AT&T

UNIX® SYSTEM V/386 RELEASE 4

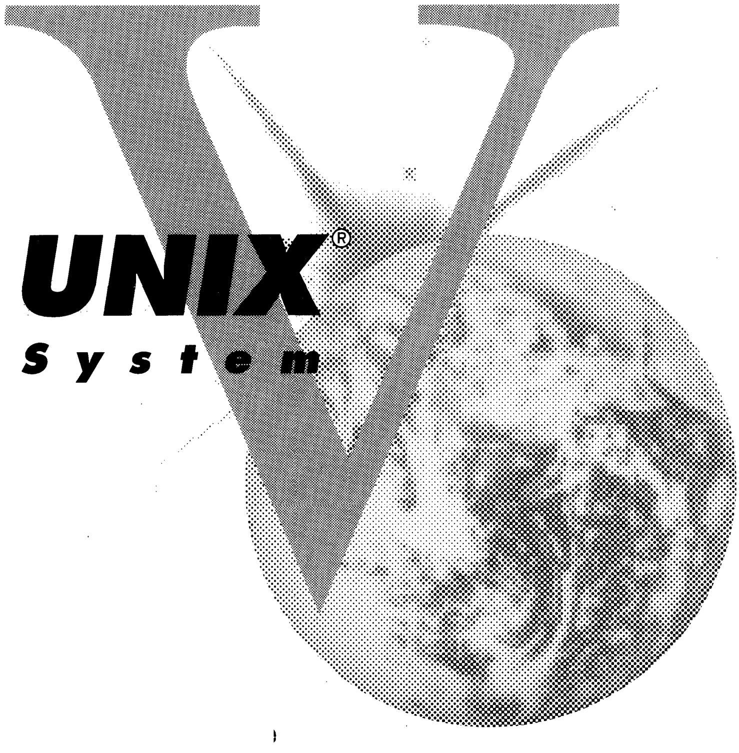
PC-Interface Administrator's Guide

UNIX Software Operation

AT&T

UNIX® SYSTEM V/386 RELEASE 4

PC-Interface Administrator's Guide



UNIX Software Operation

Published by Prentice-Hall, Inc.
A Division of Simon & Schuster
Englewood Cliffs, New Jersey 07632

ACKNOWLEDGEMENT

Portions of this book have been provided by Locus Computing Corporation.

IMPORTANT NOTE TO USERS

While every effort has been made to ensure the accuracy of all information in this document, AT&T assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document, its updates, supplements, or special editions, whether such errors are omissions or statements resulting from negligence, accident, or any other cause. AT&T further assumes no liability arising out of the application or use of any product or system described herein; nor any liability for incidental or consequential damages arising from the use of this document. AT&T disclaims all warranties regarding the information contained herein, whether expressed, implied or statutory, *including implied warranties of merchantability or fitness for a particular purpose.*

AT&T makes no representation that the interconnection of products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting or license to make, use or sell equipment constructed in accordance with this description.

AT&T reserves the right to make changes without further notice to any products herein to improve reliability, function, or design.

TRADEMARKS

EtherLink and 3Com are trademarks of 3Com Corporation
Ethernet is a registered trademark of Xerox Corporation
EXOS 205 is a trademark of Excelan Inc.
IBM and PC are trademarks of International Business Machines Corporation
MS-DOS is a registered trademark of Microsoft Corporation
PC-Interface is a trademark of Locus Computing Corporation
UNIX is a registered trademark of AT&T
VT100 is a trademark of Digital Equipment Corporation
XEROX is a registered trademark of Xerox Corporation

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-957507-3

UNIX
PRESS
A Prentice Hall Title

PRENTICE HALL

ORDERING INFORMATION

UNIX® SYSTEM V, RELEASE 4 DOCUMENTATION

To order single copies of UNIX® SYSTEM V, Release 4 documentation, please call (201) 767-5937.

ATTENTION DOCUMENTATION MANAGERS AND TRAINING DIRECTORS:
For bulk purchases in excess of 30 copies please write to:
Corporate Sales
Prentice Hall
Englewood Cliffs, N.J. 07632
Or call: (201) 592-2498

ATTENTION GOVERNMENT CUSTOMERS: For GSA and other pricing information please call (201) 767-5994.

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

AT&T UNIX® System V Release 4

General Use and System Administration

*UNIX® System V/386 Release 4 PC-Interface Administrator's Guide
*UNIX® System V/386 Release 4 Network User's and Administrator's Guide
*UNIX® System V/386 Release 4 Product Overview and Master Index
*UNIX® System V/386 Release 4 System Administrator's Reference Manual
*UNIX® System V/386 Release 4 User's Reference Manual
*UNIX® System V/386 Release 4 MULTIBUS® Reference Manual
*UNIX® System V/386 Release 4 MULTIBUS® Installation and Configuration Guide
*UNIX® System V/386 Release 4 Mouse Driver Administrator's Guide
*UNIX® System V/386 Release 4 Transport Application Interface Guide
UNIX® System V Release 4 User's Guide
UNIX® System V Release 4 System Administrator's Guide

General Programmer's Series

*UNIX® System V/386 Release 4 Programmer's Reference Manual
*UNIX® System V/386 Release 4 Programmer's Guide: SCSI Driver Interface
UNIX® System V Release 4 Programmer's Guide: ANSI C and Programming Support Tools
UNIX® System V Release 4 Programmer's Guide: Character User Interface (FMLI and ETI)
UNIX® System V Release 4 Programmer's Guide: Networking Interfaces
UNIX® System V Release 4 Programmer's Guide: POSIX Conformance
UNIX® System V Release 4 Programmer's Guide: Support Services and Application Packaging Tools

System Programmer's Series

*UNIX® System V/386 Release 4 Device Driver Interface/Driver-Kernel Interface (DDI/DKI) Reference Manual
*UNIX® System V/386 Release 4 Integrated Software Development Guide
UNIX® System V Release 4 Programmer's Guide: STREAMS

Migration Series

UNIX® System V Release 4 ANSI C Transition Guide
UNIX® System V Release 4 BSD/XENIX® Compatibility Guide
*UNIX® System V/386 Release 4 Migration Guide

Graphics Series

UNIX® System V Release 4 OPEN LOOK™ Graphical User Interface Programmer's Reference Manual
UNIX® System V Release 4 OPEN LOOK™ Graphical User Interface User's Guide
UNIX® System V Release 4 Programmer's Guide: XWIN™ Graphical Windowing System Xlib—C Language Interface
UNIX® System V Release 4 Programmer's Guide: OPEN LOOK™ Graphical User Interface
UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System NeWS
UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System Server Guide
UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System tNt Technical Reference Manual
UNIX® System V Release 4 Programmer's Guide: X11/NeWS® Graphical Windowing System XVIEW™
UNIX® System V Release 4 Programmer's Guide: XWIN™ Graphical Windowing System Addenda: Technical Papers
UNIX® System V Release 4 Programmer's Guide: XWIN™ Graphical Windowing System The X Toolkit

*386 specific titles
Available from Prentice Hall

Contents

About This Book

1 Introduction

2 About PC-Interface

3 PC-Interface Networks

4 **PC-Interface Administration**

5 **PC-Interface System Architecture**

6 **Problem-Solving Tools**

A **Appendix A**

About This Book

About This Book

This manual explains how to install and administer the LAN and RS-232 versions of the PC-Interface™ software on a host computer running a UNIX® System V operating system with TCP/IP networking services. It covers thoroughly the system administrator's responsibilities for installation and day-to-day operation and maintenance of the PC-Interface software. For information on the operation of PC-Interface from the user's point of view, refer to the *PC-Interface User's Guide.*

The PC-Interface software allows integration of personal computers into a UNIX system host computer network. Features of this integration let you:

- Run DOS® applications while using the file and print services of one or more UNIX system hosts as if the hosts were local fixed disks.
- Execute UNIX system commands from DOS on a personal computer.
- Conduct a standard UNIX operating system session on a host computer, using your personal computer as if it were a terminal.

The UNIX system host computer can support a network of personal computers including any of the IBM™ personal computer line (PC, XT, and AT), the IBM PS/2 line, and IBM-compatible computers manufactured by other vendors.

Throughout this manual, the term "personal computer" refers to any of the personal computers listed above or other compatible personal computers. The term "host" refers to a computer running the UNIX operating system. The term "UNIX" refers to a UNIX System V or other compatible operating system. The term "DOS" refers to the Microsoft Disk Operating System program (MS-DOS) (Version 3.10 or later) or the IBM Disk Operating System program (PC-DOS) (Version 3.10 or later). The term "LAN" means local area network and refers to a connection path from your personal computer to a host via a network such as Ethernet®.

Who Should Use This Manual

This manual is written for the PC-Interface system administrator—the person who manages the day-to-day operation of the system. The system administrator should also read the *PC-Interface Host Installation Guide,* which provides complete instructions on installing PC-Interface on the host system.

This guide assumes that you are familiar with UNIX system commands and basic system administration activities such as mounting a tape, making directories, adding users, and booting the host computer. It also assumes that you are familiar with the information contained in the *PC-Interface User's Guide.* Familiarity with basic DOS commands and with your network hardware is useful but not required.

Typographic Conventions

This guide uses several typographic conventions to help you distinguish between DOS and UNIX system commands and file names and to help you recognize text that you must type exactly as shown. These conventions are:

- References to UNIX system commands and file names are in a constant-width font (for example, `lpadmin`). DOS commands and file names are in uppercase (for example, DIR D:\U\BOB).
- Examples showing exactly what you type use the constant-width font for both DOS and UNIX system commands:

  ```
  on d: mv \u\mary\messagetoall \u\mary\message
  ```

- Italics indicate generic information for which you should substitute actual values for your system. For example, the following means you should supply the names of the source and target files to be used by the UNIX system `mv` command:

  `on d: mv` *file1 file2*

- The path separator and switch character for the operating system under discussion are used: the slash (/) and hyphen (−) in the UNIX operating system environment, and the backslash (\) and slash (/) in the DOS environment.

- Prompts are shown (in bold) only in examples that might be confusing without them.
- Examples do not explicitly show carriage returns. It is assumed that you type a carriage return at the end of each line.

Organization of This Manual

This manual has six chapters and one appendix as follows:

Chapter 1. "Introduction" acquaints you with the PC-Interface software and configuration requirements for your UNIX system host computer.

Chapter 2. "About PC-Interface" describes the operation of the PC-Interface software in more depth and discusses aspects of the DOS environment with which the administrator may not be familiar.

Chapter 3. "PC-Interface Networks" discusses the networks that support PC-Interface operation.

Chapter 4. "PC-Interface Administration" tells you how to get started with the administration of PC-Interface, how to support day-to-day operation of PC-Interface, and how to tailor your host system to better suit the way your system is used.

Chapter 5. "The PC-Interface System Architecture" is a thorough description of the system architecture for the experienced administrator who wants to know more about PC-Interface operation.

Chapter 6. "Problem-Solving Tools" suggests some trouble-shooting procedures for you to follow when PC-Interface users experience problems they can't solve themselves.

Appendix A. "PC-Interface Messages" lists and explains most of the error messages you might receive from PC-Interface. (Messages relating to software installation and removal procedures are listed in the *PC-Interface Host Installation Guide.)*

Related Publications

1. Other PC-Interface publications:
   - *PC-Interface User's Guide*
   - *PC-Interface Host Installation Guide*
   - *PCILIB (PC-Interface Extended Library)*
2. UNIX System V/386 Release 4 publications:
   - *User's Reference Manual*
   - *User's Guide*
   - *System Administrator's Guide*

NOTE

Related publications includes documentation that accompanied your personal computer.

1 Introduction

PC-Interface

The PC-Interface software package allows you to bridge the gap between your personal computer and UNIX system hosts. With PC-Interface and either an RS-232 or a LAN connection to a UNIX system host, you can run DOS applications using data from the UNIX file system. You can store files and even DOS applications on the UNIX system host and access them just as if they were on a local fixed disk.

When you use the host file services of PC-Interface, the file security capabilities of the UNIX system allows you to share files throughout a personal computer network while protecting your files from unauthorized access. You can also take advantage of shared host resources, such as a laser printer.

Additionally, you can use PC-Interface to enable a personal computer to emulate a VT220™ terminal, allowing you to conduct a regular UNIX operating system session.

For all its power, PC-Interface is remarkably simple to use. DOS users need to know nothing about the UNIX system to use host file services. Just treat the UNIX system as an enhanced disk drive connected directly to your personal computer. Users familiar with both the DOS and UNIX operating system environments can combine host file services and terminal emulation, toggling back and forth between the two modes. For example, you could create a text file in host file services mode using a DOS word-processing package, then switch to terminal emulation mode to include that file in a UNIX system mail message.

Minimum Requirements

Host and Network Requirements

For LAN-Based Communication to use the PC-Interface software over a LAN, you must have the following:

- One or more host computers.
- A UNIX System V operating system, with TCP/IP service available through either a BSD socket interface or System V TLI.
- One LAN controller card for each host.
- Enough LAN cable to connect the host and personal computers.
- Appropriate adapters to connect the LAN board, the LAN cable, and the transceiver (if used).
- Terminating resistors (or "terminators") for the ends of the network cable.

Host Requirements for RS-232-Based Communication

To use the PC-Interface software over RS-232, you must have the following:

- One or more host computers.
- A UNIX System V operating system.
- One RS-232 port for each personal computer connected.
- Each personal computer connected via RS-232 requires an RS-232 cable with appropriate connectors to attach to an RS-232 port on the host and to an asynchronous communications adapter in the personal computer.

Personal Computer Requirements

- One or more IBM personal computers (PC, XT, or AT), IBM PS/2 computers, or other IBM-compatible personal computers configured as described in Chapter 1 of the *PC-Interface User's Guide.*
- If installing for a LAN, a network interface board, as described in Chapter 1 of the *PC-Interface User's Guide.*
- The PC-Interface personal computer software, contained on the PC-Interface distribution diskette.
- PC-DOS or MS-DOS, Version 3.10 or later.

Installation of Adapters in Host

For PC-Interface service, install the LAN controller card in your host according to the card manufacturer's instructions. Asynchronous communications boards for RS-232 service should be installed according to the instructions in your host system *Guide to Operations* or in the product manual for the adapter.

Installing PC-Interface

The PC-Interface host-side distribution tape or diskette contains all the files required for your host to support PC-Interface. It also contains an installation program that simplifies the installation process. Before reading further in this manual, you should read and follow the instructions in the *PC-Interface Installation Guide.* Once PC-Interface is installed on your host, it is automatically initialized whenever you boot the host system.

This guide assumes that you have installed PC-Interface exactly as described in the *PC-Interface Installation Guide.*

PC-Interface System Configuration

The maximum number of personal computer users that a PC-Interface network can support depends on several factors. The number of PC-Interface RS-232 users is limited by the RS-232 hardware installed in your host. (The RS-232 hardware required by PC-Interface is the same as RS-232 hardware required for standard asynchronous terminals.) The number of simultaneous PC-Interface terminal emulation sessions allowed is limited and varies from installation to installation. See the *Release Notes* for information on the number of sessions available at your installation.

If you use a LAN, performance is likely to be the primary consideration bearing on system configuration. We cannot recommend a specific maximum number of personal computers for your PC-Interface network because of the variables that affect performance. Here are some issues you should consider when designing your network:

- Host systems with greater processing capacity can support more PC-Interface users.
- Typically, PC-Interface file service users place a lower load on the host system than standard terminal users performing similar operations.
- Typically, PC-Interface terminal emulation users place a somewhat heavier load on the host system than standard terminal users performing similar operations.
- Performance over a LAN is better than performance over RS-232.
- As on any machine that runs the UNIX operating system, system response depends on the number and type of applications being used.

Refer to "Tailoring Your PC-Interface System" in Chapter 4 for more information on performance issues.

2 About PC-Interface

Introduction

This chapter discusses the operation of PC-Interface and describes the two major PC-Interface services as well as the PC-Interface utilities that enhance the interface between the DOS and UNIX operating system environments. The discussion here is introductory. You should also be thoroughly familiar with the description of PC-Interface in the *PC-Interface User's Guide* to effectively administer the PC-Interface system. Some comments on the DOS environment are included at the end of this chapter for administrators who are not familiar with DOS.

