options. See ping in the Command Reference in Chapter 2 of Part 3 for more details.

PING0002 packet size too large

Cause: The user specified a packet size larger than the allowable maximum packet size of 4088 bytes.

Action: Specify a packet size that is less than 4088 bytes.

SLCONFIG ERROR MESSAGES

Error messages of 1000 or over are common error messages. Please refer to the Common Error Message section for causes and action.

SLCONFIG0001 <operation>: <error-message>

slconfig failed to execute an operation. <operation> indicates the slconfig operation that failed and is one of the following:

- get interface flags
- serial port set up
- modem command

<error-message> indicates the reason for failure and is one of the following:

- No such interface

Cause: The specified serial-line interface does not exist in the TCP/IP driver.

Action: Use the netstat -i command to see if you have entered a valid serial-line interface supported by the installed TCP/IP driver.

- Can't assign requested address

Cause: A user attempted to assign an interface to an IP address, but the specified interface does not exist in the TCP/IP driver.

Action: Use the netstat -i command to verify whether the desired interface exists in the TCP/IP driver. If it does, TCP/IP probably encountered an internal error and needs a reload.

- No buffer space available

Cause: The TCP/IP driver ran out of MBUF buffers.

Action: Use the netstat -m command to monitor the MBUF statistics. Report the problem to LANERA's technical support staff.

- Operation not supported on a non socket
- Permission denied
- Invalid argument

Cause: An application or TCP/IP driver error.

Action: Report to LANERA's technical support staff.

SLCONFIG0002 interface is down. Command aborted

Cause: slconfig attempted to issue a command that requires the interface to be up. However, the interface is currently marked down.

Action: Use the ifconfig command to bring the interface up and then retry the aborted command.

SLCONFIG0003 modem is already attached

Cause: A user attempted to attach a modem to a serial interface. However, the interface driver's set-up indicates that the modem is already attached.

Action: Since the modem is already attached, you do not need to issue the attach command again. However, if you want to reattach the modem, you must detach it first.

SLCONFIG0004 modem is detached. Attach it first

Cause: A user issued a command that requires the modem to be attached to the specified interface. However, the interface driver's set-up indicates that the modem is not attached.

Action: Attach the modem then re-issue the command.

SLCONFIG0005 unknown modem command: <command>

Cause: A user issued a non-supported modem command. The following commands are valid modem commands:

- attach
- detach
- dial
- command
- hangup

Action: Use one of the supported modem commands, or refer to the TCPOpen User's Guide or the Command References for more information.

SLCONFIG0006 <entry> is not found

Cause: Either a modem set-up entry or a phone number entry is not found in the set-up files.

Action: Check the set-up files:

- {INSTALL}\SETUP\MODEMS for modem set-up information
- [INSTALL}\SETUP\PHONES for phone numbers

If necessary, follow the guidlines in the TCPOpen Command Reference to add a new entry to one of the files.

SLCONFIG0007 invalid baudrate: <baudrate>

Cause: A user entered an invalid baudrate while attempting to configure a serial port.

Action: Refer to the TCPOpen Command Reference (Chapter 2 of Part 3) for a list of valid data bit numbers, then retry the command with a valid baudrate.

SLCONFIG0008 invalid No. of Data Bits: <data-bit>

Cause: A user entered an invalid number of data bits while attempting to configure a serial port.

Action: Refer to the TCPOpen Command Reference (Chapter 2 of Part 3) for a list of valid data bit numbers, then retry the command with a valid number of data bits.

SLCONFIG0009 invalid No. of Stop Bits: <stop-bit>

Cause: A user entered an invalid number of stop bits while attempting to configure a serial port.

Action: Refer to the TCPOpen Command Reference (Chapter 2 of Part 3) for a list of valid stop bit numbers, then retry the command with a valid number of stop bits.

SLCONFIG0010 invalid Parity Select: <parity>

Cause: A user entered an invalid parity checking method while attempting to configure a serial port.

Action: Refer to the TCPOpen Command Reference (Chapter 2 of Part 3) for a list of valid parity selections, then retry the command with a valid parity checking method.

SLCONFIG0011 field length exceeds maximum allowable of <no.> characters

Cause: A field in either the MODEMS or PHONES database is longer than permissible. <no.> indicates the maximum allowable length for that particular field. This message is followed by the field detected with a bad length.

Action: Use a text editor to reduce the length of the involved field to its permissible length.

SLCONFIG0012 invalid phone number: <phone-number>

Cause: A phone number format specification does not begin with the pound sign (#) as required.

Action: Edit the {INSTALL}\SETUP\PHONES file using a text editor. Add the pound sign (#) in front of the phone number. For example, if the existing number is:

1234567

Then, the correct phone number format is:

#1234567

SLCONFIG0015 invalid modem response: <modem-response>

Cause: An invalid modem response is detected. Refer to the slconfig command in the TCPOpen Command Reference (Chapter 2 of Part 3) for the valid modem code.

If the modem response is not one of the above responses, the following are the possibilities:

- the modem is not a Hayes-command compatible modem, or
- the TCP/IP driver is in a bad state and returns bad codes.

Action: The following actions are suggested:

- check whether your modem is a Hayes-compatible modem. The TCPOpen serial line driver supports only this type of modem.
- if the modem is, in fact, Hayes-compatible, there is a good chance that the TCP/IP driver is in a bad state and needs to be reloaded.

SLCONFIG0020 unexpected end of parameter list

Cause: slconfig expects more parameters than were specified in the command line.

Action: Check the TCPOpen Command Reference (Chapter 2 of Part 3) for the correct parameter.