Host/Personal Computer

(UNIX/DOS) Integration when the network hardware and software are installed, users can easily access both host (UNIX) and personal computer (DOS) resources. PC-Interface supplies two major services that integrate network hosts and personal computers, as well as a number of special-purpose utilities that enhance the UNIX/DOS interface. These services and utilities are described briefly below.

Host File Service

When you log into PC-Interface from your personal computer, you have access to the same UNIX system files as you do when you access the UNIX system host in any other way. These files are available to DOS on a virtual drive that is specified by PC-Interface in the LOGIN connection message. From the user's point of view, the virtual drive has the characteristics of a local DOS drive attached to the user's personal computer. With PC-Interface the user can:

- Create or access files on the virtual drive using DOS commands or applications by specifying the virtual drive identifier as part of the file name.
- Change the default drive to the virtual drive.
- Copy files from diskette or local fixed disk to the virtual drive following the same procedures used to copy to a local disk.
- Execute noninteractive UNIX system processes from DOS.
- Print DOS files on a remote printer.
- Create directories on the virtual drive using the DOS MKDIR command.
- Set DOS search paths that specify directories on the virtual drive.

Terminal Emulation Service

The PC-Interface software enables a personal computer to emulate a standard VT220 terminal. Personal computer users can therefore connect to the UNIX system host and conduct standard interactive UNIX operating system sessions as though they were using terminals rather than personal computers running DOS. The advantages of using terminal emulation are:

- users more familiar with the UNIX operating system environment than with DOS can work in the environment with which they are comfortable
- all users have access to UNIX system commands that are not directly accessible from the DOS environment.

Users can freely intermix file service and terminal emulation sessions. PC-Interface allows you to suspend your work with either service at any time and switch to the other service. Unless you explicitly terminate your current service, you will find any ongoing jobs in the state you left them when you return to that service.

PC-Interface Utilities

PC-Interface provides several utility programs that enhance the user interface between the DOS and UNIX operating system environments, as listed below. All of them are supplied on the PC-Interface personal computer distribution diskette and can be invoked from the DOS environment of the personal computer. UNIX2DOS and DOS2UNIX are also supplied in versions that run in UNIX operating system environment.

- DOS2UNIX and UNIX2DOS convert text files from DOS text format to UNIX operating system text format and vice versa.
- ON executes UNIX system commands on the host computer.
- JOBS displays the job table of ON-initiated UNIX system tasks, clears it of completed tasks, and reattaches the DOS console to detached tasks.
- KILL terminates ON-initiated jobs.

- PRINTER directs DOS printing to a local (personal computer) printer or a remote (UNIX system) printer.
- UDIR lists files and directories on the virtual drive in UNIX operating system format.

For information on PC-Interface utilities of special interest to the system administrator, refer to Chapter 4.

The DOS Environment

As system administrator, you should be familiar with the UNIX operating system environment—including commonly used commands, files, messages, and procedures. Some familiarity with the DOS environment is also helpful. Following are brief definitions of some common DOS terms used in this guide. For more information on the DOS environment, refer to the related publications listed in the preface.

Application software: Programs designed for specific purposes, such as accounting or word processing, and generally commercially distributed.

Disk: Loosely, a magnetic disk. In this guide, the term *disk* specifically refers to a fixed disk, as opposed to a diskette. (See also "Fixed disk")

Diskette: A thin, flexible magnetic disk permanently enclosed in a semirigid protective jacket. Synonymous with flexible disk and floppy disk.

Distribution diskette: One of the diskettes packaged with the *PC-Interface User's Guide* containing the PC-Interface personal computer software. (PC-Interface *host* software is distributed on either a distribution diskette or a distribution tape—depending on the particular host system—and is packaged with the *PC-Interface Administrator's Guide.*) (See also "Working diskette")

DOS drive: The personal computer hardware associated with DOS diskettes or fixed disks. DOS drives are named with letters followed by a colon. By convention, diskette drives are usually designated A: and B:, and a local fixed disk is designated C:. The contents of the disk or diskette on each drive are presented to the DOS user as separate file systems (each having its own root), unlike the UNIX operating system environment, which presents the contents of mounted drives as part of a single, integrated file system. (See also "DOS volume" and "Virtual drive")

DOS executable files: DOS executable files are invoked by name, like UNIX system executable files. DOS, however, requires executable files to end with one of the three-letter extensions .BAT, .COM, or .EXE. (For example, executable files familiar to DOS users include AUTOEXEC.BAT and COMMAND.COM.)

.BAT files are equivalent to UNIX system shell files—they are text files containing a list of DOS commands that are executed when the file is invoked. AUTOEXEC.BAT is a special file contained in the root directory of the fixed disk or PC-Interface working diskette that is executed automatically whenever the personal computer is booted. .COM and .EXE files are machine-readable

DOS programs that DOS recognizes and distinguishes on the basis of the file name extension (.COM or .EXE) and internal format.

DOS search path: The directories searched by DOS for commands and files. The DOS search path is analogous to the UNIX system search path, except:

- Under DOS, the current directory is *always* searched first.
- DOS search paths can include different DOS drives, each containing an independent file system.

DOS volume: The contents of a DOS disk or diskette. DOS allows you to identify the contents of a disk or diskette with a volume label (see the DOS FORMAT and VOL commands). *Volume* is sometimes used interchangeably with *drive*, but *drive* more precisely refers to the personal computer hardware associated with a disk or diskette. (See also "DOS drive")

Emulate: To simulate one system with another, so that the simulating computer system accepts the same data, executes the same programs, and achieves the same results. The PC-Interface system includes terminal emulation software that allows a personal computer to emulate a VT220 terminal.

Fixed disk: A rigid magnetic disk housed in either the system unit or an expansion unit of a personal computer, used for mass storage.

Function keys: Special keys on the personal computer keyboard that perform tasks normally requiring more than one keystroke or tasks that cannot be performed with keys representing standard characters. The function keys are labeled F1 through F10 for standard keyboards and F1 through F12 on enhanced keyboards.

Virtual drive: Any one of the drives used to access a UNIX system host through host file services. The PC-Interface LOGIN program identifies the drive (by drive letter) when the user logs in to PC-Interface. These drives are called *virtual* because they don't use the hardware of a local DOS drive but can be treated as though they did. (See also "DOS drive")

Working diskette: The diskette from which DOS is booted and from which PC-Interface software is loaded. (If these programs are on the fixed disk, no working diskette is needed.) The PC-Interface user creates the working diskette by combining the DOS system files with the contents of the PC-Interface distribution diskette. (See also "Distribution diskette")

3 PC-Interface Networks

Introduction

This chapter describes the functional characteristics of the LAN and RS-232 hardware and software as they relate to the PC-Interface system. For further general information on network administration, consult appropriate LAN and host system hardware documentation. Network administration required specifically for PC-Interface is covered in Chapter 4.

LAN Characteristics

A LAN (such as Ethernet) is a high-bandwidth, low-delay communication medium and link-level data transmission protocol that allows a computer to send packets of data to other computers. The maximum bandwidth of a LAN is typically 10 megabits per second. The bandwidth is shared by all computers using the network. Depending on which drivers are configured into your system, the LAN may use either the XEROX® protocols or the IEEE 802 protocols.

The PC-Interface software uses a LAN, when selected, for all data traffic between the personal computer and the host system.

For use with terminal emulation, the PC-Interface software creates special virtual serial lines over the LAN between the personal computers and the host. These virtual serial lines operate independently of and concurrently with the PC-Interface file-sharing mechanism.

UNIX Host LAN Hardware and Software

The UNIX system host LAN controller gives the UNIX system host the ability to communicate over a LAN. The controller is a board that plugs into the UNIX system host backplane and is accessible under kernel control using the UNIX system host LAN driver.

The UNIX system host LAN driver provides the interface between the LAN hardware and the various processes that use the LAN. It can handle a fixed number of *network ports,* or logical connections. Each network port has a set of *input buffers* associated with it, where incoming packets are temporarily placed until a process asks for them. Each network port can be independently opened and have reads and writes performed on it. The driver is responsible for demultiplexing incoming LAN packets and placing the packets into input buffers reserved for the various network ports based on a data field in the packet or on the network address.

The network can support multiple high-level protocols simultaneously.

Personal Computer LAN Hardware and Software

PC-Interface personal computers can use any of a wide range of network interface boards. A complete list of the currently supported boards appears in Appendix A.

The board plugs into a personal computer expansion slot according to instructions contained in the manufacturer's documentation. It is accessible from the PC-Interface software via the PC-Interface LAN driver software.

The PC-Interface LAN driver is the interface between the LAN hardware and the rest of the PC-Interface software. The driver is interrupt-driven and tuned for maximum network performance. It performs a demultiplexing function for the PC-Interface software, allowing flexibility of operation and greater system performance.

RS-232 Characteristics

RS-232 is a low-bandwidth, point-to-point communication medium and data transmission protocol that supports data transfer between two computer systems. Typical RS-232 bandwidths are under 10 kilobits per second.

PC-Interface uses the host system's terminal ports in the same manner as a normal terminal would. No physical RS-232 lines need be reserved for PC-Interface service. While file service is being provided, PC-Interface uses a simple frame-based communications protocol with moderate-sized packets. For the sake of performance, whenever terminal emulator traffic is encountered, the protocol is switched to a much lower overhead method, so that single-character transmissions do not require sending a large number of bytes. Users can arbitrarily intermix terminal emulation and file service messages.

Multiple Hosts and Networks

The PC-Interface host software can support multiple simultaneous personal computer users. Each concurrent personal computer user is handled independently of all others. The only limits are those imposed by the operating system and by PC-Interface license agreement. The operating system may limit the number of processes in the system and the number of simultaneous network connections allowed.

The PC-Interface host software can also simultaneously support both LAN and RS-232 connections with personal computers. Because the host software is designed to handle LANs and RS-232 independently, your PC-Interface system can use a LAN, RS-232 only, or both a LAN and RS-232 at once.

PC-Interface networks using a LAN can have multiple UNIX system hosts as well as multiple personal computers. PC-Interface personal computer users can access any host on the network for which they have a UNIX system account.

The PC-Interface personal computer user can connect to more than one host at a time. If the personal computer is connected to a network that includes more than one host, the user can change from one host to another simply by changing drives.

The PC-Interface personal computer software supports both RS-232 and LAN connections. If the personal computer has the necessary hardware for both media, the user can choose either medium for a PC-Interface session.

TCP/IP Networking

PC-Interface uses the standard ARPA Internet Protocol (IP) and User Datagram Protocol (UDP) for communication between network hosts and personal computers. IP routes packets between machines whereas UDP provides simple process-to-process datagram service based on IP.

Internet Addresses

All hosts and personal computers on the network must have unique internet addresses at which they can receive data. An internet address is a 32-bit (4-byte) numeric value that specifies a particular network and a particular machine on that network.

Internet addresses are specified in a standard format of four fields separated by periods—for example, 72.235.82.101 (decimal values). Each of the four fields represents one byte of the complete internet address.

NOTE Internet addresses can also be specified in octal or hexadecimal notation, using the standard convention of leading zero to imply octal and a leading 0x or 0X to imply hexadecimal.

To allow flexibility in interconnecting multiple networks and multiple machines within a network, internet addresses can use one, two, or three of the four fields to specify the network. The portion of the internet address not used to specify the network is available to specify a particular host within that network.

This flexibility is achieved by dividing internet addresses into *classes,* as follows:

Class	Fields to Identify Network	Fields to Identify Host
A	1	3
B	2	2
C	3	1

Class A addresses allow a large number of network hosts to be identified with unique host addresses, since three of the four internet address fields are devoted to the host address. Class C addresses, on the other hand, provide for a larger number of networks. Class B addresses compromise between the two extremes.

The address class is coded in the first byte (or first field) of the internet address. Class A addresses use 0 in the high-order bit, Class B addresses use 10 in the high-order two bits, and Class C addresses use 110 in the high-order three bits. The following table shows the equivalent decimal values and an example address in each class.

<table>
<tr><th>Address Class</th><th>Value of First Byte (Decimal)</th><th colspan="4">Example Internet Address (Decimal)</th></tr>
<tr><td></td><td></td><td>Network Portion</td><td colspan="3">Host Portion</td></tr>
<tr><td>A</td><td>0-127</td><td>72</td><td>. 235</td><td>. 82</td><td>. 101</td></tr>
<tr><td></td><td></td><td colspan="2">Network Portion</td><td colspan="2">Host Portion</td></tr>
<tr><td>B</td><td>128-191</td><td>184</td><td>. 132</td><td>. 119</td><td>. 23</td></tr>
<tr><td></td><td></td><td colspan="3">Network Portion</td><td>Host Portion</td></tr>
<tr><td>C</td><td>192-223</td><td>205</td><td>. 198</td><td>. 198</td><td>. 2</td></tr>
</table>

PC-Interface machines running on a Class A network might have the following internet addresses:

Host1	89.0.2.19
Host2	89.0.2.4
PC1	89.0.2.77
PC2	89.0.1.1

The network machines use *internet broadcast network addresses* for messages to be received by all machines on the network. This address is derived from the internet address in a simple way: it uses the same bits as the internet address to identify the network, but replaces the host bits with zeros or ones (depending on the implementation). For example, the internet broadcast network address for all machines shown in the previous example is 89.0.0.0 or 89.255.255.255.

Internet Address–Physical Address Mapping

Network addresses are unique addresses that are physically encoded in each network adapter board in both the host and the personal computer. The network address is assigned by the hardware manufacturer and cannot be changed after the board is manufactured. The TCP/IP protocols use the network address to identify each machine on the network. When a personal computer communicates with hosts, the network address of the adapter board in the personal computer is transmitted to the hosts along with the internet address. The network hosts map the transmitted internet address to the network address and retain this information in a table. When any host needs to transmit data to a particular personal computer, the host refers to the table mapping internet addresses to network addresses. The host identifies the personal computer to which it addresses data based on the correlation of the internet and network addresses.

4 PC-Interface Administration

Introduction

This chapter discusses managing your PC-Interface system. The topics are organized in three sections, as follows:

- Getting Started with PC-Interface: starting and stopping host servers, assigning internet addresses to personal computers, and adding and removing PC-Interface users.
- Supporting PC-Interface Operation: `/etc/termcap` or `/usr/lib/terminfo` entries for terminal emulation, administering network printers, managing PC-Interface UNIX system processes, backing up system data, copy protection, file sharing and record locking, and error messages.
- Tailoring Your PC-Interface System: managing UNIX system disk resources and load, establishing shared personal computer software libraries, changing default file permission modes, enabling and disabling terminal emulation ports, and DOS software compatibility issues.

To support everyday operation of the PC-Interface system, you must be familiar with the topics discussed in the first two sections. The final section, "Tailoring Your PC-Interface System," tells you how to customize and configure your host to suit the needs of your installation.

Starting and Stopping Host Servers (Daemons)

Once the host software is installed according to the instructions in the *PC-Interface Installation Guide,* PC-Interface is operational each time you boot the host system. Occasionally, you may need to stop PC-Interface operation—for example, to change the default `umask` by modifying `pcistart`. You can stop PC-Interface without shutting down the entire system. To resume PC-Interface service, you can use a special utility for restarting the PC-Interface daemons or you can reboot the host system.

The utilities `/usr/pci/bin/pcistart` and `/usr/pci/bin/pcistop` start and stop the PC-Interface LAN daemons `/usr/pci/pcimapsvr.eth` and `/usr/pci/pciconsvr.eth`. (For a more complete explanation of these utilities, refer to Chapter 5.) Whenever you intend to stop PC-Interface service, you should inform all PC-Interface users who are currently logged in that service is about to be terminated and then invoke `/usr/pci/bin/pcistop`. The `pcistop` utility does not stop PC-Interface RS-232 processes. To stop them, PC-Interface RS-232 users should log out. If necessary, you can kill PC-Interface RS-232 processes manually by running the UNIX system `ps -ef` command to find out the process ID (pid) of any currently executing PC-Interface processes. Look for any processes in the `ps` list that start with `pci` and kill the relevant pid. To resume PC-Interface service, either invoke `/usr/pci/bin/pcistart` or, if the system was shut down, reboot.

Assigning Internet Addresses to Personal Computers

NOTE

If you are running PC-Interface over RS-232 only, skip this section.

PC-Interface networks with hosts running the UNIX system with TCP/IP networking services use the ARPA UDP/IP protocols for communication between network hosts and personal computers. For these protocols to work properly, all personal computers and hosts on a PC-Interface network must have internet addresses that are both compatible and unique. Assigning internet addresses to UNIX system hosts is discussed in the *PC-Interface Installation Guide.* Personal computer requirements are discussed here.

In general, the term *host* is used in this guide to refer to a host computer running a UNIX System V operating system with TCP/IP networking services and supporting a network of PC-Interface users. The UDP/IP protocols, however, treat *both* UNIX systems and PC-Interface personal computers as network hosts. PC-Interface personal computers must therefore have internet addresses that meet the same requirements that apply to any UNIX system hosts on the network.

PC-Interface personal computer users must know their assigned internet addresses when they create their PC-Interface working diskettes or install PC-Interface on their fixed disks. (See Appendix A of the *PC-Interface User's Guide.*) The unique internet address associated with each personal computer is stored in a file called \ETC\HOSTS, which is created automatically by the PC-Interface INSTALL program.

NOTE If the personal computer already has a HOSTS file and the HOSTS variable in the AUTOEXEC.BAT file points to that file, no new HOSTS file is created. Instead, the existing HOSTS file is used.

The HOSTS file has the following format:

internet_address host_name `localhost`

Whenever the user establishes a PC-Interface session, the PCIINIT program searches the HOSTS file for the internet address associated with the host labeled "localhost" and assigns that address to the personal computer.

Assigning internet addresses to personal computers is normally a three-step process:

1. Reserve a block of internet addresses for future assignment to personal computers.
2. Assign a unique address from the reserved block of addresses to each network personal computer as network users are added.
3. Inform users of their internet addresses and assist them as necessary in installing PC-Interface.

The procedure for reserving a block of internet addresses is described in the following paragraphs. For further information on assigning internet addresses to personal computers, see the next section, "Adding PC-Interface Users." Internet address requirements for personal computers are the same as for hosts. Both hosts and personal computers must have:

- Internet addresses in the same class and network.
- Unique host portions of the internet address.

To meet these requirements, follow these steps:

1. Determine the internet address of any UNIX system host on the PC-Interface network. To do this, type:

   ```
   grep hostname /etc/hosts
   ```

   where *hostname* is the name of a network host. (If you do not know the name of a network host, use the uname -n command to determine it.) The system displays the internet address together with the name of the host and (if applicable) any aliases that refer to the host. For example, the system might print:

   ```
   89.0.2.1    hostl    happy
   ```

   The internet address has a standard format of four numeric fields separated by periods, as shown in this example.

   Make a note of this internet address.

2. Determine the internet address *class* by examining the first field of the internet address and comparing it to the table below:

First Field of Internet Address	Class
0-127	A
128-191	B
192-223	C

NOTE Values shown are decimal. Internet addresses can also use octal or hexadecimal notation using the standard convention of a leading zero to imply octal and a leading 0x or 0X to imply hexadecimal.

Our example hostl address of 89.0.2.1 is a Class A address, since the value of the first field is between 0 and 127.

3. Determine whether all hosts are in the same class and network (that is, whether they all have the same network portion of the internet address). Depending on the class of the address (which you determined in step 2), the network portion of the address is specified in the first field, the first two fields, or the first three fields of the internet address, as follows:

Network Class	Fields Devoted to Network Portion of Address
A	1
B	2
C	3

Refer to "Assigning Internet Addresses in a Multihost PC-Interface Network" in the *PC-Interface Installation Guide* for further information on determining the network portion of the internet address.

Since it is a Class A address, the network portion of our example internet address of 89.0.2.1 is 89. The remainder
of the address (0.2.1 in our example) is the host portion of the internet address.

Make a note of the network portion of the internet address used by your network. It will be part of the internet address that you assign to all personal computers using the PC-Interface network.

4. Determine which internet *host* addresses are available for assignment to personal computers. Since internet host addresses must be unique, you must find out which internet host addresses have already been assigned. The file `/etc/hosts` on any PC-Interface network UNIX system host lists all internet addresses currently assigned to any network machines. Inspect this file, looking for all internet addresses with the same network portion.

For example, you might find that the following internet addresses with a network portion of 89 have already been assigned:

89.255.1.1
89.255.1.2
89.255.1.3
89.255.1.4
89.255.1.6
89.255.1.8

In this example, you could reserve for assignment to personal computers any internet addresses that have 89 in the first field and *do not* have values of 255.1.1, 255.1.2, 255.1.3, 255.1.4, 255.1.6, or 255.1.8 for the host portion of the address.

Note that the number of internet address fields devoted to the host portion of the address (and therefore the number of addresses that can be reserved for personal computers or UNIX system hosts) depends on the class of the network. Class A networks (as in the preceding example) can have as many hosts as can be uniquely addressed in three host address fields. Class B networks limit the number of hosts to those that can be uniquely addressed in two host address fields, and Class C networks limit the number of hosts to the number of unique addresses in a single host address field.

In addition to the requirement for a unique host address and the limitations imposed by network class, the following requirements apply to the host portion of internet addresses:

- The value of each field must be less than 256 (decimal).
- You cannot assign zero or 255 (decimal) to *all* host address fields in an internet address. This means that addresses such as 89.0.0.0 and 89.255.255.255 cannot be used. Equivalent octal or hexadecimal values are also prohibited.

(See the *PC-Interface Installation Guide* for a more detailed description of these limitations.)

Now consider the previous example in which the six internet addresses 89.255.1.1, 89.255.1.2, 89.255.1.3, 89.255.1.4, 89.255.1.6, and 89.255.1.8 are already assigned to existing network machines. Based on all the requirements that apply to internet addresses, the following addresses are available for assignment to personal computers:

- All addresses between 89.0.0.1 and 89.255.1.0

- 89.255.1.5 and 89.255.1.7
- All addresses between 89.255.1.9 and 89.255.255.254

Make a note of the internet addresses that are available for assignment to personal computers on your PC-Interface network.

5. Reserve a block of internet addresses from the list of available addresses from which you will assign addresses as personal computers are added to the network. This block can be chosen arbitrarily, but you should choose a range that won't conflict with foreseeable UNIX system host additions to the network or other existing or planned network interfaces.

   In our example, you might reserve the internet addresses between 89.255.255.0 and 89.255.255.254 for assignment to personal computers. This block is clearly distinct from the addresses already assigned to UNIX system hosts and, with 255 legal values, allows ample room for network growth.

Adding PC-Interface Users

To add new users to your PC-Interface network, follow these steps:

1. Create a UNIX system account for the user in whatever is the normal way for your system, usually either via `sysadm` or by making an entry in `/etc/passwd` and creating a home directory for the user.
2. Attach the user's personal computer to the host via a LAN or RS-232.
3. If using a LAN, assign an internet address chosen from the block of addresses reserved for personal computers to the user's personal computer. Enter this address in the file `/etc/hosts` and inform the user that he or she should use the assigned address along with his or her personal computer's host name when installing PC-Interface. Each computer in the network must have a unique internet address.

   For example, assume you have reserved the internet addresses from 89.255.255.0 through 89.255.255.254 for network personal computers. The first personal computer to be added to the network would be assigned the internet address 89.255.255.0, the second would be assigned

89.255.255.1, and so on. When you add these addresses to /etc/hosts, use the standard format:

internet address *official host name* *aliases*

where *internet address* is the address in the standard four-field format with dots separating the fields; *official host name* is the name of the personal computer being added to the network; and *aliases* (an optional field) is a list of any names other than the official personal computer name that the machine is known by. The "official host names" of your personal computers on the network are arbitrary, but you may wish to establish a meaningful naming convention.

For example, assume the personal computer "pc5" with the internet address 89.255.255.4 is being added to the network. Add the following entry to /etc/hosts:

```
89.255.255.4      pc5
```

Any number of spaces or tabs can separate the parts of the /etc/hosts entry, and the entry can appear anywhere in the /etc/hosts file.

You should be prepared to assist users as necessary when they install PC-Interface. Refer to Appendix A of the *PC-Interface User's Guide* for complete instructions on installing PC-Interface on personal computers.

NOTE

Because PC-Interface UNIX system hosts associate an internet address with a specific personal computer, it is important for users not to use the same PC-Interface working diskette to boot PC-Interface sessions on two or more personal computers. If two personal computers are booted from the same PC-Interface working diskette, they both have the same internet address, which violates the requirement that all network machines have unique addresses. To avoid such problems, it is safest to associate an internet address (and therefore a working diskette) with a particular personal computer on the network. If it is necessary to use a working diskette on more than one personal computer, the user should modify the HOSTS file to specify a different internet address before a different machine is booted.

4. When you add users, you may need to enable additional terminal emulation ports. Refer to "Enabling and Disabling Terminal Emulation Ports" later in this chapter.

Removing PC-Interface Users

You can remove a PC-Interface user from the network in any of the following ways:

- Delete the user's UNIX system account following the usual UNIX system procedure.
- Disconnect the user's personal computer from the network.
- Collect the user's PC-Interface working diskette (and also the PC-Interface distribution diskette).

Once the user's account has been removed, you should perform the same functions you would when removing any UNIX system account. This would typically include deleting, archiving, or transferring files.

When you remove users, you may also want to disable terminal emulation ports. Refer to "Enabling and Disabling Terminal Emulation Ports" later in this chapter.

Supporting PC-Interface Operation

Entries in /etc/termcap or /usr/lib/terminfo for Terminal Emulation

The PC-Interface terminal emulator requires no special entry in /etc/termcap or /usr/lib/terminfo. Since the emulator simulates a standard VT220 terminal, use the same entry you use for a VT220 terminal. The UNIX system system is preconfigured with the correct entry.

Administering Network Printers

With the PC-Interface PRINTER command, users can specify explicit print programs for up to three print streams—LPT1, LPT2, and LPT3. When the user sets PRINTER with no print program in effect for the specified print stream, PC-Interface automatically spools print jobs from the personal computer to the UNIX system print stream LPT1. The default print program for LPT1 is defined in /usr/pci/bin/pcistart, which is installed with the PC-Interface host software.

As system administrator, you need to be sure that the default settings in the pcistart file are appropriate for your system. In addition, you may want to change the default system printer to suit the needs of your user community.

Changing the default system printer involves editing the pcistart file and then stopping and restarting PC-Interface to make the change effective. Be sure to warn PC-Interface users before you run pcistop and pcistart. Editing pcistart is described below:

The pcistart file defines two variables, PRINTPATH and PRINTPROG, as follows:

```
PRINTPATH=/bin:/usr/bin

PRINTPROG=lp
```

PRINTPROG is the default print program and PRINTPATH is a list of directories the system should search for print programs. The print program specified for PRINTPROG, as well as the print programs invoked by the user with the PRINTER command, must reside in one of these directories.

Change these variables to specify the desired default print program and search paths.

When you have edited `pcistart` to specify the appropriate default print program and search paths, stop and restart PC-Interface.

For further information on administering UNIX system printers, refer to your host system documentation.

Managing PC-Interface UNIX Processes

All PC-Interface users who establish terminal emulation or file service sessions cause UNIX system processes to be executed on the host. When UNIX system commands are invoked during a terminal emulation session, the host executes these commands just as though they were issued from a terminal connected directly to the host. Managing these processes is no different from managing any other processes invoked from a UNIX system terminal.

When file service users execute DOS commands on the virtual drive, the PC-Interface software translates the commands into equivalent UNIX system commands and executes them on the UNIX system. In general, the PC-Interface software manages these UNIX system processes transparently to the user, and they rarely require intervention by the system administrator.

With the PC-Interface ON command, however, PC-Interface file service users can invoke UNIX system processes that are not under the control of the PC-Interface software. A user who is unfamiliar with the UNIX operating system might use ON inappropriately and start a runaway process—a defective process that consumes UNIX system resources and never terminates. If the user cannot terminate the process himself, the administrator should intervene and kill the process.

When necessary, use the UNIX system ps command to find out information about PC-Interface UNIX system processes, just as you would to find out about any other UNIX system processes. Note that processes invoked via ON are not associated with a terminal, so you must invoke ps with the appropriate option to display status information for *all* processes.

For example, assume that a user with the user name fred has invoked a process /u/fred/bin/doit that needs to be killed. To find out the process number that needs to be killed, issue the ps -ef command. Among the displayed data, you might see lines similar to this:

```
UID   PID  PPID  C   STIME TTY  TIME COMMAND
fred  346   345  1   Apr 7   ?  9:48 /u/fred/bin/doit
```

From this display you can see that the process you need to kill is number 346.

System Backup Procedures

Standard system backup procedures are not affected by the presence of the PC-Interface software and PC-Interface user files on the UNIX system host. PC-Interface users can, of course, use their personal computers to back up files from the UNIX system host to their own local disks. However, such procedures should not be used as a substitute for regular backups of the entire UNIX system file system done by the system administrator.

Copy Protection

You may have occasion to assist PC-Interface users who have violated the restriction against simultaneous use of multiple PC-Interface disks with the same serial number. The following paragraphs describe the PC-Interface copy protection system and recommend corrective procedures to be used when a user's personal computer is disabled by the copy protection mechanism.

Using PC-Interface simultaneously from two personal computers requires two PC-Interface disks created from two different licensed PC-Interface distribution diskettes. Users may make backup copies of their distribution diskettes or working diskettes, but they may not use more than one copy at a time. **It is the system administrator's responsibility to ensure that PC-Interface is used correctly.**

Copy protection prevents the illegal duplication and use of PC-Interface software. It is illegal to simultaneously operate two or more personal computers with PC-Interface software derived from one licensed distribution diskette. Each diskette is distributed with some unique data in the BRIDGE.DRV file that must be present in the file for correct operation.

When the PC-Interface system detects two personal computers simultaneously using PC-Interface disks created from the same distribution diskette, the second personal computer to log in to PC-Interface is halted with a displayed message stating that a copy protection violation has occurred. This condition can be recovered from by powering off the personal computer and rebooting.

File Sharing and Record Locking

File sharing means that while one user has a file open no other user can access it; the file is "locked." By contrast, record locking means that rather than a whole file being locked when one user accesses it, only the record (or, depending on the application, the group of records) being used is locked.

For example, a networked word-processing program prevents two users from modifying the same file at the same time (file sharing), whereas a networked database management program lets two or more users access the database file at once but prevents simultaneous record access (record locking).

How and whether a particular application implements file sharing or record locking is application dependent. For instance, one application may not lock records at all, another may lock only the record that is being accessed, and a third application may lock all records from the one in use to the bottom of the file. Some standalone applications lock files; others do not. If an application is designed to open a file, make its own copy of that file to work on, and close the file, then the file is not locked. However, if the application keeps the file open, then it usually locks it to other users. Attempts to access a locked file yield either an "Access denied" or "File locked" error message.

A file set to read-only mode with the DOS ATTRIB or CHMOD commands is always open to reading, although nobody can write to it. For this reason, we recommend that program files (used to execute commands) be set to read-only to avoid conflicts in their use.

File-sharing and record-locking capabilities are built into PC-Interface and become available automatically when PC-Interface is installed and booted. File sharing and record locking use UNIX system shared memory and semaphores for their operation.

PC-Interface supports DOS 3.10 file sharing and record locking used by many networked DOS applications. If a choice of network types is offered when installing networked applications, we recommend that you first choose "IBM PC Network" or another MS-NET compatible network type. However, you may have to experiment with other network types to find the one that best supports file sharing and record locking for a particular application.

If you have problems with file sharing or record locking, refer to Chapter 6, Some Problem-Solving Tools For PC-Interface.

PC-Interface Error Messages

The PC-Interface host software responds to all serious and fatal PC-Interface error conditions by displaying an error message on the host system console and mailing the message to root. Refer to Appendix A of this guide for a list of these messages and recommended recovery procedures.