Appendix A : ODI Driver Installation

The ODI version of the TCP/IP kernel supports an interface conforming to ODI driver specification. ODI or Open-Data Link Interface specification was designed and developed by Novell, Inc. It provides a flexible data-link interface; that is a network adapter card supporting multiple frame types can support multiple network protocols and multiple network adapter cards can support a single network protocol. By conforming to the ODI specification, the TCP/IP kernel can coexist with Novell Netware network operating system on the same network adapter card and work with many network adapter cards that come with an ODI driver. The ODI interface implements the ODI specification version 1.10. A description of the specification can be obtained by contacting Novell's customer service group.

ODI specifies three network modules handling low-level networking services. They are:

- MLIDs (Multiple Link Interface Drivers),
- LSL (Link Support Layer) module,
- and Protocols

The MLID driver is ODI-conforming network adapter card driver. The driver is usually shipped with the adapter card that you purchased. Please check with the network adapter card vendor to make sure they supply a driver conforming to the ODI specification for your adapter card.

Protocols are network protocols such as Softerm TCP/IP protocols, Novell IPX, etc.

LSL acts as the "middle-man" between MLIDs and Protocols. It passes receive and transmit data to/from MLIDs and Protocols. It also handles the passing of control information to/from MLIDs and Protocols.

ODI requires LSL to be loaded before any MLID (network adapter card driver) driver or Protocol can be loaded. The Protocol is normally loaded last.

Configure TCP/IP Kernel with ODI Interface

To configure for loading the ODI TCP/IP kernel, use the TCPSETUP program and select the ODI option under Interface menu (see Chapter 2 of Part 2 for how to use TCPSETUP to setup TCPOpen). Note that TCPSETUP supports only two adapters of which ODI drivers are shipped with TCPOpen/Kernel. If your adapter is none of them, you can either:

- select any one of the available adapter and proceed with the configuration and later manually reconfigure the NET.CFG, ODI drivers' configuration, file.

- or create a new entry, import the driver and then proceed with the configuration process as instructed in Chapter 2 of Part 2.

Once all the fields and parameters selected, TCPSETUP will generate the following files as you press the [F10] key to save the selected configuration:

- {INSTALL}\ODI\NET.CFG
- {INSTALL}\BIN\LOADTCP.BAT
- {INSTALL}\BIN\RMTCP.BAT

The batch files, LOADTCP.BAT and RMTCP.BAT, contain all the commands to load the required ODI drivers and the TCP/IP kernel and to unload them when they are no longer needed, respectively. The NET.CFG contains the binding instructions and the adapter card parameters that you specified while in

TCPSETUP.   If you have faked an adapter, you have to modify this file to revise the parameters specific to the faked adapter with the parameters that reflect the configuration of your adapter card. You also have to modify the batch files to replace the driver command for the faked adapter with the driver command for your adapter.

Lastly, you need to copy the LSL.COM driver and the ODI driver for   to the   adapter card to the {INSTALL}\ODI directory. Since the LSL.COM and the ODI driver must be in the same directory as the NET.CFG file is.

Load ODI Drivers and ODI-conformed TCP/IP Kernel

To load the ODI-related drivers as well as the ODI TCP/IP kernel, simply type:

```
C> LOADTCP
C>
```

at a DOS prompt. Note that, for simplicity, all output are ignored in the above example.

Unload ODI Drivers and ODI-conformed TCP/IP Kernel

All ODI drivers can be dynamically loaded and unloaded. You can unload the ODI TCP/IP kernel and other ODI-related drivers from system memory as shown below:

```
C> RMTCP
C>
```

Sample NET.CFG File

The ODI specification allows users to configure network software modules by entering configuration information in the NET.CFG file, which resides under the directory {INSTALL}\ODI. The file contains information for network adapter card parameters such as IRQ, IObase address, RAM address, Frame type, Protocol ID, etc . . . and information for network protocols such as binding instructions. Configuration information for ODI software modules such as LSL can be entered as well. The NET.CFG entry syntax is described in the ODI specification. However, the following sample NET.CFG file shows how to use the NET.CFG file to configure Softerm ODI TCP/IP kernel, and an ODI-conforming network adapter driver. In this configuration, we have a 3C503 Ethernet network interfaces. The sample NET.CFG file instructs Softerm ODI TCP/IP kernel to bind to the 3C503 network interface MLID driver.

```
Protocol IP
        Bind #1
Protocol ARP
        Bind #1

# Setup for 3C503 card
Link driver 3C503
        INT.
        MEMDC000
        PORT 300
        FRAME ETHERNET_II
        PROTOCOL IP 800 ETHERNET_II
        PROTOCOL ARP 806 ETHERNET_II
```

Figure A1. Sample NET.CFG file

Note that Softerm ODI TCP/IP kernel currently supports ETHERNET_II and TOKEN-RING_SNAP frame types for DIX and TOKEN-RING with SNAP envelope only.

Appendix B : NDIS Driver Installation

The NDIS version of the TCP/IP kernel, referred to as TCPIP/NDIS, supports the TCP/IP protocol with the capability to interface with an NDIS-conformed MAC driver. The MAC driver, referred to as MAC/NDIS, is supplied by most hardware vendors.

TCP/IP stands for Transport Control Protocol/Internet Protocol. It consists of a set of protocols, including TCP, IP, and UDP, and corresponds to layers 3 and 4 of the ISO/OSI model. NDIS stands for Network Device Interface Specification, which is jointly defined and developed by 3Com Corporation and Microsoft Corporation. NDIS provides a standard way to develop network adapter card drivers and network protocols. Therefore, if a network protocol and a network adapter card driver conform to the NDIS specification, changing to a different network adapter card will not affect the operation of a network protocol.

Network Device Interface Specification also resolves:

- the problem of binding multiple networking protocols to a single network adapter card driver, and

- the problem of binding multiple network adapter card drivers to one networking protocol.

Therefore, multiple protocols can share a single network adapter card, or a single protocol can use multiple network adapter cards. The former is normally seen where multiple protocol suites coexist on the same transmission media such as TCP/IP and Netware's IPX, or TCP/IP and LAN Manager's NetBEUI protocol stack. The later is seen where a protocol suite has access to more than one network, such as a gateway.