Tailoring Your PC-Interface System

As soon as you install your PC-Interface system, users can begin to enjoy the UNIX system host resources made available by PC-Interface file service, terminal emulation, and the special PC-Interface utilities. No special configuration of the PC-Interface system is required. As you become familiar with PC-Interface operation, however, you may want to tailor the system to accommodate the way you and the system users prefer to use the system. This section offers tips on the following topics:

- Managing Host System Disk Resources and Load: Issues concerning the host resources required for PC-Interface use.
- Establishing Shared Personal Computer Software Libraries: How to store DOS utilities and application programs on the UNIX system host.
- UNIX Operating System Permission Modes: How to alter the default read, write, and execute permissions assigned to files created with PC-Interface.
- Enabling and Disabling Terminal Emulation Ports: How to increase or decrease the number of terminal emulation sessions supported by your host.
- DOS Software Compatibility Issues: Technical requirements for DOS software to operate concurrently with PC-Interface.

Managing Host System Disk Resources and Load

Although the PC-Interface system requires no special procedures for managing host system disk space, you should advise inexperienced users that they are sharing a disk with other people and that they should conserve disk space. If your PC-Interface system uses multiple hosts, you can use them to your advantage by distributing user accounts among the different hosts so that resource consumption (system load and disk space use) on the different machines is about equal.

The task of load management on PC-Interface hosts is essentially the same as for any UNIX system machine. Every PC-Interface user puts some load on the host, just as each user who is logged directly into the UNIX system does. Terminal emulation puts more of a load on the host than file service does, so you may want to limit the number of simultaneous terminal emulation sessions allowed. (See "Enabling and Disabling Terminal Emulation Ports," later in this chapter.)

PC-Interface allows users the flexibility of alternating between local work (done on a local diskette or fixed disk) and remote work (done on the UNIX system virtual drive). The load on the remote UNIX system machine is heavier, of course, when PC-Interface users are using the virtual drive. Depending on the requirements of your installation, you may either encourage or discourage the use of the virtual drive for DOS programs, which can be run either from the user's local disk or from the virtual drive (assuming the programs are installed on the virtual drive). For example, to discourage use of the virtual drive for DOS programs, you can limit the number of publicly accessible directories containing DOS programs. (Individual users can still copy DOS programs to their own directories on the UNIX system disk, however.)

Establishing Shared Personal Computer Software Libraries

The PC-Interface system administrator is responsible for creating and administering libraries of DOS software used by the network-wide community of PC-Interface users. The software libraries managed by the system administrator are analogous to personal libraries created on the UNIX system disk by individual PC-Interface users to store their own copies of personal computer software. The difference is that the shared libraries created by the system administrator are accessible to all system users rather than to only a single user.

To create a publicly accessible library of DOS software on the UNIX system disk, follow these procedures:

1. Choose the DOS software you wish to store and execute on the UNIX system disk. PC-Interface utilities such as DOS2UNIX, EM, JOBS, ON, KILL, PRINTER, and UNIX2DOS, for example, can be installed on the UNIX disk. (These utilities are distributed with the PC-Interface personal computer software.) You can also install on the UNIX system disk any other internally developed, site-licensed, or public domain DOS software that can be stored on and executed from a fixed disk. Such DOS programs might be obtained off-the-shelf or created with any of the widely available utilities that run under DOS for writing, compiling, and linking DOS programs.

NOTE Some copy-protected DOS programs cannot be installed on the UNIX system disk. Such programs are generally not internally developed, site-licensed, or public domain, however, and therefore may be illegal to install in a publicly accessible UNIX system directory.

2. Log into a host UNIX system from a personal computer using PC-Interface.

3. Make a directory on the UNIX system disk to store the software you have selected. This directory must be easy to find and execute for all PC-Interface users. We recommend `/usr/pci/pcbin` as the standard directory for DOS executable programs.

4. Install the DOS programs in `/usr/pci/pcbin` on the UNIX system virtual drive just as you would install them in a subdirectory on a local fixed disk. During the installation procedure, refer to the virtual drive by the name PC-Interface returns to you when you log in (usually C: or D:).

5. Set the UNIX system permission modes for the DOS files and the directory containing them so that PC-Interface users can access the files. The DOS files must be readable but need not have UNIX execute permission. Directories must be readable and executable. Write permission is also often required for directories containing DOS applications, since many applications create temporary files when executed. You can use the PC-Interface ON command with the UNIX system chmod command to set the modes correctly. To make a file on virtual drive D: readable by all other users but writable only by you, type:

   ```
   on d: chmod 644 filename
   ```

   To make a directory on virtual drive D: readable, writable, and executable by you and all other users, type:

   ```
   on d: chmod 777 directory
   ```

6. Inform PC-Interface users of the location of the newly installed software and be prepared to assist them in setting their DOS search paths appropriately. For example, assume a PC-Interface user has a personal computer with one fixed disk and normally has a DOS search path set to C:\BIN. The PC-Interface virtual drive for this user is normally drive D:.

To make `/usr/pci/pcbin` easily accessible, the user's search path should be set with the command:

```
path=c:\bin;d:\usr\pci\pcbin
```

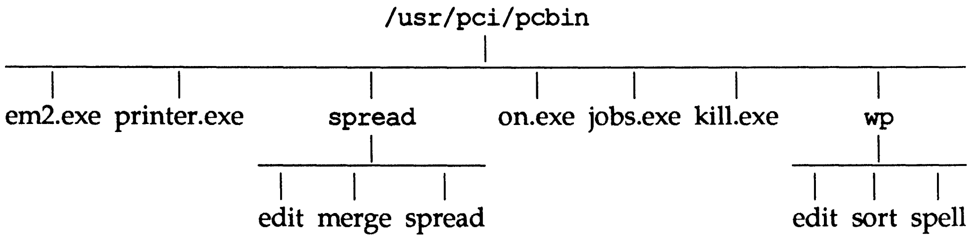
The PATH command can be added to the user's AUTOEXEC.BAT file to make it effective every time the user boots the personal computer. For further information on using the AUTOEXEC.BAT file with PC-Interface, refer to the *PC-Interface User's Guide.*

Tailoring Your Software Libraries

Sometimes it can be useful to create subdirectories within `/usr/pci/pcbin` so that related DOS files can be grouped together. Here are some factors to consider when evaluating the need for subdirectories of DOS software:

- A single directory containing all personal computer software is convenient because it is easily accessible to all PC-Interface users. The directory can be included in the search paths of all PC-Interface users, and there is little danger that users needing personal computer software will not be able to find it.
- However, a single directory containing all personal computer programs can become unmanageable if it contains a large number of personal computer programs. For example, a single personal computer application program can contain several files (sometimes including both executable and data files). If many such programs are in the same directory, it can be difficult to keep track of which files belong to which programs. Moreover, with a large number of files in the same directory, there is a risk of filename conflicts between existing files and new ones that you want to add.
- To reduce confusion, you can create separate directories for individual personal computer programs or groups of related programs. When you add such directories, however, bear in mind that users must be informed of the existence of these directories and add them to their search paths.

As an example, consider the following use of the /usr/pci/pcbin directory:



In this example, separate subdirectories are created for a spreadsheet program (spread) and a word-processing program (wp). The subdirectories allow the files for the two different applications to be grouped separately. Since both applications have files named edit, using two subdirectories also avoids a file name conflict. The standard PC-Interface utilities remain immediately subordinate to /usr/pci/pcbin. For this example, DOS search paths for PC-Interface users would be set to include /usr/pci/pcbin, /usr/pci/pcbin/wp, and /usr/pci/pcbin/spread.

Software Libraries for PC-Interface UNIX System Commands

The directory /usr/bin is recommended for storing utilities executable under the UNIX system that are of interest to PC-Interface users. The UNIX executable versions of dos2unix and unix2dos are installed in this directory during the PC-Interface host software installation. You can add any other UNIX programs you choose to this directory. In general, the same tailoring issues apply as for libraries of DOS software. We do not recommend creating subdirectories within /usr/bin, however. Following are considerations for UNIX system software libraries that do not apply to DOS libraries:

- UNIX system programs must be executable, unlike DOS programs which must only be readable.
- Search paths for UNIX system programs are relevant only for users who log in as UNIX system users—that is, users logging in from terminals and PC-Interface terminal emulation users. Do not set UNIX system search paths for PC-Interface file service users.

UNIX Operating System Permission Modes

PC-Interface supports the standard UNIX system file permission modes. Both file service and terminal emulation users have the same privileges and restrictions regarding access to host files as other UNIX system users.

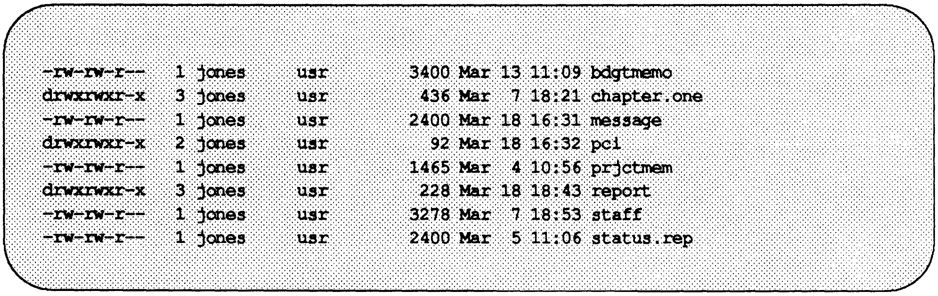
The user's `umask` determines the permission mode with which a UNIX system file is created. Users connected directly to the UNIX system use the default UNIX system `umask`, or they can change their own `umask` to reflect their preferred default permission modes. The PC-Interface `umask` for all RS-232 users (both file service and terminal emulation) and for LAN terminal emulation users is determined in the same way: they either use the default UNIX system `umask` or specify their own `umask`.

PC-Interface file service users connected to the host via LAN all have the same `umask`, which cannot be changed by individual users. The system administrator, however, can change the system-wide `umask` that affects these users.

The script `/usr/pci/bin/pcistart` sets the `umask` when PC-Interface is initialized. This `umask` is inherited by the connection server (`/usr/pci/pciconsvr`) and applies to all PC-Interface LAN file service users. The initial `umask` is 002, which means:

- Files are readable by all system users and writable by both the owner and others in the owner's group.
- Directories are readable and executable by all system users and writable by both the owner and others in the owner's group.

The following example displays the contents of a typical UNIX system directory with files and subdirectories created using PC-Interface file service over LAN. The list is produced with the `ls -l` command. Read, write, and execute permissions are shown in the first column.



```
-rw-rw-r--   1 jones    usr        3400 Mar 13 11:09 bdgtmemo
drwxrwxr-x   3 jones    usr         436 Mar  7 18:21 chapter.one
-rw-rw-r--   1 jones    usr        2400 Mar 18 16:31 message
drwxrwxr-x   2 jones    usr          92 Mar 18 16:32 pci
-rw-rw-r--   1 jones    usr        1465 Mar  4 10:56 prjctmem
drwxrwxr-x   3 jones    usr         228 Mar 18 18:43 report
-rw-rw-r--   1 jones    usr        3278 Mar  7 18:53 staff
-rw-rw-r--   1 jones    usr        2400 Mar  5 11:06 status.rep
```

To change the umask for PC-Interface users:

1. Modify the file /usr/pci/bin/pcistart to change the line that reads:

   umask 002

   to reflect your desired umask.
2. Stop the execution of PC-Interface.
3. Restart PC-Interface.

There are no special PC-Interface considerations associated with UNIX system groups.

For further information on permission modes and groups, refer to the descriptions of chmod, chown, umask, group, and setuid in your host system documentation and the discussion of changing permission modes in the *PC-Interface User's Guide.*

Enabling and Disabling Terminal Emulation Ports

PC-Interface terminal emulation can be used in these four ways:

1. Over a LAN concurrently with PC-Interface file services. (The user first logs into PC-Interface file services with the LOGIN command and then invokes EM.)
2. Over a LAN independently of PC-Interface file services. (The user invokes EM after booting the PC-Interface working diskette, without logging into PC-Interface file services.)

3. Over RS-232 concurrently with PC-Interface file services (as described in item 1 for the LAN case).
4. Over RS-232 independently of PC-Interface file services (as described in item 2 for the LAN case).

   Note that in this case, the UNIX system host treats the user's personal computer like a standard asynchronous terminal in all respects; the PC-Interface host software is not involved. Administration of such personal computers therefore requires no special procedures.

For the remaining three cases, PC-Interface uses host files that can be modified to change the maximum number of simultaneous terminal emulation sessions allowed. In the following description, references to emulation over LANs apply to emulation both concurrently with file services and independently of file services. References to emulation over RS-232 apply only to emulation concurrent with file services. For a more thorough description of the operation of terminal emulation, see Chapter 5, PC-Interface System Architecture. Here we concentrate on the files you need to know about and the operations that must be performed to reconfigure the available terminal emulation ports.

Files that Control Terminal Emulation Ports

The following factors determine the number of available terminal emulation ports:

1. `Entries in the file /usr/pci/pciptys`— This file is accessed by PC-Interface servers to determine which pseudo-ttys (ptys) are available for use in terminal emulation. The number of entries in the file determines the number of possible simultaneous terminal emulation sessions.

   The file lists the ptys used for terminal emulation in the form:

   control_half:*slave_half*: [*optional* getty *command*]

   An example of entries in a `pciptys` file appears below:

   ```
   pciptyp0:pcittyp0:
   pciptyp1:pcittyp1:
   pciptyp2:pcittyp2:/etc/getty pcittyp2 9600 none LDISC1
   ```

The third entry in the example above illustrates the optional `getty` command, which tells the PC-Interface server to start a `getty` on that pty. For some installations, this form of pty entry is used for LAN terminal emulation. Whether this form should or should not be used is discussed in the *Release Notes* accompanying this manual. This form is never used for RS-232 terminal emulation, however. For ptys used for RS-232 terminal emulation, gettys must be started in `/etc/inittab`, as described later in this section.

2. `Special files in the directory /dev` — The `/dev` directory contains pairs of special files with names corresponding to the entries in the `pciptys` file. There are separate files for the control half and the slave half of each pty. For example, the following files corresponding to the `/usr/pci/pciptys` entries shown above appear in `/dev`:

```
pciptyp0
pciptyp1
pciptyp2
pcittyp0
pcittyp1
pcittyp2
```

The number of pty devices in `/dev` available for PC-Interface terminal emulation may differ from system to system.

> NOTE
>
> On some UNIX system hosts, pty devices are used for other purposes in addition to PC-Interface terminal emulation. The `telnet` program and some windowing software, for example, use ptys. If a given pty is in use when PC-Interface tries to open it, PC-Interface tries to open each successive pty listed in `/usr/pci/pciptys` until it succeeds in opening one. For further information, see Chapter 5.

Most systems have enough pty devices to accommodate as many terminal emulation ports as the host system can adequately support. That is, simultaneously using more emulation ports than can be accommodated by the existing `/dev` special files would degrade system performance to an unacceptable level. Therefore, you should generally not need to create special files in `/dev`. If you find it necessary to increase the number of available `/dev` devices, refer to your UNIX system reference manuals, particularly the `mknod(8)` manual page.

3. Entries in the file `/etc/inittab`—For every entry in `pciptys` that does not include a `getty` command, there must be an entry in `/etc/inittab` to start a getty on that pty. An example of the `/etc/inittab` entries is:

```
                                    :
                                    :
pt0:23:respawn:/etc/getty pcittyp0 9600 none LDISC1    #PCI
pt1:23:off:/etc/getty pcittyp1 9600 none LDISC1    #PCI
                                    :
                                    :
```

   If the third field of the entry is "respawn," as in the example for `pcittyp0` above, a getty is started on that device, and the corresponding port is available for terminal emulation. If it is "off," as in the example of `pcittyp1`, a getty is not started. (For further information on `/etc/inittab`, refer to your UNIX system manual pages on `inittab(4)` and `getty(8)`.)

Procedure for Enabling and Disabling Ports

To change the number of terminal emulation ports available, follow these procedures:

1. To increase the number of terminal emulation ports available, edit the file `/usr/pci/pciptys` and add entries for the new ports to be enabled. Format your entries like the existing entries, but increment the number by one for each new entry. For example, if your system has eight ports (numbered 0-7) available when the system is installed and you want to enable two more ports, add the following lines to `/usr/pci/pciptys`:

```
    pciptyp8:pcittyp8:
    pciptyp9:pcittyp9:
```

   If the *Release Notes* accompanying this manual specify that ptys used for LAN terminal emulation should have gettys started in `pciptys`, also include the `getty` command as the third field. The `getty` command instructs the PC-Interface connection server (`pciconsvr`) to start a getty on that pty when it is needed. When PC-Interface is *not* using that pty, the getty is not running on it; therefore, it is available to other programs, such as `telnet`.

The `getty` command has the syntax:

```
/etc/getty devname 9600 [other arguments]
```

where *devname* is the pty device name (such as `pcittyp0`). The other arguments should mimic the entries in `/etc/inittab`. Do not include the comments, which are anything prefaced with a pound sign (#).

If the new port is to be used for RS-232 terminal emulation, the third field must not be present. Instead, the getty must be started in `/etc/inittab`. Ptys set up for RS-232 terminal emulation are not available for use by other programs.

2. To reduce the number of terminal emulation ports available, edit the file `/usr/pci/pciptys` and delete a line for each port you wish to disable. For example, to remove the two ports added according to the previous example, delete the lines:

```
pciptyp8:pcittyp8:
pciptyp9:pcittyp9:
```

Remember that for RS-232 service, there must be at least one entry with no `getty` command specified and a line in `/etc/inittab` to start a getty on that pty.

3. Modify `/etc/inittab` to reflect any changes made in `pciptys`. If you added an entry to `pciptys` and did not put a `getty` command in the third field, you must turn on the corresponding entry in `/etc/inittab`. If you removed such an entry from `pciptys`, you must turn off the corresponding `/etc/inittab` entry.

To turn gettys on, edit the file `/etc/inittab`. Look for the entries corresponding to the entries you have added to the file `pciptys`

NOTE

If a corresponding entry does not exist in `/etc/inittab`, you must add it.

and change the third field of these entries from "off" to "respawn." For example, if you have added the entry `pciptyp16:pcittyp16:` to `pciptys`, find the line:

```
pt16:23:off:/etc/getty pcittyp16 9600 none LDISC1  #PCI
```

and change it to:

```
pt16:23:respawn:/etc/getty pcittyp16 9600 none LDISC1  #PCI
```

The actual arguments to getty may vary from system to system. You should normally not change their values.

To turn off gettys for ports deleted from pciptys, edit the file /etc/inittab and change the third field from "respawn" to "off." For example, if you have removed the line pciptyp18:pcittyp18: from pciptys, change the /etc/inittab line:

```
pt18:23:respawn:/etc/getty pcittyp18 9600 none LDISC1  #PCI
```

to:

```
pt18:23:off:/etc/getty pcittyp18 9600 none LDISC1  #PCI
```

To communicate these changes to the system, either reboot or type:

```
init q
```

DOS Software Compatibility Issues

Most DOS software is compatible with PC-Interface operation. This means that DOS applications running on personal computers can access files on the UNIX system host through PC-Interface. DOS applications can also be stored on and executed from the UNIX system disk, as described under "Establishing Shared Personal Computer Software Libraries."

Following is a summary of the constraints on DOS software when used with PC-Interface. In general, applications will operate with PC-Interface if they *do not*:

- Overlay the DOS or BIOS area of storage.
- Program the 8259 interrupt controller in a way that interferes with PC-Interface's use.

- Disable interrupts, fail to issue an end-of-interrupt or IRET on a hardware interrupt level, or mask selected interrupt levels for more than 100 milliseconds.
- Use interrupts 13, 25, or 26 for access to the virtual drive.
- Configure hardware device registers that belong to the network hardware used by PC-Interface.
- Make incorrect use of timer interrupts.
- Open more than 125 files. This number is not affected by the value of the FILES= parameter in the CONFIG.SYS file.
- Make use of memory not assigned to them by the operating system.

This list of limitations for operation may not be complete.

We do not recommend using the following:

- Any version of IBM PC-DOS print spooling.
- The following MS-DOS or PC-DOS commands with the virtual drive: ASSIGN, JOIN, TREE, SHARE.
- Any version of DOS prior to 3.10.

5 PC-Interface System Architecture

Introduction

This chapter provides an architectural overview of the PC-Interface system. It defines the purpose of each software module on both the personal computer and the UNIX system host and describes how the modules interact during PC-Interface file service and terminal emulation sessions. First, the software modules on both the personal computer and the host are listed along with a brief statement of their purpose. Following this list is a discussion of PC-Interface operation over LANs and RS-232.

PC-Interface Software Modules

Personal Computer Modules

The following PC-Interface modules operate on the personal computer under DOS:

- PCIINIT.EXE: Initializes PC-Interface.
- BRIDGE.DRV: Provides basic PC-Interface support.
- CONFIG.SYS: Identifies device drivers to be loaded by DOS.
- *driver*.DRV: LAN device driver. *driver* represents your network interface board.
- LOGIN.EXE: Establishes a host connection for host file services.
- LOGOUT.EXE: Terminates a host file services session.
- DOS2UNIX.EXE: Converts text files in DOS format to UNIX system format.
- EM2.EXE: Establishes a terminal emulation session.
- JOBS.EXE: Displays the status of ON-initiated UNIX system tasks; reattaches the DOS console to an ON-initiated detached UNIX system task.
- KILL.EXE: Stops ON-initiated UNIX system processes.
- ON.EXE: Initiates execution of UNIX system commands on the host.
- PRINTER.EXE: Directs output to local or remote printer.
- UDIR.EXE: Lists virtual drive directory contents in UNIX system style.
- UNIX2DOS.EXE: Converts text files in UNIX system format to DOS format.
- PCICONF.EXE: Modifies the way PC-Interface responds to certain system calls.
- DOSWHAT.EXE: Returns program version numbers. This module is provided as a support tool.

Host Modules

The following PC-Interface modules operate on the host UNIX system:

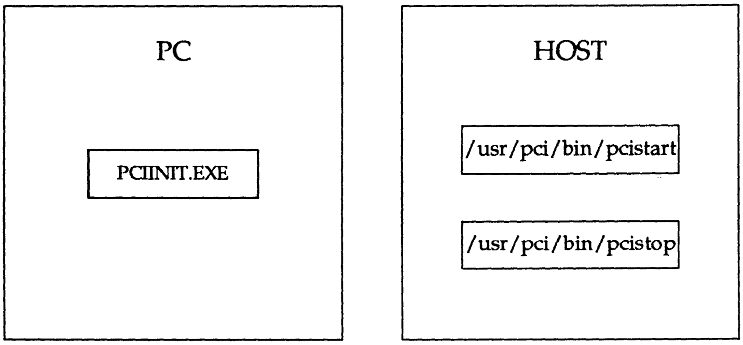
- /usr/pci/bin/pcistart: Starts PC-Interface daemons on the host.
- **/usr/pci/bin/pcistop**: Terminates execution of PC-Interface LAN processes on the host.
- /usr/pci/loadpci: Initialization program that starts the pcimapsvr and pciconsvr. Invoked by pcistart.
- /usr/pci/pcimapsvr.eth: Daemon that listens for and responds to broadcast requests for site tables from LOGIN and EM2. Also listens for broadcasts from pciconsvr on the net announcing that they are alive.
- /usr/pci/pciconsvr.eth: Daemon that listens for a connection request from LOGIN and EM2. Also broadcasts "pciconsvr *hostname* here" messages, indicating that the host is available for PC-Interface service.
- /usr/pci/pcidossvr.eth: Maintains an exclusive dialogue with the personal computer module BRIDGE.DRV. Executes UNIX system commands on behalf of the personal computer user and transmits the results back to the personal computer.
- /usr/pci/pcidosout.eth: A process created for use during terminal emulation over a LAN. It takes data passed from the UNIX system shell and other user programs via the pseudo-tty (pty) and sends it over the network to the personal computer.
- /usr/pci/pcidossvr.232: Same as pcidossvr.eth, but works over RS-232.
- /usr/pci/pcidosout.232: Same as pcidosout.eth, except it is used for terminal emulation over RS-232.
- /usr/pci/bin/sharectl: Initializes, removes, and prints information about shared memory.
- /usr/pci/pciptys: A text file listing pseudo-ttys (ptys) available to PC-Interface. Used for terminal emulation.

- /usr/pci/errlogger: A utility invoked automatically when PC-Interface encounters a serious or fatal error. It sends mail to root reporting the problem and also displays the message on the host console.
- /usr/bin/unix2dos: Converts text files in UNIX system format to DOS format.
- /usr/bin/dos2unix: Converts text files in DOS format to UNIX system format.

Software Structure for PC-Interface Over a LAN

Starting and Stopping PC-Interface Software on Personal Computers and Hosts

The term "LAN" refers to a connection between a host and a personal computer over a network adapter such as Ethernet.



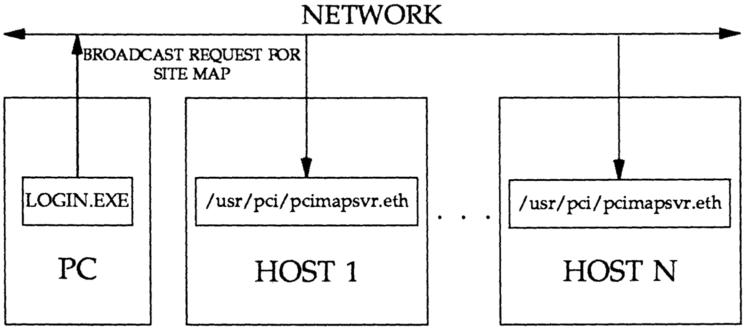
Before any interaction between personal computers and hosts can take place using PC-Interface, the PC-Interface software must be installed and initialized on both the personal computer and the host. The PCIINIT.EXE program on the personal computer initializes the personal computer side of PC-Interface. PCIINIT is normally invoked from the user's AUTOEXEC.BAT file, so it is executed automatically when the personal computer is booted.

Modules that support PC-Interface service on the personal computer are loaded as device drivers at boot time. These modules are specified in CONFIG.SYS. When a LAN is used, these modules are BRIDGE.DRV and *driver*.DRV (*driver* represents your network interface board). BRIDGE.DRV is the basic PC-Interface support module. *driver*.DRV is the LAN driver. All communication originating from the personal computer side of PC-Interface and destined for the network passes through the LAN driver.

The module `/usr/pci/bin/pcistart` initializes PC-Interface on the host. Its purpose is to start the two PC-Interface daemons that run on the host—`/usr/pci/pcimapsvr.eth` and `/usr/pci/pciconsvr.eth`. The `pcistart` module also establishes the UNIX operating system environment in which PC-Interface runs.

The module `/usr/pci/bin/pcistop` stops the two daemons and terminates the execution of PC-Interface LAN processes on the host. (Note that no equivalent module is required on the personal computer, since operation on the personal computer side is normally terminated by turning off the personal computer or booting without PC-Interface.)

Establishing a PC-Interface Connection Between a Personal Computer and a Host

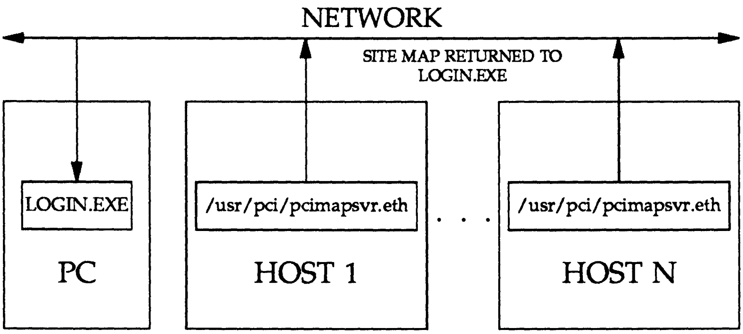


To begin a PC-Interface file services session, the personal computer user executes the LOGIN command. LOGIN.EXE (LOGIN) broadcasts over the network a request for a site map. A site map is a table listing the hosts on the network that are up and running PC-Interface—that is, all the hosts to which the personal computer can possibly connect.

The daemon `/usr/pci/pcimapsvr.eth` (pcimapsvr) on each host has two related functions:

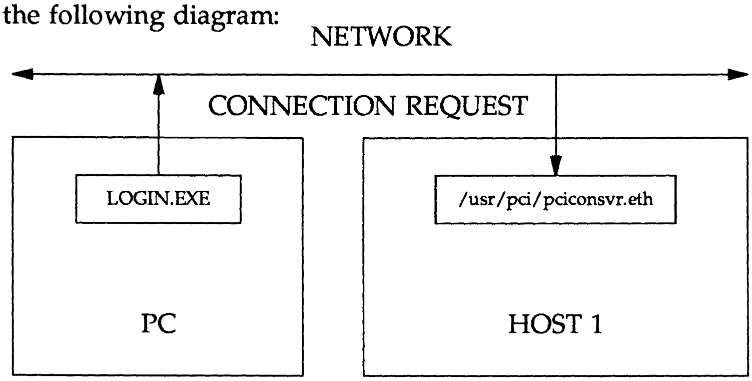
1. To listen for broadcast "pciconsvr *hostname* here" messages from each host running PC-Interface. (The module that broadcasts these messages is `/usr/pci/pciconsvr.eth`.) Each pcimapsvr keeps a table of available hosts that is updated approximately every 30 seconds based on these broadcasts.
2. To listen for broadcast requests for site maps issued by LOGIN and to send the current site map to any LOGIN process that requests it.

In response to LOGIN's request for a site map, therefore, each pcimapsvr returns its current site map, as illustrated in the following diagram:



On the personal computer side, LOGIN listens for the first returned site map and ignores any other incoming site maps from other hosts.

LOGIN displays for the personal computer user the list of available hosts. The user selects a host from the list, and LOGIN formats and sends a connection request to /usr/pci/pciconsvr.eth (pciconsvr) on the selected host, as illustrated in the following diagram:

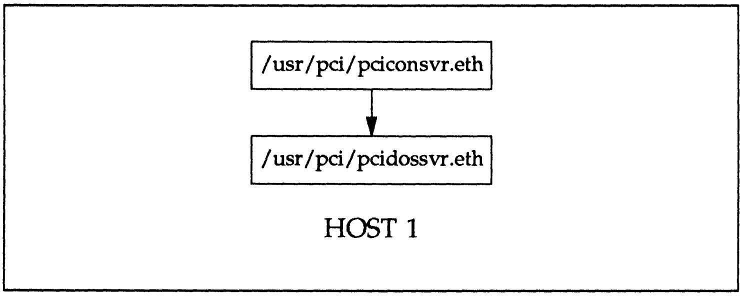


Note that the connection request need not be directed to the same host that provided the site map to LOGIN. Since all pcimapsvrs, under normal circumstances, have a current list of all available hosts, LOGIN can use the site map provided by any pcimapsvr.

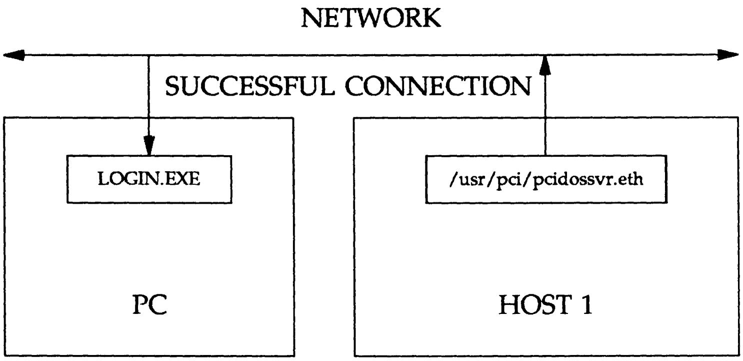
Like the pcimapsvr, each pciconsvr has more than one job:

1. To broadcast "pciconsvr *hostname* here" messages approximately every 30 seconds. These messages are used by pcimapsvrs to keep their lists of available hosts up-to-date.
2. To manage the connection between the personal computer and the host. This task includes several subordinate tasks. The connection management process begins when a pciconsvr receives a connection request, as illustrated above.