As a result of this development, TCPIP/NDIS can concurrently share a network adapter card with any other protocol stack such as LAN Manager's NetBEUI.

The following sections describe how to install, configure and set up the TCPIP/NDIS driver. Please go through these sections carefully. Should any problem or question arises, please contact LANERA's technical support group.

PREPARATION

Before the MAC/NDIS driver is installed, you need to install the TCPOpen package. If you have not done so, please refer to Part 2 - Chapter 1 for proper installation and setup procedures.

After installing the TCPOpen package, you should have the following NDIS-related files:

{INSTALL}\BIN\NETBIND.EXE	Protocol/MAC binding program
{INSTALL}\SETUP\PROTOCOL.INI	Protocol Info file
{INSTALL}\NDIS\PROTMAN.DOS	Protocol Manager driver
{INSTALL}\NDIS\??????.DOS	Various MAC/NDIS drivers
{INSTALL}\BIN\TCPIP_ND.EXE	TCP/IP (Protocol) driver with NDIS interface.

Note: Our current released version supports NDIS version 1.01. NDIS version 2.01 will be included in our next release. 3Com Corporation has not released the Protocol Manager (PROTMAN.DOS) that supports many features of NDIS version 2.01.

There should be three sections in this appendix:

1. Configure and install hardware.
2. TCPOpen Installation
3. TCPOpen Installation With Existing LAN Manager

HARDWARE INSTALLATION AND CONFIGURATION

Install an Ethernet network adapter card into an unused expansion slot of an IBM PC or compatible computer. Please refer to the vendor's adapter card installation manual for proper installation.

TCPIP/NDIS works with most adapter cards' factory default configuration. However, if you have a conflict between the default configuration and another add-on card, refer to your vendor adapter card installation manual to configure your adapter card.

> **Note:** To ensure that your network adapter does not conflict with any other adapter in the system, execute the diagnostics program that comes with the adapter. This is an extra step in the installation procedure but will save you a lot of trouble later on.

The MAC/NDIS driver is a DOS device driver. It is installed from the CONFIG.SYS file together with another driver called Protocol Manager, PROTMAN.DOS. The PROTMAN.DOS driver links the MAC/NDIS driver with our protocol driver, TCPIP_ND.EXE.

PROTMAN.DOS depends on a configuration file, PROTOCOL.INI, which guides the protocol binding process. In addition, PROTOCOL.INI also contains information about the hardware configuration that is used by the MAC/NDIS driver to initialize itself during start-up.

If your system has the Protocol Manager and the PROTOCOL.INI file already installed for an existing network adapter card, and you just want to add TCP/IP connectivity on top of it. You can skip TCPOpen INSTALLATION section and go to the next section, TCPOpen INSTALLATION WITH EXISTING LAN MANAGER.

TCPOpen INSTALLATION

This section describes the installation procedure for the MAC/NDIS driver using TCP/IP exclusively. This procedure applies for first-time installation of TCPOpen.

Step 1 Perform TCPOpen installation according to Chapter 1 of Part 2 of the manual.

Step 2 Execute TCPSETUP to configure TCPOpen databases and generate the LOADTCP.BAT file which contains the commands for loading the TCPIP protocol driver and configuring it according to the setup information. Please refer to Chapter 2 of Part 2 of the manual for detailed information on how to use the TCPSETUP program. TCPSETUP will make the following changes to your system configuration files:

- PROTOCOL.INI:

Protocol manager (PROTMAN.DOS) and NDIS MAC driver entry will be added to this file.

- CONFIG.SYS:

Protocol manager and the appropriate NDIS MAC driver will be inserted into this file.

Step 3	Reboot the PC to reload the NDIS driver.
Step 4	Type LOADTCP at DOS system prompt to load the protocol driver.

Sample CONFIG.SYS,AUTOEXEC.BAT, & PROTOCOL.INI

Using TCP/IP with 3Com Etherlink/II network adapter card:

CONFIG.SYS:

```
        DEVICE=C:\TCPOPEN\NDIS\PROTMAN.DOS /I:C:\TCPOPEN\NDIS      DEVICE=C:
\TCPOPEN\NDIS\ELNK.DOS
```

AUTOEXEC.BAT:

```
SET TCPOPEN=C:\TCPOPEN;
LOADTCP
```

PROTOCOL.INI:

```
[PROTOCOL MANAGER]
            DRIVERNAME = PROTMAN$

[TCPIP]
            Drivername = TCPIP$
            BINDINGS = "ELNKII"

[ELNKII]
            DRIVERNAME  =       ELNKII$
            INTERRUPT    =      3
            IOADDRESS    =      0x300
            MAXTRANSMITS        =       40
```

TCPOpen INSTALLATION WITH LAN MANAGER

If you are using Microsoft LAN Manager on your PC and want to add TCP/IP connectivity to your system using the same adapter, you are likely to fall into this category. The following chart shows the steps need to be taken for installing and configuring your PC so that you can concurrently access both a Microsoft LAN Manager server and a TCP/IP network on the same cabling system.

Install	Install Microsoft LAN Manager LAN Manager
Reboot PC	Reboot the PC.
Install TCPOpen	Perform TCPOpen installation.
Execute TCPSETUP	Execute TCPSETUP to configure TCPOpen databases and to create LOADTCP.BAT file.
Reboot PC	Reboot the PC.

The following setup procedures are general guidelines on how to configure TCPOpen for use in concurrent with existing installation of LAN Manager.

Step 1 If you have not yet installed Microsoft LAN Manager at this point, please do so now.

Step 2 Reboot the PC and ensure that you can access LAN Manager server.

Step 3 Perform TCPOpen installation as described in Chapter 1 of Part 2 of the manual.