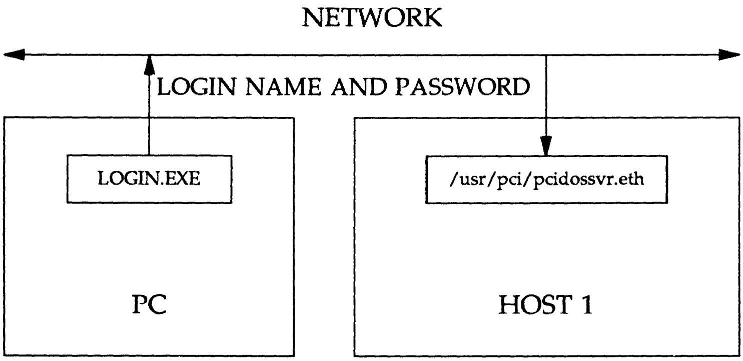
When the pciconsvr receives a connection request from LOGIN, it assigns a port number for use in subsequent communication with the personal computer and spawns a new process, /usr/pci/pcidossvr.eth (pcidossvr):



The pcidossvr inherits the port from the pciconsvr and informs LOGIN that a connection was successfully established, as illustrated in this diagram:



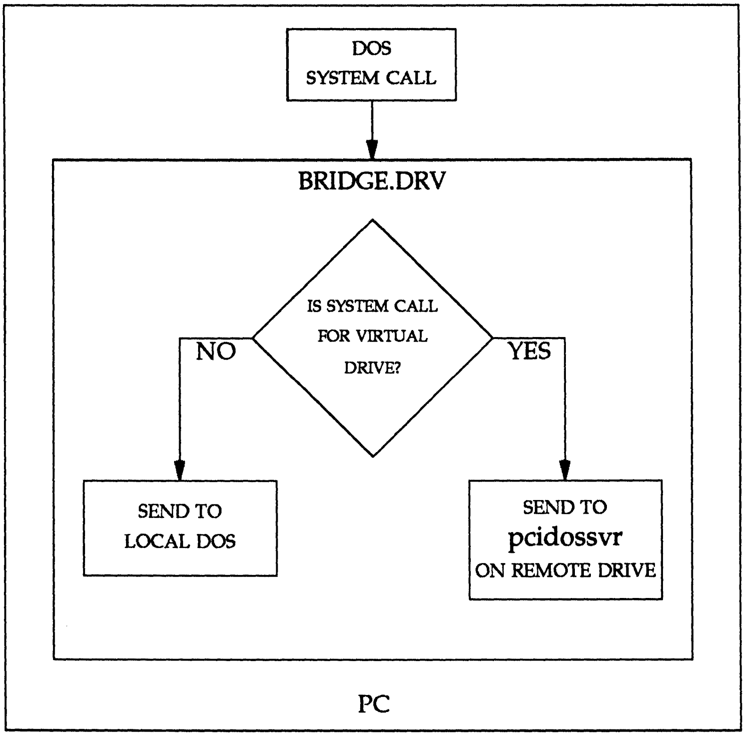
LOGIN then prompts the user for user name and password and transmits them to the pcidossvr, as the following diagram illustrates:



The pcidossvr validates the user name and password, changes its own user and group IDs to those of the logged in user, and informs LOGIN that the process has successfully completed. LOGIN tells the user on what drive the UNIX system file system is available, welcomes the user to PC-Interface, and returns the DOS prompt to the user.

Using an Established PC-Interface Connection

At this stage, bidirectional communication can occur between the personal computer and the host. On the personal computer side, the BRIDGE.DRV module takes over the task of communicating with the remote host. A top-level view of the operation of BRIDGE.DRV is as follows:



As shown in the illustration above, BRIDGE.DRV is interposed between the user's invocations and the local DOS that the user booted on the personal computer. The BRIDGE.DRV module intercepts all DOS system calls (the principal one being INT 21H). If the system call is a request for a local device, BRIDGE.DRV passes it unmodified to the local DOS. If the request is for the remote (virtual) drive, BRIDGE.DRV formats a transaction and sends it to its dossvr on the UNIX system host to be processed.

The pcidossvr module on the remote UNIX system host translates the user's standard DOS system calls into appropriate UNIX system commands, executes them on behalf of the user, translates the results back into standard DOS, and returns the results to the user via BRIDGE.DRV.

PC-Interface Utilities

When a PC-Interface file service session has been established as described above, the following special PC-Interface utilities become useful:

- ON
- JOBS
- KILL
- PRINTER
- UDIR

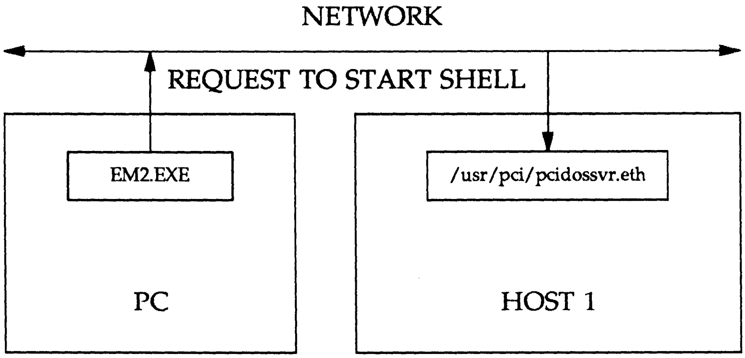
These commands allow the user to manipulate and display data on the remote UNIX system drive in ways not allowed by standard DOS. For descriptions of their operation, see the *PC-Interface User's Guide.* The following PC-Interface commands can be used independently of file services, as well as during a file service session:

- `/usr/bin/dos2unix` (UNIX system executable) and DOS2UNIX.EXE (DOS executable)
- `/usr/bin/unix2dos` (UNIX system executable) and UNIX2DOS.EXE (DOS executable)
- EM2.EXE

For a description of both the DOS and UNIX system executable versions of DOS2UNIX and UNIX2DOS, refer to the *PC-Interface User's Guide.* EM2.EXE is described from the user's point of view in the *PC-Interface User's Guide* as well, but additional relevant information appears below.

Using Terminal Emulation

To use PC-Interface to communicate with the remote UNIX system through a standard UNIX system shell rather than through DOS commands, you use the terminal emulation program EM2.EXE (EM2). When you invoke this program (by typing EM2), it broadcasts a request for a site map, just as LOGIN does. EM2 displays the returned site map, and you choose the host on which your terminal emulation session is to be established. EM2 then requests the pcidossvr on the remote host to start a UNIX system shell:

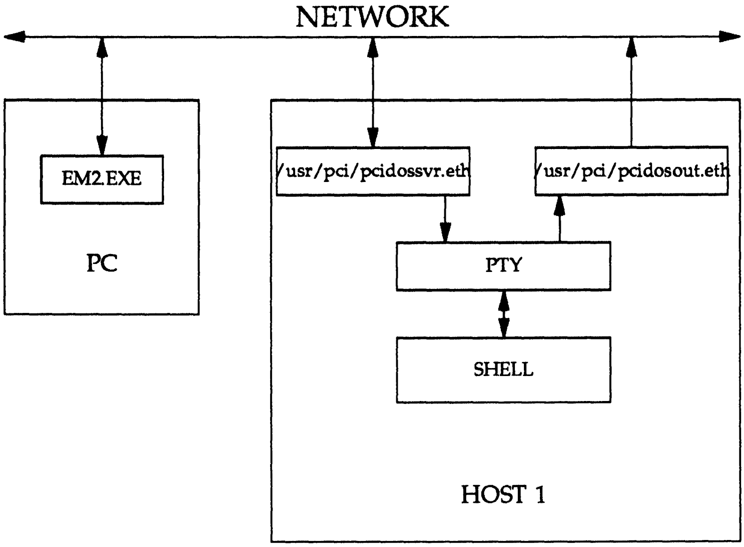


The UNIX system shell is presented to your personal computer through a pseudo-tty (pty) on the UNIX system side.

The events that occur when the pcidossvr receives the request from EM2 to start a UNIX system shell are as follows:

- The pcidossvr opens the control side of the pty. To do this, the pcidossvr attempts to open each pty listed in the file /usr/pci/pciptys until it succeeds in opening one.
- Once the control side of the pty is opened, getty starts a login operation on the slave side of the same pty.
- The pcidossvr spawns another process, /usr/pci/pcidosout.eth (pcidosout), which sends characters from the UNIX system shell (via the pty) back to the user.

The following illustration depicts the end result of this process:



The pcidosout process is required because the pcidossvr cannot listen for PC-Interface file service messages from the network at the same time as it listens for characters sent from the pty. The pcidosout process is therefore assigned the task of reading characters from the pty and sending them back to the user.

When you invoke EM2 during a PC-Interface file services session, the terminal emulation process does not terminate file services. You can alternate freely between standard PC-Interface file services and terminal emulation. Data sent by EM2 to the remote host is marked as terminal emulation data and is processed as described above. To suspend terminal emulation (without terminating the emulation session), press the ALT and D keys in combination. This allows you to return to DOS and continue using file services. To return to an ongoing terminal emulation session, reinvoke EM2. To terminate the emulation session, press ALT-L. A request to terminate emulation goes to the pcidossvr, which kills the pcidosout process. The pcidossvr closes the control side of the pty. The pty driver sends a SIGHUP signal to the UNIX system processes associated with that pty. The UNIX shell closes the slave side of the pty and exits. You can then use ALT-D to return to DOS and continue using PC-Interface file services.

Using the Terminal Emulator

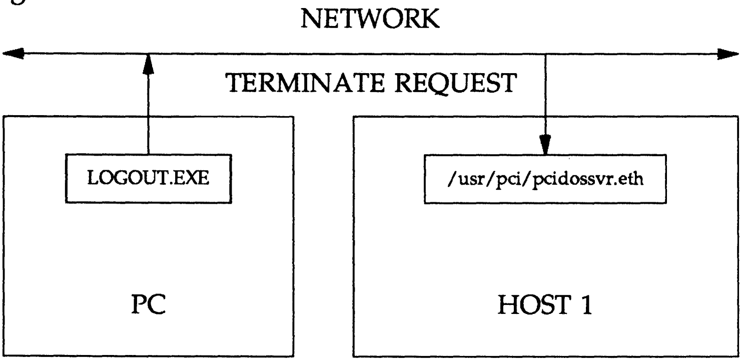
You can use the terminal emulator without establishing a file services session—that is, without logging into PC-Interface. PC-Interface must be initialized (using PCIINIT.EXE) for emulation to work, however.

When you invoke EM2 independently of a file services session, it broadcasts a request for a site map, just as it does when invoked during a file services session. When you choose a host for terminal emulation, EM2 requests that pciconsvr invoke a pcidossvr since, without a file services session, there is no pcidossvr already executing on behalf of the personal computer user on the remote host. You can then log into the UNIX system host and use terminal emulation as described above.

The only difference when you invoke the emulator outside of a file services session is that you cannot alternate between terminal emulation and file services. DOS is still available locally, and you can alternate between terminal emulation and local DOS by pressing the ALT and D keys in combination and invoking EM2 as described above.

Ending a PC-Interface File Services Session

When you execute the PC-Interface LOGOUT.EXE command from your personal computer, a terminate request is sent to the pcidossvr, as illustrated in the following diagram:



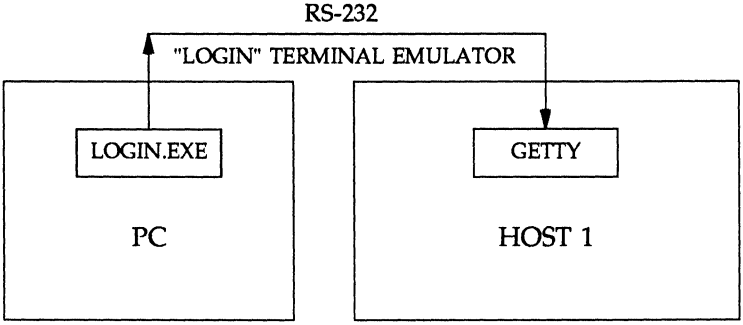
The pcidossvr sends a terminate request signal to any pcidosout process that might still exist (thus terminating any EM2 session to that host) and then exits. When the pcidosout and pcidossvr processes exit, the pciconsvr frees the network port and cleans up its status tables.

Software Structure for PC-Interface Over RS-232

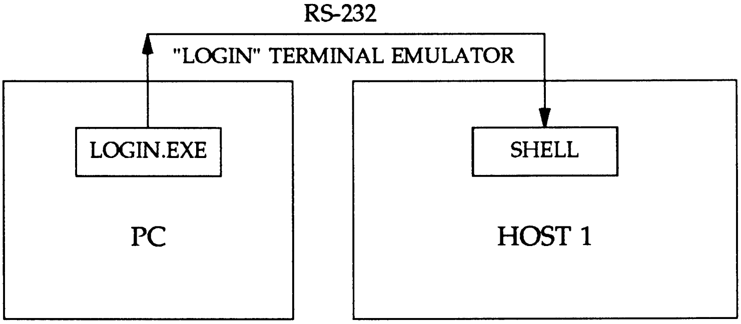
To use PC-Interface over RS-232, the PC-Interface software on both the personal computer side and the host side must be installed and initialized just as in the LAN case. BRIDGE.DRV is loaded on the personal computer side, as in the LAN case. No network device driver is required, however.

When you invoke the LOGIN command, a site map is requested and returned, again as in the LAN case. The list of available hosts displayed shows RS-232 serial ports as com ports. There may be one or two serial ports. (If no LAN is available, only the serial ports are shown.)

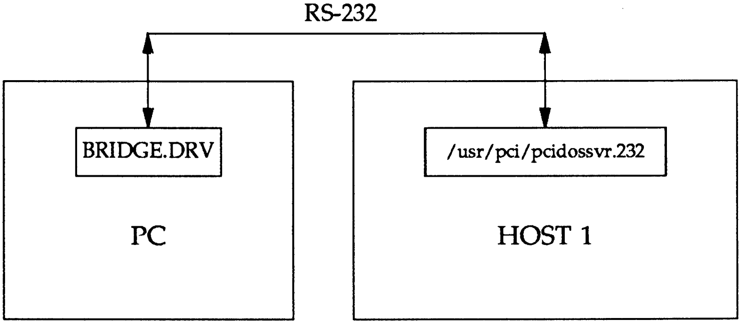
No `pciconsvr` is used with RS-232. Instead, when you select the RS-232 line, LOGIN invokes a special-purpose terminal emulator specifically for the purpose of logging into the remote host, as illustrated in the following diagram:



The RS-232 cable is connected to the UNIX system host just like any RS-232 cable coming from a standard terminal. Using the special-purpose emulator, you log in to the UNIX system host as though from a terminal. You now have a normal UNIX system shell running, as illustrated below:



When the login is finished, press F9. This terminates the special-purpose emulator and invokes /usr/pci/pcidossvr.232 on the host. As in the LAN case, LOGIN then informs you on which drive the UNIX operating system is available, welcomes you to PC-Interface, and returns the DOS prompt. (Unlike the LAN case, the pcidossvr for RS-232 does not have to validate your user name and password or change the user and group IDs. Because you have logged directly into the UNIX system as a standard UNIX system user, the UNIX login process does the validation, and the pcidossvr automatically inherits your user ID and group ID.) At this stage, you can communicate with the remote host and use PC-Interface utilities just as in the LAN case, as shown in the following diagram:



The software structure and operation for terminal emulation over RS-232 is the same as for terminal emulation over a LAN with one exception: the process spawned by the `pcidossvr` for output to DOS is `/usr/pci/pcidosout.232`, not `/usr/pci/pcidosout.eth`.

6 Problem-Solving Tools

Introduction

As system administrator, you may be called upon to assist users with problems they encounter using PC-Interface. The problem determination procedures in this chapter can help you determine if a problem is hardware related, software related, or caused by user errors.

For hardware problems, refer to the *Technical Reference* or *Guide to Operations* for the user's personal computer. For user error and software problems that are not described in this chapter, refer to the DOS reference manual or user's guide. This chapter concentrates on errors that are attributable to PC-Interface.

The approach to problem determination taken by this chapter is to break down the problem into its components and address the individual problems in order.

During this process you need to carefully collect the clues provided by the system (such as error messages and directory contents) to determine the exact nature of the problem.

Before using the procedures in this chapter, try to reproduce the problem you are having on a different personal computer. Make sure that you use the same software and hardware environment in running your duplication test.

If you cannot reproduce the problem, it may have been caused by a loose connection between the personal computer and the network or by hardware problems internal to the personal computer. If ensuring that the network is connected properly does not correct the problem, your personal computer may need repair. If the problem persists on another personal computer, continue with this procedure to classify your problem. Most PC-Interface problems occur in one of five major areas: system startup, establishing communication, proper usage, record locking, and normal operation. You can determine in which area to begin problem determination by following this procedure:

1. Does the failure occur during loading or PC-Interface initialization, before you attempt to log in?

   Yes: Start with "Problems with System Startup."

   No: Continue with the next step.

2. Does the failure prevent a session from being established?

   Yes: Start with "Problems Establishing Communication."

   No: Continue with the next step.

3. Does the failure occur during a PC-Interface session?

   Yes: Start with "Problems with Usage." If your problem is not described there, continue with "Problems with Record Locking" and "Problems with Normal Operation."

   No: The problem is not PC-Interface related. Refer to your *Guide to Operations* or specific application manual.

Refer to the proper heading in this section, as determined above, to find the cause of and solution for your problem. If you try the corrective actions described and the same problem persists, report the problem to your PC-Interface vendor.

Problems with System Startup

This section helps you determine the cause of a problem that occurs during system initialization, before you attempt to log in.

The PC-Interface system must be installed on a working diskette (that is, a diskette that contains the DOS system files) or fixed disk. The installation procedure given in the *PC-Interface User's Guide* combines the required PC-Interface files on a diskette (or fixed disk) with the DOS operating system, creating a working diskette.

Attempt to boot the personal computer from the fixed disk or working diskette containing PC-Interface by turning on the personal computer or by holding down the CTRL and ALT keys while you press the DEL key. When first powered on or booted, the personal computer runs a Power-On Self Test (POST). On some personal computers, a successful POST prints no messages; on others, it prints a series of messages of the form "xx KB," indicating the amount of memory that has been tested. Compare the POST when booting with your PC-Interface working diskette to a POST when booting from your DOS system disk without PC-Interface.

Use the following procedure to determine if your problem can be attributed to a system startup error:

1. Did the POST complete correctly without producing any error messages when the personal computer was powered on or booted?

   Yes: Continue with the next step.

   No: Refer to the *Guide to Operations.*

2. Was a "Non-System disk or disk error" message displayed?

   Yes: The disk does not have the correct operating system files on it for DOS to load. Refer to your *Guide to Operations* and the *PC-Interface User's Guide* for the procedures for creating a PC-Interface working disk.

   No: Continue with the next step.

3. Were the banners "PC-Interface @(#) Release 3.0" and "*driver*,@(#) Release 3.0" (where *driver* is the name of the network driver) displayed? (If PC-Interface is installed for RS-232 only, the second banner is not displayed.)

   Yes: Continue with the next step.

No: The file CONFIG.SYS must be present in the root of the PC-Interface disk and must contain the following lines:

```
device = bridge.drv
device = driver.drv
```

DRIVER.DRV represents the PC-Interface device driver. (This line may not be there if PC-Interface is being used over RS-232 only.) There may be directories specified for the devices (for example, you may see DEVICE = \PCI\BRIDGE.DRV).

Repeat the PC-Interface installation process if necessary to correct this problem.

4. Was a "Bad or missing BRIDGE.DRV" or "Bad or missing *driver*.DRV" message displayed?

   Yes: The file BRIDGE.DRV or *driver*.DRV was not found where specified in the CONFIG.SYS file. Ensure that the files are present in the locations specified in CONFIG.SYS. Repeat the PC-Interface installation process if necessary to correct this problem.

   No: Continue with the next step.

5. Was the message "Bad command or file name" displayed?

   Yes: The program PCIINIT.EXE may not have been found. It must be in a place where it can be executed at startup from the file AUTOEXEC.BAT. It must either be in the root of the PC-Interface disk, or in a subdirectory specified in the search path. Repeat the installation process to place the file in the correct location.

   No: Continue with the next step.

6. Was the message "pciinit: cannot find correct BRIDGE device" displayed?

   Yes: The PCIINIT program cannot find the PC-Interface bridge. To correct this problem, reinstall PC-Interface on the personal computer.

   No: Continue with the next step.

7. Was the message "pciinit: cannot open *path*\hosts" displayed?

   Yes: PCIINIT cannot find the HOSTS file. Either the file is not there, or there is no pointer to it in your AUTOEXEC.BAT file. If you have no HOSTS file, create one. The file should contain your personal computer's address, its name, and the word `localhost`. See your system administrator if you need help creating this file. You also need to add the following line to your AUTOEXEC.BAT file:

   ```
   HOSTS=path
   ```

   where *path* is the path of your HOSTS file. Usually the line should look like this:

   ```
   set hosts=\etc
   ```

   No: Continue with the next step.

8. Was the message "PC-Interface Release 3.0 (*serial*) Initialized" displayed?

   Yes: The resident part of PC-Interface initialization has succeeded. Continue with the section on establishing communication.

   No: If you have verified the contents of AUTOEXEC.BAT, then PCIINIT should have been executed. If an error message was displayed (other than the ones described above), report the problem to your PC-Interface vendor.

Problems Establishing Communication

This section helps you determine the cause of a problem that occurs during attempts to establish communication with your UNIX system host. Depending on the type of service that you want to use, one of two programs is run to establish communication. If you want to initiate host file services, you run the LOGIN program. If you want to use your personal computer to emulate a VT220 terminal, you run the EM2 program.

Refer to Appendix B of the *PC-Interface User's Guide* for errors that are not covered here.

Problems with LOGIN

1. After you executed the LOGIN command, was the message "Bad command or file name" displayed?

   Yes: The LOGIN.EXE file is not in the current directory nor in a directory that is listed in your DOS search path. The path is normally set during system startup by AUTOEXEC.BAT. To set the path yourself, see the DOS user's guide. If LOGIN.EXE is not on your disk, repeat the PC-Interface installation process.

   No: Continue with the next step.

2. Did you get a host selection menu?

   Yes: Continue with the next step.

   No: Go to step 4.

3. Did the host selection menu display only serial ports (com1 or com2)?

   Yes: If you are using RS-232 only (no LAN), go to the next step. If you are using a LAN, check the following:

   a. Check that all network hardware is installed and connected properly.

   b. Check that you specified the correct hardware when you ran the INSTALL program to install PC-Interface.

   c. Check that there is a PC-Interface host up and running on the network.

d. If your network is IP-based, make sure the host and PC are on the same logical network.

If all of the above appear OK, contact your PC-Interface vendor.

No: Continue with step 5.

4. Was the message "PC-Interface Not Initialized" displayed?

   Yes: You have a problem with system startup. Go to that section to determine the exact nature of the problem.

   No: Continue with the next step.

5. After you selected a host, was the message "Selected Host Not Available" or "Selected Host Not Available—Try Again?" displayed?

   Yes: You selected a host that became unavailable after it was included in the host selection menu, or your selected host cannot accommodate any more users, or there was a network communication problem.

   Verify that the network is working and that your personal computer is properly connected. Try to select the same host again. If it fails again, try a different host if one is available. If that works, check that the desired host is up and running PC-Interface and that the hardware between the personal computer and the host is operational.

   No: Continue with the next step.

6. Was the message "Login incorrect. Try Again? (y or n):" displayed?

   Yes: There are two possible problems in this case:

   a. You entered an incorrect user name or password at the prompt. Check that you are using the proper user name and have the correct password. Enter y to try again.

   b. The shared memory segments needed for file sharing and record locking on the host have not been initialized. See "Problems with Record Locking" in this chapter for instructions on how to initialize the files.

   No: Check Appendix B of the *PC-Interface User's Guide* for any further LOGIN errors.

Problems with EM2

1. After you executed the EM2 command, was the message "Bad command or file name" displayed?

   Yes: The EM2.EXE file is not in the current directory nor in a directory that is listed in your DOS search path. The path is normally set during system startup by AUTOEXEC.BAT. To set the path yourself, see the DOS user's guide. If EM2.EXE is not on your disk, reinstall PC-Interface on the personal computer.

   No: Continue with the next step.

2. Did the personal computer hang after you executed EM2?

   Yes: Check the current directory on the drive from which you invoked EM2 for an obsolete EM.SES file left from a previous, aborted session. If such a file exists, delete it and try executing EM2 again. There may be a host connection problem. Check your personal computer's connectors to ensure that you are connected to the network. Verify that the network is operational, and restart your system to try again.

   You may not have enough memory left to execute the emulator. Remove from your AUTOEXEC.BAT file the invocation of memory-resident programs. Restart your system and try EM2 again with no other memory-resident programs installed.

   If none of the above suggestions work, report the problem to your PC-Interface vendor.

   No: Continue with the next step.

3. Did the host selection menu display only serial ports (com1 or com2)?

   Yes: If you are using RS-232 only (no LAN), go to step 5. If you are using a LAN, continue with the next step.

   No: Go to step 5.

4. Was the message "PC-Interface software not initialized" displayed?

   Yes: You have a problem with system startup. Go to that section to determine the exact nature of the problem.

No: You do not have a LAN connection to any hosts. Perform the following:

a. Check that all network hardware is installed and connected properly.

b. Check that you specified the correct hardware when you ran the INSTALL program to install PC-Interface.

c. Make sure there is a PC-Interface host up and running on the network.

d. If your network is IP-based, make sure the host and PC are on the same logical network. Make sure that the IP addresses of the PC and the UNIX system machine do not conflict with other machines on the network.

If all of the above appear OK, contact your PC-Interface vendor.

5. After you selected a host from the menu and pressed return, did you get a blank screen but no UNIX system login prompt?

Yes: Terminate the terminal emulation session by pressing ALT-L, and try selecting the host again.

If you are running PC-Interface over RS-232, check the system console. Is a message similar to the one below displayed?

```
INIT:Command is respawning too rapidly. Check
   for possible errors.
id:01"/etc/getty tty01 9600"
```

If so, your terminal driver has a getty respawn problem, and you should follow steps a and b, below.

If you have a problem with gettys respawning, do the following:

a. Check the file `/etc/inittab` to determine the speed parameter to `/etc/getty`. In the example below, the speed parameter is 9600.

```
                   .
                   .
tty0:3:respawn:/etc/getty tty0 9600
tty1:3:respawn:/etc/getty tty1 9600
tty2:3:respawn:/etc/getty tty2 9600
                   .
                   .
```

b. Edit /etc/gettydefs and find the entry whose label corresponds to the speed parameter. Edit that entry, putting the word CLOCAL at the end of the initial-flags field. For example, this entry:

```
9600# B9600 HUPCL PARENB CS7 # B9600 SANE IXANY TAB3 #
Oogin: #9600
```

becomes this (the change is highlighted):

```
9600# B9600 HUPCL PARENB CS7 CLOCAL # B9600 SANE IXANY TAB3 #
Oogin: #9600
```

No: Continue with the next step.

6. Did the message "*hostname* Not Available -- Try another? (y or n)" appear below the connection attempt message on the host selection screen?

Yes: You may have selected a host that became unavailable after it was included in the host selection menu, or your selected host may be unable to accommodate any more users, or there may be a network communication problem.

Verify that the network is working and that your personal computer is properly connected. Try to select the same host again. If it fails again, try a different host if one is available. If that works, check that the desired host is up and running PC-Interface and that the hardware between the personal computer and the host is operational. Check that the pciptys file is correct, as described in "Tailoring Your PC-Interface System" in Chapter 4. If it is correct, contact your PC-Interface vendor.

No: Check Appendix B of the *PC-Interface User's Guide* for any further EM2 errors. If you cannot find an explanation for the problem you are experiencing, report the problem to your PC-Interface vendor.

Problems with Usage

This section describes some specific PC-Interface interactions that may result in problems if PC-Interface is not used correctly. Use the following procedure to determine if your problem is attributable to improper usage of PC-Interface:

1. Was a message indicating that you are out of room on your disk displayed?

   Yes: There are several possible reasons for this message. You may actually be out of disk space. Use the UNIX system `df` command to determine if this is the case. If so, you must remove some files from your disk.

   Or, you may be trying to create or remove a file in a directory for which you do not have write permission. Note that you may be doing this indirectly if you are using an application that needs to create temporary files as part of its operation. Check the permissions of the directory you are working in. If necessary, change directories, or change the permissions of the current directory. See "UNIX File Permissions" in Chapter 2 of the *PC-Interface User's Guide.*

   No: Continue with the next step.

2. Did you try unsuccessfully to print a file on the virtual drive? (Make sure that the printer is operational and not jammed.)

   a. If you entered PRINTER and then used one of the following:

      The print-screen key.

      COPYing your file to the print device (LPT1).

      did your file fail to print?

      Yes: Check that the UNIX system print spooler is enabled. Try the other printing methods. If your file still doesn't print, report the problem to your PC-Interface vendor.

      No: Continue with the next step.

   b. Are you printing from within an application program and expecting to see the printout before you leave the program?

      Yes: Your jobs do not print until you exit the application program unless you set a printer timeout, using the /T option to the PRINTER command. See the description of the PRINTER command in Chapter 6 of the *PC-Interface User's Guide* for further information.

No: Continue with the next step.

c. When you use the PRINT program, are your listings being printed with other listings embedded in them, or are they being broken up into multiple parts?

   Yes: The PRINT program should not be used for remote printing. See the *PC-Interface User's Guide*.

   No: If your printing problem is not described here, consult your PC-Interface vendor.

3. When using file services, do you see strange file names with apostrophes (') or other special characters embedded in them?

   Yes: These file names are a result of PC-Interface's "mapping" of UNIX system file names that cannot be translated directly to DOS file names. See "Naming Files" in Chapter 2 of the *PC-Interface User's Guide* for details of file-name mapping.

   No: Continue with the next step.

4. Are there files whose names are visible when you use the terminal emulator but invisible when you use host file services, even if you run the UDIR program?

   Yes: When logged in to the UNIX system you may use directory commands that allow you to see hidden files (files that begin with a dot). To see these files while using host file services, use UDIR with the -h option. If you wish, you can rename hidden files (using the PC-Interface ON command with the UNIX system `mv` command) to names that are visible to the DOS operating system. See your DOS user's guide for a description of legal DOS names.

   No: Continue with the next step.

5. Do you see a file in a directory listing, but you cannot access it?

   Yes: The file may belong to another user, and you may not have permission to read or write it. Or, you may not have execute permission for the directory that contains the file.

Change the file permissions or move the file into a directory for which you have write permission. Refer to "UNIX File Permissions" in Chapter 2 of the *PC-Interface User's Guide* for more information.

No: Continue with the next step.

6. Did you just edit a file, but when you printed it or edited it again, it was changed?

   Yes: The application used to edit the file did not support file sharing, and PC-Interface has no other mechanism to prevent two users from concurrently writing to a file for which they both have write permission. In addition, PC-Interface cannot prevent one user from overwriting a file that has just been updated by another user.

   The usual result of concurrent updating is that the changes made by the last user writing and closing the file are preserved and those made earlier are lost. Multiple users must coordinate their efforts and serialize the updates they make.

   If you have files in your own directories that should not be available to other users, change the file permissions accordingly. Refer to Chapter 4 of the *PC-Interface Administrator's Guide* for more information.

   No: Continue with the next step.

7. Do you see a file on the virtual drive whose permissions appear to change?

   Yes: There are differences in the DOS and UNIX system file permissions that can be visible to users. This is especially true when using the UNIX system chmod command to change UNIX system permissions to control certain types of file sharing.

   If file permissions are set in a particular way by using the UNIX system chmod command and then the file is updated by using almost any DOS program, the UNIX system chmod operation must be repeated. This is because most DOS programs perform updates by renaming the existing file and then creating a new file with the updated contents of the original file. The new file is created with default permissions according to the UNIX system umask environment variable, which is set when PC-Interface is started up. A common default for this variable sets all permissions to read and write for files and to read/write/execute for directories.

No: Continue with the next step.

8. Is there a directory on the virtual drive that you cannot access or write to, or whose permissions appear to change?

   Yes: A confusing situation can occur when PC-Interface users access directories for which they have execute permission but do not have read permission. If the programs or files they are accessing are opened with a DOS open system call, the open succeeds but a directory search for a file fails. Some applications perform directory searches even when they don't need to, which results in certain programs failing to open files if the user doesn't have read permission for the directory.

   No: Continue with the next step.

9. Are you unable to perform certain operations using the DOS CHMOD or ATTRIB command?

   Yes: The UNIX system file system does not support the DOS SYSTEM and ARCHIVE attributes. It also does not allow you to change the HIDDEN and VOLUME attributes on files on the virtual drive. You may only do this on DOS files stored on local drives. All other attributes work correctly with the DOS CHMOD and ATTRIB commands.