Step 4 Execute TCPSETUP to configure TCPOpen and generate the LOADTCP.BAT batch file which contains the commands to load the TCP/IP protocol driver and to configure it according to the setup information Please refer to Chapter 2 of Part 2 of the manual for detailed information on how to use the TCPSETUP program. TCPSETUP will make the following changes to your system configuration files:

- PROTOCOL.INI:
  TCP/IP protocol entry will be added to the existing LAN Manager PROTOCOL.INI file.

- CONFIG.SYS:
  Protocol manager (PROTMAN.DOS) and the appropriate NDIS MAC drivers will be inserted into this file if they are not there.

AUTOEXEC.BAT:
If you have the NETBIND statement in the AUTOEXEC.BAT file, replace this statement with the following statements:

```
SET TCPOPEN=C:\TCPOPEN
LOADTCP
```

Assume TCPOpen/Kernel was installed under the TCPOPEN directory in drive C.

Step 5 Reboot the PC again for all the changes to take effect. When the system comes up, TCP/IP driver is automatically loaded, configured and bound to the MAC driver. Sample Using NetBEUI and TCP/IP with 3Com Etherlink/II network adapter card.

CONFIG.SYS ,AUTOEXEC.BAT,& PROTOCOL.INI

CONFIG.SYS:

```
DEVICE=C:\LANMAN.DOS\DRIVERS\PROTMAN\PROTMAN.DOS
                              /I:C:\LANMAN.DOS
DEVICE=C:\LANMAN.DOS\DRIVERS\ETHERNET\ELNKII\ELNKII.DOS  DEVICE=C:
\LANMAN.DOS\DRIVERS\PROTOCOL\NETBEUI\NETBEUI.DOS
```

AUTOEXEC.BAT:

```
SET TCPOPEN=C:\TCPOPEN;
LOADTCP
REM == LANMAN 2.0 == DO NOT MODIFY BETWEEN THESE LINES
SET PATH=C:\LANMAN.DOS\NETPROG;%PATH%
C:\LANMAN.DOS\DRIVERS\PROTMAN\NETBIND
NET START WORKSTATION
REM == LANMAN 2.0 == DO NOT MODIFY BETWEEN THESE LINES
```

PROTOCOL.INI:

```
[PROTOCOL MANAGER]
            DRIVERNAME = PROTMAN$

[TCPIP]
            Drivername = TCPIP$
            BINDINGS = "ELNKII"

[NETBEUI_XIF]
            Drivername = NETBEUI$
            BINDINGS = "ELNKII"

[ELNKII]
            DRIVERNAME  =     ELNKII$
            INTERRUPT   =     3
            IOADDRESS   =     0x300
            MAXTRANSMITS      =     40
```

Appendix C : Packet Driver Installation

This appendix describes in details how to install and configure your PC to run TCPOpen/Kernel using Packet drivers. Beside the Clarkson University's packet drivers that come with TCPOpen/Kernel package, there are also shell drivers from BYU which allows a PC to access both Novell Netware (version 2.x or 3.x) and TCP/IP networks concurrently. You should find two sections in this appendix:

1. Clarkson University Packet Drivers

2. Using TCPOpen/Kernel and Novell Netware 2.1x & 3.x concurrently

CLARKSON UNIVERSITY PACKET DRIVERS

Included in TCPOpen/Kernel is a number of packet drivers for the most popular PC networking adapter cards.
TCPOpen/Kernel has been tested with these packet drivers. Note that the packet drivers included in this diskette are not developed or maintained by LANERA Corporation. Therefore, we assume no responsibility or liability for these drivers. We only distribute the drivers as a service to our customers.

Users who decide to use these drivers can contact Clarkson University for update information (see the HOWTOGET.IT file under the directory PDS on the distribution diskette). We will update this diskette at every new release of TCPOpen. Note that the file DRIVERS.DOC on the distribution diskette contains loading instructions for many Packet drivers included with the diskette.

Many hardware vendors also supply Packet drivers for their own adapters. You may want to use these drivers instead.

The TCPSETUP program supports many Packet drivers from Clarkson University - Please refer to Chapter 1 of Section 2 for more detailed information on how to use the TCPSETUP program.
However, if you are using a Packet driver not listed in the TCPSETUP program, you should do the followings:

Step 1 Add a new entry to the Packet driver list in TCPSETUP. You can enter all the field's value, but only the Adapter Model field is needed.

Step 2 After the LOADTCP.BAT file is generated, replace the following comment line in this batch file:

```
REM insert Packet driver command for <adapter-model> here
```

with the actual Packet driver command.

Step 3 Copy the new Packet driver to {INSTALL}\PDS.

USING TCP/IP AND NOVELL NETWARE CONCURRENTLY

Brigham Young University shell drivers allow Novell Netware to share a network adapter in the PC workstation with other type of networking systems such as TCP/IP. This is accomplished by using software that supports the Packet Driver Specification. This interface specification, proposed by FTP Software, Inc., allows various products to concurrently utilize the network adapter card installed in the PC. Please read the following pages carefully to configure your PC for this particular environment.

TCPOpen with Novell Netware 2.1x & 3.1x

INSTALL Perform TCPOpen installation.

Execute TCPSETUP Execute TCPSETUP to set up TCPOpen databases and to create the LOADTCP.BAT file.

Configure Ethernet Frame Configure the Novell Netware file server to use Ethernet
II
packet frame instead of IEEE 802.3.

Reboot Novell server Reboot the Novell Netware file server.

Create SHGEN/WSGEN Backup Diskette Create a backup of the original Novell
Netware SHGEN (Netware version 2.15 & 2.20) or
WSGEN (Netware
version 3.1x) diskette using the DOS DISKCOPY command.

Copy all .LAN and .OBJ files to SHGEN/WSGEN Backup Diskette
Copy all *.LAN and *.OBJ files from the {INSTALL}\BYU directory into the SHGEN (Netware version 2.15 & 2.20) or WSGEN (Netware version 3.1x) backup diskette.