   Move any files you need to change using DOS CHMOD to the local drive and perform the action there.

   No: Continue with the next step.

10. When using the DOS BACKUP command on a virtual drive, do you get unexpected results?

    Yes: Because of the nature of the UNIX file system, the BACKUP command works differently on a virtual drive than it does when used on a local drive.

    - The /M option to BACKUP has no meaning when used on a virtual drive. On a local drive, the command:

      ```
      backup /m
      ```

      tells the BACKUP program to back up only files that have the ARCHIVE attribute set. Since the UNIX file system does not support the ARCHIVE attribute, the command shown above, when used on a virtual drive, backs up *all* files in the specified directory.

- When you use BACKUP to back files up onto a virtual drive, you get the error message "Target cannot be used for backup." This occurs because the BACKUP program is attempting to create a directory in the UNIX system root, which is not writable. In order to back up files onto the virtual drive, you must use SUBST to create a drive specifier that points to a drive whose root is writable. See Appendix D of the *PC-Interface User's Guide* for further information.
- When you use BACKUP with the /L option to back up files *from* a virtual drive, you get the error message "error opening logfile." The situation in this case is similar to the one described above. The /L option tells the BACKUP program to create a log file in the root directory, but the root directory on the virtual drive is not writable. To use the /L option with the BACKUP command, you must first use SUBST to create a drive specifier that points to a drive whose root is writable. See Appendix D of the *PC-Interface User's Guide* for further information.

No: Continue with the next step.

11. Are you getting messages like "cannot create file," "cannot open file," or "file not found," or do you find that you cannot execute a file on the virtual drive?

Yes: These messages may result from several different problems.

You may be out of disk space. Use the UNIX system `df` command to determine if this is the case. If so, you must remove some files from your disk.

You may be trying to create or remove a file in a directory for which you do not have write permission. Note that you may be doing this indirectly if you are using an application that needs to create temporary files as part of its operation. Check the permissions of the directory you are working in. If necessary, change directories or change the permissions of the current directory. See "UNIX File Permissions" in Chapter 2 of the *PC-Interface User's Guide.*

No: Continue with the next step.

12. Do you see a file on the virtual drive, but when you try to access it you get a "File locked" or "Access denied" message *or* do you find that another user is able to access a file that should have been locked?

    Yes: If other users are logged in to PC-Interface when this happens, you are probably unable to access the file because another user has opened it with an application that locks the file. In this case, you must wait until the other user has finished and the file is closed before you can access it.

    However, if no other users are logged in and you get one of these messages, there may be something wrong with record locking. For assistance in diagnosing and correcting the problem, see "Problems with Record Locking."

    No: Continue with the next step.

13. When using the virtual drive, do you see miscellaneous error messages that appear to be incorrect?

    Yes: Some DOS programs perform operations that cannot be supported by PC-Interface. For example, there are BIOS functions that perform read and write of physical disk blocks. These operations cannot be mapped into corresponding operations on a remote volume. Programs that attempt to do this produce an error diagnostic. Fortunately there are only a small number of programs used today that do not operate correctly under PC-Interface. Use the section on "Problems with Normal Operations" to isolate this type of error.

    No: Continue with the next step.

14. Have you changed the date and time on your personal computer, but files created later still show the old date and time?

    Using the DOS DATE and TIME commands only changes the date and time on your personal computer. It does not reset the UNIX system clock. Files created on the virtual drive take the UNIX system date and time, not the DOS date and time.

    If you need to change the date and time for the UNIX system, do so in the UNIX operating system environment.

    No: Check Appendix B of the *PC-Interface User's Guide* to see if the error you are investigating is explained. Also see "DOS Software Compatibility Issues" in Chapter 4 and check the *Release Notes* accompanying this manual. If the error is not explained in any of those places, use the section on "Problems with Normal Operations" to determine whether or not your problem is related to PC-Interface.

Problems with Record Locking

There are several ways in which record-locking shared memory or files can become corrupted or otherwise damaged. For example, the kernel may lose a shared memory segment or semaphore, which corrupts the shared memory but may or may not affect user files. Alternatively, a signal that the server cannot catch (such as `kill -9`) can terminate the server process. If this happens, locks may be left on, potentially making files or records inaccessible.

Another way files can be damaged is if a user issues the DOS COPY or BACKUP command on a virtual drive file being accessed by another user through an application that uses record locking. The user receives a DOS error message indicating that no copy was made; however, a partial (and therefore corrupt) version may actually exist.

Some possible indicators of shared memory corruption include:

- A file seems to be locked, but no other users are logged in.
- Access to a file is allowed when the file should be locked.

If shared memory is corrupted, the solution is to remove it. Once the shared memory is removed, the system automatically creates new, uncorrupted shared memory. The procedure below describes how to remove shared memory.

WARNING

Be sure all PC-Interface users are logged out before you proceed; otherwise, file damage may result.

1. Request that all PC-Interface users log out.
2. Type:

   ```
   ps -ef
   ```

   to make sure that all processes whose names start with `pci` are gone.
3. Making sure you are logged in as root, execute the following command:

   ```
   /usr/pci/sharectl -r
   ```

   This command removes shared memory. The system can now create new, uncorrupted shared memory.

4. If you are running Locus Computing Corporation's Merge 386, Version 1.2 or earlier, on the same host, also execute the following command:

   ```
   /usr/pci/sharectl -o -c
   ```

   This commands recreates old style shared memory, which is required by older software.

If removing the shared memory does not solve your problems with file sharing and record locking, contact your PC-Interface vendor for assistance.

Problems with Normal Operations

This section addresses problems that occur during an established PC-Interface session. A procedure is provided in this section for determining if your problem is indeed a PC-Interface problem.

If your problem is with one of the utilities provided with PC-Interface (other than PCIINIT, LOGIN, and EM2), read the appropriate section in the *PC-Interface User's Guide* that describes the utility. PC-Interface internal problems are the main focus of this section.

Most PC-Interface operation problems fall in two main groups—problems that are caused by PC-Interface and problems that are the result of the presence in memory of PC-Interface. The basic approach to isolating your problem is to recreate the conditions that produced the problem, but without the PC-Interface software being present. If this test passes, try again with PC-Interface present but not active. Use the procedure below as a guide for doing these tests.

Before you continue, carefully note the steps that led up to the appearance of the problem when running your application program. You will be repeating these steps later.

1. Copy to a DOS disk all the files that were in use when the problem occurred. This may include data files, DOS programs, and configuration files. The DOS disk can be a diskette or a fixed disk. We call this disk the test volume.
2. Disable PC-Interface, as follows:
    a. Make a copy of CONFIG.SYS, so that you don't lose your original file. Using any text editor, remove the following two lines from the CONFIG.SYS file:

    ```
    device = bridge.drv
    device = driver.drv
    ```

    DRIVER.DRV represents the PC-Interface device driver. (This line may not be there if PC-Interface is being used over RS-232 only.) There may also be directories specified for the devices (for example, you may see DEVICE = \PCI\BRIDGE.DRV).

    b. Make a copy of AUTOEXEC.BAT, so that you don't lose your original file. Edit AUTOEXEC.BAT using any text editor and remove the following line:

    ```
    pciinit
    ```

3. Fix the path or environment. If the path or any environment variables are set up to use the virtual drive, they must be changed to reflect the locations of the files you moved to the DOS test volume. This may require changes to AUTOEXEC.BAT or other configuration files, or changes to the manual procedure that you go through to run the program. Below are some examples of things that are likely to change. There may be other changes that are not listed below but are specified in the application program manual.

   a. Display your path. Does your path contain references to the virtual drive?

      Yes: Set your path as follows:

      `path=`*n:dir1;m:dir2 ...,*

      where *n* and *m* are drive specifiers not on the virtual drive, and the *dir* entries are directories where you want the command line processor to search.

      No: Continue with the next step.

   b. Depending on your application program, you may need to have all the data files in the same directory as your application program. This directory may need to be on your current drive. Are all programs and data files for this test located on the test volume so that they can be found using the path?

      Yes: Continue with the next step.

      No: Correct the path or move the program or data files appropriately.

4. Now reboot DOS without PC-Interface and confirm that it is not present by noting whether there are any banners printed during the boot process that say "PC-Interface." If any such banners appear, something was not done in the steps above and should be corrected before proceeding.

5. Now perform the sequence you noted earlier that produced the error. Is the error still present?

Yes: The error has nothing to do with PC-Interface, and the user should determine if the error is in the application program or in DOS itself. This is best done by referring to the trouble-shooting sections of the application program manual or by referring to the *Guide to Operations* for your personal computer.

No: If the error does not occur, the problem is either in PC-Interface or because PC-Interface is present but not actually responsible for the error. The latter case can occur if programs or devices disregard DOS guidelines or sound programming practices. Continue with the following steps to determine which is the case.

6. Test whether PC-Interface presence causes the error, as follows:

   a. Restore the file CONFIG.SYS to its original contents by copying the file you saved back onto CONFIG.SYS.

   b. Restart the system noting that this time banners should appear for the device drivers that are installed as part of PC-Interface. However, no banner should appear that says that PC-Interface has been initialized.

   c. Do not run the program PCIINIT from the AUTOEXEC.BAT file or manually, since this would activate PC-Interface.

   d. PC-Interface is now present in the system memory but is passive. It has no effect on the system other than taking up a certain amount of memory.

   e. Now perform the sequence that produced the error.

      Is the error still present?

      Yes: The DOS program that fails in this case (but succeeds when PC-Interface is NOT resident in memory) is written incorrectly but may work without PC-Interface. Some programs depend on being loaded at certain absolute memory locations. Adding DOS drivers, the PC-Interface program, or TSR programs to memory can cause such programs to fail.

      No: If you have not made any mistakes, you have found a PC-Interface problem and should contact your PC-Interface vendor for assistance. Please be prepared to give the vendor all relevant information concerning the problem.

A Appendix A

PC-Interface Messages

This appendix lists all PC-Interface host messages that the administrator might see. During execution of PC-Interface, the error logger (`/usr/pci/errlogger`) mails messages to root on the host system and also displays them on the host system console.

Each message is accompanied by an explanation of the cause of the message and a recommended response. In the messages, italics denote generic information for which the system substitutes actual values. For example, *device name* in a message is replace by the name of an actual device and *amount* and *n* are replaced by numbers.

The abbreviation *errno* stands for UNIX system error number. When an error number appears in a PC-Interface message, it means there was an error condition reported by the underlying UNIX system. To find out the cause of the problem, consult the list that correlates the error number with a description of the cause. On many systems this list is on-line in the file `/usr/include/sys/errno.h`. If your system does not have this file, consult your host system documentation.

PC-Interface Execution-Time Messages

These are messages that can be seen during execution of PC-Interface.

□ `Cannot create message queue ID, errno =` *errno*
□ `Cannot create semaphore, errno =` *errno*
□ `Cannot create shared mem segment, errno =` *errno*

Explanation: If *errno* is 28, the UNIX system has run out of the specified resource.

User Response: Reconfigure the kernel for more of the specified resource. If *errno* is a number other than 28, contact your PC-Interface vendor.

□ `Cannot create logger /usr/pci/errlogger`

Explanation: A serious or fatal error has occurred in PC-Interface, but the error-logging program could not be invoked.

User Response: Contact your PC-Interface vendor.

□ `Cannot find hostname` *hostname* `in /etc/hosts.  PCI aborting.`

Explanation: This happens when PC-Interface is started if the `/etc/hosts` file does not include the name of the host on which PC-Interface is being started.

User Response: The system administrator should add the host name and internet address to the `/etc/hosts` file.

□ `Can't exec DOS server '/usr/pci/pcidossvr.eth' :` *errno*

Explanation: The `pciconsvr` could not start the `pcidossvr` process.

User Response: Check that the file `/usr/pci/pcidossvr.eth` exists and has execute permission for all UNIX system users. The personal computer user should try logging in to PC-Interface again.

□ `Can't exec /usr/pci/pcidosout.eth:` *errno*
□ `Can't exec /usr/pci/pcidosout.232:` *errno*

Explanation: This is a terminal emulation error. The `pcidossvr` could not start the `pcidosout` process.

User Response: Check that the file `/usr/pci/pcidosout.eth` or `/usr/pci/pcidosout.232` exists and has execute permission for all UNIX system users. The personal computer user should retry terminal emulation.

□ `Can't initialize record lock data` *description_of_data*

Explanation: The record locking data described cannot be initialized.

User Response: Clear the shared memory, using the procedure described in "Problems With Record Locking" in Chapter 6. If the problem persists, contact your PC-Interface vendor.

□ `Can't open network - Bye`

Explanation: This is a PC-Interface initialization problem. The `pciconsvr` tried to open the network for normal PC-Interface communication and failed. The PC-Interface files are probably corrupted.

User Response: Check that all PC-Interface files are installed properly, are accessible, and are not corrupted. If necessary, deinstall and reinstall the PC-Interface host software. Restart PC-Interface by invoking `/usr/pci/bin/pcistart`.

□ `Connection server can't open network`

Explanation: The PC-Interface connection server (`/usr/pci/pciconsvr.eth`) could not open a network port. The most likely cause is that the port is already open because the connection server is already running. Only one connection server process can be running on a host.

User Response: None required. Do not try to execute the connection server when it is already running.

□ `Couldn't enter segment`

Explanation: An operation on shared memory failed during an attempt to establish a terminal emulation session. Note that this problem is *not* related to record-locking shared memory.

User Response: The user should retry terminal emulation. If the problem persists, contact your PC-Interface vendor.

□ `Error getting current working directory`

Explanation: PC-Interface log in over RS-232 failed because the UNIX system call `getpwuid` or `getuid` failed.

User Response: Verify that the user has a valid UNIX system account, including a proper entry in `/etc/passwd` and a home directory. The user should retry logging in.

□ `Error` *errno* `on get_tty`

Explanation: The initialization of the terminal port for terminal emulation over RS-232 failed.

User Response: Refer to the reported error number for a precise description of the problem. Take appropriate action to restore UNIX system resources.

□ `Error` *errno* `on STDIN set_tty`

Explanation: An attempt to set terminal modes on the serial port failed.

User Response: Refer to the reported error number and take appropriate action.

□ `IO ERROR ON PIPE.error` *errno*

Explanation: This is a terminal emulation error. An error condition on the pipe between the `pcidossvr` and the `pcidosout` process has occurred.

User Response: Retry terminal emulation.

□ `memory: Can't get` *amount* `bytes`
□ `memory: Can't resize to` *amount*

Explanation: The PC-Interface server process cannot allocate memory required for its tables. The UNIX system call `malloc` failed.

User Response: Retry the operation. If the problem persists, the system may be in an unstable state and should be rebooted.

□ `PIPE Read error:` *errno*

Explanation: This is a terminal emulation error. There was an I/O error on the pipe between the `pcidossvr` and the `pcidosout` process.

User Response: Refer to the reported error number and take appropriate action to restore system resources. Retry terminal emulation.

□ `PTY Read error:` *errno*

Explanation: This is a terminal emulation error. The `pcidosout` process encountered an error trying to read from the pty.

User Response: Check the reported error number for the cause of the problem and take appropriate action to restore system resources. Retry terminal emulation.

□ `Too many reXMITS`

Explanation: This is a terminal emulation error. The `pcidosout` process tried to transmit data to the personal computer, but the personal computer did not acknowledge the transmissions. The terminal emulation process is therefore terminated.

User Response: Retry terminal emulation

□ `TTY Write error:` *errno*

Explanation: This is a terminal emulation error. The terminal emulator tried to write to a terminal device and failed.

User Response: Refer to the reported error number for the cause of the problem and take appropriate action to restore system resources.

□ `vfInit: Can't calloc (`*number, number*`) for vfCache`

Explanation: PC-Interface was unable to allocate memory for an internal data area. This is a problem with the UNIX system, not with PC-Interface.

User Response: Retry the operation. If the problem persists, the UNIX system may be in an unstable state and should be rebooted. Alternatively, the UNIX system may be running out of memory.

Index

320–707

UNIX
PRESS

A Prentice Hall Title

ISBN 0-13-957507-3