Generate Netware Shell Perform Netware shell generation to create IPX.COM according to the Novell workstation installation documentation.

Execute LOADTCP.BAT Run LOADTCP.BAT batch file to load the PACKET
driver,
TCP/IP driver, and IFCONFIG program as previously configured by TCPSETUP.

Load IPX.COM and NETx.COM Load Novell's IPX.COM and NETx.COM, where "x" denotes the DOS version installed on the local PC.

The following setup procedures are general guidelines on how to configure TCPOpen for use in concurrent with Novell Netware version 2.15, 2.20 or 3.x. The exact options used for each command are site-specific and required careful review of the TCPOpen and/or Novell Netware manual.

Step 1 Perform TCPOpen installation as described in Chapter 1 of Part 2 of the manual.

Step 2 Execute TCPSETUP to configure TCPOpen databases and generate a LOADTCP.BAT batch file. Please refer to Chapter 1 of Part 2 of the manual for detailed information on how to use TCPSETUP program.

Step 3 Configure the Ethernet frame type on the Novell Netware file server to Ethernet II instead of the default IEEE 802.3. This is a requirement since TCP/IP network only uses Ethernet II frame type.

a) For Novell Netware version 2.15 or 2.20, ECONFIG.EXE utility must be run to modify the NET$OS.EXE file for Novell file server to use Ethernet II. Follow these steps:

- From a Novell Netware workstation, logs in the server as "supervisor"
- Remove the read-only attribute from NET$OS.EXE file

- Execute ECONFIG.EXE program as follows:

  ```
  ECONFIG F:NET$OS.EXE A:E 8137
  ```

Type code for Ethernet_II Ethernet configuration. The network to be configured Network drive where NET$OS.EXE file resides.

- Add back the read-only attribute to the NET$OS.EXE file

b) For Novell Netware version 3.1x, the Ethernet frame type can be changed by assigning ETHERNET_II to the FRAME parameter in the LOAD LAN driver command, such as:

```
LOAD 3c503 FRAME=ETHERNET_II PORT=300 INT=3 MEM=CC000
```

Step 4 Reboot the Novell Netware file server now for the changes to take effect.

Step 5 Create a backup of the original Novell Netware SHGEN (Netware 2.15 or 2.2) or WSGEN (Netware 3.1x) diskette using DOS DISKCOPY command.

Step 6 Copy all *.LAN and *.OBJ from the {INSTALL}\BYU directory into the SHGEN (Netware version 2.15 or 2.20) or WSGEN (Netware version 3.1x) backup diskette.

Step 7 Generate the Netware shell by execute SHGEN.EXE (Netware version 2.15 or 2.2) or WSGEN.EXE (Netware version 3.1x) from the backup diskette. From the shell generation menu, select:

"Packet Driver v2.01. BYU Copyright 1989"

A new IPX.COM will be created upon completion.

Step 8 Execute the LOADTCP.BAT batch file. This file should include the PACKET driver, TCP/IP packet driver interface program (TCPIP_PD.EXE) and the network interface configuration program (IFCONFIG.EXE).

Step 9 Load IPX.COM and NETx.COM where "x" denotes the version of DOS installed on the PC.

Appendix D : Serial Line Interface Protocol (SLIP)

Softerm TCP/IP Kernel also supports TCP/IP networking over PC serial line ports. The TCP/IP serial line driver conforms to the SLIP protocol.

The serial line interface provides the most cost effective way to connect your PC to a host computer with TCP/IP, allowing your PC to connect using its serial communications ports (COM1-COM4). No additional hardware is required. With this interface, you can:

- directly connect to another host computer using a Null modem cable (see Figure D4).
- dial a remote host computer through a modem, provided the remote host supports dial-up TCP/IP connection.

Figures D1 and D2 illustrate the use of serial ports for connecting to another host: directly or indirectly through a modem. The SLIP interface can be configured to use one of the available COM ports or all of them for transferring and receiving data packets to and from another computer system. TCPOpen supports up to two serial line interfaces:

- sl0 which uses the COM1 port by default,
- sl1 which uses the COM2 port by default.

Since COM1 is widely used by other communications programs, we recommend using other COM ports for Softerm's TCP/IP driver.

WARNING
After configuring a port for TCP/IP Kernel, other communication programs can still use the port. These programs must restore the interrupt vector to TCP/IP's COM port interrupt handler and associated port configuration when they terminate.

Using SLIP with Direct RS-232 Connection

SLIP works with a direct RS-232 (Null Modem) cable connection between computers, as illustrated in figure E1.

To configure SLIP for a direct connection, do the following:

Step 1: Decide:

- which COM port to use for the connection
- what baud rate and data format to configure
- what SLIP Internet address to use.

Step 2: Use an editor to add the following entries to the HOSTS database file:

- name and Internet address of the local SL interface
- name and Internet address of the remote SL interface.

You also need to add your host entry associated with the serial line interface to the HOSTS database on the remote machines.

Step 3: Use an editor to add the newly-created serial line network to the NETWORKS database file. This addition allows the netstat program to display the network identification by name instead of by network address.

Step 4: Connect your PC's serial port and the remote host's serial port with a Null modem cable.

Step 5: Load the TCP/IP protocol driver. For example,

```
C> tcp_sl
```

Step 6: Assume you are using sl0 which associates with COM port 1 by default. If you want to use other COM ports instead of COM1, you can do so by assigning to desired port to the sl0 interface. Use the following command to reassign a COM port to a serial line interface:

```
C> slconfig sl0 port 3
```

The above command reassigns sl0 to COM port 3. If you want to use the default port assignment, you can skip the above step.

Use the slconfig command to configure the selected port's baud rate and data format, if necessary. For example,

```
C> slconfig sl0 set 2400,8,1,N
```

The above command sets the port to operate at 2400 baud, 8 data bits, 1 stop bit, and no parity bit.

Step 7: Use the ifconfig command to assign an Internet address to the serial line interface and to activate it.

```
C> ifconfig sl0 sl_dos sl_remote up
```

This example assumes that sl_dos is your PC host name associated with the serial line interface and sl_remote is the remote system's host name.

Step 8: You can now run TCP/IP applications or use the ping program to verify connectivity between the two hosts.

Using SLIP with a Dialup Modem

To connect to a remote computer using SLIP and a modem, as illustrated in figure E2, do the following:

Step 1: Decide:

- which COM port to use for the connection
- what baud rate and data format to configure
- what SLIP Internet address to use.

Step 2: Use an editor to add the following entries into the HOSTS database file:

- name and Internet address of the local SL interface
- name and Internet address of the remote SL interface.

You also need to add your host entry associated with the serial line interface to the HOSTS database on the remote machines.

Step 3: Use an editor to add the newly-created serial line network to the NETWORKS database file. This addition allows the netstat program to display the network identification by name instead of by network address.

Step 4: Use a Modem cable (see Figure E4) to attach the modem to the COM port, and connect the modem to the phone jack as shown in figure D3.

Step 5: Load the TCP/IP protocol driver. For example,

```
C> tcp_sl
```

Step 6: Assign another COM port to the serial interface if desired and then, configure the selected port's baud rate and data format if necessary.

```
C> slconfig sl0 set 2400,8,1,N
```

Step 7: Assign an Internet address to the SLI and activate it.

```
C> ifconfig sl0 sl_dos sl_remote up
```

This example assumes that slÄdos is your PC host name and slÄremote is the remote system host name.

Step 8: Use an editor to prepare the MODEMS and PHONES database files as outlined in Chapter 1 of Part 4 (Required Files).

Step 9: Attach the modem to the COM port.

```
C> slconfig sl0 attach Smartmodem
```

This example assumes that you prepared an entry for Smartmodem in the MODEMS file, as mentioned previously.

Step 10: Make the call to the remote host using the slconfig command.

```
C> slconfig sl0 dial #1234567
```

This example assumes that 1234567 is the remote phone number.

Step 11: After the call is answered, you can run TCP/IP applications or use the ping program to verify connectivity between the two hosts.

Step 12: Do not forget to hang up the call after you are done. To hang up a call, type:

```
C> slconfig sl0 hangup
```

TROUBLESHOOTING

If you are unable to communicate with the remote host, look for the following problems:

- Incompatible port configurations.

Make sure that the port configurations on both ends of the serial link are the same. A mismatch may cause the UARTs to incorrectly decode signals on the link.

- Wrong cable type.

If the connection between the two hosts is a direct connection as shown in figure E1, you must use a NULL cable, or a NULL modem attached to one end of a straight RS-232 cable to connect the two systems.

If you are connected to the remote host by modem, you must use a straight RS-232 cable to connect the COM port to the modem.

- Conflicting IRQ or I/O base addresses.

  Make sure no other adapter in the system uses the same IRQ.

COM Port	IRQ	BASE I/O ADDRESS
COM1, COM3	4	0x3f8, 0x3e8
COM2, COM4	3	0x2f8, 0x2e8

  If possible, use a program that tests whether the port is functioning.

- Inappropriate route configuration.

Check the route table, by using netstat -r, to ensure you have correct routing information on both systems.

Cabling Diagrams

	MODEM		NULL MODEM	
	Host PC	Computer	Host PC	Computer
Transmit Data	2	2	2	2
Receive Data	3	3	3	3
Request to Send	4	4	4	4
Clear to Send	5	5	5	5
Data Set Ready	6	6	6	6
Signal Ground	7	7	7	7
Carrier Detect	8	8	8	8
Data Terminal Ready	20	20	20	20

Figure D4. Cable Configurations

The Modem cable is used when connecting the PC to a modem. The Null modem cable is used when connecting a remote host's serial port directly to the local PC's serial port.

Appendix E : PC Interrupt Vector and I/O Port Usages

PC Interrupt Vector Usage

Interrupt usage in the range 0x60 through 0x80:

60	-	-	reserved for user interrupt
61	-	-	reserved for user interrupt
62	-	-	reserved for user interrupt
63	-	-	reserved for user interrupt
64	-	-	reserved for user interrupt
65	-	-	reserved for user interrupt
66	-	-	reserved for user interrupt
67	-	-	LIM EMS. . .
67	DE	00	Virtual Control Program Interface - INSTALLATION CHECK. . .
68	01	-	APPC/PC. . .
69	-	-	unused
6A	-	-	unused
6B	-	-	unused
6C	-	-	system resume vector (CONVERTIBLE)
6D	-	-	DOS 3.2 Realtime Clock update
6E	-	-	unused
6F	-	-	Novell NetWare - PCOX API (3270 PC terminal interface)
6F	00	-	10-NET - LOGIN. . .
70	-	-	IRQ8 - AT/XT286/PS50+ - REAL-TIME CLOCK
71	-	-	IRQ9 - AT/XT286/PS50+ - LAN ADAPTER 1
72	-	-	IRQ10 - AT/XT286/PS50+ - RESERVED
73	-	-	IRQ11 - AT/XT286/PS50+ - RESERVED
74	-	-	IRQ12 - PS50+ - MOUSE INTERRUPT
75	-	-	IRQ13 - AT/XT286/PS50+ - 80287 ERROR
76	-	-	IRQ14 - AT/XT286/PS50+ - FIXED DISK
77	-	-	IRQ15 - AT/XT286/PS50+ - RESERVED
78	-	-	not used
79	-	-	not used
7A	-	-	Novell NetWare - LOW-LEVEL API
7A	-	-	AutoCAD Device Interface
7B	-	-	not used
7C	-	-	not used
7D	-	-	not used
7E	-	-	not used
7F	-	-	HDILOAD.EXE - 8514/A VIDEO CONTROLLER INTERFACE
7F	-	-	HLLAPI (High-Level Language API)
80	-	-	reserved for BASIC

I/O Port Usage

I/O		
Address	AT	PS/2
200-20F	Game I/O adapter	UNDOCUMENTED
210-21F	21F RESERVED	UNDOCUMENTED
220-24F	AVAILABLE FOR I/O	UNDOCUMENTED

250-25F	AVAILABLE FOR I/O	UNDOCUMENTED
260-26F	AVAILABLE FOR I/O	UNDOCUMENTED
270-27F	Parallel printer port 2	Parallel port 3
280-28F	AVAILABLE FOR I/O	UNDOCUMENTED
290-29F	AVAILABLE FOR I/O	UNDOCUMENTED
2A0-2AF	AVAILABLE FOR I/O	UNDOCUMENTED
2B0-2BF	Alternate EGA	UNDOCUMENTED
2C0-2CF	Alternate EGA	UNDOCUMENTED
2D0-2DF	Alternate EGA	UNDOCUMENTED
2E0-2EF	GPIB 0, data acquisition 0	UNDOCUMENTED
2F0-2FF	Serial port 2	Serial port 2 (RS-232-C)
300-30F	Prototype card	UNDOCUMENTED
310-31F	Prototype card	UNDOCUMENTED
320-32F	AVAILABLE FOR I/O	UNDOCUMENTED
330-33F	AVAILABLE FOR I/O	UNDOCUMENTED
340-34F	AVAILABLE FOR I/O	UNDOCUMENTED
350-35F	AVAILABLE FOR I/O	UNDOCUMENTED
360-36F	PC network (low address)	UNDOCUMENTED
370-37F	Parallel printer 1	Parallel port 2
380-38F	SDLC or second bisync.	UNDOCUMENTED
390-39F	Cluster adapter	UNDOCUMENTED
3A0-3AF	First bisync controller	UNDOCUMENTED
3B0-3BF	Monochrome display adapter	Video subsystem, parallel 1
3C0-3CF	Enhanced graphics adapter	Video subsystem
3D0-3DF	Color graphics adapter	Video subsystem
3E0-3EF	AVAILABLE FOR I/O	UNDOCUMENTED
3F0-3FF	Floppy disk adapter,	Diskette drive controller, 1st async Serial 1

Glossary of Terms

802.X An IEEE committee responsible for standards describing network cabling, protocol access, electrical and physical topology. The standards describe the data-link and physical layers of a LAN. IEEE 802.3 is from the 802 subcommittee and also defines the IBM Token-Ring architecture.

ACK An abbreviation for Acknowledgment.

acknowledgment A message from a receiver indicating a successful transmission. An acknowledgment can come from the physical wire voltage level, the hardware link level, or the program to program level.

address resolution The conversion of an IP address to a physical address. The resolution may require broadcasting over a local network.

ANSI American National Standards Institute. The group that sets standards for the U. S. computer industry, including networks.

ARCnet Attached Resources Computing. A PC networking architecture, usually on coaxial cable. ARCnet uses a token-passing bus architecture, usually implemented in the hardware of a network adaptor card.

ARP Address Resolution Protocol. A TCP/IP protocol that dynamically binds a low-level hardware address to a high-level IP address. ARP is limited to single physical networks that support hardware broadcast.

ARPA Advanced Research Projects Agency. This government agency, formerly named DARPA, funded the ARPANET and later connected Internet.

ARPANET One of the first widely used networks, set up in 1969 and serving as the base for Internet development. ARPANET was built by BBN and funded by ARPA. ARPANET used individual packet switching nodes connected by leased lines.

baseband Baseband is typical for networks such as Ethernet that use a single carrier frequency and require all stations on the network to participate in every transmission.

baud The number of times per second that a signal can change on a transmission line. Transmission lines usually have only two signal states, making the baud rate equal to the number of bits per second transferred by the line. The transmission may also use some of the bandwidth, so data may not transfer at the line's rated capacity. For example, an asynchronous line needs 10 bit-times to send an 8 bit character, so a 9600 bit per second line can transfer only 960 characters per second.

Berkeley broadcast A non-standard IP broadcast address that uses all zeros in the host address instead of all ones.

bps bits per second. A measure of data transmission rates.

bridge A computer that connects networks and forwards packets from one network to another. A bridge usually operates at the physical network level, and uses physical addresses, not IP addresses.

broadcast A delivery of a packet copy to all hosts attached to the network. A broadcast can be done with hardware, as in Ethernet, or with software.

connection The path between two protocol processes that provides stream delivery services. With TCP/IP, the connection extends from a TCP process on one machine to a TCP process on another machine.

connectionless service A packet delivery service offered by the Internet Protocol (IP) and most hardware. This service handles each packet as a complete communication that contains its source and destination addresses. Connectionless services can drop packets or deliver packets out of sequence.

DARPA (See ARPA.)

Bandwidth The range of frequencies of data transmission over a connection. The difference between the highest and lowest frequencies transmitted.

Baseband A way to allow only one signal at a time on a distribution medium. A medium can be shared with time division multiplexing.

BSD Berkeley Software Distribution.

CSMA Carrier-Sense Multiple Access. An access control technique for multiple-access transmission media. A transmitting machine transmits only if it senses that the transmitting medium is idle.

CSMA/CD Carrier-Sense Multiple Access with Collision Detection. Collision detection allows the network node to resend data after a data collision on the transmitting media.

Cheapernet A baseband LAN using less expensive components and thinner cable than Ethernet or the first IEEE 802.3 standard. Cheapernet uses fewer stations and less of a network span than Ethernet, but gives the same data rate (10 Mbps).

Client-server A distributed system interaction in which a program at one site sends a request to a program at another site and waits for a response. The client is the requesting program and the server is the responding program.

DARPA Internet (See Internet.)

datagram (See IP datagram.)

DNS Domain Name System. An on-line distributed database that maps machine names into IP addresses. Internet uses DNS servers. DNS allows separate mappings between mail destinations and IP addresses.

domain One part of a DNS naming hierarchy. The domain name is a series of names, called labels, separated by periods (dots).

dotted decimal notation (dot notation) The syntax that represents four 8-bit base 10 numbers with periods (dots) separating the numbers. TCP/IP applications usually accept dotted decimal notation for destination machine names.

Encapsulation A layered protocol technique in which a lower level protocol accepts a message from a higher level protocol and places the message in the data portion of the low level frame. Encapsulated packets on a physical network may have a sequence of headers from the physical network frame, the Internet Protocol (IP), the transport protocol, and so on.

Ethernet A local area network (LAN) from the Xerox Corporation Palo Alto Research Center.

flow control Controlling the rate of hosts or gateways injecting packets into a network or internet. Flow control can be used at various levels of the network.

fragment A piece of a datagram. A fragment results when an IP gateway divides an IP datagram for transmission across a network. Fragments and complete datagrams have the same format. IP receiving software re-assembles fragments into complete datagrams.

FTP File Transfer Protocol. The standard TCP/IP high-level protocol that transfers files from one machine to another. FTP uses the TELNET and TCP protocols. A server side client must supply a login identifier and password.

gateway A dedicated computer that connects two or more networks and routes packets from one to the other. An IP gateway routes IP datagrams to and from the networks it connects. A gateway can route packets to another gateway. A gateway is sometimes called an IP router.

hardware address A low-level address used by a physical network. Ethernet uses 48-bit vendor assigned hardware addresses.

hop count A count of the gateways between two points in an internet.

host An end-user computer that connects to a network.

ICMP Internet Control Message Protocol. The part of the Internet Protocol (IP) that handles error and control messages. ICMP carries problem reports from gateways and hosts back to the datagram sender. An ICMP echo test can determine whether a destination is active.

internet A collection of packet switching networks connected by gateways and protocols. The networks can act as a single large network. Internet in upper case denotes the connected Internet and its TCP/IP protocols.

Internet A collection of networks and gateways using the TCP/IP protocols that functions as a single network. Universities, government research labs, military installations, and private companies use Internet.

Internet address (See IP address.)

Internet Protocol (See IP.)

interoperability The ability of differing software and hardware from different vendors to communicate. Interoperability is the goal of internetworking: to interact at the network transport level without knowing the underlying technologies.

IP Internet Protocol. The standard TCP/IP protocol defining a datagram as the basic transmission unit for an internet. IP includes the ICMP control and error message protocol, and transmits information across the internet.

IP address A 32-bit address assigned to hosts in a TCP/IP internet. The IP address has a host component and a network component.

IP datagram The basic information unit on a TCP/IP internet. A datagram contains data, a source address, and a destination address.

kbps kilo bits per second. A measure of the rate of data transmission.

LAN A physical network operating at high speed over short distances. Ethernet is an example.

mbps millions of bits per second. A measure of the rate of data transmission.

IEEE Institute of Electrical and Electronic Engineering.

packet The unit of data on a packet switching network. Some documentation describes IP datagrams as packets.

PING Packet InterNet Groper. A program that tests a TCP/IP destination response by sending an echo request.

Protocol A description of the rules and procedures for using a network. A common protocol allows different networking programs to use the same network.

RFC Request For Comments. An on-line collection of notes including requested and changed TCP/IP protocol standards, ideas, observations, and informal techniques. RFC notes are available from the Network Information Center.

rlogin Remote Login. A Berkeley 4BSD UNIX system service allowing a user to login to UNIX systems across an internet. rlogin passes user environment information, such as terminal type, to the remote machine.

route A network path from a source to a destination. A route can include many gateways and physical networks.

RS232 An EIA standard giving the electrical characteristics of slow speed connections between terminals and computers or between two computers. The speed limit is 20 kbps and the distance limit is 500 feet, but most manufacturers support greater speed and longer distances. The commonly used RS232C standard is usually called RS232.

socket A Berkeley 4BSD UNIX system tool that gives a program access to the TCP/IP protocols. The program opens a socket, requests a specific service, binds the socket to a destination, and then sends or receives data.

TCP Transmission Control Protocol. The standard TCP/IP transport level protocol for a full duplex stream service. TCP allows a process on one machine to send data to a process on another machine. TCP software is usually in the operating system and uses the IP protocol to transmit information across an internet. TCP can be used as a one-way simplex connection.

TCP/IP Internet Protocol Suite The complete TCP/IP protocol group.

TELNET The standard TCP/IP remote terminal connection protocol. A user can login to a remote machine and work on the remote machine as if the user had a direct terminal connection.

TFTP Trivial File Transfer Protocol. The standard TCP/IP protocol for simple, connectionless datagram delivery.

time to live (See TTL.)

token ring A type of network technology that controls media access by passing a distinguishing packet, called a token, from machine to machine. A machine can transmit a packet only when holding the token. Specifically, token ring refers to the token ring network hardware produced by IBM.

TTL Time To Live. A technique that avoids endlessly looping packets. Each IP datagram is given a time to live at creation, and each gateway decrements the time when it processes a packet. A gateway discards a datagram if the time to live is zero.

UDP User Datagram Protocol. A standard TCP/IP protocol allowing a program on one machine to send a datagram to a program on another machine using the Internet Protocol. UDP datagrams include a protocol port number that identifies the destination program on the remote machine. UDP also includes a checksum for the data.

well-known port A pre-assigned protocol port number used by transport level protocols such as TCP and UDP. Clients use well-known port assignments to locate servers. Examples are ports assigned to echo servers, time servers, remote login servers, and file transfer servers.